



UNIVERSITAS UDAYANA

Volume 2, Nomor 3, Mei 2024

# JNATIA

Jurnal Nasional Teknologi Informasi dan Aplikasinya

Program Studi Informatika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Udayana

**Susunan Dewan Redaksi**  
**Jurnal Nasional Teknologi Informasi dan Aplikasinya (JNATIA)**  
**Volume 2, Nomor 3, Mei 2024**

**Penanggung Jawab**

Prof. Dra. Ni Luh Watiniasih, M.Sc., Ph.D.

**Ketua Redaktur**

I Gede Surya Rahayuda, M.Kom.

**Editor**

Drs. I Wayan Santiyasa, M.Si.  
Dr. Dra. Luh Gede Astuti, M.Kom.  
Ida Bagus Made Mahendra, S.Kom., M.Kom.  
I Gede Arta Wibawa, S.T., M.Kom.  
I Dewa Made Bayu Atmaja Darmawan, S.Kom., M.Cs.  
I Made Widhi Wirawan, S.Si., M.Si., M.Cs.

**Desain Grafis**

I Gede Yogananda Adi Baskara  
I Gusti Agung Ayu Gita Pradnyaswari Mantara

**Fotografer**

I Kadek Agus Candra Widnyana  
I Komang Dwiprayoga

**Sekretariat**

Ni Ketut Alit Widiastuti, S.Kom.  
Anak Agung Raka Darmawan, S.Kom.  
I Putu Herryawan, S.Kom.

**Reviewer**

Dr. Ir. I Ketut Gede Suhartana, S.Kom., M.Kom., IPM., ASEAN.Eng.  
I Gede Arta Wibawa, S.T., M.Kom.  
I Made Widiartha, S.Si., M.Kom.  
Ida Bagus Made Mahendra, S.Kom., M.Kom.  
Ida Bagus Gede Dwidasmara, S.Kom., M.Cs.  
Gst. Ayu Vida Matrika Giri, S.Kom., M.Cs.  
I Gusti Agung Gede Arya Kadyanan, S.Kom., M.Kom.  
Dr. Ngurah Agus Sanjaya ER, S.Kom., M.Kom.  
I Dewa Made Bayu Atmaja Darmawan, S.Kom., M.Cs.

Luh Arida Ayu Rahning Putri, S.Kom., M.Cs.  
Agus Muliantara, S.Kom., M.Kom.  
Dra. Luh Gede Astuti, M.Kom.  
Cokorda Rai Adi Pramartha, S.T., M.M., Ph.D.  
I Gusti Ngurah Anom Cahyadi Putra, S.T., M.Cs.  
Dr. Anak Agung Istri Ngurah Eka Karyawati, S.Si., M.Eng.  
I Gede Santi Astawa, S.T., M.Cs.  
Dr. Made Agung Raharja, S.Si., M.Cs.  
I Komang Ari Mogi, S.Kom., M.Kom.  
Ida Ayu Gde Suwiprabayanti Putra, S.Kom., M.T.  
I Putu Gede Hendra Suputra, S.Kom., M.Kom.  
Dr. Drs. I Wayan Santiyasa, M.Si.  
I Gede Surya Rahayuda, M.Kom.  
I Wayan Supriana, S.Si., M.Cs.

## Daftar Isi

Implementasi Algoritma A* (Star) untuk Menentukan Rute Jarak Terpendek Yukandri, Melisa Tryastie, Lia Nelda, Ressa Priskila, Viktor Handrianus Pranatawijaya	449-458
Prediksi Kenaikan Penduduk Jawa Timur Menggunakan Metode Long Short Term Memory Muhammad Rohman Irsyadi, Atiqur Rozi, Sandy Nicholas, Anggraini Puspita Sari	459-468
Deteksi Objek pada Citra Menggunakan Model YOLO Intara Pratama Harahap, Agus Muliantara	469-474
Memprediksi Kelulusan Mahasiswa: Graduate dan Dropout dengan Support Vector Machine dan GridSearchCV Ni Putu Eka Marita Anggarini, Agus Muliantara	475-480
Klasifikasi Ekspresi Wajah Menggunakan Metode CNN: Studi Kasus Dataset Kaggle I Wayan Restama Yasa, Anak Agung Istri Ngurah Eka Karyawati	481-488
Klasifikasi Penyakit Gagal Jantung Dengan Algoritma XGBoost I Gusti Gde Bagus Bhadraka Artawibawa, Anak Agung Istri Ngurah Eka Karyawati	489-494
Klasifikasi Hewan Berbasis Fitur Gray Level Co-Occurrence Matrix Dengan Artificial Neural Network Ryan Hangralim, Cokorda Rai Adi Pramatha	495-500
Deteksi Pneumonia Dengan Ekstraksi Fitur Gray-Level Co-occurrence Matrix (GLCM) Dan Support Vector Machine (SVM) I Gusti Bagus Sutha Arianata Putra, Gst. Ayu Vida Matrika Giri	501-510
Klasifikasi Citra Elektrokardiogram untuk Deteksi Penyakit Jantung Menggunakan Metode GLCM dan SVM Andreas Panangian Tamba, I Gede Arta Wibawa	511-520
Pengklasifikasian Kualitas Pisang dengan Deep Learning CNN Arsitektur VGG16 Vodka Joe Junior, I Gede Santi Astawa	521-530
Perancangan Algoritma Caesar Cipher Dalam Pengembangan Game-Based Learning Putu Arya Dharma Kesuma, I Gede Surya Rahayuda	531-536
Memanfaatkan NodeMCU dalam Serangan Jaringan Wi-Fi pada Frekuensi 2,4 GHz: Tinjauan Keamanan dan Tindakan Pencegahan Ida Bagus Wahyu Semara Kamajaya, I Gusti Agung Gede Arya Kadyanan	537-544
Rancangan Machine Learning Untuk Mendeteksi Lagu Plagiat Dominggo Pratama Sidauruk, I Gusti Ngurah Anom Cahyadi Putra	545-554
Sistem Monitoring Kamar Tidur Pintar dan Suhu Berbasis IoT dengan Cisco Packet Tracer Ni Made Ayu Wirasih, I Ketut Gede Suhartana	555-562

Optimalisasi Sistem Pencarian Produk Sunscreen Melalui Rancangan Ontologi Semantik I Gusti Agung Ayu Gita Pradnyaswari Mantara, I Komang Ari Mogi	563-572
Algoritma K-Means Untuk Clustering Provinsi Di Indonesia Berdasarkan Kasus Stunting Syelvia Julianti, I Made Widiartha	573-582
Implementasi Metode Design Thinking Dalam Perancangan UI/UX Aplikasi Wisata Bali I Kadek Agus Candra Widnyana, I Made Widiartha	583-590
Perbandingan Neural Network MLP, KNN, dan Decision Tree untuk Klasifikasi Penyakit Diabetes I Made Prenawa Sida Nanda, I Putu Gede Hendra Suputra	591-600
Perlindungan Seed Phrase Dengan Enkripsi Dual-Layer Menggunakan Algoritma AES dan Caesar Cipher Raihan Akbar Maulana, I Wayan Santiyasa	601-608
Analisis Perbandingan Kualitas Citra Hasil Steganografi DCT Dan LSB Berdasarkan Parameter RMSE Dan PSNR I Putu Krisnawan Putra, I Wayan Supriana	609-616
Klasifikasi Jenis Obat Berdasarkan Gejala Yang Dimiliki Pasien Menggunakan Metode K-Nearest Neighbors (KNN) Ngakan Putu Bagus Ananta Wijaya, Ida Ayu Gde Suwiprabayanti Putra	617-624
Perlindungan Transkrip Akademik Mahasiswa dengan Kombinasi Algoritma Rijndael dan SHA-3 Amsal Hamonangan Butarbutar, Ida Bagus Gede Dwidasmara	625-630
Klasifikasi Kematangan Sayuran Pare Dengan Metode KNN I Gede Yogananda Adi Baskara, Luh Arida Ayu Rahning Putri	631-638
Analisis Sentimen Ulasan Aplikasi RedBus Menggunakan Metode Random Forest dan AdaBoost Shelomita Putrinda Cudio, Luh Gede Astuti	639-648
Analisis Sentimen pada Ulasan Aplikasi myIM3 Menggunakan Multinomial Naive Bayes dengan TF-IDF Ni Komang Ayu Juliana, Made Agung Raharja	649-656
Perlindungan pada Citra Motif Kain Songket dengan Teknik Watermarking menggunakan RSA Encryption dan MSB Steganography I Wayan Gede Gemuh Raharja RL, Ngurah Agus Sanjaya ER	657-664

# Implementasi Algoritma A\* (Star) untuk Menentukan Rute Jarak Terpendek

Melisa Tryastie, Yukandri, Lia Nelda, Ressa Priskila, Viktor Handrianus Pranatawijaya

Program Studi Teknik Informatika, Fakultas Teknik,  
Universitas Palangka Raya,  
Jalan Yos Sudarso, Palangka, Kec. Jekan Raya, Kota Palangka Raya, Indonesia  
tryastie@mhs.eng.upr.ac.id  
yukandri01@mhs.eng.upr.ac.id  
lia003@mhs.eng.upr.ac.id  
ressa@it.upr.ac.id  
viktorhp@it.upr.ac.id

## Abstract

*In this research, the implementation of the A star algorithm is used to find efficient routes, reduce the travel time, and optimize the use of existing road infrastructure at waypoints. The A star algorithm uses the concept of Open List and Close List, which helps to reduce the number of rechecks on the points travelled, thus speeding up the search process. The A star algorithm stops when there are no more points on the open list or when the end point has been determined. This research method uses primary data, which consists of five street locations in Palangka Raya city, as nodes and combines the distance between the street points. The implemented program uses the A Star algorithm to calculate the shortest route and displays the path along with the distance. The purpose of this research is to achieve the shortest route and calculate the distance travelled for the waypoints.*

**Keywords:** *shortest route, A\* algorithm, waypoint.*

## 1. Pendahuluan

Dalam pengembangan sistem navigasi, penentuan rute terpendek antara dua lokasi merupakan salah satu aspek penting. Algoritma pencarian jalur terpendek, atau yang dikenal sebagai *shortest path*, digunakan untuk menemukan rute tercepat dalam sebuah *graph* [1]. Ada dua pendekatan umum dalam pencarian rute terpendek, yaitu pencarian buta (*blind search*) dan pencarian heuristik (*heuristic search*). Pencarian buta lebih sederhana namun membutuhkan waktu lebih lama, sedangkan pencarian heuristik memberikan hasil yang lebih optimal dalam waktu yang lebih singkat. Salah satu teknik pencarian heuristik yang efektif adalah algoritma A\* [1].

Algoritma A\* merupakan algoritma pencarian jarak terpendek yang optimal dan berfitur lengkap. Dengan menggunakan konsep *Open List* dan *Close List*, algoritma ini dapat mengurangi jumlah pengecekan ulang pada titik-titik yang dilalui, sehingga mempercepat proses pencarian. Algoritma A\* berhenti saat tidak ada lagi titik pada open list atau saat titik akhir telah ditentukan [2].

Implementasi algoritma A\* untuk menentukan rute jarak terpendek diharapkan dapat membantu meningkatkan efisiensi, mengurangi waktu tempuh, serta mengoptimalkan penggunaan infrastruktur jalan yang ada. Dengan demikian, implementasi ini diharapkan dapat memberikan kontribusi positif. Dalam penelitian ini, kami mengimplementasikan algoritma A\* untuk menentukan rute terpendek. Data yang digunakan dalam penelitian ini adalah data titik jalan, jarak antar jalan, dan estimasi waktu tempuh. Algoritma ini diimplementasikan dalam bahasa Python untuk memudahkan integrasi dengan sistem navigasi yang ada.

## 2. Metode Penelitian

### 2.1. Data Penelitian

Data yang dipergunakan dalam penelitian ini adalah data primer dan melibatkan informasi terkait lokasi titik-titik jalan di wilayah Kota Palangka Raya. Setiap titik jalan diwakili sebagai simpul dalam graf, dengan lokasi sebagai simpulnya, dan jarak antara titik-titik jalan sebagai sisi yang menghubungkannya. Jarak antara titik-titik jalan yang dimasukkan ke dalam sistem dan jumlah simpul bervariasi tergantung pada tujuan titik jalan yang ditetapkan. Evaluasi rute didasarkan pada perhitungan jarak, tanpa mempertimbangkan faktor kemacetan lalu lintas atau kondisi geografis rute. Penentuan posisi menggunakan Google My Maps sebagai panduan, dengan menghitung jarak antar titik berdasarkan koordinat lintang dan bujur lokasi yang dipilih [3].

### 2.2. Graph

Menggunakan *graph* pada peta untuk menyediakan jalur yang terhubung ke *node* untuk keseluruhan rute pada objek penelitian, dan mengimplementasikan algoritma A\* untuk menentukan rute terpendek dengan cepat dan efisien [4].

### 2.3. Tahapan Algoritma A Star

Algoritma A\* merupakan algoritma pencarian yang menghitung rute paling efisien dengan mengevaluasi *node* terkecil. Ini dirancang untuk menemukan jalur tercepat antara dua titik menggunakan teori *graph* atau grafik dan kumpulan *node* yang mewakili titik awal dan titik akhir [5]. Untuk mencari jarak terpendek pada suatu peta, peta tersebut harus direpresentasikan dalam bentuk diagram. Setiap titik pada *graph* mewakili persimpangan atau lokasi tertentu pada peta, sedangkan tepi *graph* mewakili jalur antara persimpangan tersebut. Tepi dari *graph* tersebut memiliki bobot yang mencerminkan panjang jarak antara titik-titik tersebut. Algoritma A\* digunakan untuk mengevaluasi setiap *node* dengan menggabungkan  $g(n)$ , yang merupakan biaya untuk mencapai *node* tersebut, dan  $h(n)$ , yang merupakan perkiraan biaya yang diperlukan untuk mencapai tujuan dari *node* tersebut. [6]. Dalam notasi matematika dapat direpresentasikan sebagai berikut:

Rumus:

$$f(n) = h(n) + g(n) \quad (1)$$

Keterangan:

- $f(n)$  adalah biaya estimasi terendah untuk mencapai *node* n.
- $h(n)$  adalah biaya dari *node* awal ke *node* n.
- $g(n)$  adalah perkiraan biaya dari *node* n ke *node* akhir.

Dalam penerapannya algoritma A\* mempunyai beberapa istilah dasar antara lain titik awal (*starting point*), simpul (*nodes*), S, *open list*, *closed list*, harga (*cost*) [1]. Dapat dijelaskan sebagai berikut:

- "Titik awal (*starting point*)" adalah titik awal dalam menentukan algoritma A\*.
- "Simpul (*nodes*)" Adalah sebuah titik diagram yang mewakili untuk mencari rute terpendek.
- "S" adalah *node* yang digunakan sebagai titik awal dalam pencarian rute terpendek.
- "*Open List*" adalah kumpulan *node* yang belum dieksplorasi sepenuhnya dan masih dapat diakses dari titik awal atau *node* yang sedang dieksplorasi.
- "*Close List*" adalah tempat menyimpan data sebelum *node* S, yang juga merupakan bagian dari jalur terpendek yang berhasil diperoleh.
- "Harga (*Cost*)" adalah nilai yang diperoleh dari penjumlahan nilai setiap *node* pada jalur terpendek dari titik awal ke S, dan penjumlahan nilai estimasi dari *node* hingga *node* tujuan.

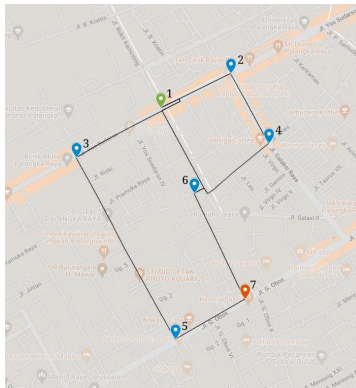
Inti dari algoritma A\* adalah untuk menemukan rute terpendek dari titik awal ke node tujuan dengan meminimalkan biaya atau *cost*.

Langkah – langkah untuk mencari rute terpendek dengan algoritma A\* [1] adalah sebagai berikut:

- a. Langkah pertama adalah memulai dari node S sebagai titik awal..
- b. Selanjutnya, tambahkan semua *node* yang terhubung dengan *node* S ke dalam *Open List*.
- c. Pilih *node* dengan nilai terkecil dari *Open List*, yaitu *node-node* yang terhubung dengan *node* S.
- d. Pindahkan *node* S ke *node* dengan nilai *Cost* terkecil. Simpan *node* sebelum S sebagai *node* induk dari S dan tambahkan ke dalam *Close List*. Jika terdapat *node* lain yang terhubung dengan S (yang sudah dipindahkan) namun belum ada di *Open List*, masukkan *node* tersebut ke dalam *Open List*.
- e. Bandingkan nilai *g(n)* dengan nilai *g(n)* sebelumnya. Jika nilai *g(n)* yang baru lebih kecil, kembalikan S ke posisi sebelumnya atau titik awal. *Node* yang sudah dimasukkan ke dalam *Close List*. Ulangi langkah ini sampai solusi ditemukan atau tidak ada lagi *node* yang tersisa di *Open List*.

### 3. Hasil dan Pembahasan

Dalam penelitian ini, pencarian rute terpendek dilakukan dengan mengidentifikasi lokasi pada peta menggunakan informasi koordinat geografis, yaitu latitude dan longitude. Implementasi algoritma A\* untuk mencari rute terpendek antara titik jalan atau tujuan dilakukan menggunakan bahasa pemrograman Python. Gambar 1 dan Tabel 1 di bawah ini menunjukkan perjalanan dari titik awal (ditandai dengan warna hijau) ke titik akhir (ditandai dengan warna merah) yang digunakan dalam penelitian.



Gambar 1. Goggle My Maps Titik awal dan Titik akhir

Tabel 1. Koordinat latitude dan Longitude dari titik-titik jalan.

No	Titik Jalan	Latitude	Longitude
1	Jl. Yos Sudarso	-2.21916	113.89514



No	Titik Jalan	Latitude	Longitude
2	Jl. Yos Sudarso	-2.21769	113.8981
3	Jl. Yos Sudarso	-2.22125	113.89156
4	Jl. Galaksi Raya	-2.22064	113.8997
5	Jl. G. Obos	-2.22894	113.89576
6	Jl. G. Garu	-2.22275	113.89655
7	Jl. G. Obos	-2.22723	113.8987

Berdasarkan koordinat titik jalan lintang (Latitude) dan titik jalan (Longitude) yang diperoleh dari Goggle My Maps, Jarak antara satu titik dengan titik lainnya dihitung menggunakan rumus berikut:

Rumus [3] :

$$d_{ij} = \left( 69 \cdot \sqrt{(lon_i - lon_j)^2 + (lat_i - lat_j)^2} \right) \cdot 1,60934 \quad (2)$$

Keterangan:

- " $d_{ij}$ " adalah jarak dalam satuan kilometer antara dua titik koordinat i dan j.
- " $lon_i$ " adalah nilai longitude titik i.
- " $lon_j$ " adalah nilai longitude titik j.
- " $lat_i$ " adalah nilai latitude titik i.
- " $lat_j$ " adalah nilai latitude titik j.

Rumus di atas menggunakan rumus Haversine, di mana konstanta 69 digunakan untuk mengubah perbedaan lintang menjadi jarak dalam mil. Konstanta ini didasarkan pada asumsi bahwa jarak sejajar satu derajat lintang sekitar 69 mil, atau sekitar 111,045 kilometer, pada garis lintang rata-rata Bumi. Setelah menghitung jarak dalam mil dengan konstanta 69, kita dapat mengonversinya menjadi kilometer dengan menggunakan faktor 1,60934. Faktor ini merupakan hasil dari konversi 1 mil menjadi kilometer (1 mil = 1,60934 kilometer), sehingga kita dapat mengalihkan jarak dalam mil dengan faktor ini untuk mengubah satuan menjadi kilometer [6].

Program Python mencari jarak semua titik ke titik akhir:

```
import math
class Node:
    def __init__(self, x, y):
        self.x = x
        self.y = y

def haversineDist(n1, n2): # fungsi untuk menghitung jarak antara dua node
    # Mengubah koordinat menjadi radian
    lat1 = math.radians(n1.x)
    lon1 = math.radians(n1.y)
    lat2 = math.radians(n2.x)
    lon2 = math.radians(n2.y)

    # Menghitung jarak menggunakan rumus Haversine
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon / 2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
    R = 6373.0 # Perkiraan radius bumi dalam km
    distance = R * c
```

```
# Membulatkan jarak menjadi 2 tempat desimal
distance = round(distance, 2)
return distance

n1 = Node()
n2 = Node()
distance = haversineDist(n1, n2)
print(f"Jarak antara ((n1.x), (n1.y)) dan ((n2.x), (n2.y)) adalah (distance) km.")
```

Berdasarkan perhitungan sebelumnya tentang jarak antar titik, jumlah jarak akan dinyatakan dalam satuan kilometer dan dibulatkan menjadi dua angka di belakang koma, sehingga hasilnya adalah sebagai berikut:

**Tabel 2.** Jarak Masing-masing Titik Tujuan

x	1	2	3	4	5	6	7
1	0.00	0.37	0.46	0.53	1.09	0.43	0.98
2	0.37	0.00	0.83	0.37	1.28	0.59	1.06
3	0.46	0.83	0.00	0.91	0.97	0.58	1.04
4	0.53	0.37	0.91	0.00	1.02	0.42	0.74
5	1.09	1.28	0.97	1.02	0.00	0.69	0.38
6	0.43	0.59	0.58	0.42	0.69	0.00	0.55
7	0.98	1.06	1.04	0.74	0.38	0.55	0.00

Setelah mengetahui jarak antara setiap titik ke titik lainnya, langkah selanjutnya adalah menghitung pencarian rute terpendek dengan mengimplementasikan algoritma A\* pada program berikut:

Program pencarian rute terpendek:

```
def aStarAlgo(start_node, stop_node):
    open_set = set(start_node)
    closed_set = set()
    g = {} # Menyimpan jarak dari node awal
    parents = {} # parents berisi peta kedekatan dari semua node
    # Jarak node awal dari dirinya sendiri adalah nol
    g[start_node] = 0
    # start_node adalah node root yaitu tidak memiliki node induk
    # jadi start_node diatur sebagai node induknya sendiri
    parents[start_node] = start_node
    while len(open_set) > 0:
        n = None
        # node dengan nilai f() terendah ditemukan
        for v in open_set:
            if n == None or g[v] + heuristic(v) < g[n] + heuristic(n):
                n = v
        if n == stop_node or Graph_nodes[n] == None:
            pass
        else:
            for (m, weight) in get_neighbors(n):
                # node 'm' yang tidak ada dalam set pertama dan terakhir ditambahkan ke
                # set pertama
                # n diatur sebagai induknya
                if m not in open_set and m not in closed_set:
                    open_set.add(m)
                    parents[m] = n
                    g[m] = g[n] + weight
                # untuk setiap node m, bandingkan jaraknya dari start yaitu g(m) dengan
                # dari start melalui node n
            else:
```

```
        if g[m] > g[n] + weight:
            # update g(m)
            g[m] = g[n] + weight
            # ubah induk m menjadi n
            parents[m] = n
            # jika m ada dalam set tertutup, hapus dan tambahkan ke set
            # terbuka
            if m in closed_set:
                closed_set.remove(m)
            open_set.add(m)

    if n == None:
        print('Path does not exist!')
        return None

    # jika node saat ini adalah stop_node
    # maka kita mulai merekonstruksi jalur dari node tersebut ke start_node
    if n == stop_node:
        path = []
        total_distance = 0
        while parents[n] != n:
            path.append(n)
            total_distance += g[n] - g[parents[n]]
            n = parents[n]
        path.append(start_node)
        path.reverse()
        print('Rute Terpendek: {}'.format(path))
        print('Total Jarak Tempuh: {:.2f}'.format(total_distance))
        return path

    # hapus n dari open set, dan tambahkan ke closed_set
    # karena semua tetangganya telah diperiksa
    open_set.remove(n)
    closed_set.add(n)
    print('Path does not exist!')
    return None

# mendefinisikan fungsi untuk mengembalikan tetangga dan jaraknya
# dari node yang dilewati
def get_neighbors(v):
    if v in Graph nodes:
        return Graph nodes[v]
    else:
        return None

def heuristic(n):
    H_dist = {
        'S': 0.98,
        'A': 1.06,
        'B': 1.04,
        'C': 0.74,
        'D': 0.38,
        'E': 0.55,
        'Z': 0
    }
    return H_dist[n]

Graph nodes = {
    'S': [('A', 0.37), ('B', 0.46), ('E', 0.43)],
    'A': [('C', 0.37)],
    'B': [('D', 0.97)],
    'C': [('E', 0.42)],
    'D': [('Z', 0.38)],
    'E': [('Z', 0.55)],
}

aStarAlgo('S', 'Z')
```

Berdasarkan proses pencarian rute terpendek di atas, maka rute terpendek ditentukan berdasarkan titik awal (warna hijau) dan titik akhir (warna merah) yang diinginkan.

Hasil output program:

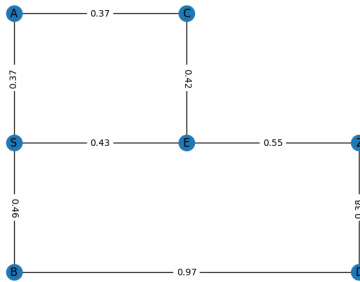
```
Rute Terpendek: ['S', 'E', 'Z']  
Total Jarak Tempuh: 0.98
```

Hasil eksekusi program di atas ditentukan oleh proses pencarian rute terpendek menggunakan algoritma A\*. Langkah selanjutnya adalah memberikan input agar dapat diimplementasikan dalam graf. Implementasi ini bertujuan untuk memudahkan visualisasi jalur rute pada peta dengan lebih detail.

Program implementasi graph dalam mencari rute terpendek:

```
import networkx as nx  
import matplotlib.pyplot as plt  
  
# Membuat objek graf  
G = nx.Graph()  
  
# Menambahkan node ke graf  
G.add_node('S', pos=(0, 1))  
G.add_node('A', pos=(0, 2))  
G.add_node('B', pos=(0, 0))  
G.add_node('C', pos=(2, 2))  
G.add_node('D', pos=(4, 0))  
G.add_node('E', pos=(2, 1))  
G.add_node('Z', pos=(4, 1))  
  
# Menambahkan edge (sambungan) antara node  
G.add_edge('S', 'A', weight=0.37)  
G.add_edge('S', 'B', weight=0.46)  
G.add_edge('S', 'E', weight=0.43)  
G.add_edge('A', 'C', weight=0.37)  
G.add_edge('B', 'D', weight=0.97)  
G.add_edge('C', 'E', weight=0.42)  
G.add_edge('D', 'Z', weight=0.38)  
G.add_edge('E', 'Z', weight=0.55)  
  
# Mendapatkan posisi node  
node_pos = nx.get_node_attributes(G, 'pos')  
  
# Menampilkan graf dengan posisi node yang telah ditentukan  
nx.draw(G, pos=node_pos, with_labels=True)  
  
# Menampilkan jarak pada edge  
labels = nx.get_edge_attributes(G, 'weight')  
nx.draw_networkx_edge_labels(G, pos=node_pos, edge_labels=labels)  
  
plt.show()
```

Hasil output program:



Gambar 2. Hasil Tampilan Graph

Hasil perhitungan lengkap rute terpendek dari setiap titik awal ke titik akhir dapat lihat dalam tabel di bawah berikut:

Tabel 3. Hasil perhitungan pengujian

No	Titik Awal	Titik Akhir	Rute	Jarak (Km)
1	S (Jl. Yos Sudarso)	Z (Jl. G. Obos)	S → A → C → E → Z	1.71
2	S (Jl. Yos Sudarso)	Z (Jl. G. Obos)	S → B → D → Z	1.81
3	S (Jl. Yos Sudarso)	Z (Jl. G. Obos)	S → E → Z	0.98

Berdasarkan hasil perhitungan pengujian, dapat disimpulkan bahwa program ini dibuat untuk menerima input berupa graph agar dapat menghitung rute terpendek dan menampilkan rute tersebut beserta jaraknya.

#### 4. Kesimpulan

Dalam penelitian ini, sistem yang menggunakan algoritma A\* untuk menentukan rute terpendek pada titik jalan telah berhasil dibangun dan diuji coba. Hasil dari sistem menunjukkan bahwa algoritma A\* efektif digunakan untuk mencari rute terpendek dalam suatu peta. Setiap lokasi direpresentasikan sebagai simpul dalam graf, dengan jalur yang menghubungkan lokasi tersebut sebagai tepi graf. Bobot atau jarak setiap tepi digunakan untuk menghitung nilai heuristik. Proses pencarian rute dengan algoritma A\* melibatkan pembentukan pohon pencarian dan penggunaan antrian prioritas. Dengan menggunakan algoritma ini, sistem dapat mengolah graf masukan, menghitung rute terpendek, dan menampilkan hasil rute terpendek beserta jaraknya. Sebagai contoh, jika titik awalnya adalah S (Jl. Yos Sudarso) dan titik akhirnya adalah Z (Jl. G. Obos), maka rute terpendek adalah 0,98 km melalui rute S → E → Z.

#### Daftar Pustaka

- [1] Ida Bagus Gede Wahyu Antara Dalem, "Penerapan Algoritma A\* (STAR) Menggunakan Graph Untuk Menghitung Jarak Terpendek," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 1, no. 1, pp. 41–47, Apr. 2018, Accessed: Apr. 17, 2024. [Online]. Available: <https://ejournal.instiki.ac.id/index.php/jurnalresistor/article/view/253>

- [2] Yusra Fernando, Muhammad Ativ Mutsaqov, and Dyah Ayu Megawaty, "Penerapan Algoritma A-STAR Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android," *Jurnal Teknoinfo*, vol. 14, no. 1, pp. 27–34, Jan. 2020, doi: 10.33365/jti.v14i1.509.
- [3] Rasita Natasya Br Sitepu and I Gusti Ngurah Anom Cahyadi Putra, "Penentuan Rute Terpendek Menggunakan Algoritma a Star (Studi Kasus: Distributor Barang)," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 1, no. 1, pp. 431–440, Nov. 2022, Accessed: Apr. 17, 2024. [Online].  
Available: <https://ojs.unud.ac.id/index.php/jnatie/article/view/92697>
- [4] Rizal Ahmad Fauzi and Rizal Rachman, "Implementasi Algoritma A\* Menggunakan Graph Pada Aplikasi Route at Location Berbasis Web," *eProsiding Sistem Informasi (POTENSI)*, vol. 2, no. 1, pp. 121–129, Jun. 2021, Accessed: Apr. 17, 2024. [Online].  
Available: <https://eprosiding.ars.ac.id/index.php/psi/article/view/369>
- [5] Susilawati, Robby Rizky, Sri Setiyowati, and Aghy Gilar Pratama, "Penerapan Metode A\*Star Pada Pencarian Rute Tercepat Menuju Destinasi Wisata Cagar Budaya Menes Pandeglang," *Geodika: Jurnal Kajian Ilmu dan Pendidikan Geografi*, vol. 4, no. 2, pp. 192–199, Dec. 2020, doi: 10.29408/geodika.v4i2.2754.
- [6] I Putu Andi Wiratama Putra and I Gede Arta Wibawa, "Implementasi Algoritma A\* (Star) dengan Graf untuk Menentukan Rute Terpendek Distributor Kopi," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 1, no. 4, pp. 1053–1062, Aug. 2023, Accessed: Apr. 17, 2024. [Online].  
Available: <https://ojs.unud.ac.id/index.php/jnatie/article/view/102421>

Halaman ini sengaja dibiarkan kosong

# Prediksi Kenaikan Penduduk Jawa Timur Menggunakan Metode Long Short Term Memory

Atiqur Rozi<sup>a1</sup>, Muhammad Rohman Irsyadi<sup>a2</sup>, Sandy Nicholas<sup>a3</sup>, Anggraini Puspita Sari<sup>a4</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Jawa Timur, Indonesia

<sup>1</sup>22081010010@student.upnjatim.ac.id

<sup>2</sup>22081010024@student.upnjatim.ac.id

<sup>3</sup>22081010237@student.upnjatim.ac.id

<sup>4</sup>anggraini.puspita.if@upnjatim.ac.id

## Abstract

*This research aims to develop a prediction model for population increase in East Java using the Long Short Term Memory (LSTM) method. Historical population data from the previous period will be used as input to train the LSTM model. This approach is expected to produce accurate predictions about population growth in the East Java region. The LSTM method was chosen due to its ability to handle sequential data and long-term memory, which is in line with the characteristics of demographic data. This research will involve data pre-processing, LSTM model building, and model performance evaluation using relevant metrics. The results of this research are expected to contribute to a better understanding of population growth trends in East Java and provide a basis for more informed decision-making in future regional development planning and social policy.*

**Keywords:** Population Prediction, East Java, LSTM

## 1. Pendahuluan

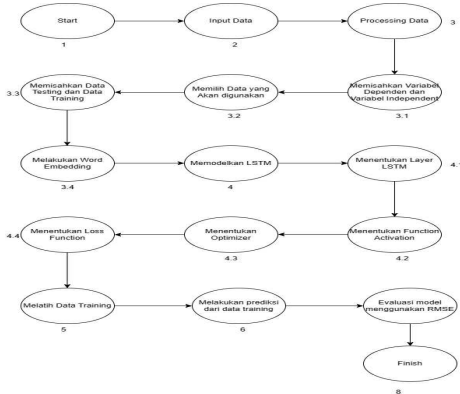
Pertumbuhan penduduk adalah perubahan suatu populasi dari waktu ke waktu dan dapat dihitung sebagai perubahan jumlah individu yang diukur [1]. Pertumbuhan penduduk sebenarnya ditentukan oleh keseimbangan yang dinamis antara dua kekuatan yang dapat menambah ataupun mengurangi jumlah penduduk, baik itu oleh jumlah bayi yang dilahirkan, sekaligus dikurangi oleh jumlah kematian yang dapat terjadi pada semua kelompok umur [2]. Dengan jumlah penduduk 248,8 juta jiwa berdasarkan data dari Badan Pusat Statistik pada tahun 2013 menjadikan Indonesia sebagai negara yang memiliki penduduk terbesar keempat di dunia [3]. Wilayah Jawa Timur, salah satu provinsi terpadat di Indonesia, menunjukkan pertumbuhan penduduk yang cepat, menjadikannya fokus utama dalam penelitian ini. Pertumbuhan populasi yang cepat di Jawa Timur mendorong perlunya pengembangan model prediksi yang dapat membantu pemerintah dan pemangku kepentingan lainnya dalam merencanakan infrastruktur, layanan publik, dan kebijakan sosial yang efektif. Pada saat yang sama, kemajuan dalam bidang kecerdasan buatan, khususnya dalam jaringan saraf tiruan, telah membuka peluang baru dalam memprediksi tren demografis. LSTM merupakan salah satu pengembangan dari metode RNN (Recurrent Neural Network) yang sangat baik untuk menganalisa data deret waktu dan juga memiliki kestabilan yang bagus untuk mempelajari ketergantungan dalam jangka waktu yang panjang, hal ini dikarenakan metode LSTM ini sendiri didukung dengan strukturnya yang terdiri dari tiga gerbang utama yakni : gerbang masukan, gerbang lupa dan gerbang keluaran [4]. Gerbang lupa digunakan untuk mengontrol sejauh mana nilai tetap berada di sel memori, gerbang masukan untuk mengontrol berapa banyak informasi yang harus disimpan, dan gerbang keluaran untuk menentukan nilai yang digunakan untuk menghitung keluaran dari hasil penelitian [5]. Dalam konteks demografi, khususnya dalam pemodelan pertumbuhan penduduk, LSTM menjanjikan pendekatan yang kuat. Karena arsitektur LSTM dapat menangani penyimpanan memori untuk jangka waktu yang lama yang disebabkan oleh faktor sel memori yang diperbesar untuk menangani terjadinya gradien yang hilang di RNN saat memproses data kontinu yang panjang [6]. Penelitian ini bertujuan untuk mengembangkan dan menguji model prediksi



menggunakan metode LSTM untuk memproyeksikan pertumbuhan penduduk di Jawa Timur. Data historis jumlah penduduk dari periode sebelumnya akan digunakan sebagai masukan untuk melatih model LSTM. Pendekatan ini diharapkan dapat menghasilkan prediksi yang akurat terkait pertumbuhan penduduk di wilayah tersebut. Selain memberikan kontribusi pada pemahaman kita tentang tren pertumbuhan penduduk di Jawa Timur, penelitian ini juga memiliki implikasi praktis yang signifikan. Hasilnya dapat digunakan oleh pemerintah daerah, lembaga penelitian, dan organisasi non-pemerintah dalam perencanaan pembangunan wilayah, alokasi sumber daya, dan pengambilan keputusan kebijakan. Dengan memiliki prediksi yang lebih akurat tentang pertumbuhan penduduk, pihak berkepentingan dapat mengantisipasi kebutuhan infrastruktur, layanan kesehatan, pendidikan, dan sosial lainnya, serta mengelola sumber daya secara efisien. Penelitian ini akan terstruktur dalam beberapa tahap. Tahap pertama adalah pengumpulan data dari sumber data terpercaya seperti Badan Pusat Statistik dan pusat data lainnya. Berikutnya melakukan pre-processing data, dimana data historis tentang jumlah penduduk akan disiapkan untuk analisis. Lalu setelah itu, kami melakukan word embedding untuk mengubah data format teks menjadi data format numerik. Langkah selanjutnya adalah pembangunan model LSTM, di mana data yang telah diproses akan digunakan untuk melatih model guna menghasilkan prediksi pertumbuhan penduduk. Terakhir, kami melakukan evaluasi performa model akan dilakukan menggunakan metrik yang relevan, seperti Root Mean Squared Error (RMSE). Metode evaluasi performa model berguna untuk mengukur akurasi prediksi, prediksi yang baik adalah dengan memilih metode yang memberikan prediksi dengan tingkat keakuratan yang besar dan tingkat kesalahan yang kecil, keakuratan model prediksi tidak terlepas dari faktor-faktor pengukuran akurasi prediksi [7]. RMSE adalah ukuran tingkat kesalahan untuk menentukan keakuratan hasil prediksi, semakin kecil nilai dari RMSE (mendekati 0), maka hasil prediksi akan semakin akurat [8]. Diharapkan bahwa temuan dari penelitian ini akan memberikan wawasan yang berharga bagi pemangku kepentingan dan mendukung pengambilan keputusan yang lebih baik dalam perencanaan dan pengelolaan pembangunan wilayah di Jawa Timur.

## 2. Metode Penelitian

Metode penelitian menurut Sugiono (2008), merupakan suatu cara ilmiah untuk mendapatkan data dengan tujuan dan fungsionalitas tertentu [9]. Berikut alur metode yang kami lakukan dalam mengerjakan penelitian kami untuk memprediksi kenaikan penduduk Jawa Timur:



Gambar 1. Alur Program

## 2.1 Melakukan Pengumpulan Data

Langkah pertama kelompok kami adalah dengan melakukan pengumpulan data atau dataset untuk melakukan metode penelitian. Kumpulan data atau dataset sendiri merupakan objek-objek yang mewakili data dan hubungannya dalam memori, strukturnya terlihat seperti dengan data yang ada dalam database, dataset berisi sekumpulan data dan catatan tabel [10]. Kami menggunakan data yang bersumber dari Badan Pusat Statistik Indonesia, yaitu jumlah penduduk menurut jenis kelamin dari berbagai Kabupaten dan Kota yang ada di Provinsi Jawa Timur.

## 2.2 Melakukan Preprocessing Data

Setelah melakukan pengumpulan data, langkah selanjutnya adalah melakukan preprocessing data. Preprocessing data merupakan suatu bentuk proses dalam mempersiapkan sekumpulan data yang nantinya akan digunakan dalam penelitian. Preprocessing data bertujuan untuk membersihkan data yang tidak perlu, menormalisasikan data yang ada, dan mempersiapkan data supaya dapat dikelola secara efisien oleh metode penelitian yang digunakan [11]. Memproses awal data mencakup langkah-langkah berikut.

### a. Memisahkan Variabel Independen dengan Variabel Dependen

Variabel Independen merupakan variabel yang menyebabkan terjadinya atau menjadi faktor terpengaruhnya variabel yang terikat [12]. Sedangkan Variabel Dependen adalah variabel yang menjadi fokus utama penelitian dan dapat dipengaruhi oleh variabel bebas [13].

### b. Memilih Data yang Akan Digunakan Dalam Penelitian

Memilih dan memilih data mana saja yang diperlukan dalam penelitian, karena memang tidak semua data benar benar dibutuhkan, hanya data tertentu saja yang penting untuk dipakai selama melakukan penelitian, sehingga proses pengolahan data akan jauh lebih efisien dan menghasilkan hasil penelitian yang lebih akurat sesuai dengan tujuan dan manfaat dari penelitian yang sedang dikerjakan oleh kelompok kami.

### c. Memisahkan Data Training dan Data Testing

Memisahkan antara data data dari BPS yang akan digunakan untuk training menggunakan metode LSTM dengan data dari BPS yang akan digunakan untuk testing atau untuk prediksi. Hal ini untuk memudahkan dalam memprogram data sesuai dengan data yang akan dilakukan training dengan data yang akan dilakukan testing.

### d. Melakukan Word Embedding

Setelah melakukan preprocessing data, selanjutnya kami melakukan proses word embedding, dimana kami mengubah format data berupa teks menjadi berupa format numerik. Pada tahun 2003, Bengio dkk, memperkenalkan istilah word embedding. Word embedding sendiri merupakan suatu fungsi parameter yang memetakan setiap teks ke vektor numerik berdimensi tinggi [14].

## 2.3 Memodelkan LSTM

Pemodelan Long Short Term Memory (LSTM) merupakan pendekatan yang cukup akurat dalam teori jaringan syaraf tiruan (JST) terkhususnya dalam metode jaringan saraf berulang atau biasa disebut dengan Recurrent Neural Network (RNN) [15].

Berikut beberapa langkah yang kami lakukan dalam membuat pemodelan LSTM dari program kami :

**a. Menentukan Layer LSTM**

Layer pada metode LSTM ini merupakan sebuah komponen yang memiliki fungsi mengolah data sekuensial dan mempertahankan informasi dalam jangka panjang. Dimana pemodelan kami ini menggunakan enam belas layer secara keseluruhan.

**b. Menentukan Function Activation**

Function activation merupakan sebuah fungsi yang digunakan untuk mengubah sinyal masukan menjadi sinyal keluaran yang akan menjadi sebuah masukan untuk layer selanjutnya [16]. Pada penelitian ini kami menggunakan function activation ReLU, dimana function ini sendiri merupakan function activation yang sederhana namun efektif. Pada function ReLU ini sendiri mengubah semua nilai negatif menjadi nol dan mempertahankan nilai positif. Adapun kelebihan function ini sendiri yakni mampu mengatasi masalah vanishing gradient [17].

**c. Menentukan Function Optimizer**

Function optimizer merupakan sebuah algoritma yang digunakan untuk mengurangi bias atau bobot atau lebih besarnya merubah atribut jaringan dalam neural network. Pada penelitian ini kami menggunakan function activation Adam (Adaptive Moment Estimation). Dimana Adam ini sendiri menghitung kecepatan pembelajaran adaptif untuk setiap parameter dengan memanfaatkan momen pertama dan momen kedua dari gradien, maksudnya ialah nilai rata-rata dari kuadrat Gradient [18].

**d. Menentukan Loss Function**

Loss function merupakan sebuah komponen penting dalam pemodelan LSTM dimana function ini berfungsi mengetes seberapa akurat pemodelan pada penelitian. Kami menggunakan Mean Squared Error (MSE) sebagai loss function pada program kami. MSE (Mean Squared Error) digunakan karena menghasilkan error yang sedang dan biasanya lebih disukai untuk prediksi yang menghasilkan error kecil, namun terkadang menghasilkan error yang sangat besar [19]. Berikut detail pemodelan LSTM dari program kami :

Parameter	Value	Parameter	Value
<b>Layer 1 LSTM</b>	-	<b>Layer 6 LSTM</b>	-
Jalur LSTM	50	Jalur LSTM	45
Function Activation	ReLU	Function Activation	ReLU
Bentuk input	(1,1) Input satu dimensi	Return Sequences	True
Return Sequences	True	<b>Layer 7 LSTM</b>	-
<b>Layer 2 LSTM</b>	-	Jalur LSTM	32
Jalur LSTM	45	Function Activation	ReLU
Function Activation	ReLU	Return Sequences	True
Return Sequences	True	<b>Layer 8 LSTM</b>	-

Parameter	Value	Parameter	Value
<b>Layer 3 LSTM</b>	-	Jalur LSTM	28
Jalur LSTM	20	Function Activation	ReLU
Function Activation	ReLU	Return Sequences	True
Return Sequences	True	<b>Layer 9 LSTM</b>	-
<b>Layer 4 LSTM</b>	-	Jalur LSTM	15
Jalur LSTM	30	Function Activation	ReLU
Function Activation	ReLU	Return Sequences	-
Return Sequences	True	Layer Dense I	50
<b>Layer 5 LSTM</b>	-	Layer Dense II	45
Jalur LSTM	25	Layer Dense III	30
Function Activation	ReLU	Layer Dense IV	6
Return Sequences	True	Function Optimize	Adam Optimizer
		Loss Function	MSE

#### 2.4 Melatih Data Training

Setelah pada processing data melakukan pemisahan antara data training dan data testing, selanjutnya kita melakukan training terhadap pemodelan yang telah dibuat sebelumnya. Maksud dari melatih data training agar pemodelan yang telah dibuat sebelumnya dapat melakukan pemrosesan data sesuai dengan inputan yakni berupa data training yang telah dipisah sebelumnya dengan data testing.

#### 2.5 Melakukan Prediksi dari Data Training

Setelah melakukan training dari pemodelan yang telah dibuat sebelumnya menggunakan data training, selanjutnya kami melakukan prediksi untuk mengetahui berapa jumlah penduduk Jawa Timur pada tahun 2023. Dimana jumlah penduduk di tahun 2023 ini merupakan data testing dari program yang diharapkan menjadi patokan dari hasil prediksi dari data training selanjutnya.

#### 2.6 Evaluasi Menggunakan Metode RMSE

RMSE digunakan sebagai fitur untuk membedakan nilai prediksi dan nilai aktual. Semakin tinggi nilai RMSE yang dihasilkan maka semakin rendah akurasi, dan semakin rendah nilai RMSE yang dihasilkan maka semakin tinggi akurasi [20].

### 3. Hasil dan Pembahasan

#### 3.1. Melakukan Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data yang berasal dari BPS Provinsi Jawa Timur, yakni berupa data Jumlah Penduduk Menurut Jenis Kelamin dan Kabupaten Kota Provinsi Jawa Timur. Dalam dataset tersebut terdapat 19 label dan 38 data. Berikut dataset yang digunakan dalam penelitian ini akan ditampilkan pada gambar 2:

Kabupaten/Kota Se Jawa Timur	Penduduk Menurut Jenis Kelamin dan Kabupaten/Kota Provinsi Jawa Timur (Jawa)										
	Unamed: 2	Unamed: 3	Unamed: 4	Unamed: 5	Unamed: 6	Unamed: 7	Unamed: 8	Unamed: 9	Unamed: 10		
NaN	Laki-laki	NaN	NaN	NaN	NaN	NaN	Perempuan	NaN	NaN	NaN	
	NaN	2018	2019.0	2020.0	2021.0	2022.0	2023.0	2018	2019.0	2020.0	2021.0
Kabupaten Pacitan	270708	271113.0	293714.0	295163.0	297014.0	298826.0	283686	284191.0	292396.0	293945.0	
Kabupaten Ponorogo	435169	435818.0	474257.0	477428.0	481540.0	485603.0	435538	435752.0	475061.0	478411.0	
Kabupaten Trenggalek	345282	346008.0	367372.0	369196.0	371527.0	373810.0	349620	350287.0	363753.0	365682.0	
Kabupaten Tulungagung	504804	506702.0	545254.0	548563.0	552835.0	557045.0	530486	532582.0	544521.0	548025.0	

Gambar 2. [Dataset Dari BPS](#)

### 3.2. Melakukan Preprocessing Data

Data di atas kemudian akan diproses agar menjadi sebuah data yang sesuai dengan metode LSTM yang digunakan dalam penelitian ini. Dalam processing data ini sendiri terdapat beberapa tahapan yang perlu dilakukan agar data mentah sebelumnya yang berasal dari BPS Provinsi Jawa Timur dapat digunakan dalam pemodelan LSTM yang digunakan dalam penelitian ini. Tahapannya antara lain sebagai berikut :

#### a. Memisahkan Variabel Independen dengan Variabel Dependen

Variabel Independen dalam penelitian ini berupa data Kabupaten Kota di Provinsi Jawa Timur. Dimana dari data kabupaten kota akan mempengaruhi data kependudukan di setiap kabupaten kotanya. Variabel dependen dalam penelitian ini berupa data kependudukan masing-masing kabupaten kota di Provinsi Jawa Timur dalam rentang periode 2001-2023. data kependudukan ini dependen atau terikat dengan data Kabupaten Kota yang telah dibuat sebelumnya.

#### b. Memilih Data yang Akan Digunakan Dalam Penelitian

Tidak semua data yang telah diproses sebelumnya akan digunakan dalam penelitian ini, data yang digunakan dalam penelitian ini yakni data kependudukan secara menyeluruh untuk satu Provinsi Jawa Timur. Jadi data Tersebut merupakan data penjumlahan masing-masing kabupaten atau kota yang terdapat di Jawa Timur. Adapun alasan mengapa menggunakan data tersebut karena data tersebutlah yang relevan dengan judul penelitian yang kami lakukan yakni memprediksi jumlah kependudukan untuk wilayah Provinsi Jawa Timur.

#### c. Memisahkan Data Training dan Data Testing

Dari data yang sudah terpilih sebelumnya sebagai data yang akan digunakan dalam penelitian ini selanjutnya kami memisahkan data tersebut menjadi data training dan data testing. Untuk meyakinkan metode ini sendiri kami memisahkan menjadi dua tahapan, yakni uji coba yang pertama dimana kami memisahkan data kependudukan untuk di provinsi Jawa Timur dari tahun 2001-2022 menjadi data training untuk pengujian yang pertama dan data kependudukan Provinsi Jawa Timur di tahun 2023 sebagai data testing. Kemudian di pengujian kedua untuk data training yang digunakan yakni data kependudukan provinsi Jawa Timur dari tahun 2001 sampai 2021 dan data kependudukan Provinsi Jawa Timur di tahun 2022 sebagai data testing.

### 3.3 Melakukan Word Embedding

Kelompok kami juga perlu melakukan word embedding terhadap data pada tahun 2018 dikarenakan dari excel milik Badan Pusat Statistik Jawa Timur disediakan data berformat teks atau string, sehingga dalam program prediksi buatan kelompok kami, perlu diubah dalam format

numerik atau float agar dapat dilakukan proses penghitungan prediksi yang akurat dan sesuai. Berikut data 2018 sebelum di Embedding:

2	270708
3	435169
4	345282
5	504804
6	579925
7	787023
8	1302771
9	507781

**Gambar 3.** Data Tahun 2018 Sebelum di Embedding

Berikut data 2018 setelah di Embedding:

2	283686.0
3	435536.0
4	349620.0
5	530486.0
6	577575.0
7	781090.0
8	1289024.0
9	532013.0

**Gambar 4.** Data Tahun 2018 Setelah di Embedding

### 3.4 Pemodelan Menggunakan LSTM

Dari pemodelan LSTM yang telah dibuat sebelumnya mendapatkan hasil yang cukup bagus dalam pemodelan ini. Dimana dari dua uji coba yang telah dilakukan mendapatkan hasil sebagai berikut:

Uji coba 1 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2022) data testing (data kependudukan Provinsi Jawa Timur tahun 2023).

Prediksi jumlah penduduk di Provinsi Jawa Timur untuk tahun 2023: 41416408.0

**Gambar 5.** Hasil Prediksi Untuk Uji Coba 1

Uji coba 2 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2021) data testing (data kependudukan Provinsi Jawa Timur tahun 2022).

Prediksi jumlah penduduk di Provinsi Jawa Timur untuk tahun 2022: 41149968.0

**Gambar 6.** Hasil Prediksi Untuk Uji Coba 2

Pada penelitian ini kami juga menggunakan optimizer yang lain yakni nadam sebagai perbandingan pemodelan mana yang menghasilkan prediksi paling baik dalam studi kasus ini dan berikut hasil uji cobanya:

Uji coba 3 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2022) data testing (data kependudukan Provinsi Jawa Timur tahun 2023).

Prediksi jumlah penduduk di Provinsi Jawa Timur untuk tahun 2023: 75240136.0

**Gambar 7.** Hasil Prediksi Untuk Uji Coba 3

Uji coba 4 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2021) data testing (data kependudukan Provinsi Jawa Timur tahun 2022).

Prediksi jumlah penduduk di Jawa Timur untuk tahun 2022: 77469376.0

**Gambar 8.** Hasil Prediksi Untuk Uji Coba 4

Dari Hasil uji coba yang sudah dilakukan dapat ditemukan hasil bahwa function optimizer adam lah yang cocok dengan studi kasus kami ini.

### 3.5 Evaluasi Menggunakan Metode RMSE

Root Mean Squared Error (RMSE) merupakan salah satu metode matriks evaluasi yang merupakan pengembangan dari MSE yang telah digunakan dalam pemodelan LSTM sebelumnya. Adapun penggunaan metode RMSE ini kami gunakan sebagai final checking bagi pemodelan LSTM yang telah dibuat sebelumnya. Berikut Hasil RMSE dari Sistem Prediksi Kami:

Uji coba 1 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2022) data testing (data kependudukan Provinsi Jawa Timur tahun 2023).

**RMSE : 1.0**

**Gambar 9.** Hasil RMSE Uji Coba 1

Uji coba 2 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2021) data testing (data kependudukan Provinsi Jawa Timur tahun 2022).

**RMSE : 6.0**

**Gambar 10.** Hasil RMSE Uji Coba 2

Untuk uji coba yang menggunakan optimizer nadam dalam pemodelannya mendapatkan hasil RMSE sebagai berikut :

Uji coba 3 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2022) data testing (data kependudukan Provinsi Jawa Timur tahun 2023).

**RMSE : 33823729.0**

**Gambar 11.** Hasil RMSE Uji Coba 3

Uji coba 4 : data training (data kependudukan Provinsi Jawa Timur tahun 2001-2021) data testing (data kependudukan Provinsi Jawa Timur tahun 2022).

**RMSE : 36319402.0**

**Gambar 12.** Hasil RMSE Uji Coba 4

#### 4. Kesimpulan

Penelitian ini menghasilkan model prediksi kenaikan penduduk Jawa Timur menggunakan metode Long Short Term Memory (LSTM), yang menggabungkan prinsip-prinsip jaringan saraf tiruan dengan kemampuan untuk menangani data sekuensial. Melalui proses pre-processing data, pembangunan model LSTM, pelatihan model, dan evaluasi performa model, berbagai temuan penting telah dihasilkan yang dapat memberikan wawasan yang berharga bagi pemangku kepentingan dalam perencanaan pembangunan wilayah dan pengambilan keputusan kebijakan. Hasil analisis penelitian kami menunjukkan bahwa model LSTM mampu menghasilkan prediksi yang cukup akurat tentang kenaikan penduduk di Jawa Timur. Dengan menggunakan data historis jumlah penduduk sebagai masukan, model mampu menangkap pola-pola kompleks dalam data sekuensial dan memprediksi tren pertumbuhan penduduk dengan tingkat kesalahan yang relatif rendah. Evaluasi performa model menggunakan metrik Root Mean Squared Error (RMSE) menunjukkan bahwa model memiliki tingkat akurasi yang memadai, meskipun masih ada ruang untuk perbaikan. Pembahasan atas hasil analisis menunjukkan bahwa model LSTM dapat menjadi alat yang berguna dalam mendukung perencanaan pembangunan wilayah di Jawa Timur. Prediksi yang dihasilkan oleh model dapat digunakan oleh pemerintah daerah, lembaga penelitian, dan organisasi non-pemerintah dalam merencanakan infrastruktur, alokasi sumber daya, dan pengambilan keputusan kebijakan yang lebih efektif. Dengan memiliki gambaran yang lebih akurat tentang pertumbuhan penduduk di masa depan, pemangku kepentingan dapat mengantisipasi kebutuhan masyarakat dan mengalokasikan sumber daya dengan lebih efisien. Kesimpulan umum dari penelitian ini adalah bahwa penggunaan metode LSTM dalam memprediksi kenaikan penduduk di Jawa Timur memiliki potensi besar untuk mendukung perencanaan pembangunan wilayah dan pengambilan keputusan kebijakan. Meskipun masih ada beberapa batasan dan ruang untuk perbaikan, model LSTM telah terbukti efektif dalam menangkap pola-pola dalam data sekuensial dan memprediksi tren pertumbuhan penduduk dengan tingkat akurasi yang memadai.

#### Daftar Pustaka

- [1] N. C. Putri, N. Nurwati, "Pengaruh Laju Pertumbuhan Penduduk Berdampak Pada Tingginya Angka Kemiskinan Yang Menyebabkan Banyak Eksploitasi Anak Di Indonesia", *Jurnal Ilmu Kesejahteraan Sosial HUMANITAS*, vol. 3, no. 1, pp.1-15, March 2021.
- [2] E. Rochaida, "Dampak Pertumbuhan Penduduk Terhadap Pertumbuhan Ekonomi Dan Keluarga Sejahtera", *Jurnal Ekonomi Manajemen Dan Akuntansi*, vol. 18, no. 2, pp. 14-24, March 2016.
- [3] S. Devi, A. Fatchiya, D. Susanto, "Kapasitas Kader dalam Penyuluhan Keluarga Berencana di Kota Palembang Provinsi Sumatera Selatan", *Jurnal Penyuluhan*, vol. 12, no. 2, pp. 144-156, March 2016.
- [4] A. P. Sari, H. Suzuki, T. Kitajima, T. Yasuno, D. A. Prasetya, and R. Arifuddin, "Short-Term Wind Speed and Direction Forecasting by 3DCNN and Deep Convolutional LSTM," *IEEEJ Trans. Elec. Electron. Eng.*, vol. 17, pp. 1620-1628, 2022.
- [5] L. Wiranda, M. Sadikin, "Penerapan Long Short Term Memory Pada Data Tme Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma", *Jurnal Nasional Pendidikan Teknik Informatika*, vol. 8, no. 3, pp. 184-196, December 2019.
- [6] D. I. Puteri, "Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah", *Jurnal Ilmiah Matematika Sains dan Teknologi*, vol. 11, no. 1, pp. 35-43, June 2023.
- [7] L. Aryani, Fatmasari, Afriyudi, N. Hadinata, "Prediksi Jumlah Siswa Baru Dengan Menggunakan Metode Exponential Smoothing (Studi Kasus : Smk Ethika Palembang)", *Bina Darma Conference on Computer Science*, vol. 2, no. 3, pp. 237-244, June 2021.
- [8] I. Suprayogi, Trimajion, Mahyudin, "Model Prediksi Liku Kalibrasi Menggunakan Pendekatan Jaringan Saraf Tiruan (JST) (Studi Kasus: Sub DAS Siak Hulu)", *Jurnal Online Mahasiswa Fakultas Teknik*, vol. 1, no. 1, pp. 1-18, February 2014.
- [9] W. P. Han, Erviriani, A. R. A. Siregar, Ivone, Lisa, "Analisis Kepuasan Kerja Dan Lingkungan Kerja Karyawan Di Sekolah En Mandarin – Medan", *Jurnal Ilmiah Maksitek*, vol. 4, no. 2, pp. 11-30, June 2019.
- [10] Suhartini, L. K. Wijaya, N. A. Pratiwi, "Algoritma K-Means Untuk Pendataan Obat



- Berdasarkan Laporan Bulanan Pada Dinas Kesehatan Kabupaten Lombok Timur”, Jurnal Informatika dan Teknologi, vol. 3, no. 2, pp. 147-156, July 2020.
- [11] F. Putra, H. F. Tahiyat, R. M. Ihsan, Rahmaddeni, L. Efrizoni, “Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia”, Indonesian Journal of Machine Learning and Computer Science, vol. 4, no. 1, pp. 273-281, January 2024.
- [12] C. Christalisana, “Pengaruh Pengalaman Dan Karakter Sumber Daya Manusia Konsultan Manajemen Konstruksi Terhadap Kualitas Pekerjaan Pada Proyek Di Kabupaten Pandeglang”, Jurnal Fondasi, vol. 7, no. 1, pp. 87-98, April 2018.
- [13] T. S. Darmawan, S. Sutopo, “Pengaruh Persepsi Tentang Harga, Promosi, Dan Kualitas Pelayanan Terhadap Volume Penjualan Paket Wisata Karimunjawa Di Biro Tour Dan Travel Karimunjawa Beach Adventure”, Diponegoro Journal Of Management, vol. 4, no. 2, pp. 1-11, May 2015.
- [14] A. Nurdin, B. A. S. Aji, A. Bustamin, Z. Abidin, “Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks”, Jurnal TEKNOKOMPAK, vol. 14, no. 2, pp. 74-79, August 2020.
- [15] A. Rahmawati, W. Sulandari, S. Subanti, Y. Yudhanto, “Penerapan Metode Recurrent Neural Network dengan Pendekatan Long Short-Term Memory (LSTM) untuk Meramalkan Harga Saham Hybe Corporation”, Jurnal Bumigora Information Technology (BITe), vol. 5, no. 1, pp. 65-76, June 2023.
- [16] Siddharth Sharma, Simone Sharma, Anidhya Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS”, International Journal of Engineering Applied Sciences and Technology, Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316, April 2020
- [17] Mesran, S. R. Yahya, F. Nugroho, A. P. Windarto, “Investigating the Impact of ReLU and Sigmoid Activation Functions on Animal Classification Using CNN Models”, Jurnal Rekayasa Sistem Dan Teknologi Informasi, vol. 8, no. 1, pp. 111-118, February 2024.
- [18] M. S. Haqqi, B. Kusumoputro, “Komparasi Metode Optimasi Adam dan SGD dalam Skema Direct Inverse Control untuk Sistem Kendali Data Sikap dan Ketinggian Quadcopter”, Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 10, no. 2, pp. 458-469, April 2022.
- [19] M. A. Maricar, “Analisa Perbandingan Nilai Akurasi Moving Average dan Exponential Smoothing untuk Sistem Peramalan Pendapatan pada Perusahaan XYZ”, Jurnal Sistem Dan Informatika, vol. 13, no. 2, pp. 36-45, May 2019.
- [20] S. Sautomo, H. F. Pardede, “Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short Term Memory (LSTM)”, Jurnal Rekayasa Sistem Dan Teknologi Informasi, vol. 5, no. 1, pp. 99-106, February 2021.

Kabupaten/Kota Se Jawa Timur	Penduduk Menurut Jenis Kelamin dan Kabupaten/Kota Provinsi Jawa Timur (Jawa)										
	Unamed: 2	Unamed: 3	Unamed: 4	Unamed: 5	Unamed: 6	Unamed: 7	Unamed: 8	Unamed: 9	Unamed: 10		
NaN	Laki-laki	NaN	NaN	NaN	NaN	NaN	Perempuan	NaN	NaN	NaN	
	NaN	2018	2019.0	2020.0	2021.0	2022.0	2023.0	2018	2019.0	2020.0	2021.0
Kabupaten Pacitan	270708	271113.0	293714.0	295163.0	297014.0	298826.0	283686	284191.0	292396.0	293945.0	
Kabupaten Ponorogo	435169	435818.0	474257.0	477428.0	481540.0	485603.0	435538	435752.0	475061.0	478411.0	
Kabupaten Trenggalek	345282	346008.0	367372.0	369196.0	371527.0	373810.0	349620	350287.0	363753.0	365682.0	
Kabupaten Tulungagung	504804	506702.0	545254.0	548563.0	552835.0	557045.0	530486	532582.0	544521.0	548025.0	

Gambar 2. [Dataset Dari BPS](#)

### 3.2. Melakukan Preprocessing Data

Data di atas kemudian akan diproses agar menjadi sebuah data yang sesuai dengan metode LSTM yang digunakan dalam penelitian ini. Dalam processing data ini sendiri terdapat beberapa tahapan yang perlu dilakukan agar data mentah sebelumnya yang berasal dari BPS Provinsi Jawa Timur dapat digunakan dalam pemodelan LSTM yang digunakan dalam penelitian ini. Tahapannya antara lain sebagai berikut :

#### a. Memisahkan Variabel Independen dengan Variabel Dependen

Variabel Independen dalam penelitian ini berupa data Kabupaten Kota di Provinsi Jawa Timur. Dimana dari data kabupaten kota akan mempengaruhi data kependudukan di setiap kabupaten kotanya. Variabel dependen dalam penelitian ini berupa data kependudukan masing-masing kabupaten kota di Provinsi Jawa Timur dalam rentang periode 2001-2023. data kependudukan ini dependen atau terikat dengan data Kabupaten Kota yang telah dibuat sebelumnya.

#### b. Memilih Data yang Akan Digunakan Dalam Penelitian

Tidak semua data yang telah diproses sebelumnya akan digunakan dalam penelitian ini, data yang digunakan dalam penelitian ini yakni data kependudukan secara menyeluruh untuk satu Provinsi Jawa Timur. Jadi data tersebut merupakan data penjumlahan masing-masing kabupaten atau kota yang terdapat di Jawa Timur. Adapun alasan mengapa menggunakan data tersebut karena data tersebutlah yang relevan dengan judul penelitian yang kami lakukan yakni memprediksi jumlah kependudukan untuk wilayah Provinsi Jawa Timur.

#### c. Memisahkan Data Training dan Data Testing

Dari data yang sudah terpilih sebelumnya sebagai data yang akan digunakan dalam penelitian ini selanjutnya kami memisahkan data tersebut menjadi data training dan data testing. Untuk meyakinkan metode ini sendiri kami memisahkan menjadi dua tahapan, yakni uji coba yang pertama dimana kami memisahkan data kependudukan untuk di provinsi Jawa Timur dari tahun 2001-2022 menjadi data training untuk pengujian yang pertama dan data kependudukan Provinsi Jawa Timur di tahun 2023 sebagai data testing. Kemudian di pengujian kedua untuk data training yang digunakan yakni data kependudukan provinsi Jawa Timur dari tahun 2001 sampai 2021 dan data kependudukan Provinsi Jawa Timur di tahun 2022 sebagai data testing.

### 3.3 Melakukan Word Embedding

Kelompok kami juga perlu melakukan word embedding terhadap data pada tahun 2018 dikarenakan dari excel milik Badan Pusat Statistik Jawa Timur disediakan data berformat teks atau string, sehingga dalam program prediksi buatan kelompok kami, perlu diubah dalam format

# Prediksi Kenaikan Penduduk Jawa Timur Menggunakan Metode Long Short Term Memory

Atiqur Rozi<sup>a1</sup>, Muhammad Rohman Irsyadi<sup>a2</sup>, Sandy Nicholas<sup>a3</sup>, Anggraini Puspita Sari<sup>a4</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Jawa Timur, Indonesia

<sup>1</sup>22081010010@student.upnjatim.ac.id

<sup>2</sup>22081010024@student.upnjatim.ac.id

<sup>3</sup>22081010237@student.upnjatim.ac.id

<sup>4</sup>anggraini.puspita.if@upnjatim.ac.id

## Abstract

*This research aims to develop a prediction model for population increase in East Java using the Long Short Term Memory (LSTM) method. Historical population data from the previous period will be used as input to train the LSTM model. This approach is expected to produce accurate predictions about population growth in the East Java region. The LSTM method was chosen due to its ability to handle sequential data and long-term memory, which is in line with the characteristics of demographic data. This research will involve data pre-processing, LSTM model building, and model performance evaluation using relevant metrics. The results of this research are expected to contribute to a better understanding of population growth trends in East Java and provide a basis for more informed decision-making in future regional development planning and social policy.*

**Keywords:** Population Prediction, East Java, LSTM

## 1. Pendahuluan

Pertumbuhan penduduk adalah perubahan suatu populasi dari waktu ke waktu dan dapat dihitung sebagai perubahan jumlah individu yang diukur [1]. Pertumbuhan penduduk sebenarnya ditentukan oleh keseimbangan yang dinamis antara dua kekuatan yang dapat menambah ataupun mengurangi jumlah penduduk, baik itu oleh jumlah bayi yang dilahirkan, sekaligus dikurangi oleh jumlah kematian yang dapat terjadi pada semua kelompok umur [2]. Dengan jumlah penduduk 248,8 juta jiwa berdasarkan data dari Badan Pusat Statistik pada tahun 2013 menjadikan Indonesia sebagai negara yang memiliki penduduk terbesar keempat di dunia [3]. Wilayah Jawa Timur, salah satu provinsi terpadat di Indonesia, menunjukkan pertumbuhan penduduk yang cepat, menjadikannya fokus utama dalam penelitian ini. Pertumbuhan populasi yang cepat di Jawa Timur mendorong perlunya pengembangan model prediksi yang dapat membantu pemerintah dan pemangku kepentingan lainnya dalam merencanakan infrastruktur, layanan publik, dan kebijakan sosial yang efektif. Pada saat yang sama, kemajuan dalam bidang kecerdasan buatan, khususnya dalam jaringan saraf tiruan, telah membuka peluang baru dalam memprediksi tren demografis. LSTM merupakan salah satu pengembangan dari metode RNN (Recurrent Neural Network) yang sangat baik untuk menganalisa data deret waktu dan juga memiliki kestabilan yang bagus untuk mempelajari ketergantungan dalam jangka waktu yang panjang, hal ini dikarenakan metode LSTM ini sendiri didukung dengan strukturnya yang terdiri dari tiga gerbang utama yakni : gerbang masukan, gerbang lupa dan gerbang keluaran [4]. Gerbang lupa digunakan untuk mengontrol sejauh mana nilai tetap berada di sel memori, gerbang masukan untuk mengontrol berapa banyak informasi yang harus disimpan, dan gerbang keluaran untuk menentukan nilai yang digunakan untuk menghitung keluaran dari hasil penelitian [5]. Dalam konteks demografi, khususnya dalam pemodelan pertumbuhan penduduk, LSTM menjanjikan pendekatan yang kuat. Karena arsitektur LSTM dapat menangani penyimpanan memori untuk jangka waktu yang lama yang disebabkan oleh faktor sel memori yang diperbesar untuk menangani terjadinya gradien yang hilang di RNN saat memproses data kontinu yang panjang [6]. Penelitian ini bertujuan untuk mengembangkan dan menguji model prediksi

## Deteksi Objek pada Citra Menggunakan Model YOLO

Intara Pratama Harahap<sup>a1</sup>, Agus Muliantara<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

<sup>1</sup>redintara123@gmail.com

<sup>2</sup>muliantara@unud.ac.id

### Abstract

*Object detection is a crucial task in the field of computer vision and digital image processing, with numerous practical applications. This paper focuses on the implementation of the You Only Look Once (YOLO) model, a deep learning-based approach for object detection. The YOLO model offers several advantages over previous methods, such as simultaneous prediction of bounding boxes and object class probabilities, a relatively simple Convolutional Neural Network (CNN) architecture, and high computational speed, making it suitable for real-time applications. The study utilizes a dataset of 770 images, with 524 for training, 136 for validation, and 110 for testing, specifically focused on detecting various pet animals. The training process involves annotation of the image data, followed by training and validation of the YOLO model. The results demonstrate the model's ability to effectively detect and classify objects, achieving high performance metrics such as precision, recall, and mean Average Precision (mAP) nearing 0.8 towards the end of the training process. Additionally, a confusion matrix is presented, highlighting the model's accuracy in classifying different classes, with the highest accuracy for the 'Cat' class at 95%. The paper concludes by discussing the model's performance and potential areas for improvement.*

**Keywords:** YOLO, You Only Look Once, Citra, Object Detection,

### 1. Pendahuluan

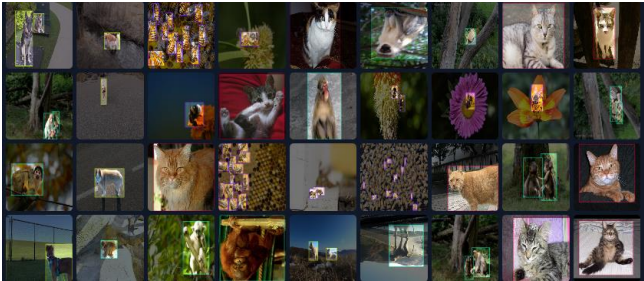
Artificial Intelligence (AI) atau Kecerdasan Buatan adalah ilmu yang dikembangkan dengan kombinasi banyak subyek. Dalam istilah sederhana, kecerdasan buatan adalah memberi mesin dengan kecerdasan manusia, mensimulasikan pemikiran manusia sehingga membantu orang memecahkan masalah dan untuk mewujudkan aplikasi yang lebih canggih seperti produksi dan kehidupan manusia yang dibantu komputer. Kecerdasan buatan, cabang ilmu komputer, dianggap sebagai salah satu dari tiga teknologi yang paling maju (Rekayasa genetika, ilmu Nano dan kecerdasan Buatan) pada abad ke-21. Kecerdasan buatan (Artificial Intelligence) memodifikasi atau mengadaptasi komputer dari tindakan manusia (termasuk prediksi atau control robot), sehingga bisa menjadi lebih akurat. Kecerdasan buatan telah mencapai perkembangan yang sangat cepat di 30 tahun terakhir dan telah banyak digunakan di banyak negara bidang akademik. Supaya komputer dapat bertindak atau serupa dengan manusia, maka komputer harus diberi bekal pengetahuan, dan mempunyai kemampuan untuk menalar. Penerapan AI beraneka ragam. Salah satu kemampuan ai saat ini adalah mengenali objek-objek pada sebuah gambar atau video, singkatnya dapat mendeteksi objek. Deteksi objek merupakan salah satu tugas penting dalam bidang visi komputer dan pengolahan citra digital. Tujuan utama dari deteksi objek adalah untuk mengidentifikasi dan melokalisasi objek-objek spesifik dalam sebuah citra atau video. Deteksi objek memiliki banyak aplikasi praktis, seperti dalam sistem keamanan, kendaraan otonom, sistem pengawasan, analisis citra medis, dan banyak lagi [1]. Dalam beberapa tahun terakhir, pendekatan berbasis deep learning, khususnya Convolutional Neural Network (CNN), telah menjadi metode yang paling populer dan efektif untuk tugas deteksi objek. Salah satu model deteksi objek berbasis deep learning yang telah mencapai popularitas adalah YOLO (You Only Look Once), yang diusulkan oleh Redmon et al. pada tahun 2016 [1]. Model YOLO menawarkan beberapa keunggulan dibandingkan metode deteksi objek sebelumnya, seperti R-CNN [2] dan Faster R-CNN [3]. Pertama, YOLO memprediksi bounding box dan

kemungkinan kelas objek secara simultan, sehingga lebih efisien dalam hal waktu komputasi. Kedua, YOLO menggunakan arsitektur CNN yang relatif sederhana, sehingga lebih mudah diimplementasikan dan dioptimalkan. Ketiga, YOLO memiliki kecepatan yang tinggi, sehingga cocok untuk aplikasi real-time seperti deteksi objek pada video atau kamera yang bergerak. Meskipun demikian, model YOLO juga memiliki beberapa keterbatasan, seperti akurasi yang sedikit lebih rendah dibandingkan metode lain pada objek kecil atau objek dengan variasi bentuk yang ekstrem. Oleh karena itu, pengembangan dan penyempurnaan model YOLO terus dilakukan, seperti versi YOLOv3 yang memiliki peningkatan dalam hal akurasi dan kemampuan mendeteksi objek kecil.

## 2. Metode Penelitian

### 2.1. Dataset

Dataset yang digunakan diambil dari Roboflow. Dataset yang kali ini saya gunakan berupa dataset tentang hewan khususnya hewan peliharaan. Dataset yang totalnya berjumlah 770 gambar. Dataset untuk train berjumlah 524, untuk validasi berjumlah 136 dan terakhir untuk testing berjumlah 110 gambar.



Gambar 1. Contoh Dataset

### 2.2. CNN (Convolutional Neural Network)

Convolutional Neural Network (CNN) adalah salah satu jenis Neural Network (NN) yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. Secara garis besar CNN tidak jauh beda dengan Neural Network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Yang membedakan antara CNN dengan NN ialah arsitektur dari CNN dibagi menjadi dua bagian besar, yaitu Feature Extraction Layer dan Fully Connected Layer [2].

### 2.3. You Only Look Once (YOLO)

Arsitektur YOLO terdiri atas 24 lapisan konvolusional (convolutional layer) dengan 4 lapisan max pooling, yang diikuti oleh 2 lapisan yang terhubung penuh (fully connected layer). Beberapa lapisan konvolusi menggunakan lapisan reduksi  $1 \times 1$  sebagai alternatif untuk mengurangi kedalaman feature maps [1].

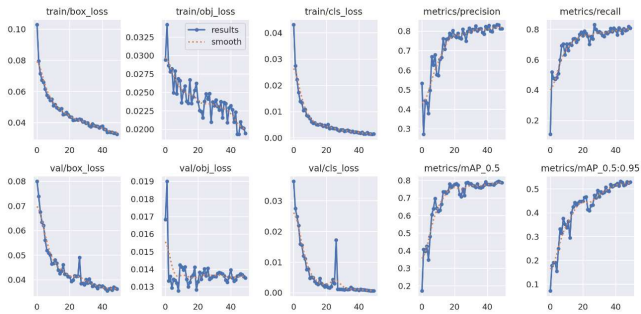
Arsitektur YOLO sesungguhnya cukup sederhana. Sistem akan menerima input citra dengan bentuk (448, 448, 3) yaitu citra berukuran  $448 \times 448$  dengan 3 channel, yang kemudian akan melewati satu kali proses convolutional network hingga menghasilkan output dengan bentuk (7, 7, 30), dimana  $7 \times 7$  merupakan ukuran grid sel ( $S = 7$ ) dan 30 merupakan nilai dari jumlah kotak



dilakukan setelah proses anotasi pada seluruh citra selesai, sehingga diperoleh kumpulan file .txt dari tiap citra yang berisi representasi hasil anotasi citra.

### 3.2. Training dan Validasi Model YOLO V

Proses training model dilakukan pada data training. Metriks evaluasi seperti Precision, Recall, dan mAP diperoleh melalui proses validasi pada model yang dihasilkan.

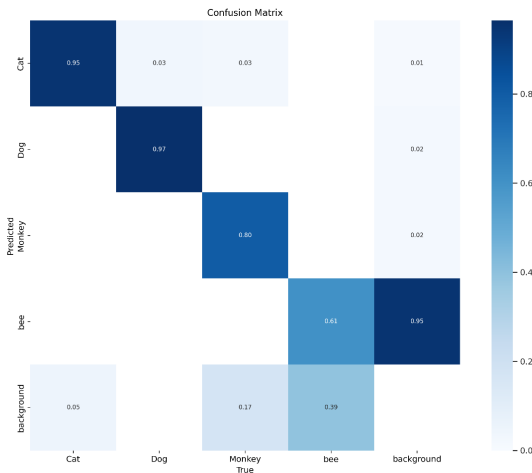


Gambar 4. Grafik Loss dan Metriks Training dan Validasi Model

Plot train/box\_loss menunjukkan penurunan loss untuk regresi bounding box dari nilai 0,08 hingga 0,04 pada akhir pelatihan. Sementara pada plot train/obj\_loss, nilai loss untuk deteksi objek menurun dari sekitar 0,0325 hingga 0,01 di akhir proses. Pada metrik klasifikasi yakni train/cls\_loss, nilainya menurun dari 0,04 hingga mendekati 0,01 di akhir pelatihan. Di sisi lain, metrik seperti precision dan recall untuk evaluasi performa model meningkat dari nilai rendah di awal hingga mendekati 0,8 di akhir proses. Begitu pula metrik mAP (mean Average Precision) untuk deteksi objek meningkat dari nilai sekitar 0,3 hingga mendekati 0,8 pada akhir pelatihan. Penurunan loss dan peningkatan metrik seperti precision, recall, dan mAP mengindikasikan peningkatan performa model dalam melakukan deteksi dan klasifikasi objek seiring berjalannya proses pelatihan.

### 3.3. Confusion Matrix

Confusion matrix atau matriks konfusi yang menunjukkan hasil klasifikasi dari sebuah model machine learning untuk mengklasifikasikan beberapa objek, yaitu Cat, Dog, Monkey, Bee, dan Background. Matriks konfusi menyajikan informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh model dalam bentuk matriks. Diagonal utama matriks menunjukkan jumlah sampel yang diklasifikasikan dengan benar untuk setiap kelas, sedangkan nilai-nilai di luar diagonal utama merupakan kesalahan klasifikasi. Berdasarkan gambar, dapat dilihat bahwa model memiliki akurasi tertinggi dalam mengklasifikasikan kelas Cat dengan nilai 0,95 atau 95%. Sementara untuk kelas Dog dan Monkey, akurasi model masing-masing adalah 0,97 atau 97% dan 0,80 atau 80%. Namun, model tampaknya kurang akurat dalam mengklasifikasikan kelas Bee, dengan nilai 0,61 atau 61% sampel Bee diklasifikasikan dengan benar, sementara 0,39 atau 39% lainnya salah diklasifikasikan. Selain itu, terdapat beberapa kesalahan klasifikasi di luar diagonal utama, seperti 0,03 sampel Cat salah diklasifikasikan sebagai Dog dan Monkey, serta 0,02 sampel Dog dan Monkey salah diklasifikasikan sebagai Bee.



Gambar 5. Confusion Matrix

#### 4. Kesimpulan

Penelitian ini berfokus pada implementasi model You Only Look Once (YOLO) untuk deteksi objek, khususnya hewan peliharaan. Model YOLO, yang berbasis deep learning, menawarkan beberapa keunggulan dibandingkan metode tradisional, termasuk prediksi bounding box dan probabilitas kelas objek secara simultan, arsitektur Convolutional Neural Network (CNN) yang relatif sederhana, dan kecepatan komputasi yang tinggi, sehingga cocok untuk aplikasi real-time. Studi ini menggunakan dataset sebanyak 770 gambar, dengan 524 untuk pelatihan, 136 untuk validasi, dan 110 untuk pengujian. Data gambar melalui proses anotasi untuk memberi label pada daerah yang menjadi perhatian dengan label kelas. Proses pelatihan melibatkan optimalisasi model YOLO menggunakan data yang dianotasi, sementara metrik kinerja seperti presisi, recall, dan mean Average Precision (mAP) diperoleh melalui proses validasi. Hasil menunjukkan kemampuan model dalam mendeteksi dan mengklasifikasikan objek secara efektif, dengan metrik kinerja seperti presisi, recall, dan mAP mendekati 0,8 di akhir proses pelatihan. Fungsi kerugian untuk regresi bounding box, deteksi objek, dan klasifikasi juga menunjukkan tren penurunan, mengindikasikan peningkatan kinerja model. Selanjutnya, sebuah confusion matrix disajikan, memberikan wawasan tentang akurasi model dalam mengklasifikasikan kelas yang berbeda. Model mencapai akurasi tertinggi untuk kelas 'Cat' sebesar 95%, sementara kelas 'Bee' menjadi yang paling menantang dengan akurasi 61%. Secara keseluruhan, implementasi model YOLO dalam studi ini menunjukkan efektivitasnya dalam mendeteksi dan mengklasifikasikan hewan peliharaan. Namun, masih ada ruang untuk peningkatan, terutama dalam meningkatkan akurasi model untuk kelas yang menantang dan mengatasi kemungkinan kesalahan klasifikasi.



**Daftar Pustaka**

- [1] R. Dwiyanto, D. W. Widodo, and P. Kasih, "Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung," in Seminar Nasional Inovasi Teknologi, 2022, pp. 102-104.
- [2] A. R. Wasril, M. S. Ghozali, and M. B. Mustafa, "Pembuatan Pendeteksi Obyek Dengan Metode You Only Look Once (YOLO) Untuk Automated Teller Machine (ATM)," *Majalah Ilmiah UNIKOM*, vol. 17, no. 1, pp. 69-75, 2019.
- [3] Janapriya, A.A.G.B., 2023. Pengenalan Jenis Rambu Lalu Lintas menggunakan Metode YOLO V5. *Jurnal Elektronik Ilmu Komputer Udayana*, 11(4), pp.1011-1018.
- [4] Priana, K.A., Karyawati, A.A.I.N.E., 2023. Sistem Pendeteksi Sampah Secara Realtime Menggunakan Metode YOLO. *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, 2(1), pp.31-36.

# Memprediksi Kelulusan Mahasiswa: Graduate dan Dropout dengan Support Vector Machine dan GridSearchCV

Ni Putu Eka Marita Anggarini<sup>a1</sup>, Agus Muliantara<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>anggarini.2208561032@student.unud.ac.id  
<sup>2</sup>muliantara@unud.ac.id

## Abstract

*In today's educational landscape, having a model to predict whether a student will graduate or drop out based on their academic statistics is highly beneficial. Such a model allows for early assessment of academic success. Human calculations alone can be time-consuming and often lack accuracy, hence the introduction of machine learning models to address this issue. This research utilizes a dataset comprising undergraduate students from various majors in higher education institutions. The data were collected while the students were still enrolled, with their grades from the first year serving as a key feature. The response variable in the dataset is labeled as either 'dropout' or 'graduate'. We employ Support Vector Machines (SVM) with GridSearchCV optimization to build the predictive model. The goal of this model is to predict a student's academic success as early as their first-year statistics are available. If a student is predicted to drop out, targeted interventions can be provided to help them overcome challenges, ultimately aiming to improve graduation rates.*

**Keywords:** siswa, akademik, dropout, graduate, SVM, hyperparameter tuning, klasifikasi, prediksi, machine learning, GridSearchCV

## 1. Pendahuluan

Mengetahui apakah seorang siswa akan drop out atau lulus sangat penting sejak dini, bahkan sejak tahun pertama mereka. Hal ini penting agar siswa bisa memahami kondisi mereka dan mendapatkan dukungan yang diperlukan untuk membantu mereka dalam akademis, sehingga pada akhirnya mereka semua dapat lulus. Prediksi dan klasifikasi awal tingkat kinerja siswa memberikan peringatan dini dan memberikan resep untuk meningkatkan kinerja siswa yang kurang baik serta untuk pengaturan manajerial lainnya [1]. Di era revolusi informasi, analisis basis data dalam lingkungan pendidikan seperti analitik pembelajaran, analitik prediktif, penambahan data pendidikan, dan teknik *machine learning* telah menjadi bidang penelitian yang populer [2]. *Machine Learning* adalah proses mempelajari serangkaian aturan dari contoh-contoh atau lebih umum lagi, menciptakan sebuah pengklasifikasi yang dapat digunakan untuk menggeneralisasi dari contoh-contoh baru. Pembuatan pengklasifikasi adalah proses dua langkah. Pada langkah pertama, model pengklasifikasi dibangun menggunakan set data yang diberikan. Langkah ini disebut *training*. Pada langkah ini, aturan-aturan klasifikasi dibuat. Langkah kedua disebut *testing*, yang menentukan akurasi aturan-aturan klasifikasi [3]. *Machine learning* digunakan untuk memprediksi, mengklasifikasikan kinerja siswa, dan menganalisis perilaku belajar mereka guna memantau kemajuan mereka di bidang akademik. Namun, yang menjadi tantangan adalah menemukan algoritma optimal yang dapat menghasilkan hasil yang memuaskan [1]. Algoritma *machine learning* seperti *naive Bayes*, *logistic regression*, *artificial neural network*, *decision trees*, *random forest*, *support vector machine*, *k-nearest neighbor*, dan lainnya, banyak digunakan untuk menganalisis dan memprediksi kinerja akademik [2]. Dalam penelitian ini, diusulkan penggunaan model pembelajaran mesin untuk memprediksi keberhasilan akademik siswa, yang akan mengeliminasi kebutuhan perhitungan manual. Penggunaan model pembelajaran mesin tidak

hanya menghemat waktu tetapi juga meningkatkan akurasi dibandingkan perhitungan manual. Model ini akan menggunakan teknologi terkini dalam pembelajaran mesin dan teknik optimasi, khususnya model Support Vector Machine (SVM) dan algoritma tuning hyperparameter GridSearchCV untuk mencapai akurasi tertinggi. Dengan memanfaatkan dataset dari University of California, Irvine, model pembelajaran mesin dilatih untuk memprediksi apakah seorang siswa akan drop out atau lulus.

## 2. Metode Penelitian

### 2.1 Dataset

Untuk melatih model pembelajaran mesin agar dapat memprediksi apakah seseorang akan lulus, putus sekolah, atau masih terdaftar sebagai mahasiswa, terlebih dahulu dibutuhkan dataset yang berisi catatan informasi pendidikan atau atribut pribadi mahasiswa dan status mereka, apakah lulus, putus sekolah, atau masih terdaftar. Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle.com, sebuah repositori sumber terbuka untuk dataset pembelajaran mesin, yang berasal dari University of California, Irvine, terkait dengan mahasiswa yang terdaftar dalam berbagai program sarjana. Dataset ini mencakup informasi yang diketahui saat pendaftaran mahasiswa (jalur akademik, demografi, dan faktor sosial-ekonomi) dan kinerja akademik mahasiswa (GDP) pada akhir tahun pertama kuliah. Data ini memiliki tiga kategori sebagai target: putus sekolah, terdaftar, dan lulus. Dengan 4424 instance, dataset ini mencakup 36 fitur. Namun dalam penelitian ini, label "Enrolled" pada variabel target dihapus, sehingga hanya 3630 baris yang akan digunakan, dan hanya akan ada dua kategori sebagai target: putus sekolah dan lulus. Fitur target adalah variabel respon yang dihitung menggunakan variabel lainnya. Deskripsi masing-masing fitur dan variabel target disajikan dalam Tabel 1.

Tabel 1. Karakteristik Basis Data

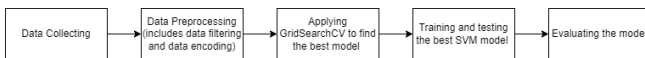
Nama Variabel	Type Variable	Deskripsi	Nilai
<i>Marital Status</i>	<i>Feature</i>	<i>Status of marriages/relationship</i>	<i>Any value from 1-6 (6 unique marital statuses)</i>
<i>Application mode</i>	<i>Feature</i>	<i>Mode of application (e.g., change of course, transfer, etc.)</i>	<i>Some value from 1-57 (17 unique application modes)</i>
<i>Application order</i>	<i>Feature</i>	<i>Order of application (e.g., first choice, last choice, etc.)</i>	<i>Any value from 0-9 (10 unique application orders)</i>
<i>Course</i>	<i>Feature</i>	<i>Type of courses</i>	<i>Some value from 33-9991 (17 unique courses)</i>
<i>Daytime/evening attendance</i>	<i>Feature</i>	<i>Attendance time (e.g., evening, daytime)</i>	0,1
<i>Previous qualification</i>	<i>Feature</i>	<i>Previous education level</i>	<i>Some value from 1-43 (17 unique education levels)</i>
<i>Previous qualification (grade)</i>	<i>Feature</i>	<i>Grade of previous qualification</i>	<i>Any values between 0-200</i>
<i>Nacionality</i>	<i>Feature</i>	<i>Nationality</i>	<i>Some value from 1-109 (21 unique nationalities)</i>
<i>Mother's qualification</i>	<i>Feature</i>	<i>Education level</i>	<i>Some value from 1-44 (29 unique education levels)</i>
<i>Father's qualification</i>	<i>Feature</i>	<i>Education level</i>	<i>Some value from 1-44 (29 unique education levels)</i>

<b>Nama Variabel</b>	<b>Type Variable</b>	<b>Deskripsi</b>	<b>Nilai</b>
<i>Mother's occupation</i>	<i>Feature</i>	<i>Types of job</i>	<i>Some values from 0-194 (32 unique jobs)</i>
<i>Father's occupation</i>	<i>Feature</i>	<i>Type of job</i>	<i>Some values from 0-195 (46 unique jobs)</i>
<i>Admission grade</i>	<i>Feature</i>	<i>Admission grade</i>	<i>Any values between 95-190</i>
<i>Displaced</i>	<i>Feature</i>	<i>1 – yes 0 – no if displaced</i>	0,1
<i>Educational special needs</i>	<i>Feature</i>	<i>1 – yes 0 – no if needs educational special needs</i>	0,1
<i>Debtor</i>	<i>Feature</i>	<i>1 – yes 0 – no if a debtor</i>	0,1
<i>Tuition fees up to date</i>	<i>Feature</i>	<i>1 – yes 0 – no if tuition fees are up to date</i>	0,1
<i>Gender</i>	<i>Feature</i>	<i>1 – male 0 – female</i>	0,1
<i>Scholarship holder</i>	<i>Feature</i>	<i>1 – yes 0 – no if a scholarship holder</i>	0,1
<i>Age at enrollment</i>	<i>Feature</i>	<i>Age of student at enrollment</i>	<i>Any values from 17-70</i>
<i>International</i>	<i>Feature</i>	<i>1 – yes 0 – no if an international student</i>	0,1
<i>Curricular units 1st sem (credited)</i>	<i>Feature</i>	<i>Number of curricular units credited in the 1st semester</i>	<i>Any values from 0-20</i>
<i>Curricular units 1st sem (enrolled)</i>	<i>Feature</i>	<i>Number of curricular units enrolled in the 1st semester</i>	<i>Any values from 0-26</i>
<i>Curricular units 1st sem (evaluations)</i>	<i>Feature</i>	<i>Number of evaluations to curricular units in the 1st semester</i>	<i>Any values from 0-45</i>
<i>Curricular units 1st sem (approved)</i>	<i>Feature</i>	<i>Number of curricular units approved in the 1st semester</i>	<i>Any values from 0-26</i>
<i>Curricular units 1st sem (grade)</i>	<i>Feature</i>	<i>Grade average in the 1st semester</i>	<i>Any values from 0-18.875</i>
<i>Curricular units 1st sem (without evaluations)</i>	<i>Feature</i>	<i>Number of curricular units without evaluations in the 1st semester</i>	<i>Any values from 0-12</i>
<i>Curricular units 2nd sem (credited)</i>	<i>Feature</i>	<i>Number of curricular units credited in the 2nd semester</i>	<i>Any values from 0-19</i>
<i>Curricular units 2nd sem (enrolled)</i>	<i>Feature</i>	<i>Number of curricular units enrolled in the 2nd semester</i>	<i>Any values from 0-23</i>
<i>Curricular units 2nd sem (evaluations)</i>	<i>Feature</i>	<i>Number of evaluations to curricular units in the 2nd semester</i>	<i>Any values from 0-33</i>
<i>Curricular units 2nd sem (approved)</i>	<i>Feature</i>	<i>Number of curricular units approved in the 2nd semester</i>	<i>Any values from 0-20</i>
<i>Curricular units 2nd sem (grade)</i>	<i>Feature</i>	<i>Grade average in the 2nd semester</i>	<i>any values from 0-18.6</i>

Nama Variabel	Tipe Variable	Deskripsi	Nilai
Curricular units 2nd sem (without evaluations)	Feature	Number of curricular units without evaluations in the 1st semester	Any values from 0-12
Unemployment rate	Feature	Unemployment rate (%)	Any values from 7.6 to 16.2
Inflation rate	Feature	Inflation rate (%)	Any values from (-0.8) to 3.7
GDP	Feature	GDP	Any values from (-4.06) to 3.51
Target	Target	Dropout, Graduate	0,1

### 2.2 Support Vector Machine

Semua prosedur dalam penelitian ini dilakukan menggunakan bahasa pemrograman Python, lingkungan pengembangan Google Colaboratory, dan pustaka pembelajaran mesin sklearn. Untuk melatih model pembelajaran mesin, diperlukan set pelatihan dan set pengujian. Oleh karena itu, dataset dibagi secara programatis. Pembagian data ini dilakukan secara acak. Konsep utama dari SVM adalah untuk mendapatkan Hyperplane Pemisah Optimal (OSH) antara sampel positif dan negatif. Ini dapat dilakukan dengan memaksimalkan margin antara dua hyperplane paralel. Dengan menemukan hyperplane ini, SVM kemudian dapat memprediksi klasifikasi sampel yang tidak berlabel dengan menentukan di sisi mana dari hyperplane pemisah sampel tersebut berada. SVM mampu menangani berbagai jenis masalah klasifikasi seperti masalah klasifikasi linier dan non-linier [4]. Linear SVM digunakan dengan data yang dapat dipisahkan secara linier, artinya data tidak memerlukan transformasi untuk memisahkan kelas yang berbeda. Jika data tidak dapat dipisahkan secara linier (*non-linear data*), SVM menyelesaikannya dengan membuat variabel baru menggunakan kernel. Kernel adalah fungsi matematis yang digunakan untuk memetakan titik data input asli ke ruang fitur berdimensi tinggi, sehingga *hyperplane* dapat dengan mudah ditemukan meskipun titik data tidak dapat dipisahkan secara linier dalam ruang input asli. Beberapa fungsi kernel yang umum digunakan adalah linear, polynomial, radial basis function (RBF), dan sigmoid. Dalam pengaplikasiannya, pertama, data akan dibagi menjadi  $X_{train}$  dan  $y_{train}$  dengan variabel 'Target' bertindak sebagai label model kami. Kemudian, data dibagi menjadi 80% sebagai data pelatihan dan 20% sebagai data pengujian. Sebelum membuat model dan memasukkan data ke dalamnya, metode *hyperparameter tuning* dengan *GridSearchCV* diterapkan untuk mengoptimalkan model agar mencapai akurasi tertinggi yang mungkin. Berikut adalah gambaran dari metode yang akan digunakan.



Gambar 1. Metode Aplikasi SVM dan GridSearchCV

### 2.3 GridSearchCV

*GridSearchCV* (*Grid Search Cross-Validation*) adalah sebuah teknik yang digunakan untuk mengatur *hyperparameter* dan pemilihan model [5]. *GridSearchCV* mengotomatiskan proses untuk menemukan set parameter yang optimal dan menghindari *overfitting* [6]. Metode ini secara sistematis menguji semua kombinasi yang mungkin dari *hyperparameter* yang disediakan untuk menemukan set yang memberikan kinerja terbaik untuk model yang diberikan. Proses ini melibatkan pembagian dataset menjadi set pelatihan dan validasi beberapa kali menggunakan *cross-validation*, yang memastikan setiap kombinasi parameter dievaluasi secara adil dan mengurangi risiko *overfitting*. Untuk setiap kombinasi, model dilatih pada set pelatihan dan dievaluasi pada set validasi. Metrik kinerja dari setiap lipatan *cross-validation* dirata-ratakan untuk menentukan efektivitas keseluruhan dari set parameter tersebut. Kombinasi yang menghasilkan

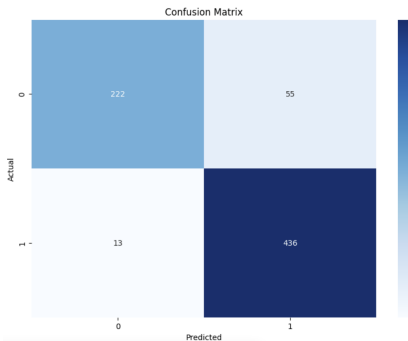
kinerja rata-rata tertinggi dipilih sebagai set *hyperparameter* optimal. Metode ini menyediakan pendekatan komprehensif untuk penyetelan *hyperparameter*, meningkatkan kinerja prediktif dan kemampuan generalisasi model. Dalam pengaplikasiannya di penelitian ini, parameter yang digunakan meliputi parameter C, degree, dan kernel dengan berbagai nilai. Ide utamanya adalah model akan dibuat menggunakan semua nilai berbeda untuk setiap parameter, dan yang memiliki akurasi terbaik akan dipilih sebagai parameter terbaik dan estimator terbaik yang akan digunakan untuk model akhir.

### 3. Hasil dan Diskusi

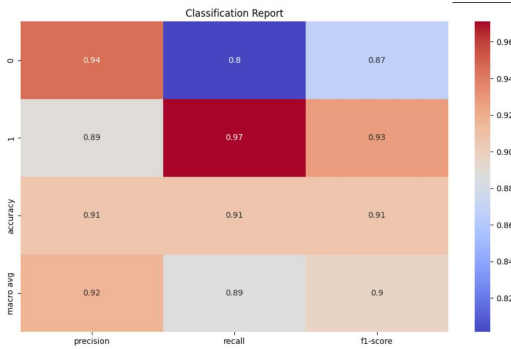
Setelah menguji berbagai kombinasi *hyperparameter* menggunakan metode *GridSearchCV*, kami menemukan bahwa parameter terbaik untuk model *Support Vector Machine* (SVM) adalah C: 100, degree: 1, dan kerne: poly. Kombinasi ini menghasilkan estimator terbaik yaitu SVC (C=100, degree=1, kernel=poly). Langkah selanjutnya adalah menerapkan model ini pada data pelatihan (X\_train dan y\_train). Model dilatih untuk mengenali pola dan hubungan dalam data pelatihan dengan menggunakan parameter optimal yang telah ditemukan. Setelah pelatihan selesai, model kemudian diuji pada data pengujian untuk mengevaluasi kinerjanya. Hasilnya menunjukkan bahwa model tersebut mencapai tingkat akurasi sebesar 0.90633608815427 pada data pengujian, yang menunjukkan bahwa model tersebut mampu memprediksi dengan tingkat ketepatan yang tinggi. Berikut adalah laporan klasifikasi yang merinci kinerja model pada data pengujian, yang meliputi metrik seperti precision, recall, f1-score, dan support untuk masing-masing kelas target:

**Tabel 2.** Classification Report

	Precision	Recall	F1-Score	Support
0 (Dropout)	0.94	0.8	0.87	277
1 (Graduate)	0.89	0.97	0.93	449
Accuracy			0.91	726
Macro Avg	0.92	0.89	0.9	726
Weighted Avg	0.91	0.91	0.9	726



**Gambar 2.** Confusion Matrix



Gambar 3. Classification Report

#### 4. Kesimpulan

Algoritma SVM dapat memprediksi kesuksesan akademik mahasiswa dengan baik menggunakan penyetalan *hyperparameter* dengan *GridSearchCV*. Model terbaik dari algoritma SVM menggunakan parameter C: 100, degree: 1, dan kernel: poly yang diperoleh dengan menggunakan metode *GridSearchCV*. Model ini mencapai akurasi hampir 91% yang akan baik untuk memprediksi keberhasilan akademik mahasiswa berdasarkan statistik tahun pertama mereka untuk menentukan apakah mereka akan lulus atau putus sekolah. Ini akan bermanfaat karena mahasiswa dengan kinerja akademik yang kurang baik dapat mendapatkan akomodasi sehingga mereka dapat berprestasi lebih baik dan tidak putus sekolah pada akhir tahun universitas.

#### Daftar Pustaka

- [1] S. Phauk and O. Takeo, "Hybrid Machine Learning Algorithms for Predicting Academic Performance," *International Journal of Advanced Computer Science and Applications*, vol. 11, no.1, pp. 32-41, 2020.
- [2] S. Slater, S. Joksimovic, V. Kovanovic, R.s Baker, and D. Gasevic, "Tools for Educational Data Mining: A Review", *Journal of Educational and Behavioral Statistics*, Vol. 42, No. 1, 2016, pp. 88-106.
- [3] W. Slamet and A. Taufiq, "Comparative Study of Machine Learning KNN, SVM, and Decision Tree Algorithm to Predict Student's Performance", *International Journal of Research - Granthaalayah*, vol. 7, no.1, pp. 190-196, 2019.
- [4] K. Motaz, A. Tarek, and S. Ghada, "A Comparison among Support Vector Machine and other Machine Learning Classification Algorithms," *International Journal of Computer Science*, vol. 3, no.5, pp. 25-45, 2015.
- [5] T. Yan, S.-L. Shen, A. Zhou, and X. Chen, "Prediction of geological characteristics from shield operational parameters by integrating grid search and K-fold cross validation into stacking classification algorithm," *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 14, no. 4, pp. 1292-1303, Aug. 2022: <https://doi.org/10.1016/j.irmge.2022.03.002>.
- [6] A. Talal, "Using Artificial Neural Networks with GridSearchCV for Predicting Indoor Temperature in a Smart Home", *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13437-13443, 2024.

# Klasifikasi Ekspresi Wajah Menggunakan Metode CNN: Studi Kasus Dataset Kaggle

Wayan Restama Yasa<sup>a1</sup>, AAIN Eka Karyawati <sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>wayanrestama20@email.com  
<sup>2</sup>eka.karyawati@unud.ac.id

## Abstract

*This research aims to implement a Convolutional Neural Network (CNN) in facial expression classification using the Kaggle dataset which consists of five types of facial expressions, namely anger, disgust, fear, happiness and sadness. This method is considered important in supporting various applications such as emotion detection, facial recognition, and better human-machine communication. In this research, data preprocessing and augmentation were carried out using ImageDataGenerator to increase data diversity and prevent overfitting. Next, a CNN architecture is built which consists of convolution layers, pooling layers, and Dense layers. The model was trained using the Adam optimizer with a categorical crossentropy loss function for 50 epochs. The results show that the model achieves approximately 51% accuracy on the validation set. However, further analysis showed variations in model performance among facial expression classes, with some classes performing better than others.*

**Keywords:** Facial Expression Classification, Convolutional Neural Network, Kaggle Dataset, Data Augmentation, Image Processing.

## 1. Pendahuluan

Wajah memiliki peran yang sentral dan penting dalam menyampaikan pesan emosional, mengenali individu, dan berkomunikasi. Dalam interaksi sosial, ekspresi wajah menjadi sarana komunikasi yang sangat penting, yang mencerminkan emosi yang dirasakan oleh seseorang. Ekspresi wajah tersebut bukan hanya sekedar pantulan dari keadaan emosi seseorang, tetapi juga dapat memberikan petunjuk terhadap perasaan emosional seseorang pada saat tertentu, sehingga memungkinkan kita untuk memahami kondisi psikologis mereka [1]. Ekspresi wajah dapat menjadi penanda yang khas dari berbagai jenis emosi yang dialami oleh seseorang. Melalui gerakan otot yang halus, wajah yang dapat menggambarkan kegembiraan, kesedihan, kejijikan, kemarahan, ketakutan dan lainnya [2]. Pengenalan ekspresi wajah, sebagai salah satu aplikasi dari *Convolutional Neural Network* (CNN), menjadi fokus utama dalam upaya memahami dan menginterpretasikan bahasa visual manusia dengan memanfaatkan teknologi deep learning [3]. Penelitian ini bertujuan untuk mengembangkan model prediktif yang mampu secara real-time mengidentifikasi ekspresi wajah manusia. Melalui pendekatan ini, diharapkan mesin juga dapat menjadi semakin cakap dalam memahami dimensi ekspresif manusia, berkontribusi dalam berbagai bidang seperti deteksi emosi, analisis sentiment, dan antarmuka manusia-mesin yang lebih responsive. *Convolutional Neural Network* (CNN) digunakan karena kemampuannya didalam mengidentifikasi fitur – fitur penting secara otomatis dari data citra, seperti tepi, warna, dan bagian-bagian parsial dari objek. Dengan demikian, penelitian ini memusatkan perhatian pada pengembangan model CNN untuk mengenali ekspresi wajah manusia, sebagai langkah dalam memahami dimensi ekspresif manusia dengan lebih mendalam serta meningkatkan kemampuan teknologi dalam memahami bahasa visual yang kompleks.



## 2. Metode Penelitian

### 2.1 Desain Penelitian



**Gambar 1.** Diagram Flow Penelitian

Diagram flow penelitian diatas mengilustrasikan langkah langkah dari pengumpulan dataset Kaggle hingga evaluasi model CNN. Langkah – langkah tersebut mencakup pengumpulan dataset, *preprocessing*, augmentasi data, Pembangunan arsitektur CNN, pelatihan model, dan evaluasi model CNNnya.

### 2.2 Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Kaggle yang memuat gambar-gambar ekspresi wajah manusia. Dataset ini mencakup lima jenis ekspresi emosi yang berbeda, yaitu marah, sedih, senang, jijik, dan takut. Tabel 1 menampilkan contoh data gambar dari masing-masing ekspresi, disertai dengan labelnya.

**Tabel 1.** Dataset Ekspresi Wajah

No	Data	Label
1		Ekspresi marah
2		Ekspresi jijik
3		Ekspresi takut

No	Data	Label
4		Ekspresi senang
5		Ekspresi sedih

Kemudian setelah seluruh dataset terkumpul dengan jumlah 1174 gambar dari seluruh ekspresi emosi. Dataset tersebut dibagi menjadi lima jenis ekspresi emosi: marah, jijik, takut, senang, dan sedih. Setiap jenis ekspresi memiliki jumlah gambar yang berbeda. Untuk keperluan pelatihan model, dataset dibagi menjadi dua bagian, yaitu data training dan data validation. Proporsi pembagian dataset untuk training dan validation adalah 70% dan 30% secara berurutan. Rincian pembagian datasetnya dapat terlihat pada Tabel 2.

Tabel 2. Spesifikasi Dataset

Ekspresi	Jumlah	Training	Validation
Marah	165	110	55
Jijik	360	240	120
Takut	324	216	108
Senang	157	105	52
Sedih	168	112	56

### 2.3 Preprocessing Data

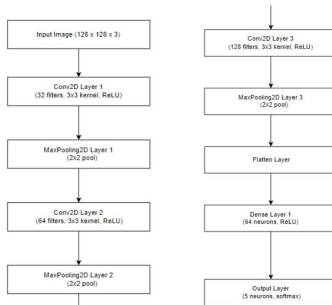
Sebelum dilakukan pelatihan model, dilakukan tahap preprocessing data terlebih dahulu. Tahap ini meliputi dua langkah utama, yaitu pengubahan ukuran gambar dan normalisasi intensitas piksel. Pertama, semua gambar dalam dataset diubah ukurannya menjadi 128x128 piksel. Hal ini dilakukan untuk memastikan bahwa semua gambar memiliki ukuran yang konsisten, memudahkan proses pelatihan model. Selanjutnya, intensitas piksel dalam gambar dinormalisasi ke rentang [0, 1]. Dengan melakukan normalisasi ini, nilai piksel dalam gambar diubah sedemikian rupa sehingga nilai piksel tertinggi menjadi 1 dan nilai piksel terendah menjadi 0

### 2.4 Augmentasi Data

Augmentasi data dilakukan menggunakan ImageDataGenerator, sebuah alat bawaan TensorFlow yang memungkinkan untuk melakukan augmentasi data secara otomatis pada gambar-gambar dalam dataset. Teknik augmentasi yang diterapkan mencakup rotasi, pergeseran, pemotongan, dan pembalikan horizontal. Augmentasi data bertujuan untuk meningkatkan keragaman data yang tersedia untuk pelatihan model, sehingga model dapat belajar dari berbagai variasi gambar dan menjadi lebih robust terhadap variasi yang mungkin ada dalam data yang sebenarnya.

## 2.5 Arsitektur CNN

Arsitektur Convolutional Neural Network (CNN) telah dibangun untuk mengekstrak fitur dari gambar-gambar ekspresi wajah. Arsitektur ini terdiri dari lapisan-lapisan konvolusi yang bertugas untuk mengekstrak fitur-fitur penting dari gambar, diikuti oleh lapisan pooling untuk mereduksi dimensi fitur, dan dilanjutkan dengan lapisan-lapisan dense sebagai pengklasifikasi. Lapisan-lapisan konvolusi menggunakan filter untuk mendeteksi pola-pola visual, sedangkan lapisan pooling digunakan untuk mereduksi dimensi peta fitur. Lapisan-lapisan dense bertugas sebagai pengklasifikasi untuk menghasilkan prediksi akhir terkait dengan kategori ekspresi wajah.



Gambar 2. Arsitektur CNN

Gambar 2 menunjukkan arsitektur CNN yang digunakan dalam penelitian ini. Setiap lapisan dalam arsitektur ini dirancang untuk secara bertahap mengurangi dimensi data sambil mempertahankan informasi penting, yang akhirnya diolah oleh lapisan dense untuk menghasilkan prediksi akhir. Lapisan konvolusi dan pooling secara berurutan mengolah gambar input, mengekstrak fitur penting, dan mengurangi dimensi data sebelum data tersebut masuk ke lapisan dense untuk klasifikasi akhir.

## 2.6 Pelatihan dan Evaluasi Model

Setelah arsitektur CNN dibangun, selanjutnya akan dilakukan tahap tuning hyperparameter pada dataset training. Pada tahap ini, beberapa hyperparameter yang diatur dan diuji meliputi batch size dan jumlah neuron pada lapisan dense.

Tabel 3. Hasil Tuning Hyperparameter

No	Batch Size	Neuron Dense Laver	Enochs	Validation Accuracv
1	32	64	10	0.31
2	64	64	10	0.35
3	32	64	25	0.38
4	64	64	25	0.30
5	32	64	40	0.45
6	64	64	40	0.31
7	32	64	50	0.51
8	64	64	50	0.40

Dari hasil tuning hyperparameter yang dilakukan, model terbaik yang dipilih memiliki berdasarkan nilai akurasi yang tinggi yaitu pada tuning ke delapan dengan nilai akurasi 0.51%

### 3. Hasil dan Diskusi

#### 3.1 Evaluasi Kinerja Model

Model yang dikembangkan mencapai akurasi sekitar 51% pada set validasi. Namun, analisis lebih lanjut menunjukkan variasi kinerja model di antara kelas-kelas ekspresi wajah, seperti yang terlihat pada Tabel 1

**Table 4.** Classification Report

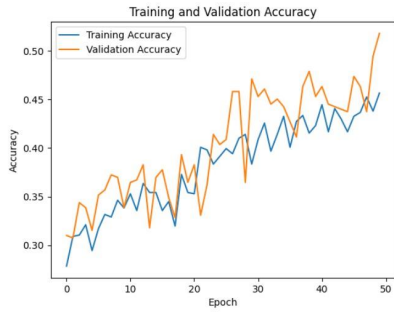
	precision	recall	f1-score	support
angry	0.07	0.02	0.03	55
disgusted	0.21	0.17	0.19	120
fearful	0.26	0.45	0.33	108
happy	0.11	0.15	0.13	52
sad	0.09	0.02	0.03	56
accuracy			0.21	391
macro avg	0.15	0.17	0.14	391
weighted avg	0.17	0.21	0.17	391

Variasi kinerja antar kelas menunjukkan bahwa beberapa kelas memiliki performa yang lebih baik daripada yang lain, sementara kinerja model cenderung rendah pada kelas dengan representasi data yang lebih sedikit atau fitur yang sulit untuk diekstrak. Tantangan dalam pengenalan ekspresi wajah muncul karena model kesulitan untuk mempelajari pola-pola yang ada dalam kelas-kelas tersebut. Hal ini mungkin memerlukan strategi tambahan untuk meningkatkan kinerja model pada kelas-kelas tersebut.

#### 3.2 Diskusi

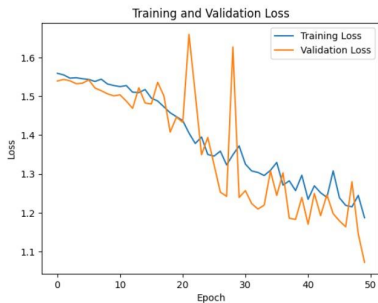
Hasil evaluasi model CNN memberikan wawasan dalam pengembangan teknologi pengenalan ekspresi wajah. Namun, tantangan yang dihadapi, terutama dalam mengenali ekspresi wajah, menunjukkan bahwa masih ada ruang untuk peningkatan kinerja model secara keseluruhan. Variasi kinerja antar kelas menunjukkan bahwa beberapa kelas memiliki performa yang lebih baik daripada yang lain, sementara kinerja model cenderung rendah pada kelas dengan representasi data yang lebih sedikit atau fitur yang sulit untuk diekstrak. Tantangan dalam pengenalan ekspresi wajah muncul karena model kesulitan untuk mempelajari pola-pola yang ada dalam kelas-kelas tersebut. Hal ini mungkin memerlukan strategi tambahan untuk meningkatkan kinerja model pada kelas-kelas tersebut.

Grafik berikut menggambarkan tren akurasi dan loss selama pelatihan dan validasi model:



**Gambar 3.** Grafik Training and Validation Accuracy

Gambar 3 menunjukkan grafik akurasi pelatihan dan validasi pada epoch 50. Dari grafik ini dapat dilihat bahwa akurasi pelatihan dan validasi meningkat pada setiap epoch, namun pada pertengahan terlihat bahwa akurasi validation mengalami penurunan namun bisa stabil kembali.



**Gambar 4.** Grafik Training and Validation Loss

Gambar 4 menunjukkan grafik loss pelatihan dan validasi selama 50 epoch. Loss pada data pelatihan terus menurun seiring bertambahnya epoch menunjukkan bahwa model semakin baik dalam memprediksi data pelatihan. Namun, berbeda dengan nilai pada validasi yang sempat mengalami kenaikan yang signifikan dan kembali turun.

#### 4. Kesimpulan

Penelitian ini berhasil menerapkan Convolutional Neural Network dalam klasifikasi ekspresi wajah menggunakan dataset Kaggle. Model yang dikembangkan mampu mengklasifikasikan ekspresi wajah dengan akurasi sekitar 51% pada set validasi. Meskipun demikian, masih diperlukan penelitian lebih lanjut untuk meningkatkan kinerja model, terutama dalam mengenali ekspresi wajah yang lebih kompleks atau jarang terjadi. Ini menunjukkan bahwa model masih memiliki kesulitan dalam mengidentifikasi pola-pola yang rumit atau jarang muncul dalam dataset.

**Daftar Pustaka**

- [1] E. J. D. B. AL Sigit Guntoro, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network," *Jurnal Informatika Atma Jogja*, vol. 3, pp. 155-160, 2022.
- [2] D. P. Derry Alamsyah, "Implementasi Convolutional Neural Networks (Cnn) Untuk Klasifikasi Ekspresi Citra Wajah Pada Fer-2013 Dataset," *Jurnal Teknologi Informasi*, vol. 4, no. 2615-2738, 2020.
- [3] I. F. R. A. Pulung Adi Nugroho, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *JURNAL ALGOR*, vol. 2, 2020.
- [4] D. I. M. D. L. Soleh Ependi, "Klasifikasi Pendeteksi Wajah Berhijab Menggunakan Metode CNN," vol. 6, no. 2614-3097, pp. 3157-3164, 2022.
- [5] I. G. A. W. Ni Putu Suci Paramita, "Penerapan Metode Kompresi Wavelet dalam Pengolahan Data Gambar untuk Mengurangi Ukuran File," *JNATIA*, vol. 1, no. 2986-3929, 2023.
- [6] I. N. P. K. Q. F. I Putu Agus Aryawan, "Analisis Perbandingan Algoritma Cnn Dan Svm Pada Klasifikasi Ekspresi Wajah," *Jurnal Teknologi Informasi dan Komputer*, vol. 9, 2023.

Halaman ini sengaja dibiarkan kosong

# Klasifikasi Penyakit Gagal Jantung dengan Algoritma XGBoost

I Gusti Gde Bagus Bhadraka Artawibawa<sup>a1</sup>, AAIN Eka Karyawati<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>artawibawa.2208561052@student.unud.ac.id  
<sup>2</sup>Eka.karyawati@unud.ac.id

## Abstract

*Heart, as one of the most important organs in the body, carries a risk of death if abnormalities occur. Heart problems are divided into two categories: heart failure and heart attacks. According to WHO data, approximately 7.3 million people worldwide die due to heart disease. This study uses a dataset of heart disease patients and applies the XGBoost algorithm. The objectives of this study are to process and analyze the data, implement the XGBoost algorithm for heart disease classification, and evaluate the performance of the XGBoost algorithm. The result of this study is the performance evaluation of the XGBoost algorithm, which achieved an accuracy of 93%.*

**Keywords:** Heart, Classification, Failure, XGBoost, Disease

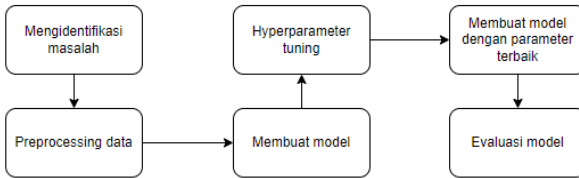
## 1. Pendahuluan

Penyakit kardiovaskular merupakan penyakit pada jantung dan pembuluh darah sebagai penyebab terjadinya gagal jantung. Penyakit ini sering terjadi dan merupakan salah satu penyebab utama kematian di dunia. Berdasarkan data dari WHO, pada tahun 2021 angka kematian akibat penyakit jantung mencapai 17,8 juta orang atau satu dari tiga kematian di dunia pada tiap tahunnya disebabkan oleh penyakit jantung. Gagal jantung adalah kondisi jantung kehilangan kemampuannya untuk memompa darah untuk jumlah yang cukup dalam memenuhi kebutuhan metabolisme tubuh atau jantung hanya mampu melakukannya dengan tekanan pengisian yang tinggi atau dapat juga terjadi kedua-duanya secara bersamaan. Beberapa faktor yang biasanya menyebabkan penyakit gagal jantung diantaranya adalah diabetes, tekanan darah tinggi pola hidup yang tidak sehat dan kurangnya aktivitas fisik [1]. Oleh sebab itu perlu adanya suatu model yang akurat untuk mengklasifikasikan gagal jantung berdasarkan informasi klinis dan gaya hidup pasien pengidap penyakit tersebut, sebagai solusi alternatif dalam pemberian obat yang tepat. Akan tetapi, tingkat akurasi klasifikasi kejadian terkait gagal jantung dalam praktik klinis biasanya kurang sensitif. Saat ini, banyak cara untuk melakukan prediksi, salah satunya dengan membuat pemodelan menggunakan algoritma XGBoost, yaitu pengembangan algoritma dan model secara statistik yang menggunakan sistem komputer serta membutuhkan data yang mengandaikan pola serta inferensi [1]. Dalam beberapa penelitian yang memiliki kemiripan dalam melakukan klasifikasi penyakit jantung, metode-metode yang digunakan umumnya menghasilkan nilai akurasi yang berbeda. Penelitian Perancangan Sistem Klasifikasi Penyakit Jantung Menggunakan Naïve Bayes yang diterbitkan pada tahun 2019[6]. Penggunaan metode Naive Bayes menghasilkan nilai rata-rata akurasi sebesar 90,61%. Penelitian Analisis Penyakit Jantung Koroner Menggunakan Decision Tree yang diterbitkan pada tahun 2019 menjelaskan bahwa dengan menggunakan metode CART Decision Tree, mendapatkan akurasi sebesar 80% [7]. Penelitian Prediction of Heart Diseases using Random Forest yang diterbitkan Maret 2021 menjelaskan bahwa penerapan klasifikasi pada 303 data penyakit jantung menggunakan algoritma Random Forest, mendapatkan hasil akurasi sebesar 86,69% [2]. Tujuan dari penelitian ini untuk mengidentifikasi pasien gagal jantung berdasarkan informasi klinis dan gaya hidup pasien dengan algoritma XGBoost dan memberikan sebuah algoritma atau model yang dapat digunakan untuk mengklasifikasi penyakit gagal jantung.



## 2. Metode Penelitian

Penelitian dibagi menjadi beberapa tahapan yaitu yang pertama ada mengidentifikasi masalah, preprocessing data, membuat model, melakukan hyperparameter tuning, membuat model dengan parameter terbaik, dan evaluasi model. Untuk gambaran dari tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

### 2.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari Kaggle, dan untuk datasetnya dapat dilihat pada Gambar 2. <https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction/data>

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingCG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	209	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
...	...	...	...	...	...	...	...	...	...	...	...	...
913	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
914	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
915	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1
916	57	F	ATA	130	236	0	LVH	174	N	0.0	Flat	1
917	38	M	NAP	138	175	0	Normal	173	N	0.0	Up	0

918 rows x 12 columns

Gambar 2. Dataset yang digunakan

Data yang digunakan berjumlah 918 baris dan 12 kolom, didalamnya terdapat beberapa parameter seperti *age*, *sex*, *cholesterol*, *chestpainType*, dan *oldpeak*.

### 2.2. Preprocessing Data

Sebelum data digunakan untuk melatih model atau algoritma XGBoost, diperlukan adanya preprocessing data agar tidak berdampak buruk pada performa dari model tersebut. Pada penelitian ini, preprocessing data mencakup pengecekan terhadap data duplikat, dan melakukan label encoder untuk data yang bersifat kategorikal. Setelah dilakukan preprocessing data, kemudian data tersebut akan dibagi menjadi data training sebesar 80% dan data testing sebesar 20%.

### 2.3. Algoritma XGBoost

Algoritma Extreme Gradient Boosting (XGBoost) merupakan salah satu metode yang unik dimana XGBoost menggabungkan proses boosting dengan gradient boosting untuk mendapatkan hasil 10 kali lebih cepat. Dimana Extreme Gradient Boosting menggabungkan tiga macam ensembles algorithm yaitu classification dan regression trees (CART) [3]. XGBoost Sebenarnya adalah algoritma yang lebih banyak digunakan dalam pemodelan prediktif, seperti regresi dan klasifikasi. Meskipun XGBoost dapat menghasilkan skor kepentingan atribut, biasanya digunakan untuk menentukan atribut yang paling penting dalam pemodelan prediktif, bukan untuk seleksi atribut untuk clustering K-Means. Meskipun demikian, XGBoost dapat digunakan sebagai alat tambahan untuk analisis atribut pada clustering K-Means. Berikut tahapan penerapannya:

- a. Melakukan clustering menggunakan algoritma K-Means untuk membagi data menjadi kluster-kluster tertentu.
- b. Menggabungkan hasil clustering dengan dataset asli
- c. Memisahkan atribut dan label kluster
- d. Membangun model XGBoost untuk memperbaiki clustering
- e. Menghitung skor kepentingan setiap atribut
- f. Menampilkan atribut yang paling penting berdasarkan skor kepentingan tertinggi [5].

### 2.4. Hyperparameter Tuning

Hyperparameter adalah parameter yang ditetapkan sebelum proses training dimulai. Parameter ini dapat disesuaikan dan secara langsung dapat mempengaruhi seberapa baik kinerja model. Untuk menemukan hyperparameter yang paling optimal, berbagai strategi diperlukan untuk menyesuaikannya. Cara yang paling mudah adalah dengan mencoba berbagai kombinasi hyperparameter. Namun, seiring waktu, banyak pendekatan telah diusulkan untuk mengoptimalkan hyperparameter ini, seperti Grid Search dan Random Search [4]. Pada penelitian ini proses tuning hyperparameter dilakukan menggunakan metode Randomized Search Cross-Validation (Randomized Search CV), yang dipilih karena efisiensinya dalam mengeksplorasi ruang hyperparameter yang besar tanpa memerlukan pencarian lengkap seperti pada Grid Search.

## 3. Hasil dan Diskusi

Pada penelitian ini digunakan algoritma XGBoost dengan hyper parameter tuning. Dataset yang digunakan dibagi menjadi 80% training dan 20% testing. Sebelum melakukan data training, dilakukan pengecekan terhadap data duplikat, dan melakukan label encoder untuk data yang bersifat kategorikal. Pembuatan model akan didasarkan pada parameter terbaik. Selanjutnya penggunaan K-fold cross validation yang bertujuan untuk mendapatkan hasil penilaian akurasi yang lebih maksimal.

### 3.1. Performa Algoritma XGBoost dengan Hyper Parameter Tuning

Berikut merupakan hasil dari algoritma XGBoost. dapat dilihat pada gambar 4 hasil metrics score dari algoritma ini menghasilkan akurasi k-fold cross validation menghasilkan sebesar 88% akurasi dari model XGBoost sebesar 83%, nilai precision sebesar 93%, Recall sebesar 93%, dan F1 score sebesar 93%. Hasil dari penilaian metrics score dari algoritma ini tergolong baik untuk proses klasifikasi. Untuk lebih lengkapnya dapat dilihat pada Gambar 3 dibawah ini.

	precision	recall	f1-score	support
0	0.92	0.93	0.92	86
1	0.94	0.93	0.93	98
accuracy			0.93	184
macro avg	0.93	0.93	0.93	184
weighted avg	0.93	0.93	0.93	184

Gambar 3. Hasil metrics score

Pada Gambar 4 menunjukkan confusion matrix dari model klasifikasi XGBoost dengan hyperparameter tuning RandomizedSearchCV



Gambar 4. Confusion Matrix XGBoost + RandomizedSearchCV

#### 4. Kesimpulan

Berdasarkan hasil penelitian, ditemukan bahwa algoritma XGBoost dapat digunakan untuk mengklasifikasikan penyakit gagal jantung. Performa dari algoritma ini dengan dilakukannya hyperparameter tuning mendapatkan akurasi sebesar 93%, precision sebesar 93%, recall sebesar 93%, dan f1-score sebesar 93%. Jadi, kesimpulannya jika dibandingkan dengan algoritma Decision Tree yang hanya mendapatkan akurasi sebesar 80%, penerapan XGBoost dengan hyperparameter tuning melalui RandomizedSearchCV adalah salah satu pendekatan yang sangat efektif untuk klasifikasi penyakit gagal jantung dengan akurasi mencapai 93%. Hasil penelitian ini menunjukkan potensi besar dalam membantu tenaga medis untuk mendeteksi dan mendiagnosis gagal jantung dengan akurasi yang tinggi, yang pada akhirnya dapat berkontribusi pada peningkatan kualitas perawatan pasien.

#### Daftar Pustaka

- [1] R. Arisandi, "Perbandingan Model Klasifikasi Random Forest Dengan Resampling Dan Tanpa Resampling Pada Pasien Penderita Gagal Jantung," *Jurnal Gaussian: Jurnal Statistika Undip*, vol. 12, no. 1, pp. 136–145, May 2023, doi: 10.14710/j.gauss.12.1.136-145.
- [2] D. H. Depari, Y. Widiastiyi, and M. M. Santoni, "Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung," *Informatik*, vol. 18, no. 3, p. 239, Dec. 2022, doi: 10.52958/iftk.v18i3.4694.
- [3] Y. F. Wijaya and S. Y. J. Prasetyo, "Model penilaian tata guna lahan dengan citra Landsat 8 OLI menggunakan algoritma XGBOOST diwilayah beresiko tsunami (Studi kasus: Kota Palu Sulawesi Tengah)," *ICM (Indonesian Journal of Computing and Modeling)*, vol. 4, no. 1, pp. 23–28, Jul. 2021, doi: 10.24246/icm.v4i1.4981.

- [4] N. F. Rahmadayana and N. Y. Sibaroni, "Sentiment Analysis of Work from Home Activity using SVM with Randomized Search Optimization," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 5, no. 5, pp. 936–942, Oct. 2021, doi: 10.29207/resti.v5i5.3457.
- [5] Bengnga, Amiruddin, and Rezqiwati Ishak. "Penerapan XGBoost untuk Seleksi Atribut pada K-Means dalam Clustering Penerima KIP Kuliah." *Jambura Journal of Electrical and Electronics Engineering* 5, no. 2 (2023): 192-196.
- [6] M. A. Bianto, K. Kusriani, and S. Sudarmawan, "Perancangan Sistem klasifikasi penyakit jantung menggunakan Naive Bayes," *Citec (Creative Information Technology) Journal/Citec Journal*, vol. 6, no. 1, p. 75, Apr. 2019, doi: 10.24076/citec.2019v6i1.231.
- [7] Akbar, Jimmi Afriando, Zidane Ibrahim Fadela, Luthfi Fachruddin, Feby Ardiansyah, and Zuhdi Mukarram Bakhri. "Analisis Penyakit Jantung Koroner Menggunakan Decision Tree." *Departemen Ilmu Komputer, IPB University, Cvd* (2019): 3-7.

Halaman ini sengaja dibiarkan kosong

# Klasifikasi Hewan Berbasis Fitur Gray Level Co-Occurrence Matrix dengan Artificial Neural Network

Ryan Hangralim<sup>a1</sup>, Cokorda Pramatha<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>ryanhangralim@email.com  
<sup>2</sup>cokorda@unud.ac.id

## Abstract

*Dogs and cats are animals commonly treated as pets by many people. Humans have the ability to differentiate various things, and this ability when converted to a form of system is called Computer Vision. Computer Vision has many applications such as image processing that can be used for various things and one of the techniques in image processing is image classification. Image classification is a problem that aims to organize objects that are then observed into predefined categories. The approach utilized to construct the model involves employing an Artificial Neural Network (ANN) using Gray Level Co-occurrence Matrix (GLCM) as the method for extracting the features. The data used are images of cats and dogs which will be extracted using GLCM using various parameters that include distance and angles. The extracted feature will then be used to train a model and accuracy of each model will be measured to find the best parameter result. In this study, the best parameter that results in the best accuracy is 1 for the distance and 0°, 45°, 90° for the combination of angles resulting in 79% accuracy.*

**Keywords:** Animal, Image, Gray Level Co-Occurrence Matrix, Classification, Artificial Neural Network

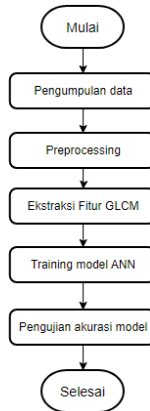
## 1. Pendahuluan

Anjing dan kucing merupakan hewan peliharaan yang cukup sering dipelihara oleh masyarakat karena mereka memiliki karakter yang beragam dan dapat menyenangkan manusia [1]. Manusia memiliki kemampuan dalam melakukan pengenalan terhadap banyak hal, mulai dari sesama manusia sampai hewan dan masih banyak lagi. Kemampuan penglihatan yang dimiliki oleh manusia kemudian dijadikan sebuah sistem yang dikenal sebagai *Computer Vision*. *Computer Vision* memiliki tujuan untuk membangun sebuah model yang didasari oleh sistem penglihatan manusia yang dapat menyelesaikan tugas-tugas yang umumnya dapat dikerjakan oleh sistem penglihatan manusia secara otomatis. [2]. *Computer Vision* memiliki berbagai aplikasi, termasuk *image processing* yang bermanfaat dalam berbagai konteks kegiatan. Salah satu teknik dalam *image processing* adalah proses klasifikasi. Tujuan klasifikasi adalah mengelompokkan objek yang diamati ke dalam kategori yang sudah ditentukan [2]. Hasil dari proses klasifikasi citra digital dapat digunakan sebagai alternatif dalam mengenali hewan seperti anjing dan kucing. Pada penelitian ini akan mengusulkan penggunaan ekstraksi fitur citra digital menggunakan *Gray Level Co-occurrence Matrix (GLCM) dengan Artificial Neural Network (ANN)* sebagai metode pengklasifikasi dalam melakukan klasifikasi citra hewan menjadi anjing dan kucing. GLCM merupakan salah satu metode ekstraksi yang menghasilkan beberapa fitur saja sebagai output dalam bentuk nilai-nilai statistik sehingga akan mereduksi waktu komputasi [3]. Hasil ekstraksi fitur pada GLCM adalah nilai-nilai statistik yang kemudian akan digunakan untuk menentukan pola dan prediksi yang sesuai dengan kategori pada data untuk proses klasifikasi [4]. *Artificial Neural Network (ANN)* merupakan sebuah model komputasi yang terinspirasi dari sistem jaringan pada saraf biologis manusia. Metode ini memanfaatkan komponen yang disebut neuron yang merupakan perhitungan non-linier, yang tersusun dalam jaringan yang saling terhubung. Konsep ini kemudian diimplementasikan dalam program komputer untuk menjalankan serangkaian proses perhitungan selama proses pembelajaran [5].

## 2. Metode Penelitian

### 2.1 Alur Penelitian

Alur penelitian dalam bentuk *flowchart* dapat dilihat pada gambar 1. Penelitian akan dimulai dengan pengumpulan data yang bertujuan untuk mengumpulkan data-data yang akan digunakan untuk training model. Setelah pengumpulan data, akan dilakukan tahap *preprocessing* yang berfungsi untuk mengolah data sebelum dilakukan ekstraksi fitur menggunakan GLCM. Hasil ekstraksi fitur GLCM akan digunakan pada proses training model ANN yang kemudian akan diuji akurasi model dalam proses klasifikasi.



Gambar 1. *Flowchart* Penelitian

### 2.2 Pengumpulan data

Dataset yang akan digunakan untuk membangun model berasal dari situs kaggle.com yang berupa data citra anjing dan data citra kucing [6], kaggle merupakan sebuah platform komunitas online yang digunakan oleh para ilmuwan data dan para penggemar pembelajaran mesin. Kaggle memungkinkan para pengguna untuk mempublikasikan kumpulan data yang dapat membantu para pengguna mencapai tujuan dalam perjalanan ilmu data dengan sumber daya yang disediakan [7]. Data yang digunakan berupa citra digital sejumlah 6000 citra dengan 2 kategori yang dibagi menjadi 3000 data citra anjing dan 3000 data citra kucing. Data yang telah dikumpulkan akan dipisah dengan rasio 80:20 dimana 80% data akan digunakan untuk training model ANN dan 20% sisanya akan digunakan untuk melakukan pengujian terhadap model yang telah dibangun.

### 2.3 Preprocessing

Pada tahap ini terdapat 2 proses yang akan dilakukan pada data sebelum dilakukan proses ekstraksi fitur yaitu proses mengubah ukuran citra dan proses perubahan citra berwarna menjadi citra *grayscale*. Pada proses mengubah ukuran data citra, citra yang awalnya memiliki dimensi ukuran 512 x 512 akan diubah menjadi ukuran yang lebih kecil yakni 256 x 256. Setelah perubahan ukuran citra, akan dilakukan proses perubahan citra menjadi citra *grayscale*.

Proses ini dilakukan dengan cara mengambil nilai 3 warna dasar pada setiap pixel yang kemudian akan dihitung untuk mendapat nilai rata-rata yang kemudian akan digunakan untuk nilai keabuan pada pixel tersebut [8].

#### 2.4 Ekstraksi Fitur

Pada penelitian ini, ekstraksi fitur akan dilakukan menggunakan pendekatan GLCM untuk melakukan analisis tekstur pada citra digital. Metode GLCM menggunakan hasil perhitungan dari matriks co-occurrence yang didapatkan melalui proses menghitung hubungan antara nilai-nilai piksel yang saling berdekatan dalam gambar, dengan mempertimbangkan dua parameter yaitu sudut (angle) sebagai arah dan jarak ketetanggaan antar pixel. Dalam metode GLCM, nilai sudut yang umum digunakan adalah  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ , sedangkan untuk jarak ketetanggaan antar pixel umumnya menggunakan nilai 1 sampai 3, dan seterusnya [3]. Setelah matriks GLCM dibentuk, nilai-nilai fitur statistik seperti *contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy* dan *correlation* dapat dihitung berdasarkan matriks ini [9].

$$a. \text{ Contrast} = \sum_{i,j=0}^{levels-1} p_{i,j} (i - j)^2 \tag{1}$$

$$b. \text{ Dissimilarity} = \sum_{i,j=0}^{levels-1} p_{i,j} |i - j| \tag{2}$$

$$c. \text{ Homogeneity} = \sum_{i,j=0}^{levels-1} \frac{p_{i,j}}{1 + (i - j)^2} \tag{3}$$

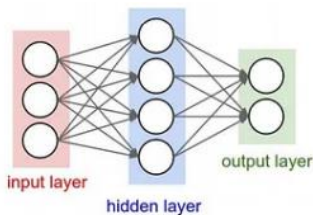
$$d. \text{ ASM} = \sum_{i,j=0}^{levels-1} p_{i,j}^2 \tag{4}$$

$$e. \text{ Energy} = \sqrt{ASM} \tag{5}$$

$$f. \text{ Correlation} = \sum_{i,j=0}^{levels-1} p_{i,j} \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \tag{6}$$

#### 2.5 Artificial Neural Network (ANN)

ANN merupakan model komputasi yang sering dipergunakan sebagai metode klasifikasi pada citra. ANN merupakan model yang terinspirasi oleh cara kerja neuron atau sel saraf yang terdapat di dalam otak manusia [10]. Algoritma ini bekerja secara *non-linear* dan terbentuk dari 3 lapisan yakni lapisan input, lapisan hidden dan juga lapisan output [3]. Diagram pada gambar 9 menggambarkan sebuah arsitektur ANN dimana setiap layer terdiri dari *neuron-neuron*. Jaringan ini menerima input data dan mengirimkannya melalui proses propagasi untuk menghasilkan *output*.



Gambar 2. Arsitektur ANN [11]



## 2.6 Evaluasi

Pengukuran kinerja dalam penelitian ini dilakukan dengan menghitung sejauh mana model dapat memprediksi hasil pengujian dengan benar atau salah. Hasil ini direpresentasikan dalam bentuk tabel yang disebut *confusion matrix*.

**Tabel 1. Confusion matrix**

		Kelas Prediksi	
		Positif	Negatif
Kelas Sebenarnya	Positif	TP	FN
	Negatif	FP	TN

Berikut adalah rumus yang digunakan untuk mengukur kinerja model dalam hal akurasi.

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + TN + FN} * 100\% \quad (7)$$

Dengan keterangan sebagai berikut:

- True Positive* (TP) = Jumlah data positif yang diprediksi dengan benar oleh model.
- False Negative* (FN) = Jumlah data negatif yang diprediksi dengan benar oleh model.
- False Positive* (FP) = Jumlah data negatif yang salah diprediksi sebagai positif oleh model.
- True Negative* (TN) = Jumlah data positif yang salah diprediksi sebagai negatif oleh model.

## 3. Hasil dan Diskusi

Kombinasi parameter yang digunakan untuk pengujian adalah jarak ketetanggaan pixel dan sudut yang digunakan pada GLCM. Jarak ketetanggaan pixel yang akan digunakan adalah 1, 2, 3, 4 dan 5. Kemudian arah GLCM dalam bentuk sudut yang digunakan adalah seluruh kombinasi yang mungkin dari sudut 0°, 45°, 90° dan 135°. Data hasil GLCM dari seluruh kombinasi parameter tersebut akan dilatih pada model ANN yang kemudian akan diujikan untuk mencari akurasi model dalam proses klasifikasi.

**Tabel 2. Hasil Uji Coba**

Jarak Ketetanggaan	Arah GLCM	Akurasi	Arah GLCM	Akurasi	Arah GLCM	Akurasi
1	0°	66%	0°, 90°	77%	0°, 45°, 90°	79%
	45°	68%	0°, 135°	74%	0°, 45°, 135°	73%
	90°	71%	45°, 90°	73%	0°, 90°, 135°	77%
	135°	66%	45°, 135°	66%	45°, 90°, 135°	73%
	0°, 45°	72%	90°, 135°	73%	0°, 45°, 90°, 135°	78%
2	0°	65%	0°, 90°	74%	0°, 45°, 90°	75%
	45°	68%	0°, 135°	73%	0°, 45°, 135°	74%
	90°	67%	45°, 90°	67%	0°, 90°, 135°	77%

Jarak Ketetangan	Arah GLCM	Akurasi	Arah GLCM	Akurasi	Arah GLCM	Akurasi
3	135°	64%	45°, 135°	66%	45°, 90°, 135°	69%
	0°, 45°	72%	90°, 135°	69%	0°, 45°, 90°, 135°	77%
	0°	63%	0°, 90°	72%	0°, 45°, 90°	71%
	45°	64%	0°, 135°	70%	0°, 45°, 135°	72%
	90°	66%	45°, 90°	67%	0°, 90°, 135°	71%
	135°	63%	45°, 135°	63%	45°, 90°, 135°	69%
	0°, 45°	71%	90°, 135°	68%	0°, 45°, 90°, 135°	73%
4	0°	61%	0°, 90°	68%	0°, 45°, 90°	70%
	45°	65%	0°, 135°	67%	0°, 45°, 135°	69%
	90°	64%	45°, 90°	66%	0°, 90°, 135°	71%
	135°	63%	45°, 135°	62%	45°, 90°, 135°	68%
	0°, 45°	66%	90°, 135°	67%	0°, 45°, 90°, 135°	71%
5	0°	61%	0°, 90°	66%	0°, 45°, 90°	68%
	45°	64%	0°, 135°	62%	0°, 45°, 135°	67%
	90°	65%	45°, 90°	65%	0°, 90°, 135°	68%
	135°	61%	45°, 135°	60%	45°, 90°, 135°	68%
	0°, 45°	64%	90°, 135°	65%	0°, 45°, 90°, 135°	71%

Berdasarkan hasil pengujian yang diperoleh, didapatkan bahwa kombinasi parameter dengan akurasi terbaik adalah jarak 1 dan kombinasi sudut 0°, 45° dan 90° yang menghasilkan akurasi 79% dengan hasil confusion matrix yang dapat dilihat pada tabel 3. Dari 1200 data yang digunakan untuk pengujian, terdapat 480 citra anjing yang dapat diklasifikasi dengan tepat, 133 citra anjing yang gagal diklasifikasi, 462 citra kucing yang dapat diklasifikasi dengan tepat dan 125 citra kucing yang gagal diklasifikasi.

**Tabel 3.** Confusion matrix hasil klasifikasi

Kelas Sebenarnya	Kelas Prediksi	
	Anjing	Kucing
Anjing	480	133
Kucing	125	462

#### 4. Kesimpulan

Dari hasil pengujian model klasifikasi hewan dengan ANN berbasis GLCM, diperoleh bahwa model ANN menghasilkan akurasi terbaik saat jarak ketetanggaan pixel yang digunakan pada GLCM adalah 1 dengan kombinasi sudut  $0^\circ$ ,  $45^\circ$  dan  $90^\circ$  dengan akurasi 79%. Dimana terdapat 480 citra anjing yang dapat diklasifikasi dengan tepat, 133 citra anjing yang gagal diklasifikasi, 462 citra kucing yang dapat diklasifikasi dengan tepat dan 125 citra kucing yang gagal diklasifikasi dari 1200 data citra yang digunakan pada proses pengujian.

#### Daftar Pustaka

- [1] A. G. Riyadi, I. P. Wardhani, and S. Widayati, "Klasifikasi Citra Anjing dan Kucing Menggunakan Metode Convolutional Neural Network (CNN)," Seminar Nasional Teknologi dan Komunikasi STI&K, vol. 5, no.1, 2019.
- [2] S. D. Salim, A. Suryadibrata, "Klasifikasi Anjing dan Kucing menggunakan Algoritma Linear Discriminant Analysis dan Support Vector Machine," ULTIMATICS, vol. 11, No. 1, 2021.
- [3] S. S. Panna, A. Alhamad, and K. C. Pelangi, "Pengenalan Ekspresi Wajah Pengemudi Berbasis Fitur *Eigenface* dan *Gray Level Co-Occurrence Matrix*," Jurnal Edukasi dan Penelitian Informatika, vol. 9, no.2, 2023.
- [4] I. P. R. Paramaditya, and C. Pramarta, "Implementasi Algoritma *Random Forest* Dalam Menentukan Kualitas Susu Sapi," JNATIA, vol. 1, No. 1, 2022.
- [5] Husdi, "Pengenalan Ekspresi Wajah Pengguna Elearning Menggunakan *Artificial Neural Network* dengan Fitur Ekstraksi *Local Binary Pattern* dan *Gray Level Co-occurrence Matrix*," Jurnal Ilmiah ILKOM, vol. 8, No. 3, 2016.
- [6] Larxel. (2020). Animal Faces, Version 1. Retrieved May 6, 2024, from: <https://www.kaggle.com/datasets/andrewmvd/animal-faces/verison/1>
- [7] C. Uslu. (2022). "Apa itu Kaggle?". [Online]. Available: <https://www.datacamp.com/blog/what-is-kaggle>
- [8] C. N. Santi, "Mengubah Citra Berwarna Menjadi Citra Grayscale dan Citra Biner", Jurnal Teknologi Informasi DINAMIK, vol. 16, no. 1, pp. 14-19, 2021.
- [9] M. Yunus, (2020). "Feature Extraction: Gray Level Co-occurrence Matrix (GLCM)". [Online]. Available: <https://yunusmuhammad007.medium.com/feature-extraction-gray-level-co-occurrence-matrix-qlcm-10c45b6d46a1>
- [10] F. Fathurrahman, "Penerapan *Artificial Neural Network* Untuk Klasifikasi Citra Teks Dalam Penerjemahan Bahasa Daerah, " SENAMIKA, 2020.
- [11] M. Yunus, (2020). "#6 Artificial Neural Network (ANN) – Part 1 (Pengenalan)". [Online]. Available: <https://yunusmuhammad007.medium.com/6-artificial-neural-network-ann-part-1-pengenalan-db487b81fd85>

# Deteksi Pneumonia dengan Ekstraksi Fitur Gray-Level Co-occurrence Matrix (GLCM) dan Support Vector Machine (SVM)

I Gusti Bagus Sutha Arianata Putra<sup>a1</sup>, Gst. Ayu Vida Mastrika Giri<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>bagussuthaap@gmail.com  
<sup>2</sup>vida@unud.ac.id

## Abstract

*Pneumonia, a prevalent lung disease globally, poses significant challenges in accurate diagnosis despite its severity. This paper proposes a novel approach leveraging Support Vector Machine (SVM) classification and Gray-Level Co-occurrence Matrix (GLCM) analysis on chest X-ray images to aid in pneumonia diagnosis. By extracting pneumonia-indicative features from digital X-ray images using Gray-Level Co-occurrence Matrix (GLCM) and employing Support Vector Machine (SVM) for classification, the study aims to enhance pneumonia diagnosis effectiveness, particularly crucial in regions with limited healthcare resources. The proposed method focuses on identifying characteristic patterns indicative of pneumonia in chest X-ray images and distinguishing between normal and pneumonia-affected images based on GLCM-extracted features. Furthermore, the study evaluates the impact of hyperparameter tuning using grid search on the proposed diagnostic system's performance, including accuracy, sensitivity, and specificity. By achieving these objectives, the research aims to contribute significantly to the development of more accurate and effective diagnostic tools for pneumonia, especially in resource-constrained areas.*

**Keywords:** Gray-Level Co-occurrence Matrix (GLCM), Machine Learning, Pneumonia, Support Vector Machine, X-Ray

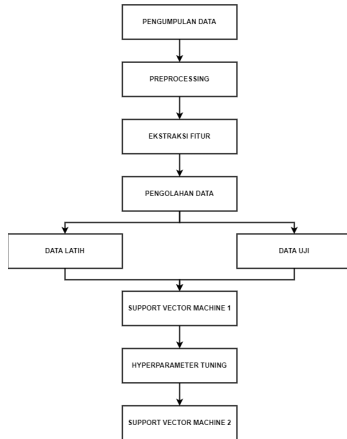
## 1. Pendahuluan

Penyakit paru-paru banyak terjadi di seluruh dunia, diantaranya berkaitan dengan penyakit paru obstruktif kronis, asma, tuberkulosis, fibrosis, dan pneumonia [1]. Pneumonia, atau dikenal juga dengan istilah paru-paru basah, adalah infeksi yang mengakibatkan peradangan pada kantong-kantong udara di salah satu atau kedua paru-paru. Biasanya pneumonia ini ditandai dengan peningkatan opasitas pada area paru-paru [2]. Pada pasien penderita pneumonia, terjadi peradangan dan pengisian cairan atau nanah pada sekelompok kecil kantong udara di ujung saluran pernapasan paru-paru, yang disebut alveoli. Akibatnya, para penderita pneumonia mengalami gejala batuk berdahak, demam, menggigil, nyeri dada, dan kesulitan bernapas. Bakteri, virus, dan jamur merupakan organisme yang dapat menyebabkan pneumonia atau paru-paru basah. Namun, pada penderita dewasa, kondisi ini lebih sering diakibatkan karena infeksi bakteri, meskipun kondisi tersebut juga dapat disebabkan oleh virus maupun jamur. Pneumonia dapat dikatakan salah satu penyebab kematian pada anak tertinggi di dunia. World Health Organization (WHO) memperkirakan bahwa penyakit ini telah menjadi pemicu 15% kematian anak-anak berusia di bawah 5 tahun [4]. Menurut Ourworldindata. Di Indonesia sendiri, data menunjukkan bahwa pada tahun 2019, estimasi tingkat kematian tahunan akibat pneumonia dan infeksi saluran pernapasan bawah lainnya adalah 28.43 per 100.000 penduduk dan estimasi tingkat kematian tahunan akibat pneumonia dan infeksi saluran pernapasan bawah lainnya untuk anak dibawah umur 5 tahun adalah 45.8 per 100,000 penduduk [3]. Studi ini bertujuan untuk mengusulkan dan mengevaluasi pendekatan dalam diagnosis pneumonia melalui analisis citra digital menggunakan metode Gray-Level Co-occurrence Matrix (GLCM) dan Support Vector

Machine (SVM). Tujuan utama adalah mengembangkan metode ekstraksi fitur berbasis GLCM untuk mengidentifikasi pola karakteristik pada citra rontgen dada yang mengindikasikan adanya pneumonia, dan menerapkan algoritma SVM sebagai classifier untuk membedakan antara citra rontgen normal dan citra rontgen yang menunjukkan tanda-tanda pneumonia berdasarkan fitur yang diekstraksi dari GLCM. Selain itu, studi ini juga bertujuan untuk mengukur pengaruh penyetalan *hyperparameter* menggunakan grid search dalam kinerja sistem diagnosis yang diusulkan dalam mendeteksi pneumonia, termasuk akurasi, sensitivitas, dan spesifisitasnya. Dengan mencapai tujuan ini, diharapkan penelitian ini dapat memberikan kontribusi signifikan dalam pengembangan alat bantu diagnosis yang lebih akurat dan efektif untuk pneumonia, terutama di daerah-daerah dengan sumber daya kesehatan yang terbatas. Beberapa penelitian sebelumnya telah mengusulkan metode-metode berbeda untuk diagnosis pneumonia. Beberapa menggunakan teknologi citra seperti pemrosesan citra digital, machine learning, dan deep learning. Metode-metode ini sering kali memanfaatkan fitur-fitur yang diekstraksi dari citra rontgen dada untuk membedakan antara citra rontgen normal dan yang menunjukkan tanda-tanda pneumonia. Pada beberapa penelitian menggunakan deep learning seperti cnn, telah menunjukkan bahwa deep learning telah sangat mampu dalam mendeteksi pneumonia dimana terutama metode CNN-ELM, yang dimana akurasinya mencapai 93,59% dan F1-Score mencapai nilai 0,9474 [2]. Penggunaan metode GLCM untuk ekstraksi fitur dan SVM sebagai classifier dalam penelitian ini dapat berpotensi memberikan kontribusi yang signifikan dalam memberikan perspektif baru dalam deteksi pneumonia menggunakan machine learning, meningkatkan akurasi dan efektivitas diagnosis pneumonia. Urgensi penelitian ini tak dapat diabaikan. Diagnosis yang cepat dan akurat adalah kunci untuk menangani pneumonia secara efektif. Keterlambatan dalam diagnosis bisa mengakibatkan risiko komplikasi serius, seperti gagal napas atau sepsis, yang berpotensi fatal. Oleh karena itu, pengembangan metode yang dapat mempercepat dan memperbaiki proses diagnosis merupakan prioritas utama dalam upaya penyelamatan nyawa. Penerapan teknologi ini diharapkan dapat menaikkan produktivitas radiolog dengan bantuan algoritma komputer dalam mendeteksi tanda-tanda pneumonia pada citra rontgen. Hal ini memungkinkan radiolog untuk lebih fokus pada kasus-kasus yang membutuhkan perhatian khusus, mengurangi beban kerja mereka dan meningkatkan efisiensi diagnosis secara keseluruhan. Dengan demikian, penelitian ini tidak hanya bertujuan untuk meningkatkan akurasi diagnosis pneumonia, tetapi juga memiliki implikasi yang luas dalam meningkatkan perawatan kesehatan masyarakat, terutama di daerah-daerah dengan sumber daya terbatas.

## 2. Metode Penelitian

Tahapan-tahapan penelitian yang ditampilkan dalam Gambar 1 yang menjelaskan langkah-langkah secara terperinci, mulai dari tahap awal persiapan data hingga tahap akhir evaluasi hasil. Setiap tahapan memiliki peran krusial dalam memastikan bahwa penelitian berjalan dengan validitas dan efektivitas yang maksimal. Mulai dari mengumpulkan dan mempersiapkan data, melakukan preprocessing untuk mempersiapkan data agar dapat digunakan dalam analisis, mengekstraksi fitur penting dari data, mengolah data dengan membagi dataset dan menjalankan algoritma machine learning, hingga melakukan evaluasi untuk mengukur kinerja model yang telah dikembangkan. Keseluruhan proses penelitian ini mengikuti rangkaian-rangkaian tahapan tersebut dengan tujuan sehingga mencapai pemahaman yang lebih mendalam tentang dataset yang digunakan serta memastikan performa yang optimal dari model yang dikembangkan.



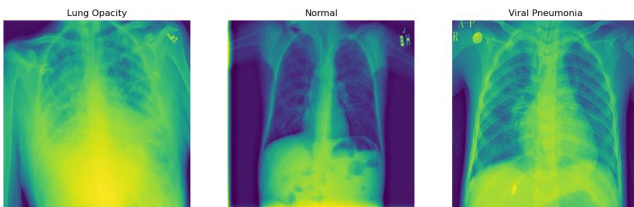
Gambar 1. Tahapan Penelitian

### 2.1. Persiapan Data

Data yang digunakan dalam penelitian ini diperoleh dari platform Kaggle dan dibuat oleh Fatemeh Mehrparvar dengan ukuran 37 mb yang dapat diakses pada link berikut:

<https://www.kaggle.com/datasets/fatemeMehrparvar/lung-disease> [10]

Pada dataset tersebut terdapat 3475 file jpg dibagi menjadi 3 klasifikasi citra yang berbeda. Berikut adalah contoh gambar dataset pada gambar 2.



Gambar 2. Dataset

### 2.2. Preprocessing

#### a. Data Balancing

Menyeimbangkan data dari setiap kelas adalah langkah penting untuk memastikan distribusi yang merata dalam dataset. Dalam kasus ini, total terdapat 3475 file JPG dengan

pembagian yang tidak merata antar kelas. Oleh karena itu, pendekatan yang diambil adalah dengan menghapus sebagian data agar setiap kelas memiliki jumlah sampel yang seimbang, yaitu sekitar 1000 file JPG per kelas.

b. Normalisasi Label

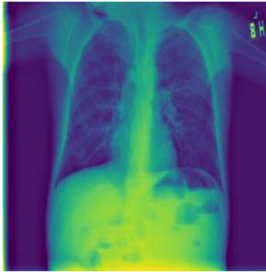
Normalisasi label digunakan untuk mengubah label dari format yang diberikan dalam nama folder menjadi format yang lebih sederhana dan standar. Proses ini menghapus spasi, tanda kurung, dan mengambil dua kata pertama dari label untuk menghasilkan label yang dinormalisasi.

c. Normalisasi Deskripsi

Normalisasi deskripsi digunakan untuk menghasilkan deskripsi yang dinormalisasi dari folder dan sub-folder. Proses ini menghapus angka, tanda titik, dan spasi berlebih, serta menggabungkan nama folder dan subfolder menjadi satu string.

d. CLAHE (Contrast Limited Adaptive Histogram Equalization)

CLAHE (Contrast Limited Adaptive Histogram Equalization) digunakan untuk meningkatkan kontras gambar dengan memperhatikan distribusi histogram lokal, sehingga membuat detail gambar yang rendah lebih jelas.



Original Image

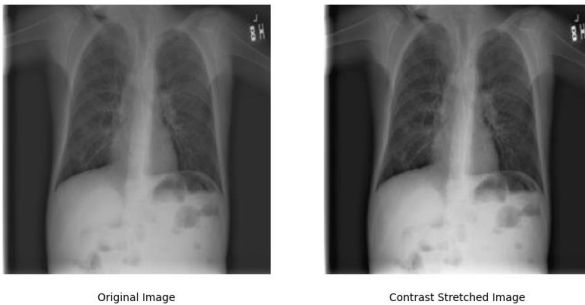


CLAHE Image

**Gambar 3.** Perbandingan Gambar Asli dan Contrast Stretched Image

e. Kontras Stretching

Kontras stretching adalah teknik yang digunakan untuk meningkatkan kontras gambar dengan meregangkan rentang intensitas piksel sehingga seluruh rentang intensitas dapat digunakan secara optimal.



**Gambar 4.** Perbandingan Gambar Asli dan Contrast Stretched Image

### 2.3. Ekstraksi Fitur

Ekstraksi fitur pada penelitian ini akan menggunakan pendekatan GLCM untuk melakukan analisis citra dari sudut pandang tekstur pada sebuah citra digital. Metode GLCM menggunakan hasil kalkulasi dari matriks co-occurrence yang didapatkan melalui proses menghitung hubungan antara nilai-nilai piksel yang saling berdekatan dalam gambar, dengan mempertimbangkan dua parameter: sudut (angle) piksel citra dan jarak ketetanggaan antar pixel. Dalam metode GLCM, nilai sudut yang umum digunakan adalah 0, 45, 90, dan 135 derajat, sedangkan untuk jarak ketetanggaan antar pixel dapat menggunakan nilai 1, 2, 3, dan seterusnya [5]. Setelah matriks GLCM terbentuk, nilai-nilai fitur statistik seperti *contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy* dan *correlation* dapat dihitung berdasarkan matriks ini. Berikut merupakan persamaan yang digunakan untuk menghasilkan nilai-nilai statistik pada metode GLCM [6].

$$a. \text{ Contrast} = \sum_{i,j=0}^{levels-1} p_{i,j} (i - j)^2 \quad (1)$$

$$b. \text{ Dissimilarity} = \sum_{i,j=0}^{levels-1} p_{i,j} |i - j| \quad (2)$$

$$c. \text{ Homogeneity} = \sum_{i,j=0}^{levels-1} \frac{p_{i,j}}{1 + (i - j)^2} \quad (3)$$

$$d. \text{ ASM} = \sum_{i,j=0}^{levels-1} p_{i,j}^2 \quad (4)$$

$$e. \text{ Energy} = \sqrt{ASM} \quad (5)$$

$$f. \text{ Correlation} = \sum_{i,j=0}^{levels-1} p_{i,j} \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (6)$$

### 2.4. Pengolahan Data

Pengolahan data dilakukan dengan membagi dataset yang telah melewati tahap sebelumnya menjadi 2 (dua) yaitu data uji dan data dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Setelah pembagian, data-data tersebut lalu dijalankan pada algoritma Support Vector Machine.

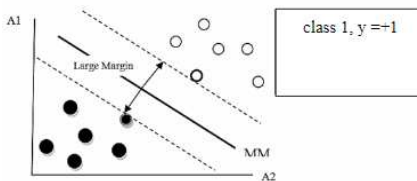


## 2.5. Grid Search

Grid search merupakan pencarian yang menyeluruh berdasarkan subset yang telah ditentukan dari ruang hiper-parameter. Hiper-parameter tersebut ditentukan dengan menggunakan nilai minimal (batas bawah), nilai maksimal (batas atas), dan jumlah langkah [8]. Grid Search akan membagi rentang parameter yang akan dioptimalkan menjadi sebuah grid dan melintasi semua titik untuk mendapatkan parameter optimal. Grid Search mengoptimalkan parameter SVM menggunakan teknik validasi silang sebagai metrik kinerja. Menurut Lin et al., menerapkan teknik validasi silang mampu mencegah overfitting [9]. Grid search bertujuan untuk mengidentifikasi kombinasi *hyperparameter* terbaik sehingga *klasifier* mampu memprediksi data-data yang tidak diketahui dengan benar. Sebelum melakukan penyetalan *hyperparameter* menggunakan Grid Search, model dilatih pada set data latih menggunakan algoritma Support Vector Machine (SVM). Prediksi kemudian dibuat dengan menggunakan set data uji. Setelah itu, untuk penyetalan *hyperparameter*, pendekatan Grid Search digunakan. Grid Search mencari kombinasi *hyperparameter* terbaik untuk model SVM dengan membagi rentang parameter yang akan dioptimalkan menjadi sebuah grid, dan akan melintasi semua titik sehingga mendapatkan parameter paling optimal. Dalam konteks penelitian ini, variabel yang diubah-ubah adalah parameter C dan gamma. Nilai-nilai yang digunakan untuk parameter 'C' adalah [0.1, 1, 10, 100, 1000] dan untuk parameter 'gamma' adalah [1, 0.1, 0.01, 0.001, 0.0001]. Rentang nilai ini dipilih untuk mencakup sejumlah besar kemungkinan nilai, sehingga memungkinkan Grid Search untuk mengeksplorasi ruang hiper-parameter dengan baik.

## 2.6. Support Vector Machine

Support Vector Machine (SVM) adalah sebuah sistem pembelajaran yang memanfaatkan ruang hipotesis berupa fungsi-fungsi linier dalam fitur yang memiliki dimensi tinggi [7]. SVM dilatih dengan menggunakan algoritma pembelajaran yang berbasis teori optimasi. Tingkat keakuratan yang dihasilkan oleh model SVM sangat bergantung pada pengaturan parameter dan jenis fungsi kernel yang digunakan. SVM dapat diklasifikasikan menjadi dua tipe, yaitu SVM linear dan SVM non-linear. SVM linear digunakan ketika data dapat dipisahkan secara linier, sementara SVM non-linear digunakan untuk data yang tidak bisa dipisahkan secara linier dan biasanya menggunakan kernel untuk memungkinkan pemisahan tersebut. Gambar 5 mengilustrasikan bagaimana SVM bekerja dengan menemukan *hyperplane* dengan margin maksimum yang memisahkan dua kelas.



Gambar 5. Ilustrasi Support Vector Machine

## 2.7. Evaluasi

Performa diukur dengan cara memeriksa berapa banyak pengujian yang berhasil diprediksi secara tepat dan berapa yang tidak tepat oleh model. Informasi ini biasanya disusun dalam sebuah tabel yang dikenal sebagai confusion matrix.

**Tabel 1. Confusion Matrix**

		Kelas Prediksi				
		Positif		Negatif		
Kelas Sebenarnya	Positif	TP	FN	Negatif	FP	TN

Berikut merupakan persamaan yang digunakan untuk mengukur performa model dalam bentuk akurasi.

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \% \tag{7}$$

Dengan keterangan sebagai berikut:

- True Positive (TP) = Jumlah data positif yang berhasil diklasifikasikan oleh model
- False Negative (FN) = Jumlah data positif yang diklasifikasikan salah oleh model
- False Positive (FP) = Jumlah data negatif yang diklasifikasikan salah oleh model
- True Negative (TN) = Jumlah data negatif yang diklasifikasikan benar oleh model

### 3. Hasil dan Diskusi

Pada pengujian yang dilakukan, kombinasi parameter yang digunakan adalah sudut yang digunakan pada GLCM dan hasil parameter C dan gamma setelah grid search. Kemudian arah GLCM dalam bentuk sudut yang digunakan adalah sudut 0°, 45°, 90° dan 135°. Data hasil GLCM sebelum dan sete tersebut akan dilatih pada model SVM yang kemudian akan diujikan untuk mencari akurasi model dalam proses klasifikasi. Akurasi dari hasil pengujian setelah dan sebelum penyetulan *hyperparameter* dapat dilihat pada tabel 2.

**Tabel 2. Hasil Uji Coba**

		Sebelum Hyperparameter Tuning				Setelah Hyperparameter Tuning			
		Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
GLCM 0	Lung Opacity	0.52	0.29	0.37	203	0.58	0.51	0.54	203
	Normal	0.47	0.62	0.53	206	0.64	0.51	0.57	206
	Viral Pneumonia	0.57	0.64	0.6	191	0.64	0.86	0.74	191
	Accuracy	-	-	0.51	600	-	-	0.62	600
	Macro Avg	0.52	0.52	0.5	600	0.62	0.63	0.62	600
	Weighted Avg	0.52	0.52	0.5	600	0.62	0.62	0.61	600
GLCM 45	Lung Opacity	0.35	1	0.52	203	0.6	0.69	0.64	203
	Normal	0	0	0	206	0.47	0.48	0.48	206

		Sebelum Hyperparameter Tuning				Setelah Hyperparameter Tuning			
		Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
	Viral Pneumonia	1	0.08	0.15	191	0.6	0.5	0.54	191
	Accuracy	-	-	0.36	600	-	-	0.56	600
	Macro Avg	0.45	0.36	0.22	600	0.56	0.56	0.55	600
	Weighted Avg	0.44	0.36	0.22	600	0.56	0.56	0.55	600
	Lung Opacity	0.35	1	0.52	203	0.68	0.72	0.7	203
	Normal	0	0	0	206	0.62	0.44	0.52	206
	Viral Pneumonia	1	0.08	0.15	191	0.69	0.85	0.76	191
	Accuracy	-	-	0.36	600	-	-	0.67	600
GLCM 90	Macro Avg	0.45	0.36	0.22	600	0.66	0.67	0.66	600
	Weighted Avg	0.44	0.36	0.22	600	0.66	0.67	0.66	600
	Lung Opacity	0.35	1	0.52	203	0.61	0.65	0.63	203
	Normal	0	0	0	206	0.5	0.5	0.5	206
GLCM 135	Viral Pneumonia	0.94	0.08	0.14	191	0.59	0.54	0.57	191
	Accuracy	-	-	0.36	600	-	-	0.56	600
	Macro Avg	0.43	0.36	0.22	600	0.57	0.56	0.56	600
	Weighted Avg	0.42	0.36	0.22	600	0.57	0.56	0.56	600
GLCM	Lung Opacity	0.35	1	0.52	203	0.73	0.67	0.7	203
COMBI NED	Normal	0	0	0	206	0.7	0.74	0.72	206
	Viral Pneumonia	1	0.08	0.15	191	0.9	0.92	0.91	191
	Accuracy	-	-	0.36	600	-	-	<b>0.77</b>	600
	Macro Avg	0.45	0.36	0.22	600	0.78	0.78	0.78	600

	Sebelum Hyperparameter Tuning				Setelah Hyperparameter Tuning			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Weighted Avg	0.44	0.36	0.22	600	0.77	0.77	0.77	600

Berdasarkan hasil pengujian, ditemukan bahwa model dengan konfigurasi GLCM dengan sudut yang dikombinasikan menghasilkan akurasi tertinggi, mencapai 0.77 setelah dilakukan penyetelan *hyperparameter* menggunakan metode grid search dimana parameternya adalah {'C': 10000, 'gamma': 0.0001, 'kernel': 'rbf'} dan score sebesar 0.7895833333333333. Sebaliknya, sebelum dilakukan penyetelan *hyperparameter parameter* model awal adalah {'C': 1.0, 'gamma': 'scale', 'kernel': 'rbf'} dengan hasil akurasi mencapai 0.36. Dimana berdasarkan data tersebut didapat bahwa rata-rata akurasi sebelum penyetelan *hyperparameter* mendapatkan rata rata akurasi sebesar 0.3975 dan setelah 0.6025 menunjukan bahwa grid search cukup efektif dalam mengoptimalkan kinerja model SVM. Perbaikan signifikan dari rata-rata akurasi sebesar 0.205 menunjukkan bahwa penyetelan *hyperparameter* menggunakan metode grid search berhasil meningkatkan kinerja model secara substansial. Hal ini menunjukkan pentingnya penyetelan *hyperparameter* dalam proses pengembangan model untuk mencapai kinerja yang optimal. Dengan menggunakan konfigurasi {'C': 10000, 'gamma': 0.0001, 'kernel': 'rbf'} setelah penyetelan *hyperparameter*, model SVM mampu mencapai akurasi sebesar 0.67, yang merupakan peningkatan signifikan dari akurasi sebelumnya sebesar 0.36. Konfigurasi ini menunjukkan bahwa nilai C yang tinggi (10000) mengindikasikan bahwa model SVM cenderung untuk memisahkan data dengan batas keputusan yang sangat ketat, sementara nilai gamma yang rendah (0.0001) menunjukkan bahwa pengaruh dari satu sampel terbatas pada jarak yang cukup jauh dari batas keputusan. Dengan demikian, hasil ini menegaskan bahwa pemilihan *hyperparameter* yang tepat sangat penting dalam meningkatkan kinerja model SVM, dan metode grid search adalah pendekatan yang efektif dalam menemukan kombinasi *hyperparameter* yang optimal.

#### 4. Kesimpulan

Dalam penelitian ini, langkah-langkah metodologi yang komprehensif telah dijalankan untuk menganalisis dataset gambar. Tahapan awal melibatkan persiapan data dengan melakukan normalisasi label dan deskripsi, serta penerapan teknik CLAHE dan kontras stretching untuk meningkatkan kualitas gambar. Selanjutnya, fitur-fitur tekstur diekstraksi menggunakan metode GLCM, yang memungkinkan analisis citra digital dari sudut pandang tekstur. Proses ini menghasilkan serangkaian fitur seperti kontras, dissimilarity, homogeneity, ASM, energy, dan correlation. Setelah data diproses, Data tersebut akan dibagi dua bagian yaitu data uji 20% dan data latih 80%, yang selanjutnya dianalisis menggunakan algoritma Support Vector Machine (SVM). Pentingnya tuning *hyperparameter* diakui dengan menerapkan metode grid search, yang memungkinkan identifikasi kombinasi *hyperparameter* terbaik untuk meningkatkan kinerja model dan mencegah overfitting. Evaluasi performa model dilakukan melalui confusion matrix dan mengukur akurasi. Hasil pengujian menunjukkan bahwa konfigurasi GLCM pada sudut yang dikombinasikan menghasilkan akurasi tertinggi setelah penyetelan *hyperparameter* menggunakan grid search, sementara model SVM sebelum penyetelan menunjukkan akurasi rendah. Dengan demikian, penelitian ini menegaskan pentingnya proses preprocessing, ekstraksi fitur, tuning *hyperparameter*, dan evaluasi model dalam mencapai hasil yang optimal dalam klasifikasi gambar menggunakan SVM.

#### Daftar Pustaka

- [1] S. Bharati, P. Podder, and R. Hossain, "Hybrid deep learning for detecting lung diseases from X-ray images," *Journal of Informatics in Medicine Unlocked*, vol. 20, 2020. DOI: 10.1016/j.imu.2020.100391.
- [2] B. Nugroho and E. Y. Puspaningrum, "Kinerja Metode CNN untuk Klasifikasi Pneumonia dengan Variasi Ukuran Citra Input," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*,

- vol. 8, no. 3, pp. 533-538, 2021. DOI: 10.25126/jtiik.202184515.
- [3] "Pneumonia death rates, age-standardized," IHME, Global Burden of Disease, 2019. [Online]. Available: <https://ourworldindata.org/grapher/pneumonia-death-rates-age-standardized?tab=table&time=2019..latest>.
- [4] "Pneumonia Fact Sheet," World Health Organization (WHO), 2020. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/pneumonia#:~:text=Pneumonia%20accounts%20for%2014%25%20of.and%20by%20addressing%20environmental%20factors>.
- [5] S. S. Panna, A. Alhamad, and K. C. Pelangi, "Pengenalan Ekspresi Wajah Pengemudi Berbasis Fitur Eigenface dan Gray Level Co-Occurance Matrice," Jurnal Edukasi dan Penelitian Informatika, vol. 9, no. 2, 2023.
- [6] M. Yunus, "Feature Extraction: Gray Level Co-occurrence Matrix (GLCM)," [Online]. Available: <https://yunusmuhammad007.medium.com/feature-extraction-gray-level-co-occurrence-matrix-glcm-10c45b6d46a1>.
- [7] I. M. Sulistiana and M. A. Muslim, "Optimasi Support Vector Machine (SVM) Menggunakan Grid Search dan Unigram untuk Meningkatkan Akurasi Ulasan E-Commerce," Journal of Self-Computing Exploration (JOSTex), vol. 11, no. 3, pp. 1234-1245, 2020. [Online]. Available: <https://jurnal.uns.ac.id/verstek>.
- [8] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM parameter optimization using grid search and genetic algorithm to improve classification performance," Telkomnika, vol. 14, no. 4, pp. 1502-1509, 2016.
- [9] S. W. Lin, K. C. Ying, S. C. Chen, and Z. J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," Expert Systems with Applications, vol. 35, no. 4, pp. 1817-1824, 2009.
- [10] F. Mehrparvar, "Lung Disease Dataset." [Online]. Available: <https://www.kaggle.com/datasets/fatemeMehrparvar/lung-disease>.

# Klasifikasi Citra Elektrokardiogram untuk Deteksi Penyakit Jantung Menggunakan Metode GLCM dan SVM

Andreas Panangian Tamba<sup>a1</sup>, I Gede Arta Wibawa<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>andreastamba04@gmail.com  
<sup>2</sup>gede.arta@unud.ac.id

## Abstract

*Heart disease is a major cause of death worldwide. Electrocardiogram (ECG) is a common method used to detect heart abnormalities. Analyzing ECG signals requires expertise and can be time-consuming. This study investigated the use of machine learning to classify ECG images for heart disease detection. The proposed method utilizes Gray Level Co-occurrence Matrix (GLCM) for feature extraction such as Dissimilarity, contrast, energy, ASM, homogeneity and Correlation. Meanwhile using Support Vector Machine (SVM) for the classification. We achieved an accuracy of 99.61% using this approach. The results suggest that the combination of GLCM and SVM can be a valuable tool for ECG image classification and potentially aid in early and accurate diagnosis of heart disease.*

**Keywords:** *Electrocardiography, Support Vector Machine, Gray Level Co-Occurrence Matrix, Classification, Myocardial Infarction*

## 1. Pendahuluan

Analisis dan interpretasi elektrokardiogram (ECG) adalah prosedur yang sangat penting dalam diagnosa dan pengawasan berbagai kondisi kardiovaskular. ECG merekam aktivitas listrik jantung dan mencatat impuls listrik yang bergerak melalui otot jantung, memberikan gambaran rinci tentang fungsi jantung (sitasi). ECG sangat penting di Indonesia, di mana penyakit kardiovaskular telah menjadi penyebab kematian terbanyak sejak tahun 1996. Keterlambatan kesadaran masyarakat tentang pentingnya menjaga kesehatan jantung telah menyebabkan peningkatan kematian akibat penyakit kardiovaskular. Namun, dengan kemajuan teknologi, ECG telah menjadi metode yang paling umum digunakan untuk mendeteksi kondisi kardiovaskular, menawarkan cara non-invasif dan relatif murah untuk diagnosa dan pengawasan kesehatan jantung [1]. Sinyal ECG normal biasanya terdiri dari tiga komponen utama: gelombang P, kompleks QRS, dan gelombang T. Gelombang P merepresentasikan aktivitas listrik atria ketika mereka berkontraksi untuk mengumpulkan darah ke ventrikel. Kompleks QRS merepresentasikan aktivitas listrik ventrikel ketika mereka berkontraksi untuk mengumpulkan darah keluar dari jantung. Gelombang T merepresentasikan aktivitas listrik ventrikel ketika mereka mengalami kontraksi. Komponen-komponen ini sangat penting dalam diagnosa berbagai kondisi kardiovaskular, termasuk aritmia, penyakit koroner, dan gagal jantung [2]. Analisis sinyal ECG adalah prosedur yang kompleks yang memerlukan pemahaman yang lebih dalam tentang morfologi sinyal ECG itu sendiri. Ini termasuk morfologi gelombang P, hubungan antara gelombang P dan kompleks QRS, serta keberadaan defleksi abnormal dalam lead seperti avR. Analisis ini sangat penting dalam diagnosa kardiovaskular, karena memberikan informasi yang sangat berharga tentang aktivitas listrik jantung dan keberadaan abnormalitas potensial. Namun, tidak semua profesional medis memiliki kemampuan untuk menganalisis sinyal ECG dengan akurat, yang mengapa sangat penting untuk memiliki pemahaman yang lebih dalam tentang ECG dan komponen-komponennya [3]. Analisis ECG dengan teknologi komputasi menggunakan teknik machine learning dan simulasi komputer untuk menganalisis sinyal ECG secara otomatis dan mendeteksi

abnormalitas. Pendekatan ini telah ditunjukkan efektif dalam mendeteksi aritmia dan kondisi kardiovaskular lainnya, serta dapat membantu mengurangi beban kerja profesional medis sementara meningkatkan akurasi diagnosa. Analisis ECG dengan teknologi komputasi juga memungkinkan pengembangan simulasi ECG yang disesuaikan, yang dapat digunakan untuk menerjemahkan hasil ECG dan menghubungkan perubahan struktural atau elektrofisiologis dengan abnormalitas ECG [4]. Melihat pengaruh dari ECG pada kondisi jantung, penulis berasumsi bahwa citra yang didapatkan dari gelombang-gelombang pada ECG dapat digunakan untuk klasifikasi menggunakan machine learning. Penelitian sebelumnya bertujuan untuk mengenali gangguan jantung, terutama fibrilasi atrium dan aritmia, melalui pemeriksaan sinyal elektrokardiogram (EKG) [5]. Dalam penelitian sebelumnya, berhasil menerapkan metode Support Vector Machine (SVM) untuk mengklasifikasi fibrilasi atrium dengan tingkat keakuratan sebesar 92,9% untuk spesifisitas dan 80,5% untuk sensitivitas. Di sisi lain, penelitian lain menggunakan SVM untuk mengklasifikasikan aritmia dengan tingkat keakuratan mencapai 92% [6]. Dalam konteks penelitian ini, menekankan penggunaan Grey Level Co-Occurrence Matrix (GLCM) untuk mengekstrak fitur tekstur dari citra rekaman EKG. Penerapan GLCM untuk ekstraksi fitur memberikan peningkatan pada hasil akurasi klasifikasi meskipun studi sebelumnya lebih menitikberatkan pada analisis sinyal langsung dari EKG.[7] Mengikuti jejak penelitian-penelitian yang serupa, penelitian ini mengadopsi metode-metode yang telah digunakan dalam penelitian sebelumnya. Pendekatan tersebut melibatkan ekstraksi fitur dari gelombang ECG menggunakan Gray Level Co-Occurrence Matrix (GLCM), yang diikuti dengan klasifikasi menggunakan model SVM. Diharapkan bahwa dengan mengoptimalkan parameter SVM, penelitian ini akan menghasilkan tingkat akurasi yang signifikan. Selain itu, diharapkan juga bahwa visualisasi yang diberikan dari Confusion Matrix akan memperjelas pemahaman terhadap hasil penelitian ini dalam konteks klasifikasi citra ECG.

## 2. Metode Penelitian

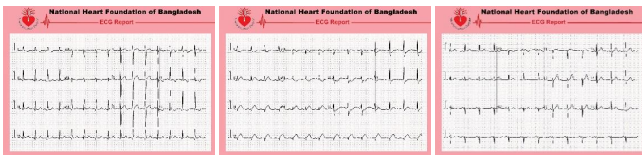
### 2.1. Akuisisi Data

Data yang digunakan pada penelitian ini adalah data sekunder. Data ini diakuisisi dari kaggle.com yaitu dataset ECG Image Dataset from the National Heart Foundation of Bangladesh. Dataset ini berisikan citra-citra yang dikategorikan menjadi 3 kelas: Abnormal Heartbeat Patients, Myocardial Infarction Patients, Normal Persons. Jumlah data tiap kelas adalah sebagai berikut:

Tabel 1. Jumlah Citra Pada Tiap Kelas

Kelas	Abnormal Heartbeat	Myocardial Infarction	Normal Person
Jumlah Citra	813	716	852

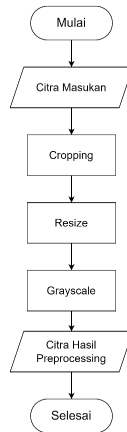
Dengan contoh citra gelombang ECG adalah sebagai berikut:



Gambar 1. Citra Gelombang ECG

### 2.2. Preprocessing

Setelah diberikan masukan berupa gambar dan label dari gambar-gambar tersebut, pada tahap *preprocessing image* akan dilakukan 3 proses seperti yang terdapat pada gambar 1.



Gambar 2. Flowchart Preprocessing

a. *Cropping*

Input berupa citra dari gambar gelombang, terdapat informasi yang tidak diperlukan yakni teks dari penjelasan citra tersebut. Hal itu tidak diperlukan karena tidak berpengaruh pada hasil klasifikasi. Maka akan dilakukan pemotongan yang akan menghasilkan area sebesar 2089x1217 px.

b. *Resize*

Image dengan ukuran 2089x1217 px tersebut merupakan ukuran yang berlebihan. Sehingga dilakukan *resize* citra menjadi lebih kecil dengan lebar 250. *Resize* dilakukan dengan tetap mempertahankan *aspect ratio* sehingga gambar hanya menjadi lebih kecil bukan dipotong.

c. *Grayscale*

Citra RGB yang merupakan matrix besar dengan 3 elemen di setiap pixelnya, diubah menjadi 1 dengan range 0-255 atau gambar menjadi hitam putih. Hal ini diperlukan untuk mendapatkan histogram dari agar dapat dilakukan ekstraksi fitur menggunakan GLCM.

### 2.3. Ekstraksi Fitur dengan GLCM

Level Co-occurrence Matrix (GLCM) adalah metode untuk mengekstrak fitur tekstur dari citra digital dengan menganalisis distribusi spasial dari intensitas piksel. GLCM memodelkan hubungan antara dua piksel pada jarak dan arah tertentu dalam citra [8]. Fitur-fitur seperti kontras, energi, homogenitas, dan korelasi dapat diekstrak dari GLCM untuk mewakili sifat tekstur citra. Ekstraksi Fitur menggunakan glcm disini menggunakan 4 sudut yang melambangkan arah yakni 135, 90, 45, dan 0 derajat [9][10]. Fitur yang diekstraksi dengan GLCM yakni *dissimilarity*, *correlation*, *homogeneity*, *contrast*, *asm*, *energy*. Setiap fitur tersebut di dapatkan menggunakan persamaan:



*Dissimilarity*

$$\sum_{i,j=0}^{levels-1} P_{i,j} |i - j| \tag{1}$$

*Correlation*

$$\sum_{i,j=0}^{levels-1} P_{i,j} \left[ \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \tag{2}$$

*Homogeneity*

$$\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{3}$$

*Contrast*

$$\sum_{i,j=0}^{levels-1} P_{i,j} (i - j)^2 \tag{4}$$

*ASM*

$$\sum_{i,j=0}^{levels-1} P_{i,j}^2 \tag{5}$$

*Energy*

$$\sqrt{ASM} \tag{6}$$

**2.4. Min-Max Normalization**

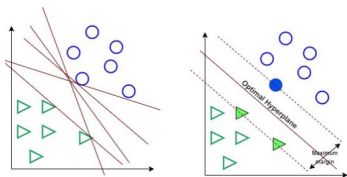
Selanjutnya, dengan menggunakan metode normalisasi min-max, nilai hasil ekstraksi fitur Co-occurrence Matrix Gray Level (GLCM) diskalakan ke rentang 0 hingga 1 [11]. Rumus standar metode normalisasi min-max adalah sebagai berikut:

$$v' = \frac{v - \text{min}A}{\text{max}A - \text{min}A} \tag{7}$$

Di mana nilai\_asli nilai\_asli adalah nilai yang diinginkan untuk dinormalisasi, nilai\_minimum adalah nilai minimum dari semua nilai, dan nilai\_maksimum adalah nilai maksimum dari semua nilai. Untuk menghindari bias yang mungkin muncul akibat perbedaan skala di antara fitur yang diekstraksi, teknik ini berguna.

**2.5. Support Vector Machine**

Algoritma pembelajaran mesin yang disebut Support Vector Machine (SVM) digunakan untuk menyelesaikan masalah klasifikasi dan regresi. SVM bekerja dengan menemukan garis pemisah terbaik, juga dikenal sebagai hyperplane, yang memisahkan data, yang memiliki nilai positif jika berada di kanan garis pemisah atau negatif jika berada di kiri garis pemisah. SVM dapat menangani regresi, klasifikasi, dan bahkan klusterisasi. Performansi algoritma ini sangat dipengaruhi oleh pemilihan hyperparameter SVM. Untuk ilustrasi, prediksi error pada SVM non-linear dipengaruhi secara signifikan oleh parameter C dan gamma. [12]. Untuk melakukan klasifikasi citra, SVM menggunakan label dari citra yang telah ditentukan sebelumnya (*supervised learning*). Input vektor dapat diubah atau dipetakan ke dimensi yang lebih besar oleh SVM. Data dipecah oleh *hyperplane* optimal dan dua *hyperplane* paralel lainnya yang terletak di tiap sisi dari *hyperplane* optimal. *Hyperparameters* adalah tiga parameter yang dimiliki SVM: kernel, C, dan Gamma. SVM, yang pertama kali dibuat oleh Vapnik dan Cortes pada tahun 1992, adalah optimal margin classifier. Salah satu metode klasifikasi pembelajaran mesin yang paling penting saat ini adalah SVM.



Gambar 3. Visualisasi Hyperplane pada SVM (sumber: Dicoding.com)

### 2.6. Grid Search

Grid Search adalah metode yang digunakan dalam pengoptimalan parameter pada model SVM, khususnya pada Support Vector Machines (SVM). Metode ini melibatkan variasi sistematis parameter dalam rentang yang telah ditentukan dan evaluasi performa model untuk setiap kombinasi. Tujuan dari Grid Search adalah untuk menemukan set parameter terbaik yang menghasilkan performa terbaik pada model. Grid Search penting karena memungkinkan eksplorasi parameter yang sangat luas, yang sangat penting dalam menemukan kombinasi parameter terbaik untuk model. Hal ini sangat signifikan pada SVM, di mana performa model sangat dipengaruhi oleh pilihan parameter dan fungsi kernel. Grid Search memastikan bahwa semua kombinasi parameter dipertimbangkan, yang dapat menghasilkan hasil yang lebih baik dibandingkan dengan metode lain yang mungkin tidak mempertimbangkan seluruh ruang parameter [13]. Grid Search bekerja dengan mendefinisikan rentang untuk setiap parameter dan kemudian membagi setiap rentang menjadi beberapa langkah yang telah ditentukan. Proses ini diulangi untuk semua parameter, menghasilkan grid dari semua kombinasi yang mungkin. Setiap kombinasi kemudian dievaluasi menggunakan metrik performa, seperti akurasi cross-validation, untuk menentukan set parameter terbaik. Proses ini biasanya diulangi beberapa kali untuk memastikan bahwa kombinasi terbaik tidak terlewatkan karena fluktuasi acak dalam data [12].

### 2.7. Confusion Matrix

Confusion Matrix adalah sebuah tabel yang digunakan untuk menampilkan hasil perhitungan dari data yang diklasifikasi sebagai benar atau salah yang dibandingkan dengan label sebenarnya pada data yang telah diklasifikasi. Tabel ini memperlihatkan jumlah prediksi yang benar dan salah yang dibuat oleh model, serta jumlah yang sebenarnya dari setiap kelas. Dengan demikian, Confusion Matrix memberikan gambaran yang lebih jelas tentang performa model dalam memprediksi kelas data yang diklasifikasi.

Tabel 2. Confusion Matrix 2 Kelas

		Kelas Prediksi	
		Positif	Negatif
Kelas Sebenarnya	Positif	TP	FN
	Negatif	FP	TN

Namun dikarenakan terdapat 3 kelas maka Confusion Matrix yang dibuat adalah sebagai berikut:

**Tabel 3. Confusion Matrix 3 Kelas**

Kelas Sebenarnya	Kelas Prediksi		
	Kelas 1	Kelas 2	Kelas 3
Kelas 1	T1	F2	F3
Kelas 2	F1	T2	F3
Kelas 3	F1	F2	T3

Dengan pengukuran akurasi adalah sebagai berikut

$$Accuracy = \frac{T1+T2+T3}{T1+F1+F1+T2+F2+F2+T3+F3+F3} \times 100\% \quad (8)$$

Dari confusion matrix tersebut, selain accuracy dapat dilakukan perhitungan nilai precision, recall, dan F1-Score, yang akan dijelaskan dengan persamaan dibawah ini [12]:

$$Precision_i = \frac{T_i}{T_i + \sum_{j \neq i} F_{ji}} \quad (9)$$

$$Recall_i = \frac{T_i}{T_i + \sum_{j \neq i} F_{ij}} \quad (10)$$

$$F1Score_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (11)$$

Keterangan:

$T_i$  : True Positif untuk kelas i

$F_{ji}$  : False Positif untuk kelas j yang diprediksi dengan kelas i

$F_{ij}$  : False Negatif untuk kelas i yang diprediksi sebagai kelas j

### 3. Hasil dan Diskusi

#### 3.1. Penentuan Parameter

Dalam penelitian ini, dilakukan pengujian parameter pada metode Support Vector Machine (SVM) dengan kernel rbf untuk membandingkan citra pada data EKG. Parameter yang diuji adalah nilai C (0.1, 1, 10, 100, 1000) dan gamma (0.0001, 0.001, 0.01, 0.1, 1). Proses grid search dilakukan untuk menemukan kombinasi parameter terbaik yang menghasilkan model SVM dengan performa optimal.

**Tabel 4. Akurasi C 0.1**

Param C	Param Gamma	Kernel	Akurasi
0.1	1	rbf	0.710871806
0.1	0.1	rbf	0.710871806
0.1	0.001	rbf	0.878177286
0.1	0.0001	rbf	0.858125989

**Tabel 5. Akurasi C 10**

Param C	Param Gamma	Kernel	Akurasi
10	1	rbf	0.994621027
10	0.1	rbf	0.994621027
10	0.001	rbf	0.995108826
10	0.0001	rbf	0.991195647

**Tabel 6. Akurasi C 100**

Param C	Param Gamma	Kernel	Akurasi
100	1	rbf	0.994621027
100	0.1	rbf	0.994621027
100	0.001	rbf	0.995108826
100	0.0001	rbf	0.994619828

**Tabel 7. Akurasi C 1000**

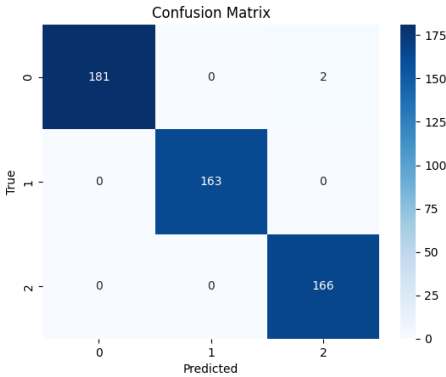
Param C	Param Gamma	Kernel	Akurasi
1000	1	rbf	0.994621027
1000	0.1	rbf	0.994621027
1000	0.001	rbf	0.995108826
1000	0.0001	rbf	0.994619828

Grid search dilakukan untuk menemukan kombinasi parameter terbaik dalam model Support Vector Machine (SVM) dengan kernel rbf untuk membandingkan citra pada data elektrokardiogram (EKG). Parameter yang diuji adalah nilai C dan gamma, dengan variasi nilai tertentu untuk setiap parameter. Dalam proses grid search, berbagai kombinasi nilai C dan gamma dievaluasi menggunakan metode cross-validation pada data pelatihan. Setiap kombinasi parameter tersebut digunakan untuk melatih model SVM dan kemudian dievaluasi kinerjanya menggunakan data uji. Hasil grid search ditampilkan dalam tabel-tabel akurasi, yang menunjukkan performa model SVM dengan berbagai nilai C dan gamma. Dari hasil ini, dapat dilihat bahwa variasi akurasi terjadi tergantung pada kombinasi parameter yang digunakan. Dari hasil pengujian parameter pada SVM dengan kernel rbf untuk membandingkan citra pada data EKG, dapat disimpulkan bahwa terdapat variasi dalam akurasi model tergantung pada nilai-nilai C dan gamma yang dipilih. Namun, pada rentang nilai yang diuji, model SVM cenderung memberikan akurasi yang tinggi, mencapai nilai tertinggi sekitar 99.5% dengan beberapa kombinasi parameter. Oleh karena itu, penting untuk melakukan grid search untuk menemukan parameter terbaik yang menghasilkan model SVM dengan performa tertinggi dalam membandingkan citra pada data EKG.

### 3.2. Confusion Matrix

Setelah menyelesaikan *grid search*, langkah selanjutnya adalah melakukan pengujian model menggunakan data testing. Tujuannya adalah untuk mengevaluasi kinerja SVM dengan parameter terbaik yang telah ditentukan sebelumnya. Pengujian ini penting untuk memastikan bahwa model mampu menggeneralisasi pola dengan baik pada data testing yang belum diuji. Evaluasi menggunakan pendekatan perhitungan beberapa metrik evaluasi standar, termasuk precision, accuracy, dan F1 score, yang akan dihitung berdasarkan confusion matrix. Melalui langkah ini, kesesuaian model dalam membandingkan citra pada data EKG dapat dipahami

secara lebih komprehensif, yang mana kritis untuk memvalidasi hasil grid search sebelumnya serta untuk memastikan bahwa model yang dihasilkan dapat diandalkan dalam konteks aplikasi praktis.



Gambar 4. Confusion Matrix

Hasil evaluasi model menunjukkan kinerja yang sangat baik, dengan angka akurasi, presisi, recall, dan F1 score mencapai tingkat yang sangat tinggi. Akurasi sebesar 99.61% menunjukkan tingkat keseluruhan keakuratan model dalam memprediksi kelas. Presisi sebesar 99.61% menggambarkan tingkat keakuratan dalam mengidentifikasi kelas positif, sedangkan recall sebesar 99.61% mengukur kemampuan model dalam menangkap semua instance yang sebenarnya positif. F1 score, yang merupakan rata-rata harmonis antara presisi dan recall, juga sangat tinggi, mencapai 99.61%. Dari hasil confusion matrix, terlihat bahwa model hanya melakukan sedikit kesalahan, dengan hanya 2 prediksi salah pada kelas pertama dan tidak ada prediksi yang salah pada kelas kedua dan ketiga. Ini menunjukkan bahwa model sangat baik dalam membedakan antara kelas dan memiliki kemampuan yang sangat baik dalam mengidentifikasi kelas yang berbeda dalam data EKG.

#### 4. Kesimpulan

Penelitian ini mengeksplorasi potensi metode GLCM dan SVM dalam mengklasifikasikan citra elektrokardiogram (EKG) untuk mendeteksi penyakit jantung. Hasil menunjukkan bahwa kombinasi GLCM dan SVM mampu mencapai akurasi klasifikasi yang tinggi, mencapai 99.61%. Hal ini menunjukkan efektivitas metode ini dalam membedakan antara kelas EKG yang berbeda, seperti Normal, Abnormal Heartbeat, dan Myocardial Infarction. Penggunaan parameter C dan gamma yang tepat dalam SVM terbukti penting untuk mencapai akurasi optimal. Meskipun penelitian ini menunjukkan hasil yang menjanjikan, beberapa keterbatasan perlu dipertimbangkan. Jumlah data yang digunakan relatif kecil dan hanya satu metode ekstraksi fitur (GLCM) yang dieksplorasi. Selain itu, model SVM belum diuji dengan data pasien nyata. Penelitian selanjutnya direkomendasikan untuk mengatasi keterbatasan ini. Penggunaan data yang lebih besar dan beragam, serta eksplorasi metode ekstraksi fitur lain, dapat meningkatkan keandalan dan generalisasi metode ini. Pengujian model SVM dengan data pasien nyata juga penting untuk memvalidasi efektivitasnya dalam konteks klinis. Secara keseluruhan, penelitian ini memberikan bukti awal bahwa metode GLCM dan SVM memiliki potensi sebagai alat bantu yang efektif untuk diagnosis penyakit jantung melalui analisis citra EKG.

#### Daftar Pustaka

- [1] M. Rifali and D. Irmawati, "Sistem Cerdas Deteksi Sinyal Elektrokardiogram (EKG) untuk Klasifikasi Jantung Normal dan Abnormal Menggunakan Jaringan Syaraf Tiruan (JST)," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 4, no. 1, pp. 49–55, 2019, doi: 10.21831/elinvo.v4i1.28242.
- [2] A. Lyon, A. Mincholé, J. P. Martínez, P. Laguna, and B. Rodriguez, "Computational techniques for ECG analysis and interpretation in light of their contribution to medical advances," *J. R. Soc. Interface*, vol. 15, no. 138, 2018, doi: 10.1098/rsif.2017.0821.
- [3] N. Rafie, A. H. Kashou, and P. A. Noseworthy, "ECG Interpretation: Clinical Relevance, Challenges, and Advances," *Hearts*, vol. 2, no. 4, pp. 505–513, 2021, doi: 10.3390/hearts2040039.
- [4] X. Xu, S. Jeong, and J. Li, "Interpretation of Electrocardiogram (ECG) Rhythm by Combined CNN and BiLSTM," *IEEE Access*, vol. 8, pp. 125380–125388, 2020, doi: 10.1109/ACCESS.2020.3006707.
- [5] M. Rofii, "Identifikasi Fibrilasi Atrium Pada Isyarat Elektrokardiogram (Ekg) Menggunakan Support Vector Machine (Svm)," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 1, pp. 231–240, 2018, doi: 10.24176/simet.v9i1.2080.
- [6] Q. Ayuni, R. Cahya Wihandika, and N. Yudistira, "Klasifikasi Aritmia Dari Hasil Elektrokardiogram Menggunakan Metode Support Vector Machine," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, pp. 2163–2170, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] M. Sofie and A. Rizal, "Klasifikasi Citra Rekaman Sinyal Elektrokardiogram Menggunakan Metode Analisis Tekstur, K-Nn Dan Multilayer Perceptron," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 7, no. 1, p. 233, 2016, doi: 10.24176/simet.v7i1.509.
- [8] N. Neneng, K. Adi, and R. Isnanto, "Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM)," *J. Sist. Inf. Bisnis*, vol. 6, no. 1, p. 1, 2016, doi: 10.21456/vol6iss1pp1-10.
- [9] S. Handayani, A. Zuhdi, and R. Shofiati, "Implementation of Gray Level Co-occurrence Matrix (GLCM) and Support Vector Machine (SVM) Methods for Recognition of Batik Bekasi Motifs Implementation of Gray Level Co-occurrence Matrix (GLCM) and Support Vector Machine (SVM) Methods for Recognition of B," *Intelmatics*, vol. 2, no. 2, pp. 67–72, 2022, doi: 10.25105/itm.v2i2.14423.
- [10] L. M. Wisudawati, "Klasifikasi Tumor Jinak Dan Tumor Ganas Pada Citra Mammogram Menggunakan Gray Level Co-Occurrence Matrix (Glcm) Dan Support Vector Machine (Svm)," *J. Ilm. Inform. Komput.*, vol. 26, no. 2, pp. 176–186, 2021, doi: 10.35760/ik.2021.v26i2.4897.
- [11] I. B. W. B. Adi Putra, L. G. Astuti, A. E. Karyawati, I. W. Santiyasa, C. R. A. Pramatha, and I. G. S. Astawa, "Diagnosis Penyakit Retinopati Diabetes Menggunakan SVM dengan Optimasi Algoritma Genetika," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 11, no. 3, p. 457, 2022, doi: 10.24843/jlk.2023.v11.i03.p01.
- [12] M. Fajri and A. Primajaya, "Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search," *J. Appl. Informatics Comput.*, vol. 7, no. 1, pp. 14–19, 2023, doi: 10.30871/jaic.v7i1.5004.
- [13] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 4, p. 1502, 2016, doi: 10.12928/telkomnika.v14i4.3956.

Halaman ini sengaja dibiarkan kosong

# Pengklasifikasian Kualitas Pisang dengan Deep Learning CNN Arsitektur VGG16

Vodka Joe Junior<sup>a1</sup>, I Gede Santi Astawa<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>bagusuthaap@gmail.com  
<sup>2</sup>santi.astawa@unud.ac.id

## Abstract

*Bananas are one of the most popular fruits consumed worldwide, valued for their nutritional benefits and versatility in various dishes. However, ensuring banana quality, including ripeness and integrity, remains crucial in meeting consumer expectations and maintaining supply chain standards. Manual classification of banana quality can be tedious, prompting the need for efficient methods. In this study, we explore the classification of banana quality using Convolutional Neural Network (CNN) with VGG16 architecture and image augmentation. Leveraging previous research and considering the superior performance of VGG16, we gathered data from Kaggle and evaluated our model's accuracy. The implementation yielded promising results, achieving a peak accuracy of 97.50% with 15 epochs and an 80%-20% training-validation data split. This surpasses previous methods, indicating the effectiveness of CNN with VGG16 in banana quality classification.*

**Keywords:** *Banana quality, Convolutional Neural Network, VGG16 architecture, Image augmentation, Classification accuracy.*

## 1. Pendahuluan

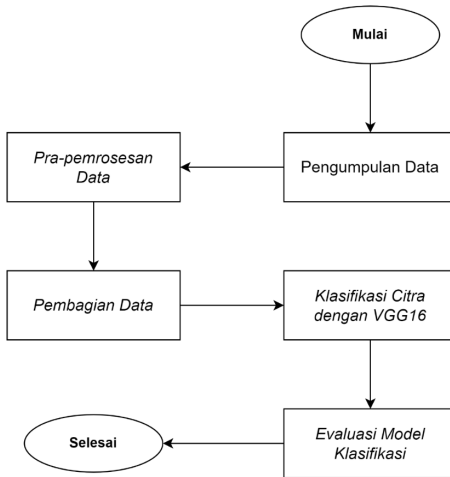
Pisang merupakan buah yang amat terkenal dan diminati di berbagai belahan dunia. Kaya akan nutrisi, Pisang merupakan sumber karbohidrat yang kaya, terdiri dari glukosa, fruktosa, dan serat makanan. Buah ini juga kaya akan vitamin dan mineral penting, seperti magnesium, vitamin C, kalium, dan vitamin B6 [1]. Berkat kandungan dan manfaat yang kaya dan beragam, pisang dianggap sebagai makanan sehat dan penghasil sumber energi ideal. Pisang bisa dikonsumsi dalam berbagai cara, baik sebagai buah utuh maupun sebagai bahan dalam berbagai hidangan seperti kue, dan roti pisang. Varietas pisang seperti pisang raja, pisang ambon, dan lainnya memberikan variasi dalam hal ukuran, rasa, dan tekstur, memberikan pengalaman memasak dan menikmati buah ini yang beragam. Permintaan global terhadap pisang meningkat, mendorong perlunya pengembangan metode yang efisien untuk mengidentifikasi kualitasnya. Kesegaran, tingkat kematangan dan aspek-aspek lainnya dari pisang merupakan faktor penting dalam memenuhi harapan konsumen dan menjaga standar kualitas produk selama distribusi. Namun, mengklasifikasikan kualitas pisang secara manual bisa menjadi pekerjaan yang sangat melelahkan [2]. Dalam era teknologi informasi saat ini, ada dampak besar terhadap cara data diproses, terutama dalam konteks citra. Teknologi informasi memiliki berbagai aplikasi, salah satunya adalah pengolahan citra digital. Pengolahan citra digital merupakan bidang ilmu yang mempelajari cara mengubah dan menganalisis gambar digital dengan memodifikasi piksel-pikselya untuk mencapai tujuan tertentu [3]. Salah satu cara menggunakan pengolahan citra digital adalah melalui proses klasifikasi. Klasifikasi citra melibatkan pengelompokan elemen citra ke dalam kelas-kelas tertentu, di mana setiap kelas menggambarkan karakteristik objek tertentu sehingga objek tersebut dapat diidentifikasi [4]. Metode yang paling populer dalam pengolahan citra digital saat ini adalah *Convolutional Neural Network* (CNN). CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP) dan termasuk dalam kategori algoritma *Deep Learning*. Sebelum CNN menjadi primadona, berbagai metode lain telah digunakan, namun dengan keterbatasannya masing-masing. Terdapat berbagai algoritma yang digunakan untuk



mengklasifikasikan gambar. Biasanya, orang akan mengekstraksi fitur dari gambar dan memasukkannya ke dalam beberapa algoritma klasifikasi seperti SVM [5]. Ada beberapa studi sebelumnya mengenai klasifikasi citra yang telah dilakukan contohnya seperti penelitian yang telah dilakukan Yusuf Amrozi, dkk yang melakukan klasifikasi jenis pisang berdasarkan citra warna menggunakan metode SVM pada tahun 2022, mendapatkan akurasi sebesar 89.86% [6]; penelitian yang telah dilakukan Rifki Kosasih yang melakukan klasifikasi tingkat kematangan pisang menggunakan ekstraksi fitur tekstur dan algoritma KNN, mendapatkan akurasi sebesar 88,89% [7]; dan penelitian yang dilakukan oleh Imam Muslem R, dkk yang melakukan klasifikasi citra ikan dengan algoritma CNN dengan arsitektur VGG-16 dan mendapatkan akurasi sebesar 99,5% [8]. Mengacu pada tinjauan literatur dan penelitian sebelumnya, penulis tertarik untuk mengeksplorasi klasifikasi kualitas pisang menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur VGG16 dan menerapkan Augmentasi Citra. Keputusan untuk menggunakan VGG16 didasarkan pada kinerjanya yang lebih baik daripada beberapa metode lainnya. Data yang digunakan diambil dari *platform* Kaggle, dan evaluasi dilakukan dengan menggunakan metrik akurasi.

## 2. Metode Penelitian

### 2.1. Alur Penelitian



Gambar 1. Alur Penelitian

Studi ini diawali dengan mengumpulkan dataset berupa gambar pisang yang telah diberi label sebagai segar atau busuk. Data tersebut kemudian dipersiapkan melalui proses pra-pemrosesan, termasuk penerapan augmentasi gambar dan penyesuaian ukuran gambar agar sesuai untuk pelatihan model. Sebelum proses pelatihan, data dibagi menjadi set pelatihan dan set validasi. Kemudian, model CNN dengan arsitektur VGG16 diimplementasikan dan disesuaikan. Model tersebut dilatih menggunakan dataset gambar untuk mengklasifikasikan kondisi pisang. Setelah

pelatihan selesai, dilakukan evaluasi untuk menilai tingkat akurasi prediksi model terhadap dataset gambar.

## 2.2. Pengumpulan Data

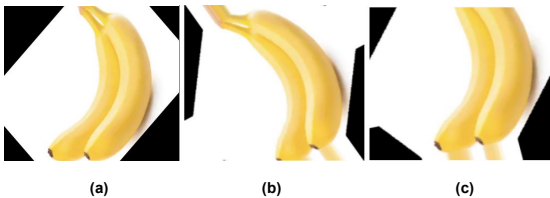
Sumber data yang digunakan untuk penelitian ini adalah data sekunder yang didapatkan melalui platform Kaggle, yang terdiri dari 1000 gambar. Data lalu dibagi menjadi dua kategori, yaitu segar dan busuk, dengan masing-masing kelas memiliki 500 gambar, dan memiliki berbagai resolusi gambar. Contoh-contoh gambar pisang asli dapat dilihat pada Gambar 2.

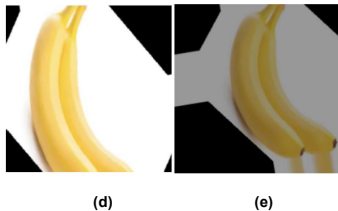


Gambar 2. (a) Gambar Pisang Segar, (b) Gambar Pisang Busuk

## 2.3. Pra-pemrosesan Data

Data citra yang telah dikumpulkan dari berbagai sumber memiliki resolusi yang berbeda-beda. Untuk memastikan konsistensi dan memenuhi persyaratan input model, semua citra disesuaikan ukurannya menjadi 224x224 piksel. Penyesuaian ukuran ini penting karena model yang digunakan, VGG16, dirancang untuk menerima input citra dengan ukuran tersebut. Setelah penyesuaian ukuran, dilakukan augmentasi data untuk meningkatkan variasi dalam dataset pelatihan, yang dapat membantu mengurangi *overfitting* dan meningkatkan kemampuan generalisasi model. Augmentasi data yang dilakukan meliputi rotasi citra secara acak dengan sudut  $40^\circ$ , citra digeser secara acak pada sumbu horizontal dan vertikal sebesar maksimal 20% dari lebar dan tinggi citra. Citra diperbesar atau diperkecil secara acak dengan faktor zoom maksimal 20% lalu citra dibalik secara horizontal secara acak dan Kecerahan citra diubah secara acak dalam rentang yang ditentukan untuk menambah variasi kondisi pencahayaan.





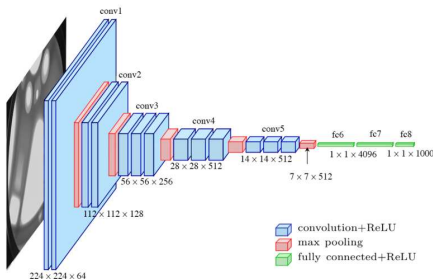
**Gambar 3.** (a) Sebelum pra-pemrosesan, (b) Rotasi 40°, (c) Pemangkasan (*cropping*), (d) *Flip Horizontal* (e) *Brightness*

#### 2.4. Pembagian Data

Penelitian ini membagi data menggunakan metode Split Validations. Metode ini memisahkan data menjadi dua bagian dengan rasio tertentu. Pada penelitian ini, data dibagi menjadi 80% untuk data latihan dan 20% untuk data validasi, yang dipilih secara acak.

#### 2.5. CNN Arsitektur VGG16 untuk Klasifikasi Citra

CNN merupakan salah satu metode populer dalam *Deep Learning* untuk mengklasifikasikan citra[6]. Ini adalah metode dalam bidang *Deep Learning* yang digunakan untuk mengelompokkan citra berdasarkan kesamaan dan juga dapat mengenali konteks dalam suatu adegan. Pada tahun 2014, Oxford Visual Geometry Group (VGG) mengeluarkan dua model CNN (*Convolutional Neural Network*) dengan 16 dan 19 lapisan, yang menerapkan filter konvolusi berukuran 3x3. [9]. Penelitian ini menggunakan model VGG16, yang terdiri dari 16-layer secara keseluruhan. Di antara *layer-layer* tersebut, 13-layer merupakan *layer* konvolusi dan 3-layer lainnya merupakan *layer fully connected*. [10]. VGG16 menerapkan filter 3x3 pada seluruh lapisan konvolusi dengan langkah satu. Setelah setiap lapisan konvolusi, diikuti oleh lapisan pooling maksimum. Ukuran masukan untuk VGG16 ditetapkan menjadi 224x224 piksel dengan tiga saluran gambar. Berikut adalah gambar dari arsitektur VGG16:



**Gambar 4.** Arsitektur VGG16

Dalam studi ini, model dilatih dengan data yang dibagi menjadi 80% sebagai data latihan.

## 2.6. Evaluasi

Evaluasi dilakukan dengan menghitung akurasi model CNN dalam melakukan klasifikasi citra pisang. Pengujian akurasi dilakukan pada tahap pengujian untuk mengukur tingkat keakuratan sistem dalam melakukan klasifikasi kualitas pisang. Evaluasi akurasi dilakukan melalui beberapa epoch dan data yang telah terbagi. Nilai akurasi dihitung menggunakan suatu persamaan (1).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Keterangan:

TP = True Positive  
TN = True Negative  
FP = False Positive  
FN = False Negative

## 3. Hasil dan Diskusi

Penulis menggunakan bahasa pemrograman python untuk mengimplementasikan teknik augmentasi data guna meningkatkan variasi dalam dataset pelatihan. Data yang digunakan berasal dari google drive, yang meliputi gambar yang telah dikategorikan untuk pelatihan model klasifikasi biner. Augmentasi data dilakukan menggunakan *ImageDataGenerator* dari *tensorflow.keras*, yang menerapkan transformasi seperti rotasi acak hingga 40 derajat, pergeseran lebar dan tinggi hingga 20%, transformasi *shear*, *zoom* acak, pencerminan *horizontal*, dan penyesuaian kecerahan. Transformasi ini bertujuan untuk membuat model lebih *robust* dan mengurangi risiko *overfitting* dengan memperkenalkan variasi realistis pada gambar yang digunakan selama pelatihan. Adapun source code dapat dilihat pada Gambar 5:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1./255)

path_to_train_data = '/content/drive/My Drive/testing'
path_to_validation_data = '/content/drive/My Drive/validation'

train_generator = train_datagen.flow_from_directory(
    path_to_train_data,
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary'
)

validation_generator = validation_datagen.flow_from_directory(
    path_to_validation_data,
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary'
)
```

Gambar 5. Augmentasi

Setelah tahap preprocessing data selesai, langkah berikutnya adalah membagi data. Pembagian ini dilakukan menggunakan split validation, dengan 80% data sebagai data latih dan 20% sebagai data validasi. Kemudian, pengujian terhadap akurasi dilakukan melalui beragam nilai epoch pada

arsitektur VGG16 yang sudah dibuat. Kombinasi yang digunakan dalam pengujian meliputi epoch 5, 10, 15 dengan batch 32. Hasil loss dan akurasi dari pembagian data dan epoch 5, 10, 15 hasil bisa dilihat pada tabel 1.

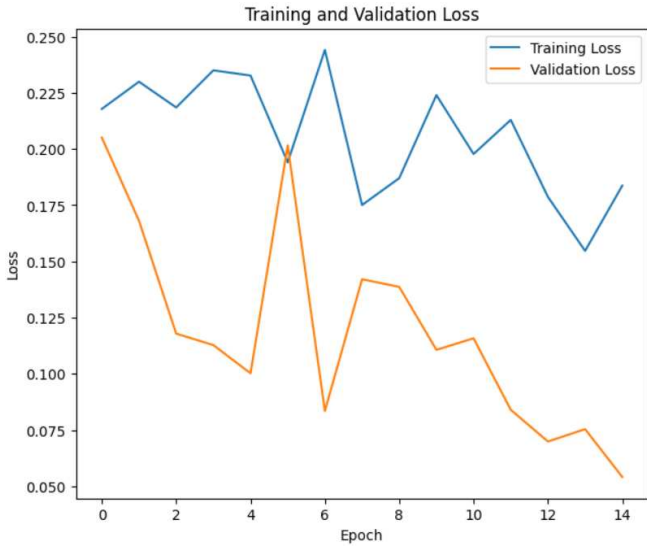
**Tabel 1.** Hasil Dari Loss dan Akurasi

Data Latih	Data Validasi	Epoch	Loss	Akurasi(%)
80%	20%	5	0.1597	0.9438
		10	0.1821	0.9250
		15	0.0541	0.9750

Berdasarkan tabel tersebut, dapat disimpulkan bahwa jumlah epoch yang menghasilkan statistik terbaik yaitu loss terendah dan akurasi tertinggi adalah model dengan 15 epoch, yang memiliki akurasi validasi sebesar 97,50%.



**Gambar 6.** Grafik Training and Validation Accuracy



**Gambar 7.** Grafik Training and Validation Loss

Penguji lalu mencoba melakukan klasifikasi kepada dua buah gambar pisang yang satu memiliki kondisi yang terlihat seperti busuk akan tetapi masih layak untuk dikonsumsi, dan yang satu lagi memiliki kondisi yang buruk dan tidak layak untuk dikonsumsi.



**Gambar 8.** Pisang Layak Dikonsumsi



Gambar 9. Pisang Tidak Layak Dikonsumsi

Penulis mendapatkan hasil memuaskan yang Dimana pada gambar 8, penulis mendapatkan hasil identifikasi bahwa pisang tergolong dengan label segar, dan pada gambar 9 penulis mendapatkan hasil identifikasi bahwa pisang tergolong dengan label busuk.

```
1/1 [=====] - 1s 759ms/step  
Model memprediksi gambar ini sebagai: Segar
```

Gambar 10. Hasil Gambar 8

```
1/1 [=====] - 1s 1s/step  
Model memprediksi gambar ini sebagai: Busuk
```

Gambar 11. Hasil Gambar 9

#### 4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penggunaan CNN (*Convolutional Neural Network*) dengan VGG16 sangat efektif untuk klasifikasi kualitas pisang. Akurasi tertinggi, sebesar 97,50%, dicapai dengan menggunakan 15 epoch dan pembagian data 80% untuk pelatihan serta 20% untuk validasi. Angka ini jauh lebih tinggi dibandingkan penelitian sebelumnya oleh Rifki Kosasih, yang menggunakan ekstraksi fitur tekstur dan algoritma KNN dengan akurasi 88,89% [7].

#### Daftar Pustaka

- [1] F. K. Arinta, F. S. Pranata, and Y. R. Swasti, "Potensi daging buah pisang dan kulit pisang (Musaceae) untuk peningkatan kualitas roti dan kue," *Teknologi Pangan: Media Informasi dan Komunikasi Ilmiah Teknologi Pertanian*, vol. 12, no. 2, pp. 185–196, Sep. 2021, doi: 10.35891/tp.v12i2.2416.
- [2] D. Armiady and I. Muslem R, "KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Kualitas Buah Pisang Berdasarkan Citra Buah Menggunakan Stochastic Gradient Descent," *Media Online*, vol. 4, no. 2, pp. 1207–1215, 2023, doi: 10.30865/klik.v4i2.1243.
- [3] S. R. Fakultas, T. Informasi, U. Islam, K. Muhammad, and A. Al Banjari, "Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm," 2020.
- [4] A. Herdiansah, R. I. Borman, D. Numaningsih, A. A. J. Sinlae, and R. R. Al Hakim, "Klasifikasi Citra Daun Herbal Dengan Menggunakan Backpropagation Neural Networks Berdasarkan Ekstraksi Ciri Bentuk," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, p. 388, Apr. 2022, doi: 10.30865/jurikom.v9i2.4066.
- [5] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," 2020. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>

- [6] Y. Amrozi, D. Yulianti, A. Susilo, N. Novianto, and R. Ramadhan, "Klasifikasi Jenis Buah Pisang Berdasarkan Citra Warna dengan Metode SVM," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 3, pp. 394–399, Dec. 2022, doi: 10.32736/sisfokom.v11i3.1502.
- [7] R. Kosasih *et al.*, "Classification of Banana Ripe Level Based on Texture Features and KNN Algorithms," 2021.
- [8] I. R. Muslem and T. M. Johan, "KLIK: Kajian Ilmiah Informatika dan Komputer Klasifikasi Citra Ikan Menggunakan Algoritma Convolutional Neural Network dengan Arsitektur VGG-16," *Media Online*, vol. 4, no. 2, pp. 978–985, 2023, doi: 10.30865/kiik.v4i2.1209.
- [9] S. Kumaresan, K. S. J. Aultrin, S. S. Kumar, and M. D. Anand, "Transfer Learning with CNN for Classification of Weld Defect," *IEEE Access*, vol. 9, pp. 95097–95108, 2021, doi: 10.1109/ACCESS.2021.3093487.
- [10] E. Tanuwijaya *et al.*, "Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia Classification of Indonesian Spices Digital Image using Modified VGG 16 Architecture Article Info ABSTRAK," vol. 21, no. 1, 2021, doi: 10.30812/matrik.v21i1.xxx.



Halaman ini sengaja dibiarkan kosong

# Pengembangan Algoritma Caesar Cipher dalam Game-Based Learning untuk Pendidikan Kriptografi pada Anak

Putu Arya Dharma Kesuma<sup>a1</sup>, I Gede Surya Rahayuda<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana

Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

<sup>1</sup>aryadharmakesuma@gmail.com

<sup>2</sup>igedesuryarahayuda@unud.ac.id

## Abstract

*The decline in children's interest in learning due to easy access to games is the main reason I designed game-based learning. To make learning more engaging, presenting materials in an interactive way is essential. Game-based learning addresses this issue by packaging educational content attractively, increasing children's interest in learning. This game will introduce the world of IT to children, particularly cyber security. One key concept is cryptography, using the Caesar cipher algorithm due to its simplicity and suitability for children. The Caesar cipher enables both encryption and decryption processes, making it perfect for teaching cryptography basics. By designing the game as an interactive mystery, it aims to enhance children's learning interest and introduce them to cyber security effectively, making learning both fun and educational.*

**Keywords:** Game-Based Learning, Caesar Cipher, Cryptography

## 1. Pendahuluan

Dalam era digital yang terus berkembang dan mudahnya akses alat elektronik, anak-anak cenderung lebih tertarik untuk menghabiskan waktu mereka dengan bermain game daripada fokus pada proses belajar. Fenomena ini tidaklah mengejutkan. Game online mempunyai daya tarik yang membuat para siswa lebih senang bermain ketimbang belajar. Game online juga dapat menyebabkan ketagihan ketika sedang bermain kemudian kalah anak-anak akan mencoba lagi sampai menang [1]. Game Based Learning merupakan metode pembelajaran yang menggunakan aplikasi permainan atau game yang telah dirancang khusus untuk membantu proses belajar dan membantu meningkatkan minat siswa dalam belajar [2]. Dengan berkembangnya zaman tentu cara penyajian materi pembelajaran harus berubah menjadi lebih menarik, salah satunya yaitu menyajikan materi pembelajaran melalui game. Pada game yang akan dikembangkan membahas mengenai pengenalan dunia IT kepada anak-anak salah satu konsep yang akan diimplementasikan adalah kriptografi, kriptografi merupakan bagian ilmu yang mempelajari tentang cara menjaga agar data atau pesan tetap aman [3], algoritma yang kriptografi yang akan digunakan adalah caesar cipher, alasan dipilihnya algoritma ini adalah mudahnya dipahami maka dari itu algoritma ini dipilih untuk membantu pengembangan game yang bertujuan untuk memperkenalkan dunia IT kepada anak-anak. Caesar cipher adalah salah satu algoritma enkripsi yang cukup mudah untuk dipahami, cara kerja dari algoritma tersebut adalah menggeser setiap huruf dalam pesan untuk menjaga keamanan data yang dikirim [4], seberapa jauh pergeseran huruf yang akan terjadi tergantung dari key yang digunakan. Diharapkan dengan dikembangkannya game-based learning ini nantinya dapat menarik minat anak-anak terhadap dunia IT dan memberikan sedikit pembelajaran tentang bagaimana dunia IT terkhususnya pada bagian cyber security. Cyber security adalah konsep dimana memproteksi informasi dari serangan cyber salah satunya adalah dengan metode kriptografi [5], dengan menggunakan kriptografi kita dapat menyamarkan pesan yang sebenarnya menjadi pesan yang sulit dibaca dan hanya dapat diubah kembali ke pesan yang dapat dibaca jika sudah mengetahui key dari pesan tersebut.

## 2. Metode Penelitian

### 2.1. Game Based Learning

Game Based Learning merupakan metode pembelajaran yang menggunakan aplikasi permainan atau game yang telah dirancang khusus untuk membantu proses belajar dan membantu meningkatkan minat siswa dalam belajar. Dalam rancangan ini akan memperkenalkan siswa tentang dunia keamanan digital, salah satu bidang ilmu yang diterapkan adalah kriptografi.

### 2.2. Algoritma Caesar Cipher

Dengan key yang telah ditentukan cara kerja algoritma yang digunakan pada penulisan ini adalah dengan menggeser huruf awal menjadi cipher. Jika Key yang digunakan adalah angka 2 maka pergeseran huruf yang terjadi adalah sebagai berikut:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Gambar 1. Pergeseran Huruf

Pada gambar 1 terjadi pergeseran 2 huruf dari A menjadi C, dari B menjadi D dan seterusnya, pergeseran 2 huruf ini terjadi dikarenakan penggunaan angka 2 pada key yang digunakan untuk melakukan enkripsi, jadi seberapa jauh pergeseran terjadi tergantung pada key yang digunakan nantinya.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 2. Konversi Huruf

Pada gambar 2 dapat dilihat konversi huruf menjadi angka, tujuan huruf diubah menjadi angka ini adalah agar mempermudah pergeseran huruf pada saat implementasi menggunakan code. berikut adalah perhitungan manual menggunakan rumus sebelum diimplementasikan dengan code.

#### Keterangan:

PT = Plain Text  
 CT = Cipher Text  
 K = key

#### Enkripsi:

PT = Arya  
 K = 2

CT =  $(PT + K) \bmod 26$   
 $= 0 + 2$   
 $= 2 \bmod 26$   
 $= 2 (C)$   
 $= C$

CT =  $(PT + K) \bmod 26$

$$\begin{aligned} &= 17 + 2 \\ &= 19 \text{ mod } 26 \\ &= 19 \text{ (T)} \\ &= \mathbf{T} \end{aligned}$$

$$\begin{aligned} \text{CT} &= (\text{PT} + \text{K}) \text{ mod } 26 \\ &= 24 + 2 \\ &= 26 \text{ mod } 26 \\ &= 0 \text{ (A)} \\ &= \mathbf{A} \end{aligned}$$

$$\begin{aligned} \text{CT} &= (\text{PT} + \text{K}) \text{ mod } 26 \\ &= 0 + 2 \\ &= 2 \text{ mod } 26 \\ &= 2 \text{ (C)} \\ &= \mathbf{C} \end{aligned}$$

**Cipher Text = CTAC**

**Dekripsi:**

$$\begin{aligned} \text{CT} &= \mathbf{CTAC} \\ \text{K} &= 2 \end{aligned}$$

$$\begin{aligned} \text{PT} &= (\text{CT} - \text{K}) \text{ mod } 26 \\ &= 2 - 2 \\ &= 0 \text{ mod } 26 \\ &= 0 \text{ (A)} \\ &= \mathbf{A} \end{aligned}$$

$$\begin{aligned} \text{PT} &= (\text{CT} - \text{K}) \text{ mod } 26 \\ &= 19 - 2 \\ &= 17 \text{ mod } 26 \\ &= 17 \text{ (R)} \\ &= \mathbf{R} \end{aligned}$$

$$\begin{aligned} \text{PT} &= (\text{CT} - \text{K}) \text{ mod } 26 \\ &= 0 - 2 \\ &= -2 \text{ mod } 26 \\ &= 24 \text{ (Y)} \\ &= \mathbf{Y} \end{aligned}$$

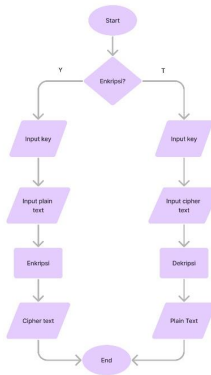
$$\begin{aligned} \text{PT} &= (\text{CT} - \text{K}) \text{ mod } 26 \\ &= 0 \text{ mod } 26 \\ &= 0 \text{ (A)} \\ &= \mathbf{A} \end{aligned}$$

**Plain Text = Arya**

### 3. Hasil dan Diskusi

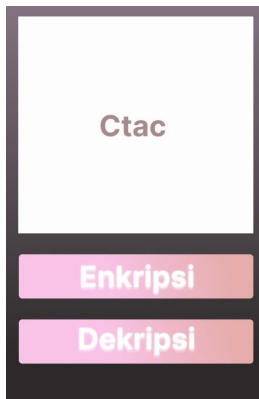
#### 3.1. Rancangan Sistem

Pada perancangan sistem ini akan menggambarkan bagaimana flow dari program nantinya dan gambaran bagaimana rancangan sistem pada game.



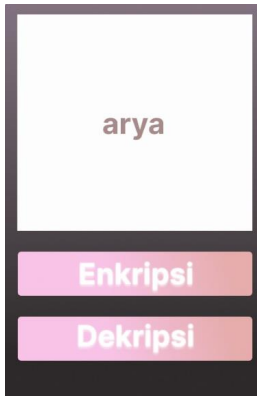
Gambar 3. Alur Sistem

Pada gambar 3 dijelaskan bagaimana alur sistem, ketika user memilih enkripsi user pertama harus memilih angka berapa yang digunakan sebagai key, lalu memasukan plain text kemudian sistem akan melakukan enkripsi kemudian memberikan output ciphertext, begitupun sebaliknya ketika user memilih dekripsi, pertama user akan memilih key kemudian memasukan ciphertext kemudian proses lainnya sama seperti bagian enkripsi.



Gambar 4. Design Input

Pada gambar 4 dapat dilihat "Text field" merupakan tempat user mengetikkan kalimat lalu dibawahnya adalah opsi yang dapat dipilih oleh user untuk melakukan enkripsi ataupun dekripsi.

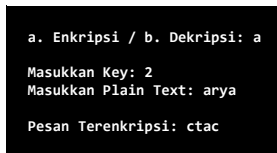


Gambar 5. Design Output

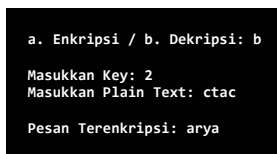
Pada gambar 5 dapat dilihat itu merupakan tempat hasil dari proses yang dipilih oleh user.

### 3.2. Pengujian Sistem

Pengujian sistem bermaksud untuk memastikan keakuratan output yang diberikan oleh sistem.



Gambar 6. Enkripsi



Gambar 7. Dekripsi

Pada gambar 6 dan 7 dapat dilihat bahwa sistem enkripsi dan dekripsi dapat berjalan dengan baik dan menghasilkan output yang akurat.

#### 4. Kesimpulan

Dari percobaan yang telah dilakukan, dapat disimpulkan bahwa algoritma Caesar cipher mudah untuk dipahami. Oleh karena itu, pemilihan algoritma ini dalam rancangan game berbasis edukasi sangat tepat, karena kesederhanaannya memungkinkan pengguna mempelajarinya dengan mudah dan dapat digunakan untuk pengembangan game yang berbasis pembelajaran.

#### Daftar Pustaka

- [1] N. Nisrinafatin, "Pengaruh Game Online Terhadap Motivasi Belajar Siswa", JENFOL, vol. 1, no. 1, pp. 135-142, Mar. 2020.
- [2] R. Oktavia, "Game Based Learning Meningkatkan Efektivitas Belajar Siswa," Center for Open Science, Jun. 2022. Accessed: May 10, 2024. [Online]. Available: <http://dx.doi.org/10.31219/osf.io/6aeuy>
- [3] Yhopi Suhelna, "Perancangan Aplikasi Penyandian Pesan Teks dengan Menggunakan Algoritma Digraph Cipher", JUKI, vol. 2, no. 1, pp. 25–34, May 2020.
- [4] M. A. F. Rachman, "Perancangan Aplikasi Memo Menggunakan Algoritma Kriptografi Caesar Cipher Dan Rsa Berbasis Android," Semin. Nas. Inov. dan Apl. Teknol. di Ind., pp. 121–127, 2018.
- [5] F. Indah, A. Sidabuntar, and N. Anissa, "Peran Cyber Security Terhadap Keamanan Data Penduduk Negara Indonesia (Studi Kasus: Hacker Bjorka)", Jurnal Bidang Penelitian informatika, vol.1,no.1, pp. 1-8,2022.

# Memanfaatkan NodeMCU dalam Serangan Jaringan Wi-Fi pada Frekuensi 2,4 GHz: Tinjauan Keamanan dan Tindakan Pencegahan

Ida Bagus Wahyu Semara Kamajaya<sup>a1</sup>, I Gusti Agung Gede Arya Kadyanan<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>kamajaya.2208561038@student.unud.ac.id  
<sup>2</sup>gungde@unud.ac.id

## Abstract

*NodeMCU, an open-source development platform based on the ESP8266, has become a significant tool in network security testing. This article investigates the use of NodeMCU in attacks against Wi-Fi networks on the 2.4 GHz frequency. We conducted experiments to launch NodeMCU's capabilities to exploit security weaknesses in Wi-Fi networks, including attacks against WEP, WPA, and WPA2 encryption. We also present preventive measures that can be implemented to protect Wi-Fi networks from attacks possible using NodeMCU.*

**Keywords:** NodeMCU, ESP8266, Wi-Fi network, security attack, encryption, WEP, WPA, WPA2

## 1. Pendahuluan

Dengan berkembangnya teknologi IoT (Internet of Things), ketersediaan jaringan Wi-Fi menjadi semakin penting. Namun, bersamaan dengan peningkatan ketersediaan, keamanan jaringan Wi-Fi juga menjadi isu yang mendesak. Sejumlah besar jaringan Wi-Fi masih rentan terhadap serangan berbagai jenis, mulai dari serangan penyusupan hingga serangan pengintaian data. Dalam beberapa tahun terakhir, NodeMCU telah muncul sebagai alat yang kuat dalam pengujian keamanan jaringan Wi-Fi. Dengan platform open-source yang dapat diprogram ulang, NodeMCU menyediakan fleksibilitas yang besar bagi peneliti keamanan untuk mengembangkan serangan baru dan memvalidasi kelemahan keamanan pada jaringan Wi-Fi yang menyebabkan terjadi banyak kelemahan yang terjadi pada jaringan Wi-Fi terutama pada frekuensi 2,4 Ghz.

## 2. Metode Penelitian

### 2.1. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental untuk mengeksplorasi dan mendemonstrasikan berbagai jenis serangan jaringan Wi-Fi menggunakan NodeMCU pada frekuensi 2,4 GHz. Studi ini juga mencakup analisis kerentanan jaringan dan evaluasi efektivitas tindakan pencegahan yang diusulkan.



## 2.2. Alat dan Bahan



Gambar 1. NodeMCU ESP 8266

- a. **NodeMCU (ESP8266):** Alat utama untuk melakukan serangan.
- b. **Router Wi-Fi:** Menyediakan jaringan Wi-Fi 2,4 GHz dengan pengaturan keamanan yang dapat diubah-ubah (WEP, WPA2, WPA3).
- c. **Laptop/PC:** Digunakan untuk pemrograman NodeMCU, analisis data, dan pemantauan jaringan.
- d. **Perangkat Lunak:**
  - **Arduino IDE:** Untuk memprogram NodeMCU.
  - **Wireshark:** Untuk menangkap dan menganalisis paket data.
  - **Aircrack-ng suite:** Untuk mendukung serangan jaringan.
  - **Alat IDS (Intrusion Detection System):** Untuk mendeteksi serangan

## 2.3. Prosedur Penelitian

### a. Persiapan dan Konfigurasi Alat

```
ESP8266-EvtTwin [Arduino 1.8.13]
File Edit Sketch Tools Help

ESP8266-EvtTwin
}

String _correct = "";
String _tryPassword = "";

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.promiscuous_enable();
  WiFi.config(IPAddress(192, 168, 4, 1), IPAddress(192, 168, 4, 1), IPAddress(255, 255, 255, 0));
  WiFi.softAP("EvtTwin", "12345678");
  dnsserver.start(53, "", IPAddress(192, 168, 4, 1));

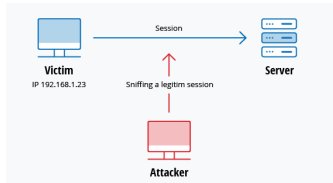
  webServer.on("/", handleIndex);
  webServer.on("/result", handleResult);
  webServer.on("/admin", handleAdmin);
  webServer.onNotFound(handleIndex);
  webServer.begin();
}

void performScan() {
  int n = WiFi.scanNetworks();
  clearArray();
  if (n == 0) {
    for (int i = 0; i < n || i < 16; ++i) {
      _network network;
      network.ssid = WiFi.SSID(i);
    }
  }
}
```

Gambar 2. Source Code Sniffing, Spoofing, DoS

- **Konfigurasi NodeMCU:** Memprogram NodeMCU menggunakan Arduino IDE untuk melakukan berbagai jenis serangan seperti sniffing, spoofing, dan denial-of-service (DoS).
- **Pengaturan Jaringan Wi-Fi:** Mengonfigurasi router Wi-Fi untuk beroperasi pada frekuensi 2,4 GHz dengan pengaturan keamanan yang bervariasi (WEP, WPA2, dan WPA3).

**b. Pelaksanaan Eksperimen**



**Gambar 3.** Alur Proses Eksperimen Sniffing

- **Serangan Sniffing:** Menangkap paket data menggunakan NodeMCU dan menganalisisnya dengan Wireshark.

Start deauthing    Start EvilTwin

SSID	BSSID	Channel	Select
samsul2	58:af:f1:a3:ac:b8	1	<input checked="" type="button" value="Selected"/>
Gauri :D	6e:ff:7b:2fc3:ad	8	<input type="button" value="Select"/>
samsul	5c:92:5e:bf:bb:81	8	<input type="button" value="Select"/>
realme C55	92:ff:47:b6:1e:99	11	<input type="button" value="Select"/>

**Gambar 4.** Pelaksanaan Eksperimen Serangan Spoofing dan DoS

- **Serangan Spoofing:** NodeMCU digunakan untuk menyamar sebagai perangkat lain di jaringan, mencoba mendapatkan akses yang tidak sah.
- **Serangan DoS:** NodeMCU diprogram untuk mengirimkan lalu lintas palsu ke jaringan Wi-Fi untuk mengganggu layanan.

**c. Analisis Kerentanan dan Dampak**

Titik Lemah	Deskripsi	Kemungkinan Eksploitasi
1. Password Wi-Fi yang lemah	Password Wi-Fi yang mudah ditebak atau tidak memenuhi standar keamanan	Tinggi
2. Konfigurasi default	Konfigurasi default pada router Wi-Fi yang tidak diubah	Sedang
3. Update firmware yang tidak terbaru	Firmware router Wi-Fi yang tidak diupdate	Sedang
4. Enkripsi yang tidak aktif	Enkripsi Wi-Fi yang tidak diaktifkan atau tidak digunakan	Tinggi
5. Port yang tidak dikonfigurasi	Port yang tidak dikonfigurasi dengan benar	Sedang

**Gambar 5.** Analisis Kerentanan

- **Analisis Kerentanan:** Mengidentifikasi titik lemah dalam pengaturan keamanan jaringan Wi-Fi yang dapat dieksploitasi oleh NodeMCU.

Dampak	Deskripsi	Tingkat Keparahan
1. Gangguan Layanan	Gangguan pada layanan Wi-Fi yang menyebabkan pengguna tidak dapat mengakses internet	Tinggi
2. Kebocoran Data	Kebocoran data pengguna yang sensitif, seperti password atau informasi pribadi	Sangat Tinggi
3. Akses Tidak Sah	Akses tidak sah ke jaringan Wi-Fi yang dapat menyebabkan kerusakan pada sistem atau data	Tinggi
4. Malware dan Virus	Penyebaran malware dan virus yang dapat menyebabkan kerusakan pada perangkat pengguna	Sedang

**Gambar 6.** Analisis Dampak

- **Dampak Serangan:** Penilaian dampak potensial dari serangan tersebut terhadap pengguna individu dan jaringan organisasi, termasuk gangguan layanan dan potensi kebocoran data.

**d. Pengujian Tindakan Pencegahan**

Tindakan Pencegahan	Deskripsi
1. Enkripsi WPA3	Menggunakan enkripsi WPA3 yang lebih aman dibandingkan dengan versi sebelumnya
2. Segregasi Jaringan	Memisahkan jaringan Wi-Fi menjadi beberapa bagian yang berbeda untuk mengurangi dampak serangan
3. Penggunaan IDS	Menggunakan sistem deteksi intrusi (IDS) untuk mendeteksi dan mencegah serangan

**Gambar 7.** Pengujian Tindakan Pencegahan

- **Implementasi Tindakan Pencegahan:** Menerapkan langkah-langkah keamanan yang direkomendasikan seperti enkripsi WPA3, segregasi jaringan, dan penggunaan IDS.

Tindakan	Deskripsi	Hasil
1. Serangan dengan NodeMCU	Mengulangi serangan dengan NodeMCU setelah tindakan pencegahan diterapkan	Gagal
2. Perbandingan dengan hasil sebelum penerapan tindakan pencegahan	Membandingkan hasil serangan dengan hasil sebelum penerapan tindakan pencegahan	Hasil serangan lebih rendah

**Gambar 8.** Pengujian Efektivitas

- **Pengujian Efektivitas:** Mengulangi serangan dengan NodeMCU setelah tindakan pencegahan diterapkan untuk mengevaluasi keberhasilan serangan dan membandingkannya dengan hasil sebelum penerapan tindakan pencegahan.

#### 2.4. Analisis Data

Data yang diperoleh dari eksperimen dianalisis secara kualitatif dan kuantitatif untuk:

- Menilai efektivitas NodeMCU dalam melaksanakan berbagai jenis serangan jaringan.
- Mengukur dampak serangan terhadap kinerja dan keamanan jaringan.
- Mengevaluasi efektivitas tindakan pencegahan dalam mengurangi risiko serangan.

#### 2.5. Validasi dan Reliabilitas

Untuk memastikan validitas dan reliabilitas hasil penelitian, eksperimen diulang beberapa kali dan hasilnya dibandingkan. Selain itu, berbagai skenario jaringan dengan pengaturan keamanan yang berbeda digunakan untuk mengevaluasi konsistensi temuan.

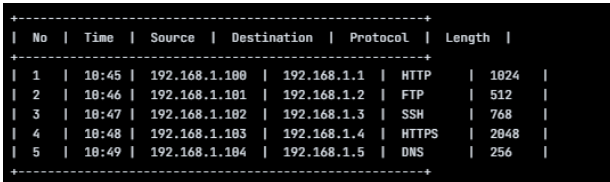
#### 2.6. Etika Penelitian

Semua eksperimen dilakukan dalam lingkungan pengujian yang aman dan terkendali tanpa membahayakan jaringan atau data nyata. Penelitian ini sepenuhnya mematuhi standar etika dan hukum terkait dengan keamanan siber dan privasi.

### 3. Hasil dan Diskusi

#### 3.1. Hasil Eksperimen

##### a. Serangan Sniffing

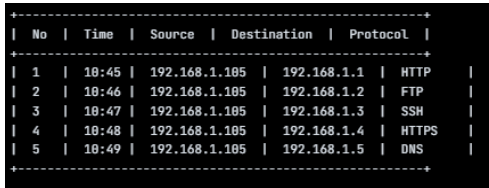


No	Time	Source	Destination	Protocol	Length
1	18:45	192.168.1.180	192.168.1.1	HTTP	1824
2	18:46	192.168.1.101	192.168.1.2	FTP	512
3	18:47	192.168.1.102	192.168.1.3	SSH	768
4	18:48	192.168.1.103	192.168.1.4	HTTPS	2048
5	18:49	192.168.1.104	192.168.1.5	DNS	256

Gambar 9. Hasil Penangkapan Paket Data Menggunakan Wireshark

Pada eksperimen sniffing, NodeMCU berhasil menangkap paket data yang dikirimkan melalui jaringan Wi-Fi 2,4 GHz. Paket yang tertangkap mencakup data yang tidak terenkripsi serta informasi tentang perangkat yang terhubung ke jaringan. Analisis menggunakan Wireshark menunjukkan bahwa beberapa paket berisi informasi sensitif yang dapat digunakan untuk serangan lebih lanjut.

### b. Serangan Spoofing

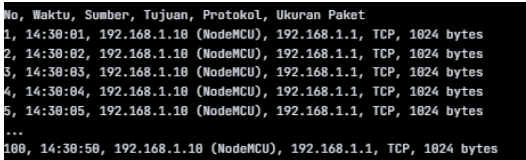


No	Time	Source	Destination	Protocol
1	10:45	192.168.1.105	192.168.1.1	HTTP
2	10:46	192.168.1.105	192.168.1.2	FTP
3	10:47	192.168.1.105	192.168.1.3	SSH
4	10:48	192.168.1.105	192.168.1.4	HTTPS
5	10:49	192.168.1.105	192.168.1.5	DNS

Gambar 10. NodeMCU Menyamar sebagai Perangkat Lain di Jaringan

Dalam contoh di atas, NodeMCU berhasil menyamar sebagai perangkat lain di jaringan dengan menggunakan alamat IP yang sama dengan perangkat asli. Dengan demikian, NodeMCU dapat memperoleh akses yang tidak sah ke sumber daya jaringan dan mengambil data yang seharusnya hanya bisa diakses oleh perangkat asli. Serangan ini dapat menimbulkan dampak besar jika tidak diidentifikasi dan dihindari dengan baik.

### c. Serangan Denial-of-Service (DoS)



No	Waktu	Sumber	Tujuan	Protokol	Ukuran Paket
1	14:38:81	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes
2	14:38:82	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes
3	14:38:83	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes
4	14:38:84	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes
5	14:38:85	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes
...					
100	14:38:50	192.168.1.10 (NodeMCU)	192.168.1.1	TCP	1024 bytes

Gambar 11. Penurunan Kinerja Jaringan Selama Serangan DoS

Eksperimen DoS menunjukkan bahwa NodeMCU dapat mengirimkan lalu lintas palsu yang cukup untuk mengganggu layanan jaringan. Selama serangan berlangsung, perangkat lain mengalami penurunan kinerja jaringan yang signifikan dan kesulitan untuk tetap terhubung.

### 3.2. Analisis Kerentanan

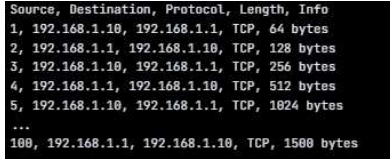
Hasil eksperimen menunjukkan bahwa jaringan Wi-Fi pada frekuensi 2,4 GHz memiliki beberapa titik lemah yang dapat dieksploitasi oleh NodeMCU. Kerentanan utama termasuk:

- a. Ketiadaan enkripsi atau penggunaan enkripsi yang lemah.
- b. Kurangnya otentikasi yang kuat.
- c. Rentan terhadap serangan DoS yang mengganggu ketersediaan layanan.

### 3.3. Evaluasi Tindakan Pencegahan

Setelah menerapkan tindakan pencegahan seperti enkripsi WPA3, segregasi jaringan, dan penggunaan IDS, efektivitas serangan berkurang secara signifikan.

**a. Efektivitas Enkripsi WPA3**



```
Source, Destination, Protocol, Length, Info
1, 192.168.1.10, 192.168.1.1, TCP, 64 bytes
2, 192.168.1.1, 192.168.1.10, TCP, 128 bytes
3, 192.168.1.10, 192.168.1.1, TCP, 256 bytes
4, 192.168.1.1, 192.168.1.10, TCP, 512 bytes
5, 192.168.1.10, 192.168.1.1, TCP, 1024 bytes
...
100, 192.168.1.1, 192.168.1.10, TCP, 1500 bytes
```

**Gambar 12.** Paket Data yang Terenkripsi dengan WPA3

Dengan menggunakan enkripsi WPA3, serangan sniffing menjadi jauh lebih sulit karena data yang ditangkap telah terenkripsi dengan kuat.

**b. Segregasi Jaringan**

Memisahkan jaringan untuk perangkat kritis dan tamu mengurangi risiko akses tidak sah. NodeMCU yang mencoba menyamar sebagai perangkat kritis gagal mendapatkan akses ke jaringan terpisah yang dilindungi.

**c. Deteksi Intrusi (IDS)**

Penggunaan IDS memungkinkan deteksi dini aktivitas mencurigakan. Selama eksperimen, IDS berhasil mengidentifikasi dan memberi peringatan tentang serangan spoofing dan DoS.

**3.4. Hasil dan Diskusi**

Hasil penelitian menunjukkan bahwa NodeMCU adalah alat yang efektif untuk mengeksploitasi kerentanan jaringan Wi-Fi pada frekuensi 2,4 GHz. Namun, dengan penerapan tindakan pencegahan yang tepat seperti enkripsi WPA3, segregasi jaringan, dan penggunaan IDS, risiko serangan dapat diminimalkan. Pentingnya menggunakan protokol keamanan yang kuat seperti WPA3 tidak bisa diremehkan. Protokol ini memberikan lapisan perlindungan tambahan yang signifikan dibandingkan dengan protokol yang lebih lama seperti WEP dan WPA2. Selain itu, segregasi jaringan membatasi dampak dari perangkat yang mungkin telah dikompromikan, memastikan bahwa kerusakan tidak menyebar ke seluruh jaringan. Implementasi IDS juga terbukti sangat efektif dalam mendeteksi dan merespons serangan secara real-time. IDS memberikan wawasan penting tentang aktivitas mencurigakan di jaringan dan memungkinkan respon cepat untuk mengurangi dampak serangan.

**4. Kesimpulan**

Berdasarkan hasil penelitian Memanfaatkan NodeMCU dalam Serangan Jaringan Wi-Fi pada Frekuensi 2,4 GHz: Tinjauan Keamanan dan Tindakan Pencegahan bahwa NodeMCU terbukti efektif dalam melaksanakan berbagai jenis serangan terhadap jaringan Wi-Fi pada frekuensi 2,4 GHz, termasuk sniffing, spoofing, dan denial-of-service (DoS). Jaringan Wi-Fi pada frekuensi ini memiliki kerentanan signifikan, terutama jika menggunakan protokol keamanan yang lemah seperti WEP dan WPA2. Namun, penerapan langkah-langkah pencegahan seperti enkripsi WPA3, segregasi jaringan, dan penggunaan Intrusion Detection System (IDS) terbukti dapat secara signifikan mengurangi risiko serangan. Oleh karena itu, untuk meningkatkan keamanan jaringan Wi-Fi, disarankan untuk menggunakan protokol keamanan yang lebih kuat, menerapkan segregasi jaringan, dan memanfaatkan IDS untuk pemantauan dan deteksi dini aktivitas mencurigakan.

**Daftar Pustaka**

- [1] Balakrishnan, M., & Goyal, P. (2019). *Security Analysis of Wi-Fi Networks: Techniques and Tools*. International Journal of Network Security, 21(4), 573-580.
- [2] Carvajal, G., Silva, J., & Ponciano, J. (2020). Exploring Wi-Fi Security Using ESP8266 and NodeMCU. Proceedings of the 2020 IEEE International Conference on Cybersecurity and Resilience, 123-128.
- [3] Choi, H., & Choi, D. (2018). *Vulnerabilities of IEEE 802.11 Wireless LANs and Countermeasures*. Journal of Communication and Networks, 20(3), 255-262.
- [4] Gupta, A., & Shukla, P. (2021). *Enhancing Wi-Fi Security Using WPA3: A Comparative Study*. Journal of Information Security and Applications, 58, 102-110.
- [5] Kumar, S., & Kaur, P. (2019). *Wi-Fi Security: Overview of Technologies and Threats*. International Journal of Computer Applications, 178(7), 22-27.
- [6] Patel, N., & Shah, V. (2020). *Using ESP8266 for Wi-Fi Sniffing and Jamming*. International Journal of Computer Networks and Applications, 7(2), 65-73.

# Rancangan Machine Learning untuk Mendeteksi Lagu Plagiat

Dominggo Pratama Sidauruk<sup>1</sup>, I Gusti Ngurah Anom Cahyadi Putra<sup>2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>dominggoprataama@email.com  
<sup>2</sup>anom.cp@unud.ac.id

## Abstract

*Plagiarism in the music industry is a serious issue that requires advanced solutions. This research proposes a Machine Learning-based system for detecting song plagiarism by combining Convolutional Neural Network (CNN) and Dynamic Time Warping (DTW). CNN is used to extract features from the visual representation of music notations, while DTW measures the temporal distance between two sequences of notations. Experimental results show that this system provides a more accurate solution with an accuracy of 92.71%, with a dataset of 4800 data points.*

**Keywords:** Music plagiarism, Convolutional Neural Network (CNN), Dynamic Time Warping (DTW), plagiarism detection, music notation, machine learning

## 1. Pendahuluan

Musik sebagai Bahasa universal, mampu menyampaikan emosi, cerita, dan pengalaman manusia dengan menggunakan berbagai elemen seperti melodi, harmoni, ritme, dan lirik. Dalam era modern ini, perkembangan teknologi digital telah memberikan perubahan pada cara kita mengakses, menciptakan, dan berbagi musik. Platform streaming seperti *Spotify*, *Apple Music*, dan *YouTube* telah membuka pintu bagi akses yang lebih mudah ke jutaan lagu dari seluruh dunia. Namun, seiring dengan kemudahan akses dan kolaborasi yang ditawarkan oleh teknologi, juga muncul tantangan baru, termasuk isu-isu seperti plagiarisme musik. Plagiarisme musik, di mana sebagian atau seluruh karya musik dicuri atau disalin tanpa izin dari pencipta aslinya, telah menjadi masalah yang semakin meresahkan dalam industri musik modern. Plagiarisme musik tidak hanya merugikan secara finansial bagi pencipta lagu asli, tetapi juga merugikan dalam Hak Kekayaan Intelektual (HKI) bagi pencipta lagu. Plagiarisme dalam "UUHC No 28 Tahun 2014" tidak diatur secara jelas di dalam undang-undang tersebut. plagiarisme yang diatur dalam UUHC ini hanya sebatas peniadaan nama pencipta serta tidak adanya izin dari pemegang atau pemilik hak cipta sehingga terjadi tindakan eksploitasi atau tindakan memperbanyak hasil cipta orang yang dilakukan tanpa izin [1]. Dalam upaya mengatasi tantangan ini, pengembangan teknologi yang dapat mendeteksi dan mencegah plagiarisme musik menjadi semakin penting. Salah Upaya untuk mengatasi plagiarisme pada industri musik yakni menggunakan Machine learning untuk pengenalan karakter dalam notasi angka. Kemampuan untuk mengenali dan menerjemahkan notasi angka menjadi teks yang dapat diproses oleh sistem komputer dapat membantu membangun sistem deteksi plagiarisme musik yang lebih efektif dan andal. Artikel ini bertujuan untuk mengeksplorasi kemampuan Machine Learning dalam pengenalan karakter dalam notasi angka, dengan mempertimbangkan implikasi hukum terkait dengan plagiarisme musik. Fokus khusus diberikan pada penggunaan Convolutional Neural Networks (CNNs) untuk mengenali dan menerjemahkan simbol-simbol musik dalam notasi angka menjadi teks yang dapat dibaca oleh manusia.

## 2. Metode Penelitian

Pada penelitian ini dilakukan beberapa tahap proses penelitian untuk mendapatkan program yang dapat mengenali notasi angka pada gambar yang diberikan.



## 2.1. Pengumpulan Data

Langkah pertama dari penelitian ini adalah pengumpulan data. Data yang saya gunakan adalah data primer berupa gambar hasil editan saya berupa notasi angka dengan latar putih. Dataset ini berjumlah total 4,800 data gambar dalam 5 kategori (note1, note2, note3, note4, note5) yang terdiri dari data train dan data test.

## 2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network adalah salah satu metode Machine Learning dari pengembangan Multi-Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi [2]. CNN merupakan hierarchical neural networks yang disusun atas beberapa convolutional layer dan beberapa subsampling layer [3]. Dalam machine learning, CNN merupakan teknik yang termasuk pada feed-forward neural network. Pada arsitektur CNN setiap individu neuron disusun sedemikian sehingga dapat memberikan respon terhadap region yang saling overlapping pada suatu area visual [4]. Convolutional Neural Network memiliki cara kerja yang sama dengan tradisional Artificial Neural Network, mulai dari masukan awal sampai hasil akhir, setiap jaringan akan tetap menampilkan sebuah skor atau yang disebut dengan weight dan loss functions pada layer terakhir [5].

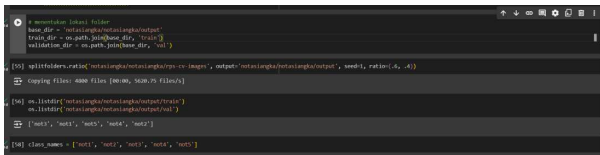
## 2.3. Dynamic Time Warping (DTW)

Dynamic Time Warping algorithm (DTW) adalah algoritma yang menghitung optimal warping path antara dua waktu. Metode ini untuk menentukan kecocokan optimal antara dua deret waktu dengan beberapa pembatasan tertentu [6]. DTW mencocokkan dua sekuensial dengan menghitung transformasi temporal sehingga keduanya dapat diselaraskan (aligned). Penyelarasan (alignment) adalah optimal jika terukur jarak kumulatif terkecil antara dua sampel yang telah diselaraskan [7]. DTW dapat membandingkan dua urutan yang memiliki panjang yang berbeda. Dalam konteks notasi angka lagu, dua lagu bisa memiliki durasi yang berbeda atau bisa dinyanyikan dengan tempo yang berbeda. DTW mampu mengatasi perbedaan panjang ini dengan cara merentangkan dan menyusutkan urutan waktu. Sebuah lagu yang sama bisa dinyanyikan atau dimainkan dengan tempo yang berbeda. DTW menyesuaikan urutan waktu sehingga bagian yang mirip dari dua urutan dapat dibandingkan secara lebih akurat, meskipun satu urutan lebih cepat atau lebih lambat dari yang lain.

## 2.4. Implementasi Program

Pada implemetansi program, kami membuat program sederhana yang bisa mengenali notasi angka menggunakan CNN dan program sederhana untuk mendeteksi dan menghitung seberapa mirip baris note suatu lagu dengan lagu lainnya. Pada program sederhana ini, program ini hanya bisa membaca 5 not saja (not1, not2, not3, not4 dan not5).

### a. Pembagian Dataset



```
PS C:\Users\user> mkdir folder
PS C:\Users\user> mkdir folder\train_dir
PS C:\Users\user> mkdir folder\validation_dir
PS C:\Users\user> splitfolders -path "folder" -in "folder\train_dir" -out "folder\train_dir" -seed 1 -rate (.4, .4)
Copying files: 4800 files (3000, 1800 files)
PS C:\Users\user> cd folder\train_dir
PS C:\Users\user> ls
note1, note2, note3, note4, note5
PS C:\Users\user> cd folder\validation_dir
PS C:\Users\user> ls
note1, note2, note3, note4, note5
```

Gambar 1. Pembagian Dataset

Dataset dibagi menjadi dua bagian utama, yaitu data pelatihan (train) dan data validasi (validation). Pembagian dataset ini dilakukan dengan menggunakan fungsi splitfolders.

ratio()) pada library splitfolders. Data pelatihan digunakan untuk melatih model, sedangkan data validasi digunakan untuk menguji kinerja model saat pelatihan.

### b. Augmentasi Data dan Normalisasi Data

```
data_train_gen = ImageDataGenerator(
    rotation_range=20,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

data_test_gen = ImageDataGenerator(
    rotation_range=20,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Gambar 2. Augmentasi Data dan Normalisasi Data

Pada tahap ini, dilakukan augmentasi data untuk memperluas dataset pelatihan dengan cara menghasilkan variasi baru dari gambar-gambar yang sudah ada. Hal ini dilakukan dengan menggunakan objek ImageDataGenerator dari library tensorflow.keras.preprocessing.image. Beberapa teknik augmentasi data yang digunakan antara lain rotasi, pergeseran, dan pembalikan gambar secara horizontal. Setelah dilakukan augmentasi data, gambar-gambar dalam dataset dinormalisasi agar memiliki rentang nilai antara 0 dan 1. Hal ini dilakukan dengan menambahkan parameter rescale=1./255 pada objek ImageDataGenerator.

```
train_generator = data_train_gen.flow_from_directory(
    train_dir, # dire train
    target_size=(160, 160), # mengubah resolusi gambar
    batch_size=32,
    class_mode='categorical', # k klasifikasi lebih dari 2 kelas maka menggunakan class_mode = 'categorical'
    color_mode='rgb',
    seed=42
)

validation_generator = data_test_gen.flow_from_directory(
    validation_dir, # dire validasi
    target_size=(160, 160), # mengubah resolusi gambar
    batch_size=32,
    class_mode='categorical', # k klasifikasi lebih dari 2 kelas maka menggunakan class_mode = 'categorical'
    color_mode='rgb',
    seed=42
)

found 2000 images belonging to 5 classes.
found 3000 images belonging to 5 classes.

[0] print(train_generator.class_indices)
[0] {'cat': 0, 'dog': 1, 'horse': 2, 'sheep': 3, 'wolf': 4}

[0] print(validation_generator.class_indices)
[0] {'cat': 0, 'dog': 1, 'horse': 2, 'sheep': 3, 'wolf': 4}
```

Gambar 3. Generator Data

Data pelatihan dan data validasi disiapkan dalam bentuk generator data menggunakan objek flow\_from\_directory() dari objek ImageDataGenerator. Generator data ini akan menghasilkan batch-batch gambar yang akan digunakan secara iteratif selama proses pelatihan dan validasi model. Selanjutnya program akan menampilkan indeks kelas dari data pelatihan dan validasi.

### c. Membuat Model CNN

```
[34] model = Sequential() # deklarasi untuk jenis model
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))

model.compile()
```

Selanjutnya kami mendeklarasikan dan membangun model Convolutional Neural Network (CNN). Pertama-tama, kami menggunakan Conv2D untuk menambahkan lapisan konvolusi yang akan mendeteksi fitur-fitur seperti tepi, tekstur, dan pola dalam gambar. Selanjutnya kami menggunakan MaxPooling2D untuk membantu mengurangi overfitting. Lalu Flatten mengubah data dari matriks 2D menjadi vektor 1D yang kemudian bisa dihubungkan ke lapisan fully connected (Dense). Dense digunakan untuk klasifikasi multi-kelas, menghasilkan probabilitas untuk setiap kelas.

### d. Compile dan Pelatihan Model

```
# compile data
model.compile(
    loss='categorical_crossentropy',
    optimizer=tf.optimizers.Adam(),
    metrics=['accuracy']
)

# train data
model.fit(
    train_generator,
    steps_per_epoch=25,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=5,
    verbose=2
)
```

Setelah membangun arsitektur model, langkah berikutnya adalah mengkompilasi model. Ini termasuk menentukan fungsi loss, optimizer, dan metrik evaluasi yang akan digunakan selama pelatihan. Selanjutnya model dilatih menggunakan data yang dihasilkan oleh generator data pelatihan dan validasi. Proses pelatihan berlangsung selama 20 epoch dengan 25 batch per epoch. `steps_per_epoch=25` menunjukkan jumlah batch yang dihasilkan oleh generator dalam satu epoch. `epochs=20` menentukan jumlah iterasi penuh melalui dataset.

### e. Evaluasi Model

```
[35] score = model.evaluate(train_generator)
print('loss: {:.4f}'.format(score[0]))
print('accuracy: {:.4f}'.format(score[1]))

1/1 [0.000] - loss: 0.1790 - accuracy: 0.9271

[36] tf.keras.metrics.MeanMetricWrapper: 1/1 [0.000] - loss: 0.1790 - accuracy: 0.9271

[37] tf.keras.callbacks.Callback: 1/1 [0.000] - loss: 0.1790 - accuracy: 0.9271
```

Setelah pelatihan, model dievaluasi pada data pelatihan untuk mendapatkan nilai loss dan akurasi. Evaluasi model memberikan metrik performa yang dapat digunakan untuk memahami seberapa baik model telah belajar dari data pelatihan. Model kemudian disimpan.

### f. Program untuk Prediksi Gambar

```
import numpy as np
from keras.preprocessing import image
import matplotlib.pyplot as plt
import tensorflow.keras as keras
from google.colab import files
import os

uploaded = files.upload()

for fn in uploaded.keys():
    path = fn
    img = image.load_img(path, target_size=(150, 150))
    imgplot = plt.imshow(img)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    images = np.vstack([x])
    classes = model.predict(images, batch_size=10)
    class_idx = np.argmax(classes)
    result = class_names[class_idx]
    print("Number: %s\nNama: %s" % (result, result))
```

Program menyediakan fungsi untuk memprediksi kelas dari gambar yang diunggah oleh pengguna. Gambar diunggah dari lokal, diubah ukurannya, dan diubah menjadi array sebelum digunakan untuk prediksi. Bagian ini memungkinkan pengguna untuk menguji model dengan gambar baru. Gambar yang diunggah akan diprediksi kelasnya oleh model, memberikan hasil yang dapat divisualisasikan.

### g. Program untuk Identifikasi Kemiripan Notasi Angka Sebuah Lagu dengan Lagu Yang Lain

```
import numpy as np
from fastdtw import fastdtw
from scipy.spatial.distance import euclidean

# Fungsi untuk memuat notasi angka dari file teks
def load_notasi_angka(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines() # Membaca seluruh konten file sebagai satu string dan menghapus spasi ekstra
        notasi = [int(x) for x in lines] # Mengonversi setiap karakter dalam string menjadi integer
    return notasi

# Fungsi untuk menghitung jarak DTW antara dua urutan notasi angka
def calculate_dtw_distance(seq1, seq2):
    seq1 = [x for x in seq1]
    seq2 = [x for x in seq2]
    distance, path = fastdtw(seq1, seq2, dist=euclidean)
    return distance

# Fungsi untuk menghitung jarak maksimum antara dua urutan
def calculate_max_distance(seq1, seq2):
    max_len = max(len(seq1), len(seq2))
    max_distance_per_pair = euclidean([1], [7]) # Asumsi rentang notasi adalah 1-7
    return max_distance_per_pair * max_len

# Fungsi untuk menghitung persentase kemiripan
def calculate_similarity_percentage(seq1, seq2):
    dtw_distance = calculate_dtw_distance(seq1, seq2)
    max_distance = calculate_max_distance(seq1, seq2)
    similarity = (1 - (dtw_distance / max_distance)) * 100
    return dtw_distance, max_distance, similarity

# Contoh penggunaan
if __name__ == '__main__':
    # Memuat notasi angka dari dua file
    notasi_lagu1 = load_notasi_angka('notasi_lagu1.txt')
    notasi_lagu2 = load_notasi_angka('notasi_lagu2.txt')

    # Contoh penggunaan
    if __name__ == '__main__':
        # Memuat notasi angka dari dua file
        notasi_lagu1 = load_notasi_angka('notasi_lagu1.txt')
        notasi_lagu2 = load_notasi_angka('notasi_lagu2.txt')

        # Menghitung persentase kemiripan antara dua notasi lagu
        dtw_distance, max_distance, similarity_percentage = calculate_similarity_percentage(notasi_lagu1, notasi_lagu2)

        # Menampilkan hasil
        print("Panjang urutan notasi lagu pertama: %s" % len(notasi_lagu1))
        print("Panjang urutan notasi lagu kedua: %s" % len(notasi_lagu2))
        print("Jarak DTW antara dua lagu adalah: %s" % dtw_distance)
        print("Jarak maksimum yang mungkin antara dua lagu adalah: %s" % max_distance)
        print("Persentase kemiripan antara dua lagu adalah: %s%%" % similarity_percentage)
```

Program ini dirancang untuk menghitung persentase kemiripan antara dua urutan notasi angka menggunakan metode Dynamic Time Warping (DTW). Fungsi `load_notasi_angka` digunakan untuk memuat notasi angka dari file teks. Fungsi ini membuka file dan membaca kontennya sebagai satu string, menghapus spasi ekstra, dan mengonversi setiap karakter dalam string menjadi integer yang kemudian disimpan dalam list notasi. Fungsi `calculate_dtw_distance` digunakan untuk menghitung jarak DTW antara dua urutan notasi angka. Setiap elemen dalam urutan diubah menjadi list satu elemen, karena `fastdtw` memerlukan input dalam bentuk vektor. Fungsi ini kemudian menggunakan `fastdtw` untuk menghitung jarak DTW dan jalur penyesuaian optimal antara dua urutan menggunakan jarak Euclidean sebagai metrik jarak. Untuk menghitung jarak maksimum yang mungkin antara dua urutan, fungsi `calculate_max_distance` digunakan. Fungsi ini menentukan panjang maksimum antara dua urutan, dan menghitung jarak Euclidean maksimum antara dua notasi dalam rentang 1-7. Hasilnya adalah produk dari jarak maksimum per not dan panjang maksimum rangkaian not. Kemudian, fungsi `calculate_similarity_percentage` menghitung persentase kemiripan antara dua urutan notasi angka. Fungsi ini menggunakan `calculate_dtw_distance` untuk mendapatkan jarak DTW dan `calculate_max_distance` untuk mendapatkan jarak maksimum antara dua urutan. Persentase kemiripan dihitung dengan rumus  $(1 - (\text{dtw\_distance} / \text{max\_distance})) * 100$ , di mana jarak DTW yang lebih kecil menghasilkan kemiripan yang lebih tinggi. Fungsi ini mengembalikan jarak DTW, jarak maksimum, dan persentase kemiripan. Bagian terakhir dari program adalah program utama. Program utama memuat notasi angka dari dua file teks menggunakan `load_notasi_angka`, menghitung persentase kemiripan antara dua notasi lagu menggunakan `calculate_similarity_percentage`, dan menampilkan hasil berupa panjang urutan notasi, jarak DTW, jarak maksimum, dan persentase kemiripan antara dua lagu.

### 3. Hasil dan Diskusi

Berdasarkan percobaan dengan program sederhana diatas, kami mendapatkan:

#### 3.1. Akurasi Setiap Epoch

```
Epoch 1/20  
25/25 - 46s - loss: 1.6250 - accuracy: 0.2013 - val_loss: 1.6141 - val_accuracy: 0.1688 - 46s/epoch - 2s/step  
Epoch 2/20  
25/25 - 44s - loss: 1.6109 - accuracy: 0.2062 - val_loss: 1.6086 - val_accuracy: 0.2625 - 44s/epoch - 2s/step  
Epoch 3/20  
25/25 - 42s - loss: 1.5798 - accuracy: 0.2750 - val_loss: 1.5937 - val_accuracy: 0.2375 - 42s/epoch - 2s/step  
Epoch 4/20  
25/25 - 41s - loss: 1.5023 - accuracy: 0.3363 - val_loss: 1.5023 - val_accuracy: 0.3500 - 43s/epoch - 2s/step  
Epoch 5/20  
25/25 - 42s - loss: 1.3828 - accuracy: 0.4075 - val_loss: 1.2726 - val_accuracy: 0.5063 - 42s/epoch - 2s/step  
Epoch 6/20  
25/25 - 44s - loss: 1.1888 - accuracy: 0.5200 - val_loss: 1.2791 - val_accuracy: 0.4125 - 44s/epoch - 2s/step  
Epoch 7/20  
25/25 - 43s - loss: 0.9842 - accuracy: 0.5950 - val_loss: 0.8256 - val_accuracy: 0.6750 - 43s/epoch - 2s/step  
Epoch 8/20  
25/25 - 42s - loss: 0.8316 - accuracy: 0.6525 - val_loss: 0.7993 - val_accuracy: 0.6562 - 42s/epoch - 2s/step  
Epoch 9/20  
25/25 - 41s - loss: 0.6610 - accuracy: 0.7325 - val_loss: 0.7580 - val_accuracy: 0.6625 - 41s/epoch - 2s/step  
Epoch 10/20  
25/25 - 41s - loss: 0.5893 - accuracy: 0.7437 - val_loss: 0.5510 - val_accuracy: 0.7688 - 41s/epoch - 2s/step  
Epoch 11/20  
25/25 - 41s - loss: 0.4718 - accuracy: 0.8250 - val_loss: 0.4925 - val_accuracy: 0.7937 - 41s/epoch - 2s/step  
Epoch 12/20  
25/25 - 41s - loss: 0.3909 - accuracy: 0.8325 - val_loss: 0.3460 - val_accuracy: 0.8562 - 41s/epoch - 2s/step  
Epoch 13/20  
25/25 - 41s - loss: 0.3386 - accuracy: 0.8500 - val_loss: 0.4337 - val_accuracy: 0.8062 - 41s/epoch - 2s/step  
Epoch 14/20  
25/25 - 43s - loss: 0.3466 - accuracy: 0.8475 - val_loss: 0.3294 - val_accuracy: 0.8562 - 43s/epoch - 2s/step  
Epoch 15/20  
25/25 - 42s - loss: 0.3270 - accuracy: 0.8600 - val_loss: 0.2744 - val_accuracy: 0.8625 - 42s/epoch - 2s/step  
Epoch 16/20  
25/25 - 40s - loss: 0.2632 - accuracy: 0.9025 - val_loss: 0.2912 - val_accuracy: 0.9062 - 40s/epoch - 2s/step  
Epoch 17/20  
25/25 - 43s - loss: 0.2472 - accuracy: 0.9050 - val_loss: 0.2580 - val_accuracy: 0.8938 - 43s/epoch - 2s/step  
Epoch 18/20  
25/25 - 41s - loss: 0.2271 - accuracy: 0.8988 - val_loss: 0.2290 - val_accuracy: 0.9062 - 41s/epoch - 2s/step  
Epoch 19/20  
25/25 - 42s - loss: 0.2049 - accuracy: 0.9137 - val_loss: 0.1492 - val_accuracy: 0.9563 - 42s/epoch - 2s/step  
Epoch 20/20  
25/25 - 42s - loss: 0.1657 - accuracy: 0.9513 - val_loss: 0.2144 - val_accuracy: 0.9000 - 42s/epoch - 2s/step  
<keras.src.callbacks.History at 0x7f83ef1df00>
```

Berdasarkan gambar diatas, kami mendapatkan nilai accuracy 0,9513. Adapun nilai accuracy setiap epoch yakni 0.2013, 0.2062, 0.2750, 0.3363, 0.4075, 0.5200, 0.5950, 0.6525, 0.7325, 0.7437, 0.8250, 0.8325, 0.8500, 0.8475, 0.8600, 0.9025, 0.9050, 0.8988, 0.9137, dan 0.9513. Walaupun sempat turun ke 0.8988, bisa dibalang rancangan machine learning kami dengan dataset yang kami gunakan, mampu berkembang dengan baik.

```
90/90 [=====] - 56s 615ms/step - loss: 0.1795 - accuracy: 0.9271  
Loss: 0.1795  
Accuracy: 0.9271
```

	Model	Data_Train	Data_Valid	Epochs	Batch_Size	Loss	Accuracy
0	CNN	2880	1920	20	32	0.17947	0.927083

Setelah melewati evaluasi model, accuracy machine learning kami mampu menyentuh aaccuracy 92%.

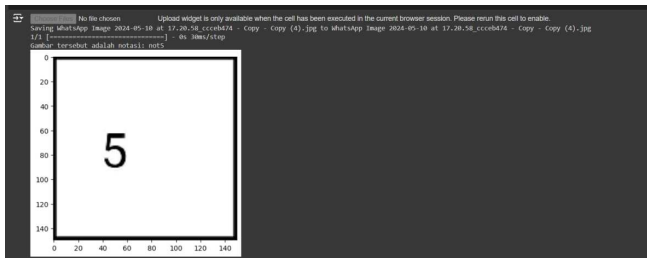
### 3.2. Matriks yang dihasilkan

```
60/60 [=====] - 40s 656ms/step  
Confusion Matrix  
[[ 81 79 74 75 75]  
 [ 94 61 82 67 80]  
 [ 73 75 84 76 76]  
 [ 74 77 81 97 55]  
 [ 92 77 89 55 71]]
```

Berikut adalah matriks yang dihasilkan dari machine learning kami berdasarkan dataset yang kami gunakan.

### 3.3. Percobaan Menginput Gambar

Karena disini kami menggunakan machine learning untuk identifikasi notasi angka, sehingga machine learning harus mampu mengidentifikasi notasi angka yang kami berikan. Walaupun begitu machine learning ini hanya mampu mengidentifikasi 1 notasi angka saja.



Bisa dilihat pada gambar diatas, machine learning mampu mengidentifikasi notasi angka walaupun hanya 1 saja pada 1 gambar.

### 3.4. Percobaan Menginput Baris Notasi Angka

Pada percobaan ini kami akan menggunakan potongan baris notasi angka dari Ibu Kita Kartini (443465653132345) dengan pembandingan notasi angka yang mirip dengan notasi angkanya (443466654132245).

```
[Running] python -u "c:\DTW\DTWnotasi.py"  
Panjang urutan notasi lagu pertama: 15  
Panjang urutan notasi lagu kedua: 15  
Jarak DTW antara dua lagu adalah: 3.0  
Jarak maksimum yang mungkin antara dua lagu adalah: 90.0  
Persentase kemiripan antara dua lagu adalah: 96.67%
```

Bisa dilihat hasil dari program kami menunjukkan lagu yang menjadi pembandingan mirip dengan potongan lagu Ibu Kita Kartini sebesar 96,67%. Bisa dilihat jarak DTW antara 2 lagu tersebut adalah 3. Jarak disini merupakan jumlah seluruhnya jarak sebuah nada dengan nada lainnya. Misal not 4 dibandingkan dengan not 1, maka not 4 memiliki jarak sebanyak 3 dari not 1. Contohnya ada baris not yakni 566432 dibandingkan dengan 564421. Program akan membandingkan satu per satu not dengan pasangan not nya. 5 dipasangkan dengan 5, 6 dipasangkan dengan 6, 6 dipasangkan dengan 4, 4 dipasangkan dengan 4, 3 dipasangkan dengan 2 dan 2 dipasangkan dengan 1. Maka jarak DTW seluruhnya adalah  $2 + 1 + 1 = 4$ . Sehingga diperoleh jarak DTW seluruhnya adalah 4. Untuk jarak maksimum antara 2 jarak diperoleh dengan mengalikan jarak DTW dengan Panjang urutan notasi dikalikan dengan rentang tangga nada (7-1) sehingga didapatkan Jarak Maksimumnya adalah  $15 \times 6 = 90$ . Presentasi kemiripannya didapatkan dengan rumus

$$\text{Persentase kemiripan} = \left( \frac{\text{Jarak DTW}}{\text{Jarak maksimum}} \right) \times 100\% \quad (1)$$

Sehingga didapatkan bahwa presentasi kemiripannya adalah 96,67%.

## 4. Kesimpulan

Adapun kesimpulan dari artikel ini antara lain:

- Akurasi tertinggi yang dicapai dari dataset setelah proses kompilasi dan pelatihan model adalah sebesar 95.13%. Hal ini menunjukkan bahwa model Convolutional Neural Network (CNN) yang kami rancang mampu mengenali pola dalam data dengan tingkat akurasi yang sangat tinggi selama fase pelatihan.
- Setelah melalui tahap evaluasi model, akurasi yang diperoleh adalah sebesar 92.71%, dengan jumlah data latih sebanyak 2880 dan data valid sebanyak 1920. Angka ini menunjukkan bahwa model tetap menunjukkan kinerja yang kuat saat diuji dengan data yang tidak digunakan selama pelatihan, meskipun terdapat sedikit penurunan akurasi.
- Rancangan kami saat ini masih berada dalam tahap program sederhana yang berfokus pada pengenalan notasi angka untuk mendeteksi plagiat musik. Kami berharap bahwa artikel ini dapat memberikan kontribusi yang berarti dalam pengembangan lebih lanjut dari teknologi machine learning untuk identifikasi musik plagiat. Kami mengharapkan pembaca yang tertarik dapat melanjutkan pengembangan ini hingga tahap yang lebih kompleks, termasuk identifikasi langsung dari data audio, sehingga teknologi ini dapat digunakan dalam aplikasi nyata di industri musik.

#### Daftar Pustaka

- [1] Yoga, P., Putra, U., Agung, A., & Indrawati, S. (2021). Perlindungan Hukum Terhadap Praktik Plagiarisme Karya Seni Lagu/Musik Berdasarkan Undang-Undang Nomor 28 Tahun 2014. *Jurnal Kertha Negara*, 9(12), 1027–1038. [2]N. I. Kurniati, A. Rahmatulloh, and D. Rahmawati, "Perbandingan Performa Algoritma Koloni Semut Dengan Algoritma Genetika – Tabu Search Dalam Penjadwalan Kuliah," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 17, 2019, doi: 10.24114/cess.v4i1.11387.
- [2] Wairata, C. R., Swedia, E. R., & Cahyanti, M. (2021). Pengklasifikasian Genre Musik Indonesia Menggunakan Convolutional Neural Network. *Sebatik*, 25(1). <https://doi.org/10.46984/sebatik.v25i1.1286>
- [3] Hakim, D. M., & Rainari, E. (2019). Convolutional Neural Network untuk Pengenalan Citra Notasi Musik. *Techno.Com*, 18(3), 214–226. <https://doi.org/10.33633/tc.v18i3.2387>
- [4] Ciresan CD, et al. 2011. Flexible, High Performance Convolutional Neural Networks for Image Classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*, Vol. Two, Pages 1237-1242
- [5] Namruddin, R., Mirfan, & Irfandi. (2023). Klasifikasi Kesegaran Buah Apel Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android. *Prosiding SISFOTEK*.
- [6] Archamadi, A., Magdalena, R., & Ramatryana, I. N. A. (2016). Analisis dan Simulasi Identifikasi Judul Lagu dari Senandung Manusia Menggunakan Ekstraksi Ciri Dct (discrete Cosine Transform). *EProceedings of Engineering*, 3(3), 4575–4584. <http://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/2894>
- [7] Iqbal, M., Supriyati, E., & Listyorini, T. (2015). Implementasi Offline Pengenalan Sistem Isyarat Bahasa Indonesia Menggunakan Metode Dynamic Time Warping Pada Perangkat Android. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 6(2), 391. <https://doi.org/10.24176/simet.v6i2.477>



Halaman ini sengaja dibiarkan kosong

## Sistem Monitoring Kamar Tidur Pintar dan Suhu Berbasis IoT dengan Cisco Packet Tracer

Ni Made Ayu Wirasih<sup>a1</sup>, I Ketut Gede Suhartana<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>wirasih.2208561014@student.unud.ac.id  
<sup>2</sup>kg.suhartana@unud.ac.id

### Abstract

*This research endeavors to develop an Internet of Things (IoT)-based smart bedroom monitoring system, leveraging Cisco Packet Tracer technology as a robust simulation platform. The system facilitates automatic monitoring and management of indoor environment parameters such as temperature, security, and lighting settings, aiming to enhance occupants' comfort and safety. Key components of the system include a temperature sensor, microcontroller, and an LCD information screen, enabling real-time display of bedroom temperature data. System validation was conducted via simulations using Cisco Packet Tracer, consistently demonstrating the system's efficacy in automating room temperature monitoring and management. These findings lay a solid groundwork for advancing IoT technology, emphasizing deeper integration, addressing challenges, proposing solutions, and exploring potential applications across diverse contexts.*

**Keywords:** Smart Bedroom, Internet of Things, Cisco Packet Tracer, Temperature Monitoring, Automatic System

### 1. Pendahuluan

Dalam era digital ini yang semakin canggih, *Internet of Things* (IoT) telah menjadi paradigma teknologi yang dominan, memberikan kemudahan, efisiensi, dan kontrol yang lebih besar terhadap lingkungan manusia. *Internet of Things* (IoT) merupakan teknologi yang canggih yang bertujuan untuk mengembangkan dan memanfaatkan konsep untuk memperluas serta meningkatkan manfaat dari koneksi internet yang terus berkembang [1]. IoT menyediakan kemampuan untuk menghubungkan berbagai perangkat fisik ke Internet, memungkinkan pertukaran data yang cepat dan efisien di antara perangkat tersebut. Salah satu manfaat IoT yaitu pengembangan konsep kamar tidur pintar yang memanfaatkan teknologi IoT pada peralatan elektronik di kamar tidur. Dengan menghubungkan peralatan elektronik tersebut melalui jaringan internet, konsep kamar tidur pintar dapat meningkatkan kenyamanan penghuni kamar dan juga menghemat energi. Pada konsep kamar tidur pintar terdapat sistem pemantauan suhu berbasis IoT yang memiliki relevansi tinggi karena dapat memantau dan mengelola lingkungan dalam ruangan secara otomatis. Solusi ini memberikan cara inovatif untuk mengintegrasikan perangkat IoT seperti sensor suhu untuk mengukur suhu ruangan dan menghubungkannya ke perangkat IoT lainnya melalui jaringan internet. Dengan demikian, sistem ini dapat memantau dan mengelola suhu ruangan secara otomatis. Pada penelitian ini aplikasi simulator Cisco Packet Tracer digunakan sebagai platform simulasi dengan antarmuka baris perintah, serta didukung oleh sensor, memungkinkan pengembangan dan pengujian sistem virtual sebelum penerapan secara fisik [2]. Memiliki tujuan untuk memberikan kontribusi yang signifikan terhadap pengembangan teknologi IoT yang lebih maju dan terintegrasi dengan membahas tantangan, solusi, dan kemungkinan penerapannya. Pada penelitian terdahulu yang dilakukan oleh Yakti pada tahun 2019 mengembangkan sistem ruangan otomatis menggunakan sensor sebagai input untuk mengendalikan perangkat secara otomatis, mengurangi campur tangan manusia. Pengguna dapat mengontrol perangkat sesuai kebutuhan, baik secara manual maupun otomatis, dan memantau kondisi ruangan melalui *streaming* internet [3]. Selain itu penelitian yang dilakukan Shombing menyatakan bahwa simulasi dapat digunakan untuk membangun jaringan *smart home*

yang efektif menggunakan IoT *home gateway*. Dalam penelitian ini, simulasi memungkinkan desain dan perencanaan implementasi yang lebih baik, sehingga memungkinkan aplikasi simulasi ini diterapkan di dunia nyata berdasarkan kemajuan teknologi saat ini [4]. Selanjutnya penelitian yang dilakukan oleh Miftah pada tahun 2018 juga menunjukkan bahwa keterbatasan dan ketersediaan perangkat keras untuk pembelajaran *Internet of Things* dapat diatasi dengan menggunakan perangkat simulasi seperti Cisco Packet Tracer, tanpa memerlukan investasi besar untuk perangkat keras. Meskipun simulasi ini tidak mencakup semua aspek seperti dalam situasi nyata, penggunaan prinsip-prinsip keilmuan dengan Cisco Packet Tracer masih dapat dipertanggungjawabkan [5].

## 2. Metode Penelitian

Penelitian ini menggunakan simulasi untuk merancang sebuah kamar tidur pintar dengan memanfaatkan perangkat lunak simulator Cisco Packet Tracer 8.2.2. Proses penelitian terbagi menjadi dua tahap, yakni: studi pustaka dan perancangan simulasi Cisco Packet Tracer.

### 2.1 Studi Pustaka

Dalam penelitian ini, studi pustaka dilakukan secara manual dengan mencari referensi secara online melalui internet dan mempelajari tentang jaringan sensor nirkabel, *Internet of Things*, dan Cisco Packet Tracer untuk memahami konsep dan teknologi yang terkait dengan kamar tidur pintar. Dalam proses ini menggunakan berbagai sumber yang relevan dan terpercaya untuk memperoleh informasi yang akurat.

### 2.2 Perancangan Simulasi

Pada tahap awal, peneliti merancang sebuah sistem kamar tidur pintar yang mengintegrasikan perangkat pembangun jaringan internet dan perangkat *Internet of Things* (IoT) khusus untuk kamar tidur. Komponen utama dari sistem IoT ini terdiri dari tiga elemen penting: pintu otomatis, monitoring kamera, dan monitoring suhu. Pintu otomatis dilengkapi dengan perangkat pembaca RFID dan kartu RFID yang sesuai dengan kode tertentu. Selain itu, ketika pintu terbuka, lampu di dalam kamar secara otomatis akan menyala untuk memberikan pencahayaan yang cukup. Sementara itu, kamera pemantauan dilengkapi dengan sensor gerak yang memungkinkannya untuk mendeteksi pergerakan dalam ruangan. Hal ini bertujuan untuk meningkatkan keamanan kamar tidur. Adapun pada pemantauan suhu, jendela dilengkapi dengan mekanisme otomatis yang dapat mendeteksi perubahan suhu. Misalnya, jika suhu ruangan meningkat, jendela akan terbuka untuk memberikan sirkulasi udara yang lebih baik, sedangkan jika suhu menurun, jendela akan tertutup untuk menjaga kenyamanan penghuni kamar tidur. Selanjutnya, sistem pemantauan suhu dirancang dengan menggunakan mikrokontroler, sensor suhu, dan dilengkapi dengan layar LCD sebagai media informasi untuk menampilkan suhu kamar tidur secara *real-time*. Semua komponen tersebut terhubung ke jaringan, memungkinkan pengendalian dan pemantauan aktivitas ruangan secara remote melalui internet, sehingga memberikan kemudahan dan kenyamanan bagi penghuni kamar tidur. Perangkat seperti *smartphone* dan laptop berperan sebagai penghubung ke internet, sedangkan perangkat *home gateway* memainkan peran penting sebagai perantara antara *smartphone* dan perangkat atau peralatan yang akan diatur dalam kamar tidur.

Tabel 1. Perangkat IoT Kamar Tidur Pintar

Perangkat IoT	Fungsi
Motion Detector	Fungsi utama motion detector dalam kamar tidur pintar adalah untuk memantau aktivitas atau gerakan yang terjadi dalam diluar ruangan tersebut.
Kamera	Kamera digunakan untuk memantau dan merekam aktivitas yang terjadi diluar kamar tidur pintar. Hal ini dapat membantu dalam pengawasan keamanan, memungkinkan pengguna untuk memantau ruangan dari jarak jauh melalui aplikasi di <i>smartphone</i> mereka.

<b>Perangkat IoT</b>	<b>Fungsi</b>
RFID Reader	Fungsi utama RFID reader adalah untuk membaca data yang terkandung dalam kartu RFID dan mengirimkannya ke komponen lain dalam sistem.
RFID Card	RFID Card adalah kartu khusus yang mengandung informasi yang dapat diidentifikasi oleh RFID reader. Kartu ini dapat didekatkan atau dipindai oleh RFID reader untuk memberikan akses atau memberikan instruksi kepada sistem kamar tidur pintar.
Lampu	Fungsi lampu adalah untuk memberikan pencahayaan yang sesuai saat pintu dibuka atau ditutup. Misalnya, lampu dapat dinyalakan saat pintu terbuka untuk memberikan penerangan yang cukup.
Pintu	Pintu adalah komponen fisik yang akan dibuka atau ditutup secara otomatis berdasarkan instruksi dari sistem otomatisasi. Ketika MCU menerima sinyal dari RFID reader yang memverifikasi kartu RFID yang valid, MCU akan mengirimkan perintah untuk membuka atau menutup pintu sesuai dengan kebutuhan.
MCU Pintu	MCU mengontrol operasi dan koneksi antara komponen-komponen lainnya dalam sistem. MCU akan menerima informasi dari RFID reader tentang kartu yang didekatkan dan memutuskan aksi selanjutnya, seperti membuka atau menutup pintu.
Temperatur Monitor	Fungsi utama temperatur monitor adalah untuk mendeteksi dan mengukur suhu udara di ruangan.
Thermostat	Fungsi thermostat adalah untuk mengontrol sistem pemanas atau pendingin udara untuk menjaga suhu dalam kisaran yang diinginkan.
Sensor Suhu	Fungsi utama sensor suhu adalah untuk mendapatkan informasi akurat tentang suhu dalam kamar tidur pintar.
MCU Suhu	MCU dapat memproses data suhu, mengontrol perangkat-perangkat terkait, dan menampilkan informasi suhu pada layar informasi.
Layar Informasi	Untuk memberikan tampilan visual yang memudahkan pengguna untuk memantau dan mengatur suhu dalam kamar tidur pintar.
Jendela	untuk mengatur ventilasi dan sirkulasi udara dalam kamar tidur pintar. Ketika suhu mencapai tingkat yang diatur, MCU dapat memberikan perintah untuk membuka atau menutup jendela sesuai dengan kebutuhan.

**Tabel 2.** Perangkat Pembangun Jaringan Internet

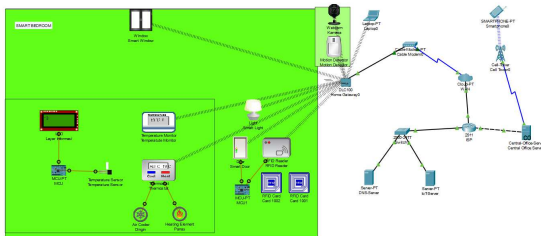
<b>Perangkat Pembangun Jaringan Internet</b>	<b>Fungsi</b>
Laptop	Digunakan sebagai perangkat komputasi yang dapat digunakan untuk mengakses dan mengendalikan berbagai perangkat dalam kamar tidur pintar.
Smartphone	Digunakan sebagai perangkat pintar yang dapat mengendalikan perangkat dalam kamar tidur pintar, memberikan akses dan kontrol jarak jauh.
Home Gateway	Berfungsi sebagai pusat kontrol untuk menghubungkan dan mengelola semua perangkat dalam jaringan kamar tidur pintar.
Cable Modem	Menghubungkan jaringan dengan penyedia layanan internet.

Perangkat Pembangun Jaringan Internet	Fungsi
Cloud PT WAN	Jaringan pribadi virtual yang memungkinkan akses dan pengaturan perangkat dalam kamar tidur pintar melalui internet.
ISP (Internet Service Provider)	Memberikan akses internet.
Switch	Digunakan untuk menghubungkan dan mengatur lalu lintas data antara perangkat dalam jaringan kamar tidur pintar.
DNS Server	Bertanggung jawab untuk menerjemahkan nama domain menjadi alamat IP, memungkinkan perangkat dalam jaringan untuk saling berkomunikasi.
IoT Server	Merupakan server khusus yang mengelola dan menghubungkan perangkat <i>Internet of Things</i> (IoT) dalam kamar tidur pintar.
Central Office Server	Merupakan server pusat yang mengelola dan mengatur fungsi-fungsi penting dalam jaringan kamar tidur pintar.
Cell Tower	Menyediakan konektivitas seluler yang memungkinkan perangkat dalam kamar tidur pintar terhubung ke jaringan seluler

### 3. Hasil dan Diskusi

#### 3.1. Implementasi Perancangan Simulasi Kamar Tidur Pintar

Dalam pembahasan ini, akan dilakukan tahapan implementasi melalui simulasi dan pengujian terhadap rancangan desain kamar tidur pintar dan suhu berbasis IoT. Tahapan ini dijalankan setelah selesai tahap perancangan, dan sebelum dilakukan implementasi secara keseluruhan.

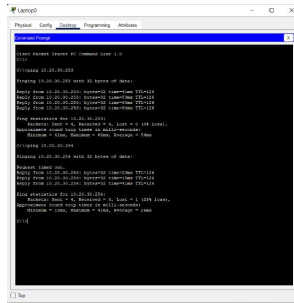


Gambar 1. Tampilan Simulasi Kamar Tidur Pintar dan Suhu pada Cisco Packet Tracer 8.2.2

#### 3.2. Pengujian Konektivitas Perangkat

##### a. Pengujian Ping pada Perangkat IoT Server dan DNS Server

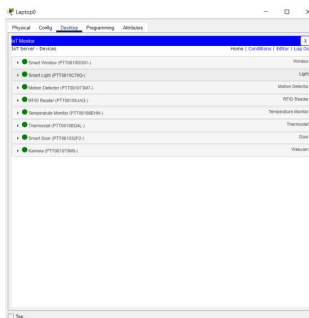
Pengujian ping perangkat dilakukan untuk memeriksa apakah koneksi dan fungsi server IoT dan server DNS berjalan dengan baik. Dalam pengujian awal, ping dilakukan ke server IoT dengan menggunakan alamat IP 10.20.30.253, yang berhasil menunjukkan bahwa laptop telah terhubung dengan server IoT. Kemudian, dilakukan pengujian terhadap server DNS dengan melakukan ping ke alamat IP 10.20.30.254, yang juga berhasil mendapatkan respons.



Gambar 2. Pengujian Ping pada Laptop ke Perangkat IoT

### b. Pengujian Konektivitas Perangkat IoT pada IoT Server

Pengujian konektivitas perangkat IoT pada server IoT dilakukan untuk memverifikasi apakah perangkat IoT yang telah direncanakan dan dikonfigurasi terhubung dengan server IoT. Ini dilakukan dengan masuk ke server IoT melalui perangkat pengguna menggunakan akun yang sudah didaftarkan sebelumnya. Jika koneksi berhasil, perangkat IoT yang sudah terhubung akan muncul di halaman utama server IoT. Pengujian pada laptop dilakukan dengan login ke server IoT menggunakan akun yang telah didaftarkan sebelumnya dengan username "kamaryuwi" dan password "kamaryuwi". Hasilnya menunjukkan bahwa perangkat IoT yang telah direncanakan dan dikonfigurasi sebelumnya muncul dalam tampilan pemantauan server IoT pada smartphone dengan status yang ditandai dengan warna hijau, menunjukkan bahwa perangkat tersebut berhasil terhubung dengan server IoT dan dapat dikendalikan dari jarak jauh. Dengan demikian, pengujian konektivitas perangkat telah sukses.

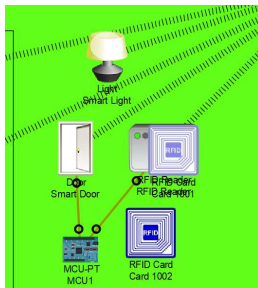


Gambar 3. Pengujian Konektivitas pada Perangkat IoT

### 3.3. Pengujian Perangkat IoT

#### a. Pengujian Pintu Otomatis

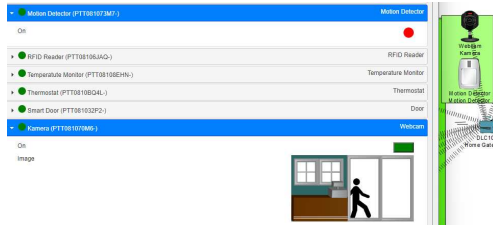
Dalam desain pintu otomatis, sistem telah diprogram untuk membuka pintu hanya ketika kode yang terdapat pada kartu RFID terbukti valid. Sebagai contoh, misalkan kode yang dianggap sah adalah "1001". Jika kode yang dipindahtangankan tidak cocok dengan kode yang diharapkan, maka pintu tidak akan terbuka karena dianggap tidak valid. Langkah-langkah keamanan ini penting untuk mengontrol akses ke dalam ruangan dengan lebih tepat dan meminimalkan risiko intrusi yang tidak diinginkan. Selain itu, sebagai tambahan fitur keamanan, jika pintu terbuka dengan menggunakan kode yang benar, sistem juga telah diprogram untuk mengaktifkan lampu secara otomatis.



Gambar 4. Tampilan Pengujian Pintu Otomatis

#### b. Pengujian Monitoring Kamera

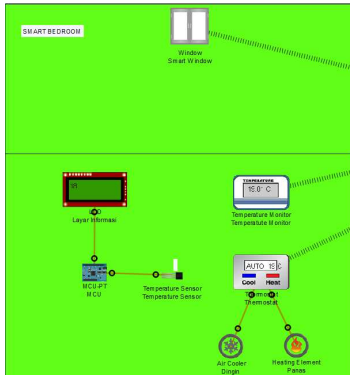
Dalam pengujian sistem monitoring kamera, komponen yang digunakan adalah motion detector yang berfungsi sebagai pendeteksi pergerakan. Motion detector ini didesain untuk secara sensitif merespons setiap pergerakan yang terdeteksi di area yang dipantau. Ketika motion detector mendeteksi adanya pergerakan, sinyal akan diteruskan ke sistem kamera yang kemudian akan memberikan respons. Saat motion detector mendeteksi pergerakan, sistem kamera akan diaktifkan secara otomatis. Kamera akan menyala dan mulai merekam gambar atau video dari area yang terpantau. Ini memungkinkan pengguna atau pengawas untuk memperoleh informasi visual tentang apa yang sedang terjadi di sekitar area yang dipantau secara real-time atau untuk merekam kejadian tersebut sebagai bukti atau referensi di masa mendatang. Penggunaan motion detector dalam sistem monitoring kamera merupakan langkah penting dalam meningkatkan efisiensi dan keandalan sistem. Dibandingkan dengan mode perekaman kontinu tanpa adanya deteksi gerakan, penggunaan motion detector dapat mengurangi jumlah data yang disimpan dan meningkatkan fokus pada kejadian yang signifikan. Hal ini tidak hanya membantu menghemat ruang penyimpanan, tetapi juga memudahkan pengguna dalam menemukan kejadian penting dengan lebih cepat dan efisien.



Gambar 5. Tampilan Pengujian Monitoring Kamera

**c. Pengujian Monitoring Suhu**

Monitoring suhu bertujuan untuk memberikan informasi tentang kondisi suhu di dalam kamar tidur serta menyediakan layar informasi yang dapat diakses pengguna. Dengan adanya monitoring suhu, penghuni kamar tidur dapat memantau perubahan suhu dan mengambil tindakan yang diperlukan untuk menjaga kenyamanan. Dalam pengujian suhu, sistem telah diprogram dengan parameter tertentu. Jika suhu di dalam kamar tidur naik di atas 20 derajat Celsius, maka sistem akan memberikan perintah kepada jendela untuk terbuka secara otomatis. Hal ini bertujuan untuk memungkinkan sirkulasi udara yang lebih baik dan mendinginkan kamar tidur saat suhu naik. Sebaliknya, jika suhu turun di bawah 19 derajat Celsius, sistem akan memberikan perintah kepada jendela untuk menutup secara otomatis. Tindakan ini diambil untuk menjaga suhu di dalam kamar tetap hangat dan nyaman bagi penghuninya.



Gambar 6. Tampilan Monitoring Pengujian Suhu

**4. Kesimpulan**

Penelitian ini berhasil dalam mengembangkan sistem monitoring kamar tidur pintar dan suhu yang berbasis IoT dengan menggunakan Cisco Packet Tracer 8.2.2 sebagai platform simulasi.



Sistem ini memungkinkan penghuni kamar tidur untuk memantau dan mengelola lingkungan dalam ruangan secara otomatis, termasuk suhu, keamanan, dan pencahayaan. Kesuksesan penelitian ini menunjukkan bahwa teknologi IoT dapat digunakan untuk meningkatkan kenyamanan dan efisiensi dalam ruangan, serta memungkinkan pengguna untuk mengontrol dan memantau aktivitas ruangan secara remote melalui internet. Dengan demikian, penelitian ini memberikan kontribusi pada pengembangan teknologi IoT yang lebih maju dan terintegrasi dengan membahas tantangan, solusi, dan kemungkinan penerapannya. Dan simulasi ini menjadi sebuah gambaran yang nantinya dapat dikembangkan secara nyata.

#### Daftar Pustaka

- [1] A. Selay *et al.*, "Internet of Things," *Karimah Tauhid*, vol. 1, no. 6, pp. 860–868, Dec. 2022, doi: 10.30997/karimahtauhid.v1i6.7633.
- [2] L. D. Samsumar and Moh. Subli, "Penggunaan Aplikasi Simulasi Jaringan Cisco Packet Tracer Dalam Design Network," *LP2M STMIK Mataram*, Jan. 01, 2019. [https://www.researchgate.net/publication/332903957\\_Penggunaan\\_Aplikasi\\_Simulasi\\_Jaringan\\_Cisco\\_Packet\\_Tracer\\_Dalam\\_Design\\_Network](https://www.researchgate.net/publication/332903957_Penggunaan_Aplikasi_Simulasi_Jaringan_Cisco_Packet_Tracer_Dalam_Design_Network) (accessed May 07, 2024).
- [3] B. K. Yakti, R. H. Prayitno, and S. Santoso, "Desain Purwarupa dan Konsep Pemanfaatan IoT pada Sistem Kamar Otomatis," *Cogito Smart Journal*, vol. 5, no. 2, pp. 148–158, Dec. 2019, doi: 10.31154/cogito.v5i2.173.148-158.
- [4] O. Sihombing *et al.*, "Smart home design for electronic devices monitoring based wireless gateway network using cisco packet tracer," *Journal of Physics: Conference Series*, vol. 1007, p. 012021, Apr. 2018, doi: 10.1088/1742-6596/1007/1/012021.
- [5] Z. Miftah, "Simulasi Pembelajaran Internet of Things menggunakan Cisco Packet Tracer 7.1.1," *Journal of Information Engineering and Educational Technology*, vol. 2, no. 1, p. 41, Jun. 2018, doi: 10.26740/jieet.v2n1.p41-46.

## Optimalisasi Sistem Pencarian Produk *Sunscreen* Melalui Rancangan Ontologi Semantik

I Gusti Agung Ayu Gita Pradnyaswari Mantara<sup>a1</sup>, I Komang Ari Mogi<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

<sup>1</sup>mantara.2208561105@student.unud.ac.id

<sup>2</sup>arimogi@unud.ac.id

### Abstract

*As a tropical country, Indonesia receives exposure to sunlight throughout the year which contains ultraviolet (UV) rays that are harmful to the skin. The use of sunscreen is important to protect the skin, but the lack of structured information regarding the characteristics of sunscreen makes it difficult to choose the appropriate product. This research aims to build an ontology model for a sunscreen product search system that is tailored to skin conditions using the Methontology method. In the implementation stage, the sunscreen ontology was developed with 1 class, 7 subclasses, 7 object property hierarchies, and 29 individuals using Protégé software. Evaluation is carried out by asking a SPARQL query that shows the ontology's ability to provide relevant answers according to the knowledge being modeled. This sunscreen product ontology model can be used as a basis for developing related knowledge management systems such as sunscreen product search systems.*

**Keywords:** *Ontology, Sunscreen, Methodology, SPARQL Query, Protégé*

### 1. Pendahuluan

Dengan kondisi geografisnya yang terletak di kawasan khatulistiwa, Indonesia menikmati penyinaran matahari yang berlangsung sepanjang tahun tanpa terputus. Sinar Ultraviolet atau UV yang terkandung didalam paparan sinar matahari ini akan bermanfaat jika diterima dalam jumlah yang terbatas. Namun jika paparan sinar matahari yang diterima dalam jumlah berlebih, tentu akan menimbulkan masalah dan berdampak buruk bagi kulit manusia. Radiasi dari Sinar Ultraviolet atau UV ini dapat merusak DNA dan memicu munculnya beberapa masalah pada kulit manusia seperti kulit terbakar, penuaan dini kulit, hingga risiko kanker kulit. Maka daripada itu, penggunaan *sunscreen* sangat penting untuk melindungi dan menghindari terjadinya kerusakan pada kulit manusia [1]. Untuk merasakan manfaatnya, *sunscreen* harus digunakan secara rutin dan tepat. Namun, masih banyak masyarakat Indonesia yang tidak menggunakannya karena merasa acuh dengan pentingnya penggunaan *sunscreen* ataupun hanya sekedar menggunakan *sunscreen* seadanya tanpa mempertimbangkan kecocokan antara *sunscreen* yang sedang digunakan dengan kondisi kulit mereka. Penggunaan *sunscreen* seperti ini berpotensi tidak memberikan perlindungan maksimal bahkan dapat menimbulkan efek merugikan bagi kesehatan kulit manusia. Masalah lainnya yaitu minimnya terdapat informasi yang tertata mengenai karakteristik dari berbagai jenis *sunscreen* yang beredar. Informasi kerap tersebar luas di berbagai situs internet dalam bentuk yang tidak tersusun rapi dan terpisah-pisah, sehingga sulit untuk memperolehnya secara utuh, praktis, dan cepat tanpa mengorbankan banyak waktu dan upaya. Kurang terstrukturnya informasi juga dapat membingungkan masyarakat untuk memilih *sunscreen* yang sesuai kondisi kulit mereka. Dengan memanfaatkan perkembangan teknologi tentu dapat menjadi solusi jitu dalam mengatasi permasalahan ini yaitu dengan mengembangkan sistem pencarian *sunscreen* yang berbasis pengetahuan. Pendekatan yang dapat diterapkan adalah dengan memanfaatkan ontologi, sebuah representasi konseptual pengetahuan, ke dalam format web yang memiliki struktur data dan semantik yang jelas. [2]. Ontologi memungkinkan pendeskripsian data-data *sunscreen* menjadi lebih terstruktur dan terkait dengan jelas. Dalam penelitian ini, akan dibangun dan berfokus dalam pemodelan ontologi pada sistem pencarian

produk sunscreen yang disesuaikan dengan kebutuhan kulit mereka. Akan digunakan metode Methontology yang merupakan salah satu pendekatan dalam pembuatan model ontologi [3]. Setelah membuat model ontologi untuk produk *sunscreen*, model tersebut akan diuji dengan berbagai pertanyaan terkait. Tujuannya adalah untuk memverifikasi apakah model ontologi tersebut telah terbentuk dengan relevan. Jika model ontologi telah terbentuk dengan baik dan relevan, maka hal ini dapat membantu dalam memilih jenis *sunscreen* yang akan digunakan oleh pengguna. Penelitian ini berpedoman pada sejumlah kajian sebelumnya yang berkaitan erat dengan topik yang dikaji. Seperti Utari dan Raharja (2023) telah melakukan penelitian yang berkaitan dengan pengembangan ontologi semantik dalam domain produk kosmetik. Selain itu, Hartani dan rekan-rekannya (2022) juga telah meneliti sistem pencarian produk perawatan kulit berbasis ontologi. Penelitian-penelitian ini memberikan kontribusi penting dalam pengembangan pengetahuan tentang ontologi dalam konteks industri kecantikan dan perawatan kulit.

## 2. Metode Penelitian

Adapun alur penelitian yang digunakan yaitu mengidentifikasi permasalahan, pengumpulan data, dan perancangan ontologi.

Tabel 1. Contoh Data Sunscreen

Nama Sunscreen	Harga	Area Pemakaian	Kadar SPF	Kadar PA	Tipe Kulit Pengguna	Tekstur Sunscreen
Wardah UV Shield Light Matte Sun Stick SPF 50 PA++++	≥50.000	Wajah, leher	50	++++	Kering, berminyak, normal, kombinasi	Stick
Azarine Ceraspray Sunscreen Spray SPF 50 PA+++	≥50.000	Wajah	50	+++	Kering, berminyak, normal, kombinasi	Spray
Implora Perfect Shield Sunscreen	≤50.000	Wajah	40	++++	Kering, berminyak, normal, kombinasi	Krim
Azarine Calm My Acne Sunscreen Moisturizer SPF 35 PA+++	≤50.000	Wajah, leher	35	+++	Berminyak	Gel
Emina Sun Battle SPF 30 PA+++	≤50.000	Wajah, leher	30	+++	Kering	Lotion

### 2.1. Identifikasi Masalah

Penelitian ini mengangkat masalah kurangnya kesadaran masyarakat tentang pentingnya menggunakan *sunscreen* untuk melindungi kulit dari paparan sinar UV yang berlebihan. Selain itu, minimnya informasi terstruktur mengenai berbagai produk *sunscreen* yang beredar di pasaran menyebabkan kesulitan bagi masyarakat dalam memilih produk *sunscreen* yang tepat sesuai dengan kondisi kulit mereka.

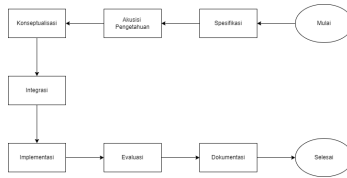
### 2.2. Pengumpulan Data

Penelitian ini menggunakan jenis data teks. Sementara itu, tipe data yang digunakan merupakan data sekunder karena data tersebut berasal dari kumpulan informasi yang sebelumnya telah dipublikasikan oleh pihak-pihak lain. Dimana informasi tersebut dikumpulkan dari situs-situs resmi

perusahaan produk *sunscreen* terkait. Informasi-informasi yang diambil disesuaikan dengan kebutuhan penelitian. Setelah itu dikelompokkan seperti dalam Tabel 1.

### 2.3. Perancangan Ontologi

Adapun metode yang digunakan dalam perancangan ontologi ini yaitu Methontology. Methontology adalah sebuah metode yang umum diaplikasikan dalam proses merancang atau membangun ontologi. Dipilih karena mampu mendeskripsikan setiap aktivitas pembangunan ontologi secara terperinci dan sistematis, sehingga memungkinkan identifikasi kebutuhan informasi, konsep, dan relasi yang relevan secara spesifik. Methontology dapat menghasilkan struktur ontologi yang akurat dan konsisten dengan representasi pengetahuan yang bermakna dalam suatu domain tertentu. Methontology tidak hanya merupakan metode untuk merancang ontologi, tetapi juga memfasilitasi integrasi atau pemanfaatan kembali ontologi yang telah dibangun sebelumnya untuk pengembangan lebih lanjut [4]. Adapun beberapa tahapan dari Methontology yaitu sebagai berikut.



Gambar 1. Alur Methontology

#### a. Spesifikasi

Tahap ini bertujuan untuk menjelaskan spesifikasi ontologi secara informal, semi formal, atau formal menggunakan bahasa alami/natural language. Representasi pengetahuan dapat dilakukan dengan menggunakan pendekatan pertanyaan kompetensi atau pendekatan menengah.

#### b. Akuisisi Pengetahuan

Pada tahap ini, akuisisi pengetahuan akan dilakukan seiring dengan spesifikasi persyaratan, dan akan berkurang sejalan dengan perkembangan perancangan ontologi. Tujuan tahap ini adalah untuk memperoleh pengetahuan yang relevan dalam merancang ontologi produk *sunscreen*.

#### c. Konseptualisasi

Tahap konseptualisasi melibatkan usaha untuk merumuskan domain pengetahuan ke dalam sebuah model konseptual, sekaligus menjaga serta mengelola pengetahuan yang didapatkan selama proses akuisisi pengetahuan. Pengetahuan dalam domain akan dimodelkan menjadi sebuah model konseptual. Pada penelitian ini domainnya yaitu *sunscreen*, akan dibuat sebuah model konseptual dari pengetahuan dalam domain produk *sunscreen*. Model ini memiliki fungsi untuk menggambarkan permasalahan dan juga solusi terkait produk *sunscreen* menggunakan istilah-istilah domain yang sudah dikenali pada tahap spesifikasi ontologi sebelumnya.

#### d. Integrasi

Tahapan integrasi merupakan evaluasi untuk memanfaatkan definisi-definisi ontologi yang sudah ada sebelumnya, lalu mengintegrasikannya ke dalam ontologi baru yang sedang dikembangkan. Ini menghindarkan perlunya memulai perancangan ontologi dari awal lagi. Proses ini melibatkan penyatuan model-model ontologi yang sudah ada dengan model ontologi terbaru yang sedang dibuat. Karena ontologi dapat digunakan kembali, dilakukan peninjauan terhadap ontologi-ontologi sebelumnya untuk memilih konsep yang relevan

yang dapat disatukan ke dalam ontologi yang sedang dikembangkan. Tujuannya adalah memastikan bahwa definisi-definisi baru dan lama yang digabungkan didasarkan pada kumpulan istilah dasar yang sama.

**e. Implementasi**

Implementasi merupakan tahap dimana rancangan ontologi yang telah disusun sejak awal direalisasikan, mulai dari spesifikasi hingga integrasi. Pada implementasi ini, rancangan ontologi didefinisikan ulang dan diterapkan dengan menggunakan perangkat lunak pengembangan ontologi yaitu Protégé. Protégé merupakan aplikasi khusus untuk membangun dan mengolah pengetahuan berbasis ontologi. Langkah-langkah implementasi ontologi menggunakan Protégé mencakup: pendefinisian class-class yang merepresentasikan konsep domain, pengaturan object property sebagai atribut kelas, pembuatan instances sebagai individu-individu, serta penentuan relasi antar elemen ontologi.

**f. Evaluasi**

Pada tahap evaluasi ini, dilakukan penilaian teknis terhadap ontologi yang telah dibuat, termasuk perangkat lunak yang digunakan serta dokumentasi yang relevan. Penilaian ini berdasarkan kerangka acuan di setiap tahap dan antar tahap dalam siklus pengembangan ontologi. Pada tahap ini, konsistensi konsep ontologi diuji dengan menggunakan fitur penalaran. Selanjutnya, serangkaian pertanyaan terkait domain dijawab menggunakan SPARQL Query, sebuah bahasa query mirip SQL yang digunakan untuk mengambil data dalam format OWL (*Ontology Web Language*) [5].

**g. Dokumentasi**

Dokumentasi berfungsi sebagai sarana untuk mengatur dan memaparkan pertanyaan-pertanyaan penting terkait dengan ontologi yang telah dihasilkan. Meski belum ada panduan baku, dokumentasi ontologi biasanya mencakup kode ontologi, penjelasan deskriptif menggunakan bahasa alami pada definisi formal, publikasi ilmiah di konferensi/jurnal terkait ontologi tersebut, serta visualisasi ontologi dalam bentuk ontograf.

**3. Hasil dan Diskusi**

Penelitian ini telah menghasilkan sebuah ontologi yang berfokus pada domain produk *sunscreen*. Berikut adalah hasil yang diperoleh dari setiap tahap metode penelitian yang telah diterapkan dalam pengembangan ontologi ini.

**3.1. Spesifikasi**

Dibawah ini merupakan spesifikasi dari ontologi yang telah berhasil dirancang:

- a. Domain: Produk *sunscreen*
- b. Tujuan: Membangun model ontologi untuk sistem pencarian produk *sunscreen* yang sesuai dengan kulit
- c. Tingkat formalitas: Formal
- d. Lingkup: Produk *sunscreen*
- e. Sumber pengetahuan: Jurnal, internet

**3.2. Akuisi Pengetahuan**

Dalam penelitian ini, tahap akuisi pengetahuan dilakukan dengan mempelajari dan mengidentifikasi pengetahuan struktur ontologi yang akan dirancang, termasuk nilai-nilai, atribut, hingga konsep. Data yang digunakan sebagai bahan dalam perancangan model ontologi produk *sunscreen* ini merupakan data latih diperoleh dari berbagai sumber yang dianggap relevan dan dapat dipercaya, seperti jurnal dan website resmi di internet. Data yang diperoleh dan dipergunakan yaitu berjumlah 10 produk *sunscreen* lokal Indonesia. Data dibagi berdasarkan

spesifikasi seperti nama *sunscreen*, harga, area pemakaian, kadar SPF, kadar PA, tipe kulit pengguna, dan tekstur *sunscreen* seperti yang dapat dilihat dalam Tabel 1.

### 3.3. Konseptualisasi

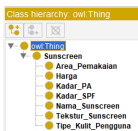
Pada tahap ini, dibuat daftar istilah lengkap yang berhubungan dengan produk *sunscreen*, meliputi konsep, instance, kata kerja, dan properti yang relevan. Tujuannya adalah untuk mengembangkan konsep-konsep yang menggambarkan masalah dan solusi terkait dengan *sunscreen*. Selanjutnya, tahap konseptualisasi mengatur domain pengetahuan menjadi model konseptual dengan mengelola pengetahuan dari proses akuisisi pengetahuan sebelumnya. Model konseptual domain *sunscreen* ini lalu diubah menjadi model formal dalam bahasa ontologi. Hasilnya adalah terbentuknya model konseptual ontologi produk *sunscreen* yang solid sebagai dasar pengembangan lebih lanjut dalam domain tersebut.

### 3.4. Integrasi

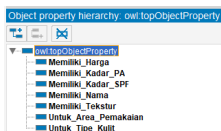
Tahap integrasi ontologi adalah proses penyatuan ontologi yang telah ada dan sesuai dengan ontologi produk *sunscreen* yang akan dikembangkan. Langkah ini penting untuk memastikan konsistensi definisi antara konsep yang baru dan yang sudah ada sebelumnya, serta mengacu pada istilah dasar yang sama. Dalam penelitian produk *sunscreen* ini, proses integrasi ontologi melibatkan evaluasi terhadap ontologi terkait produk perawatan kulit atau kosmetik yang telah dibangun sebelumnya. Tujuannya adalah untuk memilih konsep dan definisi yang relevan dengan domain *sunscreen*. Seperti terdapat ontologi perawatan kulit atau kosmetik dengan definisi nama produk, tipe kulit, tipe produk, dan informasi lain yang berkaitan dengan *sunscreen*, maka definisi tersebut dapat diintegrasikan ke dalam ontologi *sunscreen* baru sehingga tidak perlu dimulai dari awal.

### 3.5. Implementasi

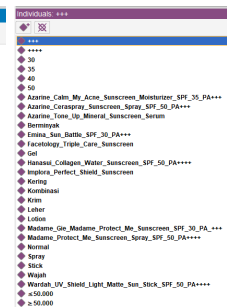
Pada tahap implementasi, digunakan perangkat lunak pengembangan ontologi yaitu Protégé dalam merancang ontologi *sunscreen*. Proses ini diawali dengan pembentukan *class hierarchy*.



Gambar 2. Class Hierarchy



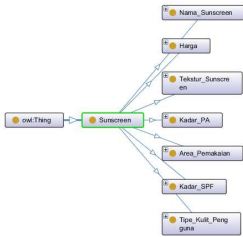
Gambar 3. Object Property Hierarchy



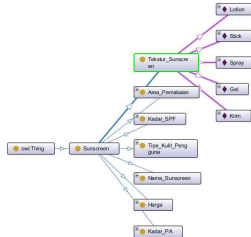
Gambar 4. Individuals

Terdapat satu *class* yaitu *Sunscreen*, dengan beberapa *subclass* meliputi *Area\_Pemakaian*, *Harga*, *Kadar\_PA*, *Kadar\_SPF*, *Nama\_Sunscreen*, *Tekstur\_Sunscreen*, dan *Tipe\_Kulit\_Pengguna* seperti yang terlihat dalam Gambar 2. Kemudian dilakukan pembentukan *object property hierarchy*, dimana ini digunakan untuk mendefinisikan dan mengatur hierarki atau relasi antara objek properties dalam ontologi. Adapun *object property hierarchy* dalam ontologi

ini yaitu Memiliki\_Harga, Memiliki\_Kadar\_PA, Memiliki\_Kadar\_SPF, Memiliki\_Nama, Memiliki\_Tekstur, Untuk\_Area\_Pemakaian, dan Untuk\_Tipe\_Kulit. Dalam Gambar 4 diatas terdapat beberapa instance atau individu yang telah dibuat untuk setiap kelas (class) dalam ontologi *sunscreen* ini. Terdapat 29 *individuals* yang terdiri atas 2 *individuals* Area\_Pemakaian, 2 *individuals* Harga, 2 *individuals* Kadar\_PA, 4 *individuals* Kadar\_SPF, 10 *individuals* Nama\_Sunscreen, 5 *individuals* Tekstur\_Sunscreen, dan 4 *individuals* Tipe\_Kulit\_Pengguna.

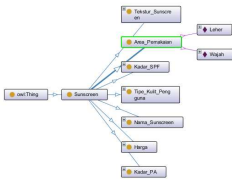


**Gambar 5.** Ontograf dari Ontologi *Sunscreen* Class *Sunscreen*

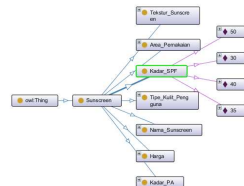


**Gambar 6.** Ontograf dari Ontologi *Sunscreen* Class *Tekstur\_Sunscreen*

Adapun visualisasi ontograf dari ontologi class *Sunscreen* yang dapat dilihat dalam Gambar 5 diatas. Visualisasi ini dibuat untuk menampilkan *class hierarchy* dengan panah biru yang menghubungkan *class* ke *subclass-subclassnya*. *Subclass* dalam ontologi mendefinisikan kategori yang lebih spesifik dibandingkan dengan *class*, dan kemudian individual diklasifikasikan sebagai anggota dari masing-masing *subclass* ini, yang ditunjukkan dengan panah ungu pada Gambar 6 hingga Gambar 12.



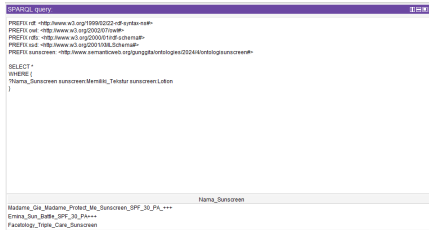
**Gambar 7.** Ontograf dari Ontologi *Sunscreen* Class *Area\_Pemakaian*



**Gambar 8.** Ontograf dari Ontologi *Sunscreen* Class *Kadar\_SPF*



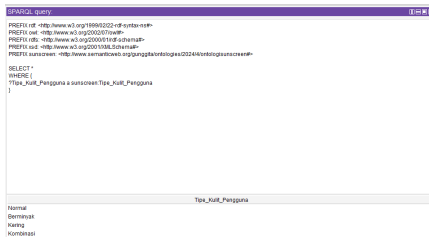




Gambar 13. Hasil Pertanyaan 1

**b. Pertanyaan 2**

Pertanyaan: Apa saja tipe kulit pengguna yang ada dalam *sunscreen*? Hasilnya dapat dilihat pada Gambar 14. Telah berhasil ditampilkan "Tipe\_Kulit\_Pengguna" yang ada dalam *sunscreen* yaitu Normal, Berminyak, Kering, dan Kombinasi.



Gambar 14. Hasil Pertanyaan 2

**c. Pertanyaan 3**

Pertanyaan: Apa saja *sunscreen* yang dapat digunakan diwajah, cocok untuk kulit berminyak, memiliki kadar SPF 40, dan memiliki tekstur lotion? Hasilnya dapat dilihat pada Gambar 15. Telah berhasil ditampilkan "Nama\_Sunscreen" "Untuk\_Area\_Pemakaian" di "Wajah" dan dapat digunakan "Untuk\_Tipe\_Kulit" "Berminyak" kemudian "Memiliki\_Kadar\_SPF" "40" serta "Memiliki\_Tekstur" "Lotion" yaitu Facetology Triple Care Sunscreen.



Halaman ini sengaja dibiarkan kosong

## Algoritma K-Means untuk *Clustering* Provinsi di Indonesia Berdasarkan Kasus Stunting

Syelvya Julianti<sup>a1</sup>, I Made Widiartha<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>syelviavia06@gmail.com  
<sup>2</sup>madewidiartha@unud.ac.id

### Abstract

*Stunting is a nutritional issue that poses a global challenge, especially in developing countries like Indonesia. According to UNICEF, Indonesia ranks among the top five countries with the highest stunting prevalence. To address this issue, clustering provinces in Indonesia each year can help ensure equitable food distribution and other resources. This can be done using the K-Means clustering algorithm, with the optimal number of clusters determined by the elbow method and evaluated using the silhouette coefficient and Davies-Bouldin index. The optimal number of clusters was found to be 3, with a silhouette coefficient of 0.50 and a Davies-Bouldin index of 0.70. In 2020, there were 15 provinces in cluster 1, 6 provinces in cluster 2, and 17 provinces in cluster 3. In 2021, 15 provinces were in cluster 1, 17 in cluster 2, and 6 in cluster 3. In 2022, there were 17 provinces in cluster 1, 14 in cluster 2, and 7 in cluster 3. In 2023, 5 provinces were in cluster 1, 14 in cluster 2, and 19 in cluster 3. By 2024, there were 18 provinces in cluster 1, 17 in cluster 2, and 3 in cluster 3.*

**Keywords:** *Stunting, K-Means, Elbow Method, Silhouette Coefficient, Davies Bouldin Index*

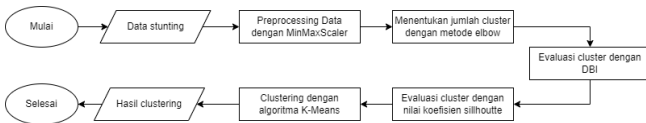
### 1. Pendahuluan

Salah satu permasalahan gizi yang menjadi permasalahan global khususnya negara berkembang seperti Indonesia adalah stunting. Menurut WHO stunting adalah gangguan kembang tumbuh anak karena terserang infeksi, kurangnya asupan gizi, ataupun stimulasi yang kurang memadai. Menurut UNICEF, Indonesia termasuk dalam lima negara dengan kasus stunting tertinggi dari 88 negara di seluruh dunia yang menjadikannya salah satu negara berkembang dengan prevalensi stunting tinggi. Wakil Ketua Komisi IX DPR RI Kurniasi Mufidayati mengatakan bahwa pada pertengahan tahun 2023 prevalensi stunting di Indonesia mencapai 21,6 persen, sedangkan target yang ingin dicapai pada tahun 2024 adalah 14 persen [1]. Sehingga, untuk mencapai target tersebut Indonesia harus mengurangi kasus stunting sebesar 7,6 persen dalam kurun waktu satu tahun. Untuk mencapai tujuan tersebut, langkah awal yang dapat dilakukan adalah mengelompokkan provinsi berdasarkan persentase kasus stunting sehingga dapat dilakukan pemerataan akses kesehatan, pangan, dan pendidikan gizi. Pengelompokkan dapat dilakukan dengan menggunakan algoritma *clustering* seperti K-Means. Algoritma K-Means digunakan karena algoritma ini memiliki kompleksitas yang rendah sehingga dinilai lebih efisien [2]. Tetapi, algoritma K-Means memiliki kekurangan dalam penentuan jumlah kluster. Oleh karena itu, dibutuhkan metode yang mampu menentukan jumlah cluster salah satunya adalah metode *elbow*. Tetapi, metode *elbow* dianggap kurang efektif karena terkadang bersifat ambigu [3]. Untuk memastikan bahwa jumlah *cluster* dengan metode *elbow* telah optimal maka dilakukan evaluasi dengan *silhouette coefficient* dan *Davies Bouldin Index* (DBI). Dalam penelitian [4] algoritma K-Means digunakan dalam klasifikasi sekolah dasar, didapatkan 5 *cluster* dengan skor *silhouette* kurang dari 0,71 dengan persentase *cluster* 1 21,43%, *cluster* 2 28,25%, *cluster* 3 14,28%, *cluster* 4 21,42%, dan *cluster* 5 14,29%, dari hasil tersebut Pemerintah Kota Pangkalpinang dapat mengklasifikasikan sekolah tingkat dasar dengan baik. Dalam penelitian [5] algoritma K-Means digunakan dalam pengelompokan daerah berdasarkan jumlah industri, didapatkan bahwa nilai DBI terbaik adalah 0,175 pada saat jumlah *cluster* 3 sehingga didapatkan kesimpulan bahwa

evaluasi nilai DBI pada algoritma K-Means menghasilkan optimasi jumlah *cluster*. Berdasarkan latar belakang permasalahan dan penelitian terdahulu, maka algoritma K-Means akan digunakan untuk *clustering* provinsi di Indonesia berdasarkan kasus stunting dengan menggunakan metode *elbow* untuk penentuan jumlah *cluster*, lalu dievaluasi dengan *silhouette coefficient* dan *Davies Bouldin Index* (DBI).

## 2. Metode Penelitian

Gambar 1 merupakan tahapan penelitian yang akan dilakukan. Input yang digunakan berupa persentase kasus stunting seluruh provinsi di Indonesia. *Preprocessing* data dilakukan dengan MinMaxScaler. Penentuan jumlah *cluster* menggunakan metode *elbow*, lalu dievaluasi dengan *silhouette coefficient* dan DBI. Selanjutnya dilakukan *clustering* dengan algoritma K-Means sehingga mendapatkan hasil *cluster*.



Gambar 1. Tahapan Penelitian

### 2.1 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah persentase kasus stunting 38 provinsi di Indonesia mulai dari tahun 2020 hingga 2024 yang didapatkan pada [website aksi.bangka.kemendagri.go.id](https://www.bangka.kemendagri.go.id). Atribut yang digunakan adalah provinsi dan persentase kasus stunting setiap tahun dimana *clustering* akan dilakukan pada setiap provinsi dalam satu tahun.

### 2.2 Preprocessing Data

Dalam penelitian ini *preprocessing* data berupa normalisasi data dengan MinMaxScaler. Normalisasi data berguna untuk penskalaan kembali terhadap nilai, sehingga proses pengolahan data menjadi lebih mudah [6]. MinMaxScaler merupakan teknik normalisasi data yang mengubah nilai pada data dalam skala nilai 0 dan 1 [7].

### 2.3 Jumlah Cluster

Dalam melakukan *clustering* penentuan jumlah *cluster* adalah hal yang utama, tetapi pada algoritma K-Means penentuan jumlah *cluster* dilakukan secara *random*. Oleh karena itu, dalam penentuan jumlah *cluster* menggunakan metode *elbow*. Tetapi, metode *elbow* terkadang ambigu dan bersifat subjektif sehingga perlu dilakukan evaluasi menggunakan *silhouette coefficient* dan DBI.

#### a. Metode Elbow

Menurut [8] metode *elbow* adalah metode untuk menentukan jumlah *cluster* berdasarkan persentase perbandingan nilai  $k$  atau jumlah *cluster* pada titik tertentu. Hasil dari metode *elbow* direpresentasikan dalam bentuk grafik dengan penentuan jumlah *cluster* terbaik apabila terdapat penurunan nilai terbesar sehingga membentuk sudut pada grafik. Grafik tersebut terbentuk dari hasil perhitungan SSE (Sum of Square Error) yang dibandingkan [9]. SSE pada K-Means dihitung menggunakan rumus pada persamaan 1 [10].

$$SSE = \sum_{k=1}^K \sum_{x_i} |x_i - c_k|^2 \quad (1)$$

Keterangan:

- $K$  = Cluster ke-c
- $c_k$  = Pusat data ke-i
- $x_i$  = Jarak data ke-i

b. *Silhouette Coefficient*

*Silhouette coefficient* mengevaluasi *cluster* dengan melihat kualitas objek pada suatu *cluster*. Nilai *silhouette coefficient* dihitung dari jumlah *cluster* 2 hingga jumlah *cluster* n-1 dengan mencari nilai tertinggi dari *Silhouette Index Global*, rumus *silhouette coefficient* terdapat persamaan 2 [10].

$$SC = maks_k SI (k) \tag{2}$$

Keterangan:

- $SC$  = *Silhouette coefficient*
- $SI$  = *Silhouette index global*
- $k$  = Jumlah *cluster*

Hasil perhitungan *silhouette coefficient* berada pada rentang nilai 1 dan -1, dengan hasil perhitungan terbaik ketika mendekati 1 karena mengindikasikan kualitas hasil *clustering* yang baik.

c. *Devies Bouldin Index (DBI)*

DBI merupakan metode untuk mengevaluasi *cluster* berdasarkan ukuran *cluster* dan penyebarannya [11]. Nilai DBI yang baik adalah nilai DBI yang mendekati 0 karena hal tersebut menunjukkan bahwa hasil *cluster* yang diperoleh baik dan optimal. Perhitungan DBI dapat dilakukan dengan menggunakan persamaan 3 [12].

$$DBI = \frac{1}{C} \sum_{i=1}^C R_i \tag{3}$$

$$R_i = \frac{\max_{j=1, \dots, C, i \neq j} R_{ij}, R_{ij}}{a(V_i, V_j)} \tag{4}$$

Keterangan:

- $C$  = Jumlah *cluster*
- $R_{ij}$  = Ukuran kemiripan *cluster* ke-i dan ke-j
- $S_i$  = Ukuran dispresi *cluster* i

**2.4. K-Means**

K-Means merupakan algoritma *clustering* yang telah umum digunakan karena mampu melakukan *clustering* dalam jumlah data besar dan mudah diimplementasikan. Algoritma K-Means bertujuan meminimalkan kemiripan data antar *cluster* dan memaksimalkan kesamaan data dalam satu *cluster* [13]. Kemiripan data dihitung berdasarkan jarak data dengan centroid (titik pusat data). Perhitungan jarak data dengan centroid dapat dilakukan dengan rumus pada persamaan 5 yang merupakan rumus *Euclidean Distance*.

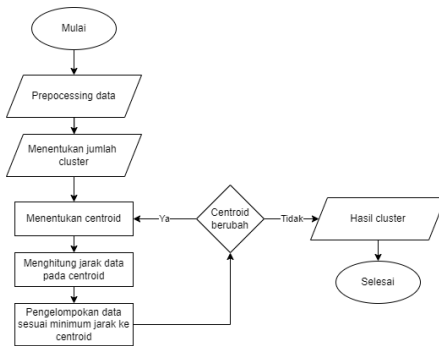
$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{5}$$

Keterangan:

- $d(x, y)$  = Jarak antar data x ke y
- $x_i$  = Data uji ke-i
- $y_i$  = Data latih ke-i

Flowchart algoritma K-Means terdapat pada gambar 2 dengan detail tahapan yang dilakukan algoritma K-Means, yaitu:

- Melakukan *preprocessing* data berupa normalisasi data atau lainnya yang berguna untuk memaksimalkan hasil *clustering*.
- Menentukan jumlah *clustering* atau pada algoritma K-Means disebut dengan nilai K.
- Menentukan *centroid* atau pusat data sesuai dengan jumlah K yang telah ditentukan.
- Menghitung jarak data dari *centroid*.
- Melakukan pengelompokan data berdasarkan jarak terdekat atau minimum antara *centroid* dengan data.
- Dilakukan pemeriksaan, apabila *centroid* mengalami perpindahan maka akan dilakukan ulang penentuan *centroid* tetapi jika tidak ada perpindahan maka hasil pengelompokan atau *clustering* akan menjadi hasil akhir dari proses *clustering*.



Gambar 2. Flowchart Algoritma K-Means

### 3. Hasil dan Diskusi

Dalam penelitian ini *clustering* data menggunakan bahasa pemrograman Python dengan memanfaatkan pustaka *scikit-learn*. Dengan tahapan berupa *preprocessing*, menentukan jumlah *cluster* menggunakan metode *elbow*, evaluasi *cluster* dengan silhouette coefficient dan DBI, lalu *clustering* dengan K-Means.

#### 3.1 Preprocessing Data

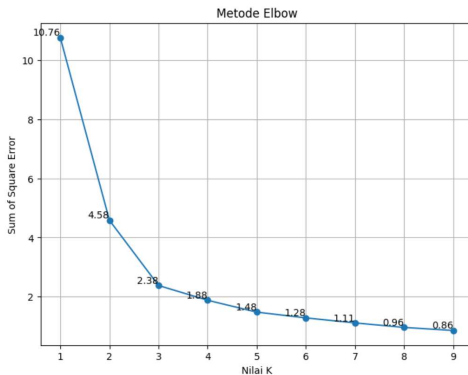
Tahapan *preprocessing* data yang dilakukan berupa normalisasi dengan menggunakan metode *MinMaxScaler*. Gambar 3 adalah data hasil normalisasi pada tahun 2020 dengan data yang diurutkan berdasarkan nama provinsi yang terurut secara menyamping sesuai pada [website aksi.bangda.kemendagri.go.id](http://website.aksi.bangda.kemendagri.go.id). Dengan menggunakan metode *MinMaxScaler* maka data akan berada pada skala nilai 0 dan 1.

Hasil Normalisasi Data 2020:  
 [0.46593497 0.23426573 0.60839161 0.26573427 0.26223776 0.07692308  
 0.28671329 0.18181818 0.11188811 0.31818182 0. 0.31118881  
 0.45454545 0.41258741 0.3986014 0.16783217 0.32517483 0.7027972  
 0.91608392 0.95454545 0.77272727 0.45454545 0.5034965 1.  
 0.18181818 0.47552448 0.41258741 0.58041958 0.38111888 0.79020979  
 0.28671329 0.32167832 0.41608392 0.16083916 0.2027972 0.39160839  
 0.57692308 0.61538462]

Gambar 3. Data Tahun 2020 Setelah Normalisasi

### 3.2 Penentuan Jumlah Cluster

Sebelum melakukan *clustering* perlu ditetapkan nilai *k* atau jumlah *cluster* menggunakan metode *elbow*. Metode *elbow* direpresentasikan dalam bentuk grafik berdasarkan nilai *Sum of Square Error* (SSE). Penentuan jumlah *cluster* ditentukan berdasarkan nilai SSE yang memiliki penurunan signifikan, pada gambar 4 dapat dilihat ketika nilai *k* = 3 terdapat penurunan signifikan. Sehingga, berdasarkan metode *elbow* dapat ditentukan jumlah *cluster* yang digunakan adalah 3.



Gambar 4. Metode Elbow

### 3.3 Evaluasi Cluster

Setelah ditentukan bahwa jumlah *cluster* adalah 3 maka dilakukan evaluasi menggunakan *silhouette coefficient* dan DBI. *Silhouette coefficient* digunakan untuk mengevaluasi kemiripan satu data dengan data lainnya dalam suatu *cluster*, lalu DBI digunakan untuk mengevaluasi pemisahan antar *cluster*. Nilai *silhouette coefficient* yang terbaik adalah yang mendekati 1, sedangkan nilai DBI yang terbaik adalah yang mendekati 0. Pada tabel 1 dapat dilihat bahwa hasil *silhouette coefficient* terbaik terdapat pada nilai *cluster* 3 dengan nilai 0,50 dan nilai DBI terbaik juga didapatkan pada saat jumlah *cluster* 3 dengan nilai 0,70 yang dapat dilihat pada tabel 2. Dari kedua hasil evaluasi tersebut dan metode *elbow* yang dilakukan sebelumnya didapatkan bahwa jumlah *cluster* yang optimal adalah 3.



**Tabel 1.** Silhouette Score

Cluster	Silhouette Score
2	0,47
3	0.50
4	0.39
5	0.39
6	0.38
7	0.28
8	0.28
9	0.27

**Tabel 2.** Davies-Bouldin Index

Cluster	Davies-Bouldin Index
2	0,80
3	0.70
4	0.90
5	0.89
6	0.83
7	0.77
8	0.88
9	0.76

### 3.4 Clustering Data

Setelah mendapatkan jumlah *cluster* yang optimal maka akan dilakukan *clustering* data menggunakan algoritma K-Means. *Clustering* dilakukan berdasarkan kasus stunting pada setiap provinsi di Indonesia setiap tahunnya dari tahun 2020 hingga 2024. Karena jumlah *cluster* yang optimal adalah tiga maka didapatkan hasil *cluster* 0, *cluster* 1, dan *cluster* 2. Hasil *cluster* pada tahun 2020 terdapat pada tabel 3 dimana terdapat 15 provinsi pada *cluster* 1, 6 provinsi pada *cluster* 2, dan 17 provinsi pada *cluster* 3.

**Tabel 3.** Hasil *Clustering* Tahun 2020

Cluster 1	Cluster 2	Cluster 3
Aceh	Nusa Tenggara Barat	Sumatera Utara
Sumatera Barat	Nusa Tenggara Timur	Riau
Jawa Tengah	Kalimantan Barat	Jambi
Di Yogyakarta	Kalimantan Tengah	Sumatera Selatan
Jawa Timur	Kalimantan Utara	Bengkulu
Kalimantan Selatan	Sulawesi Barat	Lampung

Cluster 1	Cluster 2	Cluster 3
Kalimantan Timur		Kepulauan Bangka Belitung
Sulawesi Tengah		Kepulauan Riau
Sulawesi Selatan		Dki Jakarta
Sulawesi Tenggara		Jawa Barat
Gorontalo		Banten
Papua Barat		Bali
Papua		Sulawesi Utara
Papua Tengah		Sulawesi Utara
Papua Barat Daya		Maluku
		Maluku Utara
		Sulawesi Utara

Hasil *cluster* pada tahun 2021 terdapat pada tabel 4 dimana terdapat 15 provinsi pada *cluster* 1, 17 provinsi pada *cluster* 2, dan 6 provinsi pada *cluster* 3.

**Tabel 4.** Hasil *Clustering* Tahun 2021

Cluster 1	Cluster 2	Cluster 3
Aceh	Sumatera Utara	Nusa Tenggara Barat
Sumatera Barat	Riau	Nusa Tenggara Timur
Jawa Tengah	Jambi	Kalimantan Barat
Di Yogyakarta	Sumatera Selatan	Kalimantan Utara
Jawa Timur	Bengkulu	Sulawesi Tenggara
Kalimantan Tengah	Lampung	Sulawesi Barat
Kalimantan Selatan	Kepulauan Bangka Belitung	
Kalimantan Timur	Kepulauan Riau	
Sulawesi Tengah	Dki Jakarta	
Sulawesi Selatan	Jawa Barat	
Maluku Utara	Banten	
Papua Barat	Bali	
Papua	Sulawesi Utara	
Papua Selatan	Gorontalo	
Papua Barat Daya	Maluku	
	Papua Pegunungan	
	Papua Tengah	

Hasil *cluster* pada tahun 2022 terdapat pada tabel 5 dimana terdapat 17 provinsi pada *cluster* 1, 14 provinsi pada *cluster* 2, dan 7 provinsi pada *cluster* 3.

**Tabel 5.** Hasil *Clustering* Tahun 2022

Cluster 1	Cluster 2	Cluster 3
Aceh	Sumatera Utara	Nusa Tenggara Barat
Sumatera Barat	Riau	Nusa Tenggara Timur
Jawa Tengah	Jambi	Kalimantan Barat
Di Yogyakarta	Sumatera Utara	Nusa Tenggara Barat
Jawa Timur	Riau	Nusa Tenggara Timur
Kalimantan Tengah	Jambi	Kalimantan Barat
Kalimantan Selatan	Sumatera Selatan	Kalimantan Utara
Kalimantan Timur	Bengkulu	Sulawesi Utara
Sulawesi Tengah	Lampung	Sulawesi Tenggara
Sulawesi Selatan	Kepulauan Bangka Belitung	Sulawesi Barat

Cluster 1	Cluster 2	Cluster 3
Sulawesi Tenggara	Kepulauan Riau	
Gorontalo	Dki Jakarta	
Maluku	Jawa Barat	
Maluku Utara	Banten	
Papua Barat	Bali	
Papua	Sulawesi Utara	
Papua Tengah	Papua Pegunungan	
Papua Barat Daya		

Hasil *cluster* pada tahun 2023 terdapat pada tabel 6 dimana terdapat 5 provinsi pada *cluster* 1, 14 provinsi pada *cluster* 2, dan 19 provinsi pada *cluster* 3.

**Tabel 6.** Hasil *Clustering* Tahun 2023

Cluster 1	Cluster 2	Cluster 3
Nusa Tenggara Barat	Sumatera Utara	Aceh
Nusa Tenggara Timur	Riau	Sumatera Barat
Kalimantan Barat	Jambi	Jawa Barat
Sulawesi Barat	Sumatera Selatan	Di Yogyakarta
Papua Selatan	Bengkulu	Jawa Tengah
	Lampung	Jawa Timur
	Kepulauan Bangka Belitung	Kalimantan Tengah
	Kepulauan Riau	Kalimantan Selatan
	Dki Jakarta	Kalimantan Timur
	Banten	Kalimantan Utara
	Bali	Sulawesi Selatan
	Sulawesi Utara	Sulawesi Tenggara
	Sulawesi Tengah	Kalimantan Barat Daya
	Gorontalo	Maluku Utara
	Maluku	Papua Barat
	Papua Pegunungan	Papua
		Papua Tengah

Hasil *cluster* pada tahun 2024 terdapat pada tabel 7 dimana terdapat 18 provinsi pada *cluster* 1, 17 provinsi pada *cluster* 2, dan 3 provinsi pada *cluster* 3.

**Tabel 7.** Hasil *Clustering* Tahun 2024

Cluster 1	Cluster 2	Cluster 3
Aceh	Sumatera Barat	Nusa Tenggara Timur
Sumatera Utara	Jawa Tengah	Sulawesi Barat
Riau	DI Yogyakarta	Papua Barat Daya
Jambi	Nusa Tenggara Barat	
Sumatera Selatan	Kalimantan Barat	
Bengkulu	Kalimantan Tengah	
Lampung	Kalimantan Selatan	
Kepulauan Bangka Belitung	Kalimantan Timur	
Kepulauan Riau	Kalimantan Utara	
DKI Jakarta	Sulawesi Tengah	
Jawa Barat	Sulawesi Selatan	
Jawa Timur	Sulawesi Tenggara	
Banten	Maluku Utara	

Cluster 1	Cluster 2	Cluster 3
Bali	Papua Barat	
Sulawesi Utara	Papua	
Gorontalo	Papua Selatan	
Maluku	Papua Tengah	
Papua Pegunungan		

#### 4. Kesimpulan

Pada penelitian ini dilakukan *clustering* data mengenai kasus stunting berdasarkan provinsi di Indonesia pada tahun 2020 hingga 2024 didapatkan hasil bahwa data stunting di Indonesia dapat dibagi menjadi 3 *cluster*, yaitu *cluster 1*, *cluster 2*, dan *cluster 3*. Jumlah *cluster* tersebut didapatkan dari pencarian nilai K atau jumlah *cluster* optimal menggunakan metode elbow lalu dievaluasi dengan *silhouette coefficient* dan *davies bouldin index*. Hasil terbaik *silhouette coefficient* dengan nilai 0,50 dan *davies bouldin index* dengan nilai 0,70 pada jumlah *cluster* 3. Oleh karena itu, jumlah *cluster* 3 digunakan untuk *clustering* data stunting setiap tahun di Indonesia. Didapatkan hasil bahwa pada tahun 2020 terdapat terdapat 15 provinsi pada *cluster 1*, 6 provinsi pada *cluster 2*, dan 17 provinsi pada *cluster 3*. Pada tahun 2021 terdapat 15 provinsi pada *cluster 1*, 17 provinsi pada *cluster 2*, dan 6 provinsi pada *cluster 3*. Pada tahun 2022 terdapat terdapat 17 provinsi pada *cluster 1*, 14 provinsi pada *cluster 2*, dan 7 provinsi pada *cluster 3*. Tahun 2023 terdapat 5 provinsi pada *cluster 1*, 14 provinsi pada *cluster 2*, dan 19 provinsi pada *cluster 3*. Lalu, pada tahun 2024 terdapat 18 provinsi pada *cluster 1*, 17 provinsi pada *cluster 2*, dan 3 provinsi pada *cluster 3*.

#### Daftar Pustaka

- [1] K. Mufidayati, "Kurniasih: Di Tahun 2024, Penurunan Stunting Harus Jadi Fokus Bersama," [dpr.go.id](https://www.dpr.go.id/berita/detail/id/48290/t/Kurniasih:%20Di%20Tahun%202024,%20Penurunan%20Stunting%20Harus%20Jadi%20Fokus%20Bersama#:~:text=Dikatakan%20Kurniasih%2C%20pada%20pertengahan%20tahun,adalah%2014%20persen%20pada%202024.), Accessed: May 05, 2024. [Online]. Available: <https://www.dpr.go.id/berita/detail/id/48290/t/Kurniasih:%20Di%20Tahun%202024,%20Penurunan%20Stunting%20Harus%20Jadi%20Fokus%20Bersama#:~:text=Dikatakan%20Kurniasih%2C%20pada%20pertengahan%20tahun,adalah%2014%20persen%20pada%202024.>
- [2] H. Mutiasari, T. W. Purboyo, and R. A. Nugrahaeni, "Sistem Rekomendasi Film Menggunakan Metode K-means Clustering," in *eProceedings of Engineering*, 2021, pp. 6755–6764.
- [3] S. Mutalib, N. H. Hasbullah, S. Abdul-Rahman, M. R. Shamsuddin, and A. M. A. Malik, "Herbal Plant Analysis Based on Leaf Features using K-Means Clustering," in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, May 2022. doi: 10.1088/1755-1315/1019/1/012026.
- [4] Y. Bagus Pratama and A. Setiawan, "Implementasi Machine Learning Menggunakan Algoritma K-Means Untuk Klasifikasi Sekolah Dasar," *RESOLUSI: Rekayasa Teknik Informatika dan Informasi*, vol. 4, no. 3, pp. 249–257, 2024, [Online]. Available: <https://djournals.com/resolusi>
- [5] E. Muningsih, I. Maryani, and V. R. Handayani, "Penerapan Metode K-Means dan Optimalis Jumlah Cluster dengan Index Davies Bouldin untuk Clustering Propinsi Berdasarkan Potensi Desa," *Jurnal Sains dan Manajemen*, vol. 9, no. 1, 2021, [Online]. Available: [www.bps.go.id](http://www.bps.go.id)
- [6] R. G. Whendasromo and Joseph, "Analisis Penerapan Normalisasi Data Dengan Menggunakan Z-Score Pada Kinerja Algoritma K-NN," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 4, pp. 872–876, Aug. 2022, doi: 10.30865/jurikom.v9i4.4526.
- [7] W. Purba et al., "Penerapan Data Mining Untuk Pengelolaan Data Rekam Medis Menggunakan Metode K-means Clustering Pada Rumah Sakit Royal Prima Medan," *Jurnal TEKINKOM*, vol. 6, no. 1, pp. 158–168, 2023, doi: 10.37600/tekinkom.v6i1.857.
- [8] N. Putu, E. Merliana, and A. J. Santoso, "Analisa Penentuan Jumlah Cluster Terbaik pada Metode K-Means," in *Pros. Semin. Nas. MULTI DISIPLIN ILMU CALL*, 2015, pp. 978–979.

- [9] E. Muningsih and S. Kiswati, "Sisitem Aplikasi Berbasis Optimasi Metode Elbow Untuk Penentuan Clustering Pelanggan," *JOUITICA*, vol. 3, no. 1, 2018.
- [10] D. Ayu, I. C. Dewi, and K. Pramita, "Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Pengelompokan Produksi Kerajinan Bali," *JURNAL MATRIX*, vol. 9, no. 3, pp. 102–109, 2019.
- [11] A. Putri Riani, A. Voutama, and T. Ridwan, "Penerapan K-Means Clustering Dalam Pengelompokan Hasil Belajar Peserta Didik Dengan Metode Elbow," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no. 1, pp. 164–172, 2023, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [12] I. Rosaliyah, B. Nurhakim, and J. Manajemen, "Clustering Kejadian Bencana Alam di Jawa Barat Berdasarkan Jenis Bencana Menggunakan K-means," *Jurnal Teknik Elektro dan Informatika*, vol. 18, no. 1, pp. 10–16, 2023, [Online]. Available: <https://opendata.jabarprov.go.id/>
- [13] I. Nyoman and M. Adiputra, "Clustering Penyakit DBD Pada Rumah Sakit Dharma Kerti Menggunakan Algoritma K-Means," *INSERT: Information System and Emerging Technology Journal*, vol. 2, no. 2, pp. 99–105, 2021.

## Implementasi Metode Design Thinking dalam Perancangan UI/UX Aplikasi Wisata Bali

I Kadek Agus Candra Widnyana<sup>a1</sup>, I Made Widiartha<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana

Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

<sup>1</sup>widnyana.2208561129@student.unud.ac.id@gmail.com

<sup>2</sup>madewidiartha@unud.ac.id

### Abstract

*Many tourists find it difficult to efficiently plan or find tourist destinations in Bali. They need a convenient way to search for attractions or plan their trips. A system is needed to facilitate the search and planning of their tourist trips, which can be supported by utilizing technology such as smartphones and mobile applications, making UI/UX crucial to help users solve their problems. This design is carried out using Figma tools and the design thinking method, which consists of five stages: empathize, define, ideate, prototype, and test. This research aims to analyze and implement the Design Thinking method in designing the user interface (UI/UX) of a tourist application focusing on tourist destinations in Bali.*

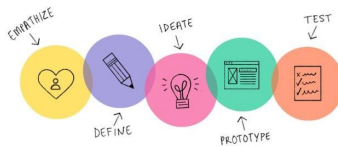
**Keywords:** Design Thinking, Pariwisata, Prototype, User Interface, User Experience

### 1. Pendahuluan

Bali merupakan pulau yang dikenal karena keindahan alamnya dan berbagai destinasi wisata menarik. Potensi pariwisata Bali sangat besar, terlihat dari banyaknya turis yang mengunjungi pulau ini. Sebagai salah satu tujuan wisata dunia, Bali terus berbenah untuk memanfaatkan peluang tersebut. Sebagai bagian dari kepulauan Indonesia, Bali memiliki sumber daya alam dan budaya yang khas, membuatnya terkenal di seluruh dunia. Bahkan, nama Bali lebih dikenal dibandingkan dengan Indonesia sendiri, dengan julukan "The Paradise Island" atau "The Island of God" [1]. Dengan kemajuan teknologi, Bali semakin dikenal luas baik oleh masyarakat dalam negeri maupun internasional. Berbagai aplikasi pariwisata yang dapat diunduh secara online tersedia untuk membantu para wisatawan merencanakan kegiatan mereka. Namun, beberapa aplikasi ini masih memiliki kekurangan fitur, sehingga penggunaannya untuk menyusun itinerary wisata di Bali belum banyak diminati oleh pelancong. Oleh karena itu, diperlukan inovasi atau kombinasi fitur dari beberapa aplikasi yang ada dengan merancang UI/UX yang lebih baik. Antarmuka yang menarik dan ramah pengguna dapat meningkatkan pengalaman pengguna dan mempengaruhi kesuksesan aplikasi. Untuk itu, penulis akan mengembangkan desain aplikasi itinerary wisata dengan menggunakan metode Design Thinking guna mengevaluasi efektivitas antarmuka yang dirancang. Pada penelitian sebelumnya yang dilakukan oleh Athiyya Mutiara Denasfi dengan judul "Perancangan User Interface dan User Experience Situs Web Aplikasi Traveling 'ANGLO' dengan Metode Design Thinking", digunakan pendekatan design thinking untuk merancang UI dan UX. Penelitian ini bertujuan memperkenalkan tempat wisata dan budaya Indonesia kepada wisatawan domestik maupun mancanegara untuk mendukung pemulihan ekonomi di sektor pariwisata. Hasil penelitian menunjukkan bahwa penerapan metode design thinking dalam perancangan UI/UX berhasil memenuhi kebutuhan calon pengguna dalam pembuatan situs web aplikasi traveling ANGLO [2].

## 2. Metode Penelitian

Dalam penelitian ini, digunakan metode design thinking, yaitu pendekatan kolaboratif yang mengumpulkan berbagai ide dari beragam disiplin ilmu untuk menemukan solusi. Design thinking adalah metode pemecahan masalah yang berfokus pada pengalaman pengguna dan bersifat iteratif. Metode ini bertujuan memecahkan masalah secara analitis melalui proses yang mendorong pengguna untuk bereksperimen dan menciptakan solusi, sehingga memungkinkan untuk mendapatkan umpan balik yang efektif dan efisien.[3]. Awalnya, terdapat tiga tahapan dalam design thinking: inspiration, yaitu identifikasi kebutuhan atau masalah yang memotivasi pencarian solusi atau inovasi; ideation, yaitu proses menghasilkan, mengembangkan, dan menguji gagasan; serta implementation, yaitu finalisasi penerapan solusi kepada pengguna. Seiring perkembangan, ketiga tahapan ini berkembang menjadi lima tahap, yang pada dasarnya serupa namun memiliki penekanan lebih spesifik pada bagian-bagian tertentu [4].



Gambar 1. Design Thinking

- Empathize adalah bagaimana dipahami emosi yang sama dengan yang dirasakan orang lain. Melalui empati dapat dirasakan perasaan mereka tentang masalah, situasi, dan keadaan.
- Define adalah suatu proses cara mendapatkan pandangan dari pengguna serta memahami kebutuhan mereka. Seperti membuat persona pengguna yang akan menjadi dasar untuk merancang dan membuat produk atau aplikasi.
- Ideate merupakan proses penggambaran solusi dari berbagai ide yang kemudian diwujudkan melalui brainstorming.
- Prototype adalah proses pembuatan desain tampilan website yang ingin dibangun dan kemudian menerapkan ide-ide tersebut untuk menghasilkan suatu prototipe atau produk yang siap diuji.
- Test adalah teknik yang digunakan untuk melakukan evaluasi, yang mencakup beberapa pertanyaan atau serangkaian tugas yang harus diselesaikan atau dijawab oleh peserta.

## 3. Hasil dan Diskusi

Hasil penelitian menggunakan metode design thinking dilakukan sesuai dengan tahapan proses design thinking yang bertujuan untuk memecahkan masalah sehingga mendapatkan solusi dan inovasi dari sudut pandang pengguna. Berikut merupakan pembahasannya.

### 3.1. Empathize

Tahap empathize adalah fase untuk memahami secara mendalam pengguna atau audiens dengan cara mengamati, berinteraksi, dan berempati terhadap pengalaman mereka. Pada tahap ini, penelitian dilakukan untuk mengetahui kebutuhan pengguna berdasarkan permasalahan yang mereka alami. Wawancara dilakukan dengan empat orang yang gemar traveling untuk mempermudah menemukan masalah yang dihadapi saat merencanakan wisata.

Terdapat beberapa pandangan pengguna mengenai aplikasi yang penulis buat:

- Pengguna lebih memilih sistem dalam bentuk aplikasi mobile.
- Pengguna kesulitan menemukan detail dari informasi tempat wisata.

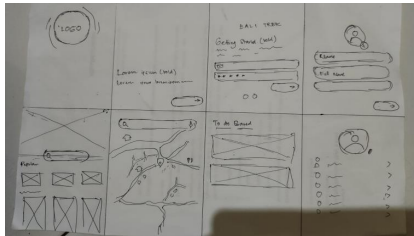
- c. Pengguna masih kesulitan menemukan tempat wisata yang jarang diketahui. orang dan tempat wisata yang terbilang baru.

### 3.2. Define

Tahap define adalah proses untuk menemukan inti permasalahan berdasarkan data yang telah dikumpulkan sebelumnya. Pada tahap ini, berbagai kebutuhan pengguna yang teridentifikasi pada tahap empathize akan dipersempit dan difokuskan [5]. Tahap define adalah proses untuk menemukan inti permasalahan berdasarkan data yang telah dikumpulkan sebelumnya. Permasalahan utama yang disimpulkan dari tahap empathize adalah minimnya penggunaan aplikasi wisata berbasis mobile, serta kurang lengkapnya informasi tentang tempat wisata pada aplikasi yang sudah ada, sehingga belum sepenuhnya memenuhi kebutuhan pengguna. Selain itu, banyak tempat wisata yang bagus namun jarang dikunjungi karena belum banyak diketahui orang. Oleh karena itu, solusi yang ditawarkan adalah merancang aplikasi serupa yang fiturnya menggabungkan keluhan dan kebutuhan pengguna.

### 3.3. Ideate

Pada tahap Ideate, dilakukan proses brainstorming untuk menemukan ide-ide yang dapat memenuhi kebutuhan pengguna serta memperhatikan batasan yang telah ditetapkan selama fase definisi. Proses ini difokuskan pada pengumpulan ide-ide guna menghasilkan solusi yang didasarkan pada data dan riset yang telah dikumpulkan sebelumnya. Hasil dari tahap ini adalah sebuah wireframe yang berfungsi sebagai kerangka dasar untuk mendukung ide-ide yang dihasilkan. Wireframe tersebut mencakup berbagai fitur, seperti halaman login yang memungkinkan pengguna yang telah memiliki akun untuk langsung mengisi email dan password. Bagi pengguna yang belum memiliki akun, mereka dapat membuat akun baru dengan mengklik tombol "Sign Up". Setelah proses sign up, pengguna akan diarahkan ke halaman formulir pendaftaran untuk mengisi data diri. Selanjutnya, terdapat menu utama yang menampilkan gambar-gambar destinasi wisata dan menu pencarian dengan peta. Kemudian, terdapat menu "To Do" yang memungkinkan pengguna untuk mencatat dan menyimpan rencana atau tempat wisata yang akan dikunjungi. Terakhir, terdapat menu profile yang memungkinkan pengguna untuk mengelola data pribadi mereka.



Gambar 2. Rancangan UI/UX

### 3.4. Prototype

#### a. Landing Page

Pengguna memasuki aplikasi lalu menunggu loading, selanjutnya pengguna memasuki landing page, pengguna dapat meng klik tombol get started untuk menuju ke halaman berikutnya.

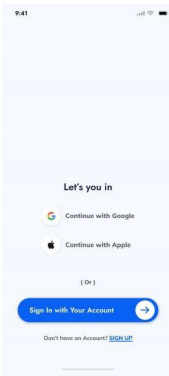




Gambar 3. Loading



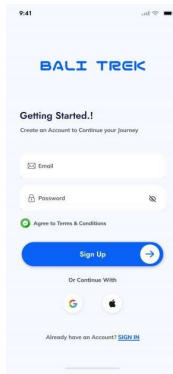
Gambar 4. Landing page



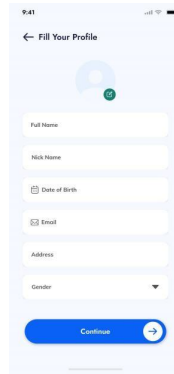
Gambar 5. Start

**b. Halaman Sign Up**

Halaman sign up digunakan untuk membuat akun pengguna, jika pengguna belum memiliki akun, pengguna dapat melakukan register dengan mengisi data diri mereka seperti Full Name, Nick Name, Date of Birth, Email, dan Password. Pada Sign Up ini juga bisa dilakukan dengan akun Google atau akun Apple.



Gambar 6. Sign Up

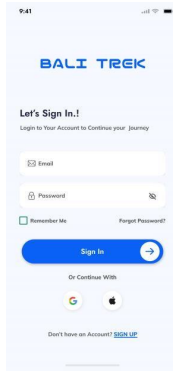


Gambar 7. Fill Data Sign Up

**c. Halaman Sign in**

Jika pengguna sudah memiliki akun atau sudah register akun, pengguna dapat melakukan log in dengan memasukkan email dan password. Di halaman sign in ini juga

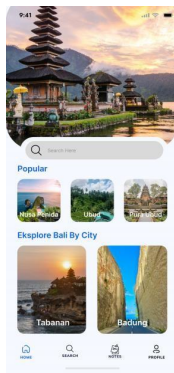
jika pengguna lupa password mereka, maka pengguna bisa mengklik forgot password yang akan diarahkan untuk mengisi email dan akan mendapatkan kode OTP.



Gambar 8. Sign In

d. Halaman Utama

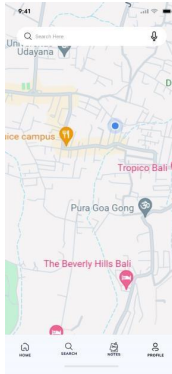
Pada menu utama merupakan menu yang menampilkan gambar gambar destinasi wisata, pada menu home ini terdapat kategori populer, explore bali by city, dan lain lain. Pada menu home ini juga pengguna bisa mengklik button search yang bisa mencari menurut pencarian pengguna.



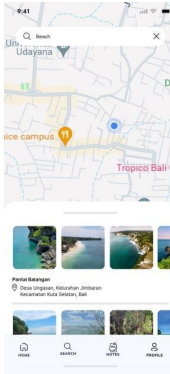
Gambar 9. Home

e. Fitur Pencarian

Pada menu pencarian pengguna dapat mencari destinasi wisata, ini sangat amat memudahkan pengguna. Pengguna dapat mencari dengan keyword yang tempat tempat yang akan dituju.



Gambar 10. Fitur Search



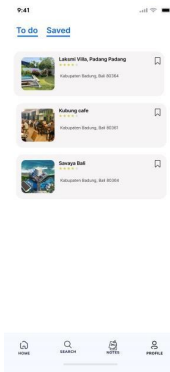
Gambar 11. Search

f. Halaman To Do (Note) dan Saved

Pada menu To Do pengguna dapat memasukkan note mereka sendiri, merencanakan perjalanan mereka, lalu untuk fitur saved, itu merupakan fitur simpanan destinasi yang sudah disimpan oleh pengguna.



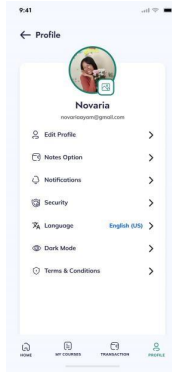
Gambar 12. Fitur To Do



Gambar 13. Fitur Saved

g. Halaman Profil

Pada halaman *profile* pengguna pengguna dapat memanipulasi atau mengubah data diri mereka



Gambar 14. Fitur Profile

### 3.5. Test

Tahap akhir dari proses design thinking ini terletak pada tahap pengujian, di mana hasil- hasil yang didapat mengarah pada pemahaman yang lebih baik tentang pengguna, cara penggunaan, serta bagaimana orang berinteraksi dengan produk secara keseluruhan. Dalam proses ini, penulis melakukan perubahan dan perbaikan untuk mengatasi masalah yang muncul dan mendalami pemahaman tentang produk dan penggunaannya. Berikut merupakan pertanyaan/ pernyataan yang diajukan:

No	Pertanyaan/pernyataan
1	Saya mungkin akan sering menggunakan aplikasi ini
2	Seharusnya aplikasi ini mudah digunakan
3	Saya mungkin nyaman menggunakan aplikasi ini
4	Saya tidak perlu belajar menggunakan aplikasi ini
5	Banyak fitur dan font yang terintegrasi dengan baik
6	Saya rasa aplikasinya tidak membosankan

### 4. Kesimpulan

Hasil dari penelitian ini menunjukkan bahwa pengguna lebih memilih aplikasi mobile untuk merencanakan perjalanan mereka, namun kesulitan dalam menemukan informasi detail tentang tempat wisata. Selain itu, banyak tempat wisata yang bagus namun belum banyak diketahui orang. Oleh karena itu, desain aplikasi Bali Trek mengintegrasikan fitur-fitur yang memperbaiki kekurangan tersebut, seperti fitur pencarian yang mudah digunakan, penyediaan informasi yang

lengkap tentang tempat wisata, serta fitur untuk menemukan tempat wisata yang kurang populer.. Setelah melakukan kuesioner mengenai desain aplikasi Bali Trek ini, penilaian rata rata untuk aplikasi ini adalah, 5.14, 6.57, 6.86, 6.43, 6.23, 6.45. Ini dapat berpotensi memberikan kontribusi yang signifikan dalam memajukan sektor pariwisata di Bali melalui pemanfaatan teknologi yang lebih efektif dan efisien.

No	Pertanyaan/pernyataan	Rata Rata Nilai
1	Saya mungkin akan sering menggunakan aplikasi ini	5.14/7
2	Seharusnya aplikasi ini mudah digunakan	6.57/7
3	Saya mungkin nyaman menggunakan aplikasi ini	6.86/7
4	Saya tidak perlu belajar menggunakan aplikasi in	6.43/7
5	Banyak fitur dan font yang terintegrasi dengan baik	6.23/7
6	Saya rasa aplikasinya tidak membosankan	6.45/7

#### Daftar Pustaka

- [1] I. W. Wiwin, "Community based tourism dalam pengembangan pariwisata Bali," *Pariwisata Budaya: Jurnal Ilmiah Agama dan Budaya*, vol. 3, no. 1, pp. 69–75, 2018.
- [2] A. M. Denasfi and E. G. Wahyuni, "Perancangan User Interface dan User Experience Situs Web Aplikasi Traveling 'ANGLO' dengan Metode Design Thinking," *AUTOMATA*, vol. 4, no. 1, 2023.
- [3] M. Azmi, A. P. Kharisma, and M. A. Akbar, "Evaluasi User Experience Aplikasi Mobile Pemesanan Makanan Online dengan Metode Design Thinking (Studi Kasus GrabFood)," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 3, no. 8, pp. 7963–7972, 2019.
- [4] A. A. Razi, I. R. Mutiaz, and P. Setiawan, "Penerapan metode design thinking pada model perancangan ui/ux aplikasi penanganan laporan kehilangan dan temuan barang tercecer," *Demandia: Jurnal Desain Komunikasi Visual, Manajemen Desain, dan Periklanan*, vol. 3, no. 02, pp. 219–237, 2018.
- [5] A. I. Pratiwi and S. Rani, "Implementasi Metode Design Thinking Dalam Perancangan UI/UX Aplikasi Itinerary Wisata," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 3, no. 6, pp. 249–258, 2023.

## Perbandingan Neural Network MLP, KNN, dan Decision Tree untuk Klasifikasi Penyakit Diabetes

I Made Prenawa Sida Nanda<sup>a1</sup>, I Putu Gede Hendra Suputra<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>nanda.2208561017@student.unud.ac.id  
<sup>2</sup>hendra.suputra@gmail.com

### Abstract

*Diabetes is one of the diseases that has received global attention due to its extensive impact on public health. Most people with diabetes are unaware that they are suffering from this condition, this situation emphasizes the need for improved understanding and more effective treatment of this disease. In an effort to address these challenges, this study compares three machine learning algorithms for diabetes classification, the three algorithms are: Multi-Layer Perceptron (MLP), K-Nearest Neighbor (KNN), and Decision Tree. Data from the Diabetes Dataset used to train and test these models will go through preprocessing first starting from data cleaning, encoding because there is string data, data distribution analysis where in this study using under sampling to equalize data and normalization using min-max normalization, Evaluation results using Confusion Matrix and Classification Report which contains precision, recall, and f1-score the results of this evaluation show that the Neural Network MLP model achieves the highest accuracy of 90.48%, followed by KNN with 88.15% accuracy, and Decision Tree with 87.24% accuracy. These findings provide important insights in selecting the optimal model for diabetes prediction applications.*

**Keywords:** Diabetes, Machine Learning, Neural Network MLP, KNN, Decision Tree

### 1. Pendahuluan

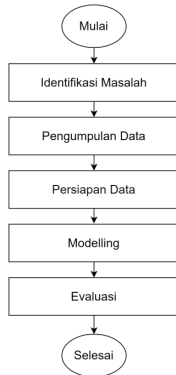
Fenomena diabetes semakin menguatkan pemahaman akan beratnya tanggung jawab global yang harus ditangani oleh individu, keluarga, dan negara. Menurut laporan IDF (Federasi Diabetes Internasional), sebanyak 10,5% dari populasi orang dewasa berusia 20-79 tahun mengidap diabetes, dengan hampir separuh dari mereka tidak menyadari kondisi kesehatan mereka yang sebenarnya [1]. Informasi yang tersedia menunjukkan bahwa angka kasus diabetes terus meningkat, pada tahun 2045 proyeksi IDF menunjukkan bahwa 1 dari 8 orang dewasa, sekitar 783 juta jiwa, akan hidup dengan diabetes, peningkatan sebesar 46% [1]. Diabetes adalah salah satu penyakit yang mendapat perhatian global karena dampaknya yang signifikan terhadap kesehatan masyarakat secara menyeluruh. Diabetes melibatkan disfungsi dalam pengaturan kadar gula darah, yang disebabkan oleh kurangnya produksi insulin yang memadai oleh pankreas atau ketidakmampuan tubuh manusia untuk menggunakan insulin dengan efektif [2]. Hal ini dapat mengakibatkan berbagai komplikasi serius yang memengaruhi kualitas hidup individu. Penyakit *metabolic* seperti diabetes adalah penyakit genetik yang biasanya disebabkan oleh penyakit keturunan, lalu penyebab lainnya bisa dari racun makanan dan infeksi. Penyakit mempengaruhi destruksi sel *beta*, *diabetogenic*, dan olahraga yang kurang. Biasanya penyakit ini menyebabkan komplikasi kronis yang terjadi pada mata, ginjal, saraf, dan pembuluh darah [3]. Klasifikasi merupakan metode yang sering digunakan untuk memprediksi kelas pada label tertentu. Ini dilakukan dengan membuat model berdasarkan kumpulan data (*training set*) dan nilai-nilai kelas (label) untuk mengklasifikasikan atribut tertentu. Ada lima kategori utama dalam klasifikasi yang didasarkan pada prinsip-prinsip matematika yang berbeda, termasuk statistik, jarak, pohon keputusan, jaringan saraf, dan aturan [4]. Beberapa algoritma yang umum digunakan dalam *machine learning* mencakup *neural network*, *decision tree*, *k-nearest neighbor*, *naive bayes*, *random forest*, dan masih banyak lagi [5]. Dalam penelitian ini, peneliti bertujuan untuk

membandingkan performa tiga algoritma yaitu *Multi-Layer Perceptron (MLP) Neural Network*, *KNN*, *Decision Tree Classifier*, untuk klasifikasi penyakit diabetes. Evaluasi dilakukan terhadap model-model ini untuk menentukan keefektifan relatifnya dalam klasifikasi penyakit diabetes. Harapannya, hasil penelitian ini dapat memberikan wawasan yang lebih mendalam terkait pemilihan algoritma *machine learning* yang optimal untuk klasifikasi penyakit diabetes, serta menjadi referensi untuk penelitian lanjutan di masa mendatang.

## 2. Metode Penelitian

### 2.1 Tahap Penelitian

Penelitian ini melibatkan lima tahap penting yang digunakan untuk memecahkan masalah yang dibahas. Tahap-tahap tersebut mencakup identifikasi masalah, pengumpulan data, persiapan data, pembentukan model, dan evaluasi model. Semua tahapan penelitian ini disajikan secara visual pada Gambar 1.



Gambar 1. Tahap Penelitian

#### a. Identifikasi Masalah

Berdasarkan konteks yang telah disajikan, masalah yang diidentifikasi dalam penelitian ini menggunakan beberapa algoritma untuk membandingkan kinerja mereka. Penelitian ini memanfaatkan tiga algoritma, yakni *Neural Network Multi-Layer Perceptron (MLP)*, *k-nearest neighbor*, dan *decision tree*, dengan tujuan memahami model klasifikasi mana yang memiliki akurasi tertinggi dalam menentukan kehadiran penyakit diabetes.

#### b. Pengumpulan Data

Data yang dipergunakan dalam studi ini merupakan data sekunder yang berasal dari Diabetes Dataset yang diunduh dari situs web *kaggle.com* [6]. Dataset ini terdiri dari file terformat ".csv" yang terdiri dari 100.000 entri data dengan 9 atribut sebagaimana tercantum dalam Gambar 2. Penjelasan mengenai setiap atribut dapat kita lihat pada Tabel 1.

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.00	0	1	never	25.19	6.00	140	0
1	Female	54.00	0	0	No info	27.32	6.00	80	0
2	Male	28.00	0	0	never	27.32	5.70	158	0
3	Female	36.00	0	0	current	23.45	5.00	155	0
4	Male	78.00	1	1	current	20.14	4.80	155	0
...	...	...	...	...	...	...	...	...	...
99995	Female	80.00	0	0	No info	27.32	6.20	90	0
99996	Female	2.00	0	0	No info	17.37	6.50	100	0
99997	Male	66.00	0	0	former	27.83	5.70	155	0
99998	Female	24.00	0	0	never	35.42	4.00	100	0
99999	Female	57.00	0	0	current	22.43	6.00	90	0

100000 rows x 9 columns

**Gambar 2.** Dataset Penelitian

**Tabel 1.** Atribut Dataset Penelitian

No	Atribut	Tipe	Deskripsi
1	<i>Gender</i>	Kategorikal	Jenis kelamin berperan dalam risiko diabetes. Wanita dengan diabetes gestasional memiliki risiko yang tinggi untuk mengembangkan diabetes tipe 2, sementara beberapa penelitian menunjukkan risiko sedikit lebih tinggi pada pria.
2	<i>Age</i>	Numerik	Usia subjek dalam tahun
3	<i>Hypertension</i>	Numerik	Hipertensi dan diabetes sering terkait. Keduanya memiliki faktor risiko yang serupa dan saling mempengaruhi. Hipertensi meningkatkan risiko diabetes tipe 2, dan sebaliknya. Ini dapat berdampak negatif pada kesehatan jantung.
4	<i>Heart Disease</i>	Numerik	Gangguan kesehatan jantung, mencakup penyakit arteri koroner dan gagal jantung, berhubungan dengan peningkatan risiko diabetes. Hubungan antara keduanya bersifat dua arah, karena memiliki faktor risiko yang serupa seperti obesitas, hipertensi, dan tingkat kolesterol yang tinggi.
5	<i>Smoking History</i>	Kategorikal	Merokok meningkatkan risiko diabetes tipe 2 karena dapat menyebabkan resistensi insulin dan gangguan metabolisme glukosa. Berhenti merokok dapat mengurangi risiko diabetes dan komplikasinya secara signifikan.
6	<i>BMI</i>	Numerik	Indeks massa tubuh, metode untuk membandingkan berat badan seseorang dengan tinggi badannya, ukuran umum untuk mengevaluasi kelebihan berat badan atau obesitas
7	<i>HbA1c Level</i>	Numerik	HbA1c adalah indikator rata-rata tingkat glukosa dalam darah selama periode 2-3 bulan terakhir, mencerminkan kontrol gula darah jangka panjang. Tingkat HbA1c yang tinggi menandakan kontrol gula darah yang buruk dan meningkatkan risiko diabetes serta komplikasinya.



No	Atribut	Tipe	Deskripsi
8	Blood Glucose Level	Numerik	Tingkat glukosa darah menggambarkan jumlah gula dalam darah pada waktu tertentu. Kadar yang tinggi, terutama setelah puasa atau makan karbohidrat, bisa menandakan gangguan regulasi glukosa dan risiko diabetes. Pemantauan teratur penting untuk diagnosis dan pengelolaan kondisi ini.
9	Diabetes	Numerik	Variabel target atau variabel yang ingin diprediksi. Bernilai 1 jika subjek didiagnosis dengan diabetes, dan 0 jika tidak.

**c. Persiapan Data**

Langkah-langkah persiapan data akan dijalankan melalui empat tahap, dimulai dengan pembersihan data untuk menghilangkan anomali atau ketidaksesuaian, dilanjutkan dengan encoding dan penyesuaian data untuk mengubah string dan mengatasi ketimpangan, normalisasi untuk memastikan konsistensi dalam rentang nilai atribut, dan terakhir, pembagian data menjadi dua bagian: data pelatihan dimanfaatkan untuk melatih model, sedangkan data pengujian digunakan untuk mengevaluasi performa model yang telah dilatih.

**d. Data Cleaning**

Proses pembersihan data dilakukan dengan melakukan pemeriksaan menyeluruh terhadap dataset untuk memastikan bahwa tidak ada atribut yang memiliki nilai null (missing value) dan juga untuk mengidentifikasi serta menghapus data duplikat.

**e. Encoding dan Penyesuaian Data**

Proses *Encoding* dan Penyesuaian Data melibatkan konversi data kategori dalam format *string* menjadi format numerik melalui teknik seperti *one-hot encoding* atau label *encoding*. Selain itu, dilakukan juga penyesuaian data untuk menangani ketimpangan atau ketidakseimbangan dalam jumlah data, seperti menggunakan teknik *undersampling* atau *oversampling*.

**f. Normalisasi**

Metode normalisasi *Min-Max* mengimplementasikan transformasi linier pada data asli. Tujuan dari proses normalisasi adalah untuk menetapkan nilai dari setiap variabel ke dalam rentang yang seragam, yaitu [0, 1]. Hal ini bertujuan agar setiap variabel memiliki pengaruh yang setara dalam perhitungan nilai kesamaan (similarity) [7]. Normalisasi ini dalam persamaan matematika seperti persamaan berikut:

$$x' = \frac{x - \text{nilai}_{\min}}{\text{nilai}_{\max} - \text{nilai}_{\min}} \tag{1}$$

x mewakili data dalam setiap kolom, sedangkan nilai<sub>min</sub> adalah nilai terendah dari data dalam setiap kolom dan nilai<sub>max</sub> adalah nilai tertinggi dari data dalam setiap kolom.

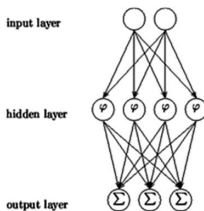
**g. Pembagian Data**

Dataset yang diolah dalam machine learning akan terlebih dahulu dibagi menjadi dua bagian, yakni data latih dan data uji. Pembagian dilakukan dengan skema 80% untuk data latih dan 20% untuk data uji. Data latih digunakan untuk melatih model klasifikasi, sementara data uji digunakan untuk pengujian.

**2.2 Neural Network Multi-Layer Perceptron (MLP)**

*Neural Network Multi-Layer Perceptron* adalah jenis Jaringan Saraf Tiruan yang bersifat *feed forward*. Ini menggunakan teknik *Backpropagation* untuk pembelajaran. Ini dilengkapi dengan lapisan neuron masukan sebagai penerima, satu atau lebih lapisan tersembunyi yang mengolah

data dan mengalami iterasi, serta lapisan keluaran untuk memprediksi hasil [8]. *Neural Network Multi-Layer Perceptron* terdiri dari tiga lapisan. Lapisan pertama terdiri dari  $n$  neuron yang merepresentasikan  $n$  variabel dari dataset. Kemudian, ada lapisan tersembunyi yang terdiri dari beberapa node. Node-node tersembunyi tidak menerima masukan langsung dari lapisan masukan, melainkan menerima sinyal yang diteruskan dari setiap node di lapisan masukan. Lapisan *output* memiliki  $m$  node yang didedikasikan untuk jumlah kelas dalam dataset. Arsitektur yang lebih kompleks ini melibatkan struktur *non-linear* dengan menggunakan fungsi aktivasi *sigmoidal* atau fungsi aktivasi buatan [9]. Gambar 3 menampilkan struktur jaringan saraf *multilayer perceptron (MLP)* yang digunakan dalam penelitian ini [8].



Gambar 3. Neural Network Multi-Layer Perceptron [8]

### 2.3 K-Nearest Neighbor

*K-Nearest Neighbor (K-NN)* merupakan metode populer dalam pengklasifikasian data. Prinsip dasarnya adalah mencari tetangga terdekat dari suatu data yang akan diklasifikasikan di dalam kumpulan data pelatihan. Algoritma ini dianggap sebagai salah satu yang paling sederhana, tetapi sering memberikan hasil yang kuat dan berarti dalam tugas klasifikasi. [10]. Jarak antara data yang akan dievaluasi dengan tetangga terdekatnya dihitung menggunakan metode jarak *Euclidean*, yang didefinisikan dalam persamaan (2).

$$d_i = \sqrt{\sum_{l=1}^p (X_{2l} - X_{1l})^2} \quad (2)$$

Keterangan:

- $X_1$  = data pelatihan
- $X_2$  = data pengujian
- $i$  = variabel data
- $d$  = jarak
- $p$  = dimensi data

### 2.4 Decision Tree

Algoritma *Decision Tree*, atau yang sering disebut sebagai 'pohon keputusan', digunakan secara luas untuk proses pengambilan keputusan. Algoritma ini berlaku baik untuk klasifikasi maupun prediksi, pohon keputusan melibatkan pembentukan struktur hirarkis yang terdiri dari simpul-simpul keputusan yang terhubung melalui cabang-cabang, mulai dari simpul akar hingga simpul daun. Model *decision tree* memiliki tiga bagian utama: *root node*, *internal node*, dan *terminal node*. Proses klasifikasi pada metode *decision tree* melibatkan pencarian dari *root node* ke *internal node* hingga mencapai terminal node berdasarkan data *query*. Konsep entropi digunakan untuk menentukan bagaimana atribut pada pohon keputusan akan dibagi, semakin tinggi entropi sampel, semakin tidak murni sampel tersebut [11]. Rumus untuk menghitung entropi sampel dapat dilihat pada persamaan (3)

$$Entropy(S) = - P_1 \log_2 P_1 - P_2 \log_2 P_2 \tag{3}$$

p1, p2, p3... pn mewakili proporsi dari setiap kelas 1, kelas 2, ..., hingga kelas n dalam output.

### 2.5 Evaluasi Model

Dalam kerangka penelitian ini, evaluasi dilakukan dengan menggunakan matriks kebingungan (*confusion matrix*). Melalui matriks kebingungan ini, nilai-nilai penting seperti akurasi, *precision*, *recall*, dan *f1-score* dihitung. Ini memungkinkan untuk mendapatkan pemahaman yang lebih komprehensif tentang kinerja model atau metode yang dievaluasi dalam konteks pengklasifikasian atau prediksi. Proses evaluasi *precision*, *recall*, *f1-score*, dan *accuracy* adalah langkah penting dalam mengukur kinerja suatu sistem. Langkah-langkah ini biasanya dilakukan dengan merujuk pada *confusion matrix*, yang didefinisikan dalam persamaan 4 hingga 6. Di sini, TP (*True Positive*) adalah jumlah prediksi positif yang tepat, sedangkan TN (*True Negative*) adalah jumlah prediksi yang tepat untuk kelas negatif. FP (*False Positive*) terjadi ketika yang diprediksi positif sebenarnya negatif [12].

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{6}$$

## 3. Hasil dan Diskusi

### 3.1 Preprocessing Data

Dataset yang digunakan dipersiapkan dengan memeriksa keberadaan nilai yang hilang. Meskipun tidak ditemukan nilai yang hilang dalam dataset, namun ditemukan keberadaan data duplikat. Karena hal tersebut, data duplikat tersebut akan dihapus. Dapat dilihat pada gambar 4 adalah hasil data setelah proses *Data Cleaning*.

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.00	0	1	never	25.19	6.80	140	0
1	Female	54.00	0	0	No Info	27.32	6.60	80	0
2	Male	28.00	0	0	never	27.32	5.70	158	0
3	Female	36.00	0	0	current	23.45	5.00	155	0
4	Male	76.00	1	1	current	20.14	4.80	155	0
...	...	...	...	...	...	...	...	...	...
99994	Female	36.00	0	0	No Info	24.60	4.80	145	0
99996	Female	2.00	0	0	No Info	17.37	6.50	100	0
99997	Male	66.00	0	0	former	27.83	5.70	155	0
99998	Female	24.00	0	0	never	35.42	4.00	100	0
99999	Female	57.00	0	0	current	22.43	6.60	90	0

96146 rows x 9 columns

Gambar 4. Dataset setelah *Data Cleaning*

#### a. Encoding dan Penyesuaian Data

Pada dataset yang digunakan, terdapat atribut yang masih memiliki nilai *string*, khususnya pada atribut *gender* dan *smoking\_history*. Oleh karena itu, perlu dilakukan proses *encoding* dengan menggunakan metode *one hot encoding*. Hal ini dilakukan untuk mengubah nilai-nilai *string* sehingga data tersebut menjadi representasi numerik yang dapat diolah oleh model. Poin ini terlihat pada Gambar 5 hasil dari dataset setelah melalui proses *encoding*.

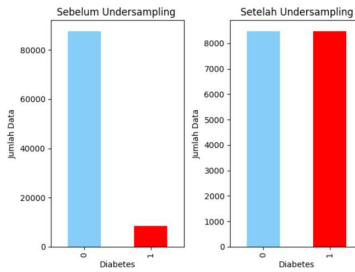
age	hypertension	heart_disease	sex	mbaz_level	blood_glucose_level	diabetes	gender_female	gender_male	smoking_history_current	smoking_history_non-smoker	smoking_history_past-smoker
0	0	0	1	25.19	0.00	0	True	False	False	True	False
1	0	0	0	27.32	0.00	0	True	False	False	True	False
2	0	0	0	27.32	0.75	100	0	False	True	False	True
3	0	0	0	25.45	5.00	100	0	True	False	True	False
4	0	1	1	25.14	4.00	100	0	False	True	True	False
...	...	...	...	...	...	...	...	...	...	...	...
10000	0	0	0	24.60	4.00	100	0	True	False	True	False
10000	0	0	0	17.37	0.00	100	0	True	False	True	False
10000	0	0	0	27.83	0.75	100	0	False	True	False	True
10000	0	0	0	35.42	4.00	100	0	True	False	True	False
10000	0	0	0	22.43	0.00	90	0	True	False	True	False

10120 rows x 12 columns

Gambar 5. Dataset setelah Encoding

**b. Analisis Distribusi Data**

Setelah melakukan pemeriksaan distribusi data pada dataset, ternyata ditemukan ketidakseimbangan yang tinggi. Oleh karena itu, penerapan *undersampling* digunakan untuk menyesuaikan data tersebut. Setelah proses *undersampling*, jumlah total data menjadi: 8482 untuk kelas 0, dan 8482 untuk kelas 1, seperti yang terlihat pada Gambar 6 yang menunjukkan hasil distribusi data setelah penerapan *undersampling*.



Gambar 6. Dataset sebelum dan setelah undersampling

**c. Normalisasi**

Tahap normalisasi data akan melibatkan metode normalisasi menggunakan teknik *min-max normalization*. Hal ini dilakukan karena metode ini dapat membantu menjaga konsistensi skala antara fitur-fitur yang ada dalam dataset. Setelah berhasil dinormalisasi, data akan dibagi menggunakan rasio 80:20 untuk data latih dan data uji. Tujuan dari pembagian ini adalah untuk memastikan bahwa model dilatih dengan data yang cukup untuk menghasilkan pembelajaran yang baik dan diuji dengan data yang cukup untuk menguji keandalan dan kinerja model secara independen. Gambar 7 menunjukkan hasil dataset setelah di normalisasi

age	hypertension	heart_disease	sex	mbaz_level	blood_glucose_level	gender_female	gender_male	smoking_history_current	smoking_history_non-smoker	smoking_history_past-smoker	diabetes
0	0.00	0.00	0.14	0.42	0.30	0.00	1.00	0.00	0.00	0.00	0.00
1	0.29	0.00	0.16	0.55	0.23	1.00	0.00	0.00	1.00	0.00	0.00
2	0.15	0.00	0.00	0.20	0.00	0.00	0.00	1.00	0.00	1.00	0.00
3	0.50	0.00	0.00	0.14	0.00	0.00	1.00	0.00	0.00	1.00	0.00
4	1.00	0.00	1.00	0.22	0.42	0.00	0.00	1.00	0.00	1.00	0.00
...	...	...	...	...	...	...	...	...	...	...	...
10000	0.81	1.00	0.30	0.05	0.27	1.00	0.00	0.00	0.00	1.00	0.00
10000	0.69	0.00	1.00	0.28	0.49	1.00	0.00	1.00	0.00	0.00	1.00
10000	0.76	0.00	0.00	0.31	0.55	0.91	1.00	0.00	0.00	1.00	0.00
10000	0.72	1.00	0.00	0.36	0.64	0.55	1.00	0.00	0.00	1.00	0.00
10000	0.76	0.00	0.25	0.49	0.73	1.00	0.00	1.00	0.00	0.00	1.00

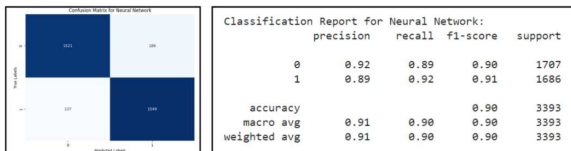
10100 rows x 12 columns

Gambar 7. Dataset setelah Normalisasi

### 3.2 Pengujian Model

#### a. Neural Network Multi-Layer Perceptron (MLP)

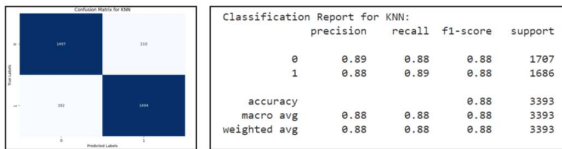
Dari *Confusion Matrix*, dapat dilihat bahwa model *Neural Network Multi-Layer Perceptron* menghasilkan 1521 prediksi yang benar bahwa pasien tidak memiliki diabetes (*True Negatives*), dan 1549 prediksi yang benar bahwa pasien memiliki diabetes (*True Positives*). Namun, terdapat 186 kasus di mana model salah mengklasifikasikan pasien sebagai memiliki diabetes ketika sebenarnya tidak (*False Positives*), dan 137 kasus di mana model salah mengklasifikasikan pasien sebagai tidak memiliki diabetes padahal sebenarnya memiliki (*False Negatives*). Melalui *Classification Report*, dapat disimpulkan bahwa model memiliki *precision* sebesar 0.92 untuk kelas tidak diabetes dan 0.89 untuk kelas diabetes, yang mengindikasikan bahwa sekitar 92% prediksi tidak diabetes dan 89% prediksi diabetes adalah benar. Selain itu, *recall* untuk kelas tidak diabetes adalah 0.89, yang berarti model berhasil menemukan sekitar 89% dari semua pasien yang sebenarnya tidak memiliki diabetes, sementara *recall* untuk kelas diabetes adalah 0.92, menunjukkan bahwa model berhasil menemukan sekitar 92% dari semua pasien yang sebenarnya memiliki diabetes. *F1-score*, yang merupakan rata-rata harmonik dari *precision* dan *recall*, adalah 0.90 untuk kelas tidak diabetes dan 0.91 untuk kelas diabetes. Dengan akurasi keseluruhan sebesar 0.90, model *Multi-Layer Perceptron* menunjukkan kinerja yang baik dalam memprediksi penyakit diabetes berdasarkan data yang diberikan. Gambar 8 menunjukkan hasil *Confusion Matrix* dan *Classification Report* dari *Neural Network Multi-Layer Perceptron*.



Gambar 8. Hasil Uji Coba Model *Neural Network Multi-Layer Perceptron*

#### b. K-Nearest Neighbor (K-NN)

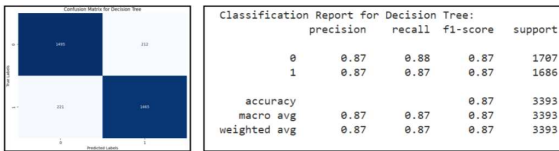
Dari *Confusion Matrix*, dapat dilihat bahwa model *KNN* menghasilkan 1497 prediksi yang benar bahwa pasien tidak memiliki diabetes (*True Negatives*), dan 1494 prediksi yang benar bahwa pasien memiliki diabetes (*True Positives*). Namun, terdapat 210 kasus di mana model salah mengklasifikasikan pasien sebagai memiliki diabetes ketika sebenarnya tidak (*False Positives*), dan 192 kasus di mana model salah mengklasifikasikan pasien sebagai tidak memiliki diabetes padahal sebenarnya memiliki (*False Negatives*). Dari *Classification Report*, dapat dilihat bahwa model memiliki *precision* sebesar 0.89 untuk kelas tidak diabetes dan 0.88 untuk kelas diabetes, yang mengindikasikan bahwa sekitar 89% prediksi tidak diabetes dan 88% prediksi diabetes adalah benar. Selain itu, *recall* untuk kelas tidak diabetes adalah 0.88, yang berarti model berhasil menemukan sekitar 88% dari semua pasien yang sebenarnya tidak memiliki diabetes, sementara *recall* untuk kelas diabetes adalah 0.89, menunjukkan bahwa model berhasil menemukan sekitar 89% dari semua pasien yang sebenarnya memiliki diabetes. *F1-score*, yang merupakan rata-rata harmonik dari *precision* dan *recall*, adalah 0.88 untuk kelas tidak diabetes dan 0.88 untuk kelas diabetes. Dengan akurasi keseluruhan sebesar 0.88, model *KNN* menunjukkan kinerja yang baik dalam memprediksi penyakit diabetes berdasarkan data yang diberikan. Gambar 9 menunjukkan hasil *Confusion Matrix* dan *Classification Report* dari *K-Nearest Neighbor (K-NN)*.



Gambar 9. Hasil Uji Coba Model *K-Nearest Neighbor (K-NN)*

**c. Decision Tree**

Dari *Confusion Matrix*, kita dapat melihat bahwa model *Decision Tree* menghasilkan 1495 prediksi yang benar bahwa pasien tidak memiliki diabetes (*True Negatives*), dan 1465 prediksi yang benar bahwa pasien memiliki diabetes (*True Positives*). Namun, terdapat 212 kasus di mana model salah mengklasifikasikan pasien sebagai memiliki diabetes ketika sebenarnya tidak (*False Positives*), dan 221 kasus di mana model salah mengklasifikasikan pasien sebagai tidak memiliki diabetes padahal sebenarnya memiliki (*False Negatives*). Dari *Classification Report*, kita dapat melihat bahwa model memiliki *precision* sebesar 0.87 untuk kelas tidak diabetes dan 0.87 untuk kelas diabetes, yang mengindikasikan bahwa sekitar 87% prediksi tidak diabetes dan 87% prediksi diabetes adalah benar. Selain itu, *recall* untuk kelas tidak diabetes adalah 0.88, yang berarti model berhasil menemukan sekitar 88% dari semua pasien yang sebenarnya tidak memiliki diabetes, sementara *recall* untuk kelas diabetes adalah 0.87, menunjukkan bahwa model berhasil menemukan sekitar 87% dari semua pasien yang sebenarnya memiliki diabetes. *F1-score*, yang merupakan rata-rata harmonik dari *precision* dan *recall*, adalah 0.87 untuk kelas tidak diabetes dan 0.87 untuk kelas diabetes. Dengan akurasi keseluruhan sebesar 0.87, model *Decision Tree* menunjukkan kinerja yang baik dalam memprediksi penyakit diabetes berdasarkan data yang diberikan.



Gambar 10. Hasil Uji Coba Model *Decision Tree*

**3.3 Akurasi Hasil Perbandingan**

Tabel 2. Akurasi Hasil Perbandingan

Model	Akurasi
<i>Neural Network MLP</i>	90.48 %
<i>Decision Tree</i>	87.24 %
<i>KNN</i>	88.15 %

Berdasarkan hasil perbandingan pada Tabel 2, dapat disimpulkan bahwa model *Neural Network MLP* menunjukkan kinerja terbaik dengan akurasi tertinggi sebesar 90.48%, menunjukkan

kemampuannya dalam memprediksi penyakit diabetes secara akurat. Model *K-Nearest Neighbors (KNN)* menempati posisi kedua dengan akurasi 88.15%, menunjukkan konsistensinya dalam memberikan prediksi yang baik. Meskipun begitu, model *Decision Tree* dengan akurasi 87.24% juga menunjukkan kemampuan yang layak dalam memprediksi penyakit diabetes. Evaluasi ini memberikan wawasan penting dalam memilih model yang paling sesuai untuk aplikasi prediksi penyakit diabetes, dengan mempertimbangkan keseimbangan antara akurasi dan kebutuhan komputasi atau interpretasi model.

#### 4. Kesimpulan

Berdasarkan hasil dari penelitian perbandingan tiga algoritma untuk klasifikasi penyakit diabetes, model *Neural Network MLP* menunjukkan performa terbaik dengan akurasi 90.48%. Diikuti oleh model *K-Nearest Neighbor (KNN)* dengan akurasi 88.15% dan *Decision Tree* dengan akurasi 87.24%. Penelitian ini memberikan pandangan penting dalam memilih model yang optimal untuk aplikasi prediksi penyakit diabetes, serta berpotensi menjadi dasar untuk penelitian dan aplikasi praktis di bidang kesehatan masyarakat.

#### Daftar Pustaka

- [1] International Diabetes Federation, "Diabetes facts and figures," *IDF Diabetes Atlas 10th edition 2021*, 2021.
- [2] M. Warke, V. Kumar, S. Tarale, P. Galgat, dan D. J. Chaudhari, "Diabetes diagnosis using machine learning algorithms," *Diabetes*, vol. 6, no. 03, hal. 1470-1476, 2019.
- [3] S. A. Agnal dan E. Saraswathi, "Analyzing Diabetic Data Using Naïve Bayes Classifier," Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, vol. 7, no. 4, hal. 2687-2698, 2020.
- [4] I. M. K. Karo dan H. Hendriyana, "Klasifikasi Penderita Diabetes menggunakan Algoritma Machine Learning dan Z-Score," *Jurnal Teknologi Terpadu*, vol. 8, no. 2, hal. 94-99, 2022.
- [5] N. R. Muntari dan K. H. Hanif, "Klasifikasi Penyakit Kanker Payudara Menggunakan Perbandingan Algoritma Machine Learning," *Jurnal Ilmu Komputer dan Teknologi*, vol. 3, no. 1, hal. 1-6, 2022.
- [6] PANNMIE, "Diabetes: EDA Random Forest," [www.kaggle.com](https://www.kaggle.com/code/tumpanjawat/diabetes-eda-random-forest-hp#2-%7C-Correlation), 2023. [Online]. Available: <https://www.kaggle.com/code/tumpanjawat/diabetes-eda-random-forest-hp#2-%7C-Correlation>. [Accessed 05 Mei 2024].
- [7] Nishom, M., "Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square," *Jurnal Informatika: Jurnal Pengembangan IT\**, vol. 4, no. 1, pp. 20-24, 2019.
- [8] Desai, M., & Shah, M. "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN)." *Clinical eHealth*, vol. 4, pp. 1-11, 2021.
- [9] Sari, N. R., & Mar'atullatifah, Y. "Penerapan Multilayer Perceptron Untuk Identifikasi Kanker Payudara." *Jurnal Cakrawala Ilmiah*, vol. 2, no. 8, pp. 3261-3268, 2023.
- [10] N. A. Nikmatun and I. Waspada, "Implementasi data mining untuk klasifikasi masa studi mahasiswa menggunakan algoritma K-Nearest Neighbor," *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 10, no. 2, pp. 421-432, 2019.
- [11] A. Tangkelayuk, "The Klasifikasi Kualitas Air Menggunakan Metode KNN, Naive Bayes, dan Decision Tree," *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, vol. 9, no. 2, pp. 1109-1119, 2022.
- [12] A. Y. Rahman, "Klasifikasi Citra Burung Lovebird Menggunakan Decision Tree dengan Empat Jenis Evaluasi," *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 688-696, 2021.

# Perlindungan Seed Phrase dengan Enkripsi Dual-Layer Menggunakan Algoritma AES dan Caesar Cipher

Raihan Akbar Maulana<sup>a1</sup>, I Wayan Santiyasa<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>raihanakbarmaulana29@gmail.com  
<sup>2</sup>Santiyasa@unud.ac.id

## Abstract

*In the digital era, securing sensitive information such as seed phrases are crucial to prevent unauthorized access and potential loss of digital assets. This paper proposes a novel approach for protecting seed phrases using dual-layer encryption with AES algorithm and Caesar cipher. The AES algorithm is utilized to encrypt the seed phrase itself, providing a strong first layer of defense. Additionally, Caesar cipher is employed to encrypt the AES secret key, adding an extra layer of security to the encryption process. By combining these two encryption techniques, the security of the seed phrase is significantly enhanced, as both the phrase and its encryption key are protected. Furthermore, the encrypted seed phrase and key can be securely stored using email as a digital storage medium, enhancing accessibility while maintaining robust security measures.*

**Keywords:** Cryptocurrency, Seed Phrase, Encryption, Advanced Encryption Standard, Caesar Cipher

## 1. Pendahuluan

Dalam ekosistem *cryptocurrency* yang terus berkembang, keamanan informasi memegang peran yang semakin krusial. Di antara berbagai jenis informasi sensitif, *seed phrase* atau frasa benih adalah elemen yang sangat penting. *Seed phrase* digunakan sebagai kunci utama untuk mengakses dan mengontrol aset digital pengguna dalam dompet kripto mereka. Saat seorang pengguna mulai menggunakan dompet kripto untuk pertama kalinya, dompet tersebut akan menghasilkan sebuah *seed phrase* panjang yang terdiri dari 12, 15, 18, 21, atau 24 kata dari daftar 2048 kata yang ditentukan oleh standar BIP39 [1]. Biasanya pengguna diminta untuk menuliskannya di sebuah kertas dan menyimpannya di tempat yang aman, *seed phrase* harus dijaga dengan sangat rahasia karena siapapun yang mengetahuinya dapat mencuri semua aset pengguna [2]. Jika komputer, laptop, atau perangkat seluler pengguna rusak, korup, dicuri, atau hancur, pengguna dapat memulihkan semua koin kripto mereka dengan menginstal ulang dompet kripto yang sama pada sistem baru dan menggunakan *seed phrase* yang sama [2]. Serta, untuk menghindari risiko seperti kehilangan atau dicurinya *seed phrase*, karena hal ini dapat berakibat pada kehilangan akses ke dana kripto yang dimiliki pengguna, penting untuk menerapkan lapisan keamanan yang kuat seperti perlindungan *seed phrase* dan menyimpannya secara digital. Salah satu pendekatan yang umum digunakan adalah pengenkripsian. Secara sederhana, enkripsi mengubah teks biasa menjadi *ciphertext* dan mendekripsi *ciphertext* menjadi teks biasa [3]. Pada penelitian ini, kami menggunakan *Advanced Encryption Standard* (AES), sebuah algoritma kriptografi simetris yang telah menjadi standar industri untuk enkripsi data. Meskipun AES memberikan tingkat keamanan yang tinggi, penggunaannya sendiri tidak selalu cukup untuk melindungi *seed phrase* secara menyeluruh. Oleh karena itu, dalam penelitian ini, kami mengusulkan sebuah pendekatan baru yang menggabungkan lebih dari satu algoritma enkripsi untuk meningkatkan keamanan *seed phrase*. Dalam pendekatan ini, kami memanfaatkan kombinasi dari AES dan *Caesar Cipher*. AES digunakan untuk mengenkripsi *seed phrase* itu sendiri, sementara *caesar cipher* digunakan untuk mengamankan kunci rahasia AES, dan penggunaan email sebagai media penyimpanan digital. Penggunaan email sebagai media



penyimpanan digital dipilih karena keunggulannya dalam keamanan dan kemudahan aksesibilitas. Dengan memanfaatkan protokol SMTP, data yang dienkripsi dapat dikirim dengan aman dan andal ke alamat email pengguna. Outlook dan Gmail dipilih karena fasilitas pemulihan bencana yang kuat dan infrastruktur yang andal, memastikan bahwa *seed phrase* pengguna tetap tersimpan dengan aman dan dapat diakses kembali saat diperlukan. Dengan menggabungkan kedua teknik enkripsi ini, kami bertujuan untuk menciptakan perlindungan yang lebih kuat terhadap *seed phrase*, mengurangi risiko kehilangan atau pencurian aset kripto pengguna. Kami percaya bahwa pendekatan ini akan menjadi langkah penting dalam memperkuat keamanan informasi di ranah *cryptocurrency*, membantu pengguna untuk merasa lebih aman dalam menyimpan dan mengelola aset digital mereka. Dalam tulisan ini, kami akan merinci metode yang diusulkan, menguji keberhasilan proses enkripsi dan dekripsi, serta menguji penyimpanan menggunakan *email*.

## 2. Metode Penelitian

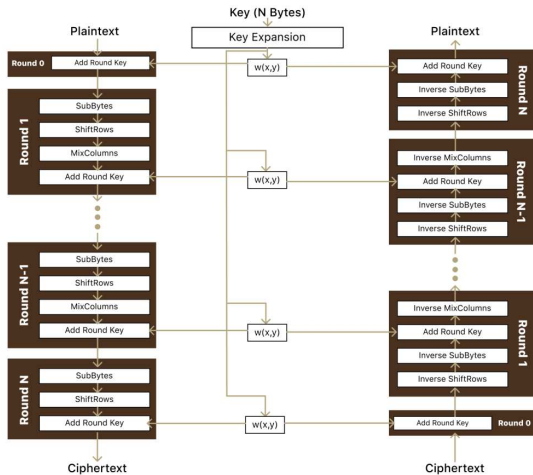
### 2.1. Gambaran Umum Sistem

Sistem ini mengimplementasikan dua tahapan enkripsi untuk meningkatkan keamanan *seed phrase*, yaitu dengan menggunakan *Advanced Encryption Standard* (AES) untuk mengenkripsi *seed phrase* dan *caesar cipher* untuk mengamankan *secret key* yang dihasilkan dari proses AES.

- a. Pertama, *plaintext* dari *seed phrase* akan dienkripsi menggunakan AES. Pada tahap ini, *plaintext* akan diproses menjadi *ciphertext* menggunakan algoritma AES dengan kunci yang disesuaikan pengguna. Proses enkripsi ini menghasilkan *ciphertext* yang tidak dapat dibaca secara langsung oleh pihak yang tidak berwenang.
- b. Selanjutnya, kunci rahasia yang dihasilkan dari proses enkripsi AES akan dienkripsi menggunakan *Caesar Cipher*. Proses ini meningkatkan keamanan *secret key* dengan melakukan enkripsi menggunakan *caesar cipher*, sehingga menghasilkan nilai yang sulit untuk diretas.
- c. Pada tahapan dekripsi, dilakukan dengan mengembalikan langkah-langkah enkripsi yang dilakukan secara terbalik: Pertama, *secret key* yang telah dienkripsi dengan *caesar cipher* akan didekripsi untuk mendapatkan *secret key* asli. Selanjutnya, *secret key* tersebut akan digunakan untuk mendekripsi *ciphertext* AES menjadi *plaintext seed phrase* yang asli.
- d. Dengan menggunakan kedua tahapan enkripsi ini, sistem bertujuan untuk memberikan perlindungan yang lebih kuat terhadap *seed phrase*, mengurangi risiko kehilangan, dicurinya aset kripto pengguna, dan memberikan keamanan yang lebih tinggi terhadap informasi sensitif dalam *cryptocurrency*.

### 2.2. Algoritma *Advanced Encryption Standard* (AES)

*Advanced Encryption Standard* (AES) adalah salah satu algoritma enkripsi yang paling umum digunakan secara global. AES menggunakan blok *cipher* yang beroperasi pada blok-blok data tetap berukuran 128 bit, dengan kunci yang dapat memiliki panjang 128, 192, atau 256 bit [4]. Pada bulan September 1997, AES (*Advanced Encryption Standard*) dipublikasikan[4]. AES telah diadopsi oleh banyak organisasi pemerintah dan industri sebagai standar enkripsi, digunakan untuk melindungi data sensitif, seperti data pribadi, transaksi keuangan, dan pesan rahasia. AES juga berjalan cukup cepat dibandingkan dengan ECC dan RSA [5]. AES menggunakan beberapa putaran transformasi linier, penggantian *byte*, pergeseran baris, dan campuran kolom untuk mengubah teks biasa menjadi teks teracak (*ciphertext*) dan sebaliknya. Hal ini membuatnya sangat kuat terhadap serangan kriptanalisis yang canggih dan cocok untuk digunakan dalam berbagai aplikasi keamanan, mulai dari komunikasi *online* hingga penyimpanan data.



Gambar 1. AES Transformation

a. Cara Kerja AES

- Inisialisasi Kunci: Tahap awal melibatkan inisialisasi kunci enkripsi, yang terdiri dari pengaturan kunci yang digunakan untuk mengacak data pada *Round 0*.
- Tahap *SubBytes*: Pada tahap ini, setiap byte dalam blok data diubah menjadi nilai baru menggunakan sebuah tabel substitusi yang disebut S-Box. Ini adalah langkah non-linear yang memperkenalkan kebingungan ke dalam proses enkripsi.
- Tahap *ShiftRows*: Di sini, setiap baris dalam blok data bergeser ke kiri sejumlah langkah tertentu. Ini menciptakan efek dispersi horizontal dalam blok data.
- Tahap *MixColumns*: Pada tahap ini, setiap kolom dalam blok data diubah menggunakan operasi matriks yang kompleks. Ini mencampur nilai-nilai dalam setiap kolom untuk memberikan dispersi vertikal dalam blok data.
- Tahap *AddRoundKey*: Ini adalah tahap kunci *round*, di mana setiap *byte* dalam blok data di-XOR-kan dengan bagian kunci rahasia yang disesuaikan oleh *Key Expansion*.

Tahapan-tahapan ini diulang secara berurutan sejumlah *round* tertentu yang tergantung pada panjang kunci (kunci 128 bits akan menghasilkan 10 *round*, kunci 192 bits akan menghasilkan 12 *rounds*, dan kunci 256 bits akan menghasilkan 14 *rounds*), kecuali pada *round* terakhir di mana tahap *MixColumns* tidak dijalankan dalam enkripsi dan sebaliknya pada dekripsi, serta di mana kunci rahasia diubah untuk setiap *Round*. Pada tahap terakhir, blok data yang dienkripsi atau didekripsi akan menjadi hasil akhir dari proses tersebut.

2.3. Algoritma Caesar Cipher

Dalam kriptografi, *Caesar Cipher* adalah salah satu teknik enkripsi yang paling dikenal dan paling sederhana. Sandi ini termasuk dalam kategori sandi substitusi, di mana setiap huruf dalam teks terang digantikan oleh huruf lain dalam alfabet dengan pergeseran posisi tertentu. Sebagai contoh, jika kita menggunakan pergeseran tiga, maka huruf A akan digantikan oleh huruf D, huruf

B akan digantikan oleh huruf E, dan seterusnya. Dalam contoh enkripsi *Caesar Cipher* dengan menggunakan pergeseran dua, setiap huruf dalam kata "snatia" akan digantikan oleh huruf yang berada dua posisi setelahnya dalam alfabet.

- a. S digantikan oleh U.
- b. N digantikan oleh P.
- c. A digantikan oleh C.
- d. T digantikan oleh V.
- e. I digantikan oleh K.
- f. A digantikan oleh C.

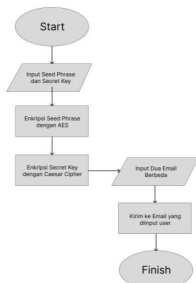
Sehingga, hasil enkripsi dari kata "snatia" dengan menggunakan *Caesar Cipher* dan pergeseran dua adalah "upcvkc". Dalam proses dekripsi, kita akan menggunakan pergeseran yang sama namun ke arah yang berlawanan untuk mengembalikan teks ke bentuk aslinya.

#### 2.4. Penyimpanan Menggunakan Email

Dalam pengembangan sistem kami untuk penyimpanan *seed phrase*, kami memanfaatkan protokol SMTP (*Simple Mail Transfer Protocol*) untuk mengirimkan data yang dienkripsi ke alamat *email* pengguna. Penggunaan SMTP memastikan pengiriman yang aman dan andal, serta memfasilitasi integrasi yang mudah dengan layanan *email* yang umum digunakan seperti Outlook dan Gmail. Kedua layanan ini dipilih karena keunggulan mereka dalam keamanan dan pemulihan data. Outlook dan Gmail menawarkan fasilitas pemulihan bencana yang kuat, yang memungkinkan pemulihan data yang cepat dan efisien dalam situasi darurat seperti kegagalan *server* atau kerusakan perangkat keras. Selain itu, keduanya memiliki infrastruktur yang sangat andal dan dapat dipercaya, dengan fasilitas replikasi dan keberlanjutan yang memastikan ketersediaan layanan yang optimal bahkan dalam kondisi yang tidak terduga. Dengan demikian, Outlook dan Gmail tidak hanya menyediakan pengiriman *email* yang aman dan handal, tetapi juga memberikan jaminan bahwa *seed phrase* pengguna akan tetap tersimpan dengan aman dan dapat diakses kembali saat diperlukan.

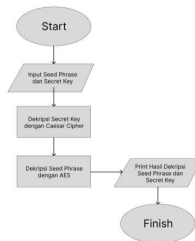
#### 2.5. Perancangan Sistem

Perancangan sistem dijelaskan dengan menggunakan diagram alur. Secara garis besar, jalannya sistem dibagi menjadi dua proses enkripsi dan proses dekripsi.



Gambar 2. Flowchart Enkripsi

- a. Proses Enkripsi:
- Program dimulai.
  - Pengguna diminta untuk memasukkan *seed phrase* dan *secret key*.
  - *Seed phrase* dan *secret key* dimasukkan ke dalam algoritma enkripsi AES untuk menghasilkan *ciphertext seed phrase*.
  - *Secret key* dienkripsi menggunakan algoritma *caesar cipher*.
  - Pengguna diminta untuk memasukkan dua alamat *email* yang berbeda untuk penyimpanan hasil enkripsi.
  - Hasil enkripsi *seed phrase* dan *secret key* disimpan secara digital di dua alamat *email* yang telah ditentukan.
  - Proses selesai.



Gambar 3. Flowchart Dekripsi

- b. Proses Dekripsi:
- Program dimulai.
  - Pengguna diminta untuk memasukkan *secret key* yang terenkripsi dan *seed phrase* terenkripsi.
  - *Secret key* dan *seed phrase* terenkripsi dimasukkan ke dalam algoritma dekripsi masing-masing.
  - Hasil dekripsi *secret key* dan *seed phrase* ditampilkan.
  - Proses selesai.

## 2.6. Pengujian Sistem

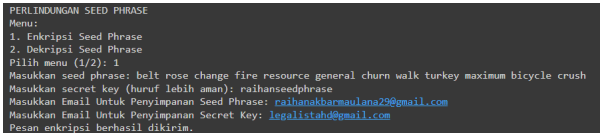
Pengujian sistem bertujuan untuk mengevaluasi tingkat keberhasilan dalam mengamankan *seed phrase* dan *secret key*, serta untuk memastikan bahwa sistem yang telah dibuat dapat berjalan tanpa mengalami *error*. Pengujian dilakukan dengan menggunakan skenario-skenario simulasi yang telah disiapkan sebelumnya.

## 3. Hasil dan Pembahasan

Penelitian ini menciptakan sebuah sistem perlindungan *seed phrase* dengan menggunakan kombinasi algoritma *Advanced Encryption Standard* (AES) dan *Caesar Cipher*. Sistem ini bertujuan untuk meningkatkan keamanan *seed phrase*, yang merupakan kunci akses utama dalam ekosistem *cryptocurrency*. Dengan menerapkan dua tahapan enkripsi, yaitu pertama, enkripsi *seed phrase* menggunakan AES, dan kedua, enkripsi kunci rahasia AES menggunakan *Caesar Cipher*, serta penggunaan email sebagai media penyimpanan digital. Sistem ini memberikan lapisan perlindungan tambahan terhadap serangan dan pencurian *seed phrase*. Dengan demikian, pengguna dapat memiliki keyakinan yang lebih tinggi dalam keamanan aset kriptoanya.

### 3.1. Pengujian Sistem

#### a. Pengujian Program Enkripsi



Gambar 4. Output Program Enkripsi

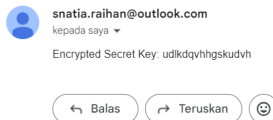
Pengujian pertama kami lakukan dengan mengenkripsi *seed phrase*, *output* dari hasil enkripsi *seed phrase* terlihat pada gambar 4. Dimasukkan *plaintext seed phrase* "belt rose change fire resource general churn walk turkey maximum bicycle crush" dengan *secret key* "raihanseedphrase" maka akan menghasilkan *ciphertext* yang dikirim ke dua *email* yang berbeda yaitu *email* 1 ke "raihanakbarmaulana29@gmail.com" dan *email* 2 ke "legalistahd@gmail.com". Tampil pesan enkripsi berhasil dikirim yang menandakan hasil enkripsi *seed phrase* dan *secret key* berhasil disimpan ke alamat *email* yang ditentukan.

#### b. Pengujian Pengiriman Email



Gambar 5. Email Seed Phrase Terenkripsi

Melanjutkan dari pengujian program enkripsi, kami menguji keberhasilan pengiriman *email* pertama. Dapat dilihat pada gambar 5, *email ciphertext seed phrase* berhasil terkirim dengan baik ke alamat *email* yang sudah ditentukan.



Gambar 6. Email Secret Key Terenkripsi

Melanjutkan dari pengujian program enkripsi, kami menguji keberhasilan pengiriman *email* kedua. Dapat dilihat pada gambar 6, *email encrypted secret key* berhasil terkirim dengan baik ke alamat *email* yang sudah ditentukan.

### c. Pengujian Program Dekripsi

```
PERINGATAN SEED PHRASE
Name:
1. Enkripsi Seed Phrase
2. Dekripsi Seed Phrase
Fill in menu (1/2): 2
Masukkan seed phrase terenkripsi: rS8gvpveGalixqhNfDyuFGiY689ZEpGb2dqf1tdxbd1f0etC5TDZJUybrNF3iL3JhLWPhZ
Masukkan secret key terenkripsi: udlkdkvvhgskudvh
Hasil dekripsi seed phrase: belt rose change fire resource general churn walk turkey maximum bicycle crush
```

Gambar 7. Output Program Dekripsi

Pengujian terakhir kami lakukan dengan mendekripsi *seed phrase*, *output* dari hasil dekripsi *seed phrase* terlihat pada gambar 7. Dimasukkan *ciphertext seed phrase* "rS8gvpveGalixqhNfDyuFGiY689ZEpGb2dqf1tdxbd1f0etC5TDZJUybrNF3iL3JhLWPhZ o4VGmQk/CBIsu64FUAYoDeo2KqxpI538sQU=" dengan *secret key* "udlkdkvvhgskudvh" yang kami dapatkan dari dua *email* yang menjadi tempat penyimpanan digital *seed phrase*. Maka akan menghasilkan *plaintext seed phrase* yang ditampilkan pada program yaitu "belt rose change fire resource general churn walk turkey maximum bicycle crush". Program berjalan dengan baik.

### 4. Kesimpulan

Dari penelitian yang telah dilakukan, kombinasi algoritma *Advanced Encryption Standard* (AES) dengan *Caesar Cipher* dapat diimplementasikan dalam enkripsi dan dekripsi *seed phrase*. Pada tahap enkripsi dijalankan menggunakan algoritma AES terlebih dahulu, lalu dilanjutkan dengan algoritma *Caesar cipher* untuk mengenkripsi *secret key* AES. Lalu hasil dari enkripsi di kirim ke dua *email* yang berbeda untuk penyimpanan secara digital. Selanjutnya, pada tahap dekripsi dijalankan dengan algoritma *Caesar cipher* dahulu untuk mendekripsi *secret key* lalu dilanjutkan dengan algoritma AES untuk mendekripsi *seed phrase*. Dengan penggabungan kedua algoritma ini membuktikan bahwa keamanan data *seed phrase* akan semakin kuat dan terjaga, karena menggunakan perlindungan enkripsi *dual-layer* algoritma untuk melindungi data *seed phrase*.

### Daftar Pustaka

- [1] Bukhari, S. T., Janjua, M. U., & Qadir, J. (2024). Secure Storage of Crypto Wallet Seed Phrase Using ECC and Splitting Technique. *IEEE Open Journal of the Computer Society*, 1–12. <https://doi.org/10.1109/OJCS.2024.3398794>
- [2] Shaik, C. (2020). Unforgettable User Defined Seed Phrase for Cryptocurrency Wallets. *International Journal on Cryptography and Information Security*, 10(4), 11–20. <https://doi.org/10.5121/ijcis.2020.10402>
- [3] Gyawali, Y. (2020). *Encryption Algorithm Advanced Encryption Standard*. <https://www.researchgate.net/publication/345684900>
- [4] Vincentrijmen, J. (2020). *Information Security and Cryptography the Design of Rijndael the Advanced Encryption Standard (AES) Second Edition*. <http://www.springer.com/series/4752>
- [5] Shaik, C. (2020). Securing Cryptocurrency Wallet Seed Phrase Digitally with Blind Key Encryption. *International Journal on Cryptography and Information Security*, 10(4), 1–10. <https://doi.org/10.5121/ijcis.2020.10401>

Halaman ini sengaja dibiarkan kosong

## Analisis Perbandingan Kualitas Citra Hasil Steganografi DCT dan LSB Berdasarkan Parameter RMSE dan PSNR

I Putu Krisnawan Putra<sup>a1</sup>, I Wayan Supriana<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>krisnawanputra09@gmail.com  
<sup>2</sup>wayan.supriana@unud.ac.id

### Abstract

*Steganography is a technique of hiding secret messages in digital media, one of the digital media that is often used is images. Two commonly used steganography methods are DCT (Discrete Cosine Transform) and LSB (Least Significant Bit). This research aims to analyze and compare the image quality of DCT and LSB steganography results based on RMSE (Root Mean Square Error) and PSNR (Peak Signal to Noise Ratio) parameters. DCT and LSB methods are implemented on several images with several variations of secret messages. RMSE and PSNR values are calculated for each stego image and analyzed to see which method produces better image quality. The results show that the LSB method produces lower RMSE values and higher PSNR values compared to the DCT method on all tested images. This shows that the LSB method is significantly better in maintaining the quality of the stego image compared to the DCT method.*

**Keywords:** Steganography, Image Quality, DCT, LSB, RMSE, PSNR

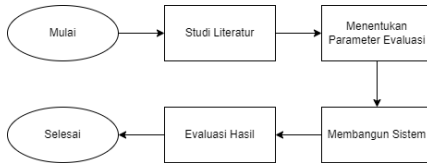
### 1. Pendahuluan

Di era kebutuhan untuk menyimpan dan mengirim informasi secara aman menjadi hal yang sangat krusial, steganografi menjadi salah satu solusi yang efektif untuk mencapai hal tersebut. Steganografi adalah ilmu dan seni menyembunyikan informasi yang dapat mencegah pendeteksian terhadap informasi yang disembunyikan[1]. Media yang digunakan umumnya merupakan suatu media yang berbeda dengan media pembawa informasi rahasia[2]. Dalam konteks citra digital, dua metode steganografi yang umum digunakan adalah metode Discrete Cosine Transform (DCT) dan Least Significant Bit (LSB). DCT adalah sebuah transformasi yang mengubah sebuah kawasan spasial menjadi kawasan frekuensi dan sebaliknya kawasan frekuensi dapat dikembalikan ke kawasan spasial dengan menggunakan invers DCT[1], sedangkan LSB bekerja dengan mengganti bit paling tidak signifikan pada citra dengan kode biner pesan sebagai nilai derajat keabuan citra pada akhir citra [3]. Pentingnya memilih metode steganografi yang tepat dan sesuai kebutuhan tidak boleh diabaikan. Salah satu parameter yang dapat digunakan untuk menilai tingkat efektivitas suatu metode steganografi dalam menyimpan dan menyembunyikan pesan pada citra digital adalah kualitas citra yang dihasilkan dari metode tersebut. Dengan memilih metode steganografi yang tepat, pengguna dapat memastikan bahwa informasi sensitif dapat disimpan dan dikirim dengan aman tanpa mengorbankan media pembawa atau meningkatkan risiko deteksi oleh pihak yang tidak berwenang. Pada penelitian ini penulis ingin menganalisis perbandingan kualitas citra hasil steganografi menggunakan metode DCT dan LSB berdasarkan parameter RMSE dan PSNR. Root Mean Square Error (RMSE) adalah metrik evaluasi yang digunakan untuk mengukur rata-rata besarnya kesalahan antara citra asli dengan citra hasil steganografi, sedangkan Peak Signal-to-Noise Ratio (PSNR) metrik untuk mengevaluasi tingkat distorsi yang dihasilkan oleh proses steganografi. Diharapkan melalui penelitian ini penulis mendapat wawasan yang lebih mendalam mengenai kelebihan dan kelemahan masing-masing metode itu, serta implikasinya terhadap keamanan informasi dalam konteks citra digital.



## 2. Metode Penelitian

Metode penelitian ini menjelaskan tahapan penelitian yang akan dilakukan oleh penulis. Pada tahap pertama, penulis melakukan studi literatur untuk memahami dan mempelajari teori-teori dari berbagai literatur yang berhubungan dengan penelitian yang dilakukan [4]. Selanjutnya, penulis menentukan parameter yang akan digunakan untuk mengevaluasi kualitas citra hasil steganografi. Setelah menentukan parameter evaluasi, penulis akan membangun sebuah sistem untuk melakukan proses steganografi pada sebuah citra digital menggunakan metode DCT dan LSB. Pada tahap akhir, penulis akan melakukan evaluasi hasil metode steganografi menggunakan parameter evaluasi yang telah ditentukan sebelumnya.



Gambar 1. Tahapan Penelitian

### 2.1 Studi Literatur

Studi literatur merupakan metode pengumpulan data dengan cara memahami dan mempelajari teori-teori dari berbagai literatur yang berhubungan dengan penelitian tersebut [4]. Pada penelitian ini penulis melakukan studi literatur untuk memahami dan mempelajari teori steganografi, metode Discrete Cosine Transform (DCT), metode Least Significant Bit (LSB), citra digital, dan parameter pengukuran kualitas citra digital Root Mean Square Error (RMSE) dan Peak Signal-to-Noise Ratio (PSNR).

### 2.2 Parameter Evaluasi Kualitas Citra Digital

Parameter evaluasi kualitas citra digital berfungsi sebagai acuan untuk mengukur tingkat kualitas suatu citra digital. Pada penelitian ini, parameter evaluasi kualitas citra digital digunakan untuk menilai kualitas citra yang dihasilkan setelah melalui proses steganografi. Parameter evaluasi yang digunakan pada penelitian ini, yaitu:

#### a. Mean Square Error (MSE)

Mean Square Error (MSE) merupakan nilai eror kuadrat rata-rata antara citra asli dengan citra hasil manipulasi[5]. MSE digunakan sebagai tolak ukur analisis kuantitatif untuk menilai kualitas sebuah citra keluaran dan keunggulan sebuah metode yang digunakan [6]. Nilai MSE dapat didefinisikan melalui persamaan berikut:

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (S_{xy} - C_{xy})^2 \quad (1)$$

Dimana  $x$  dan  $y$  adalah koordinat gambar,  $M$  dan  $N$  adalah dimensi gambar,  $S_{xy}$  adalah gambar hasil proses steganografi, dan  $C_{xy}$  adalah gambar sampel [7].

#### b. Root Mean Square Error (RMSE)

Root Mean Square Error adalah jenis lain dari teknik pengukuran kesalahan yang sangat sering digunakan, umumnya untuk mengukur perbedaan antara nilai prediksi oleh estimator dan nilai sebenarnya [8]. RMSE merupakan akar kuadrat dari MSE. RMSE mengembalikan nilai kesalahan ke skala asli, sehingga lebih mudah untuk interpretasi dalam konteks nilai asli dari data. Nilai RMSE dapat didefinisikan melalui persamaan berikut:

$$MSE = \sqrt{MSE} \tag{2}$$

c. Peak Signal-to-Noise Ratio (PSNR)

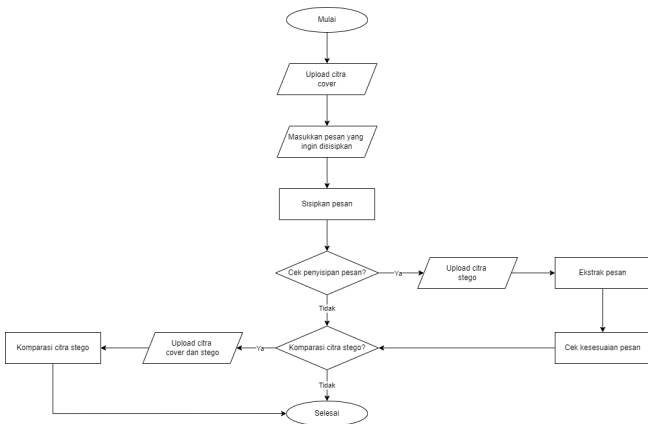
Peak Signal-to-Noise Ratio (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya *noise* yang berpengaruh pada sinyal tersebut, dalam satuan decibel (dB). Semakin besar parameter PSNR semakin mirip dengan citra asli [9]. Nilai PSNR dapat didefinisikan melalui persamaan berikut:

$$PSNR = 10Log_{10} \left( \frac{C_{max}^2}{MSE} \right) \tag{3}$$

$C_{max}^2$  adalah nilai maksimum untuk gambar. Nilai PSNR di bawah 30 dB menunjukkan kualitas yang relatif rendah dengan distorsi penyisipan yang jelas. Namun, kualitas gambar *high stay gold* adalah nilai 40 dB atau lebih [7].

**2.3 Membangun Sistem**

Sistem akan dibangun menggunakan bahasa pemrograman Python dan *library* Custom Tkinter. Sistem yang dibangun akan dapat melakukan proses *upload file* citra, melakukan penyisipan pesan pada citra, melakukan ekstraksi pesan yang ada pada citra, melakukan komparasi citra yang sudah disisipkan pesan berdasarkan parameter yang telah ditentukan sebelumnya. Metode penyisipan dan ekstraksi pesan pada citra digital menggunakan metode Discrete Cosine Transform (DCT) dan metode Least Significant Bit (LSB), sedangkan untuk proses komparasi menggunakan parameter Root Mean Square Error (RMSE) dan Peak Signal-to-Noise Ratio (PSNR).



**Gambar 2.** Alur Sistem

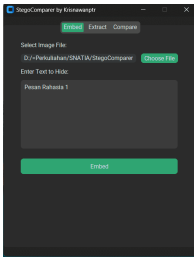
**2.4 Evaluasi Hasil**

Evaluasi hasil dilakukan dengan membandingkan citra sampul dengan citra stego berdasarkan parameter Root Mean Square Error (RMSE) dan Peak Signal-to-Noise Ratio (PSNR). Semakin

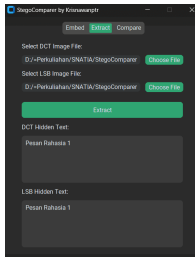
tinggi nilai RMSE dari citra stego, maka kualitas citra stego tersebut semakin rendah. Sebaliknya, semakin tinggi nilai PSNR dari citra stego, maka kualitas citra stego semakin tinggi.

### 3. Hasil dan Evaluasi

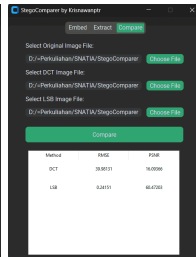
Pada tahap ini, akan ditampilkan hasil implementasi sistem dalam menyisipkan pesan pada citra digital menggunakan metode Discrete Cosine Transform (DCT) dan metode Least Significant Bit (LSB). Selanjutnya, citra stego yang dihasilkan dari kedua metode tersebut akan dievaluasi berdasarkan parameter Root Mean Square Error (RMSE) dan Peak Signal-to-Noise Ratio (PSNR).



**Gambar 3.** Proses Penyisipan Pesan



**Gambar 4.** Proses Ekstraksi Pesan

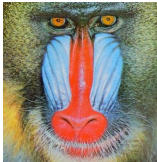



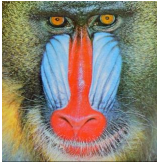
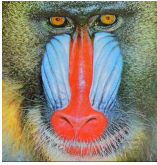
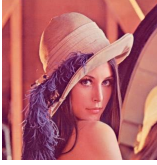
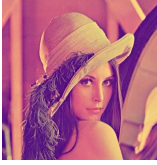
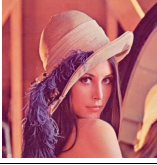
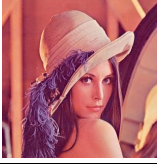




**Gambar 5.** Proses Komparasi Citra Stego

#### 3.1 Hasil Penyisipan dan Ekstraksi Pesan Terhadap Citra

Citra digital yang digunakan pada proses ini adalah citra uji standar untuk pemrosesan citra, yaitu baboon.png, lenna.png, dan tulips.png. Berikut hasil dari proses penyisipan dan ekstraksi pesan terhadap citra:

**Tabel 1.** Hasil Penyisipan dan Ekstraksi Pesan Terhadap Citra

Citra Sampul	Metode	Pesan	Citra Stego	Hasil Ekstraksi Pesan
	DCT	Pesan Rahasia 1		Pesan Rahasia 1

Citra Sampul	Metode	Pesan	Citra Stego	Hasil Ekstraksi Pesan
	LSB	Pesan Rahasia 1		Pesan Rahasia 1
	DCT	Pesan Rahasia 2		Pesan Rahasia 2
	LSB	Pesan Rahasia 2		Pesan Rahasia 2
	DCT	Pesan Rahasia 3		Pesan Rahasia 3
	LSB	Pesan Rahasia 3		Pesan Rahasia 3

Berdasarkan tabel di atas, proses penyisipan dan ekstraksi pesan berhasil dengan baik pada kedua metode steganografi. Hal ini menunjukkan bahwa kedua metode dapat diimplementasikan dengan baik pada sistem. Selain itu, terdapat perbedaan signifikan dalam hasil visual stego yang dihasilkan oleh kedua metode. Citra stego yang dihasilkan metode DCT menunjukkan perubahan yang lebih terlihat dibandingkan dengan citra sampulnya, sedangkan citra stego yang dihasilkan metode LSB tidak menunjukkan perubahan yang signifikan.

### 3.2 Hasil Evaluasi Perbandingan Kualitas Citra

Untuk melakukan evaluasi perbandingan kualitas citra, setiap citra stego yang dihasilkan akan dibandingkan dengan citra sampulnya berdasarkan parameter Root Mean Square Error (RMSE) dan Peak Signal-to-Noise Ratio (PSNR).

**Tabel 2.** Hasil Evaluasi Kualitas Citra

Citra	Metode	RMSE	PSNR
baboon.png	DCT	39.98131	16.09366
baboon.png	LSB	0.24151	60.47203
lenna.png	DCT	25.08827	20.14139
lenna.png	LSB	0.22627	61.03835
tulips.png	DCT	42.92163	15.47728
tulips.png	LSB	0.12728	66.03527

Berdasarkan hasil evaluasi kualitas citra pada Tabel 2, terlihat bahwa metode LSB menghasilkan nilai RMSE yang lebih rendah dan nilai PSNR yang lebih tinggi dibandingkan dengan metode DCT pada semua gambar yang diuji. Hal ini menunjukkan bahwa metode LSB secara signifikan lebih baik dalam menjaga kualitas citra stego dibandingkan dengan metode DCT.

### 4. Kesimpulan

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa metode steganografi LSB lebih baik dibandingkan metode steganografi DCT dalam menghasilkan citra stego dengan kualitas yang baik. Hal ini dibuktikan dengan nilai RMSE metode LSB yang sangat rendah (sekitar 0.1 hingga 0.2) dibandingkan dengan metode DCT yang memiliki nilai RMSE jauh lebih tinggi (sekitar 25 hingga 42). Ini menunjukkan bahwa metode LSB menyebabkan distorsi yang jauh lebih kecil pada citra stego dibandingkan metode DCT. Selain itu, nilai PSNR untuk metode LSB sangat tinggi (sekitar 60 hingga 66 dB), menunjukkan kualitas visual citra stego yang sangat baik dan hampir identik dengan citra asli. Sebaliknya, metode DCT memiliki nilai PSNR yang jauh lebih rendah (sekitar 15 hingga 20 dB), yang menunjukkan kualitas citra stego yang relatif rendah dan lebih mudah terlihat perbedaannya dengan citra asli. Hal ini penting untuk aplikasi steganografi yang memprioritaskan hasil citra stego yang baik agar tidak menarik perhatian pihak yang tidak berwenang. Meskipun metode DCT menghasilkan kualitas citra stego yang lebih rendah, metode ini mungkin masih berguna untuk aplikasi tertentu dimana kualitas citra stego yang sangat tinggi tidak menjadi prioritas utama.

### Daftar Pustaka

- [1] H. Ardiansyah, B. Susilo, and A. Erlansari, "Penerapan Metode Dct (Discrete Cosine Transform) Pada Aplikasi Penyembunyian Pesan Teks Berbasis Matlab," 2017. [Online]. Available: <http://ejournal.unib.ac.id/index.php/rekursif/66>
- [2] R. W. Simbolon, "Pengamanan Transkrip Nilai Mahasiswa Menggunakan Kriptografi Playfair Cipher Dan Steganografi Dengan Teknik Least Significant Bit (LSB)," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 5, no. 1, pp. 59–70, Jun. 2016.
- [3] I. Niswati, "Steganografi Berbasis Least Significant Bit (Lsb) Untuk Menyisipkan Gambar Ke Dalam Citra Gambar."
- [4] M. Nina Adlini, A. Hanifa Dinda, S. Yulinda, O. Chotimah, and S. Julia Merliyana, "Metode Penelitian Kualitatif Studi Pustaka," 2022.
- [5] A. Fahrurrosi Aziza, E. M. Imah, J. Matematika, F. Matematika, D. Ilmu, and P. Alam, "Studi Steganografi Pada Citra Digital Menggunakan Shuffled Singular Value Decomposition (SSVD)."

- [6] S. Y. Doo, S. Tena, and V. M. Ndolu, "Implementasi Pengamanan Data Menggunakan Metode Kriptografi Hill Cipher Dan Steganografi Least Significant Bit (Lsb) Pada Media Citra Digital," *Jurnal Media Elektro*, vol. VIII, no. 2.
- [7] J. Elektronik and I. Komputer Udayana, "Penerapan Steganography Untuk Perlindungan Hak Cipta Menggunakan Metode Least Significant Bit (LSB)," vol. 10, no. 4, pp. 2654–5101, 2022, [Online]. Available: <https://archive.org/details/Bali>.
- [8] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *Journal of Computer and Communications*, vol. 07, no. 03, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
- [9] E. Yudi Hidayat and K. Hastuti, "Analisis Steganografi Metode Least Significant Bit (Lsb) Dengan Penyisipan Sekuensial Dan Acak Secara Kuantitatif Dan Visual," 2013.

Halaman ini sengaja dibiarkan kosong

# Klasifikasi Jenis Obat Berdasarkan Gejala yang Dimiliki Pasien Menggunakan Metode K-Nearest Neighbors (KNN)

Ngakan Putu Bagus Ananta Wijaya<sup>a1</sup>, Ida Ayu Gde Suwiprabayanti Putra<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>anantawijaya994@email.com  
<sup>2</sup>iagsuwiprabayantiputra@unud.ac.id

## Abstract

*This research applies the K-Nearest Neighbors (KNN) algorithm to classify medicine types based on patient symptoms using a dataset from Kaggle with 200 rows and 6 columns. After preprocessing steps such as handling missing values, encoding categorical variables, and splitting data into training and testing sets, exploratory data analysis (EDA) was performed to understand the dataset's structure. The KNN model was evaluated with  $k$  values of 1, 2, and 3, finding the optimal  $k$  to be 3, achieving an accuracy of 77.50% with average precision of 0.76, recall of 0.69, and  $f1$ -score of 0.66. Lower accuracy was observed for  $k=2$  (65.00%) and  $k=1$  (67.50%), indicating that  $k=3$  is the most effective for this dataset. These results suggest that while KNN is a viable method for classifying medicine types based on symptoms, larger datasets are recommended for improved accuracy.*

**Keywords:** K-Nearest Neighbors (KNN), classify, medicine, exploratory data analysis (EDA), preprocessing

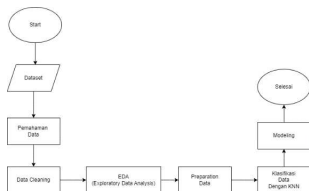
## 1. Pendahuluan

Penggunaan teknologi dalam Kesehatan telah meningkatkan efisiensi dan akurasi dalam diagnosa dan pengobatan penyakit. Salah satu metode yang sangat populer dalam klasifikasi jenis obat berdasarkan gejala yang dimiliki pasien adalah menggunakan algoritma *K-Nearest Neighbors*. Algoritma ini merupakan metode pembelajaran mesin yang digunakan untuk memprediksi kelas objek berdasarkan kelas objek yang paling dekat dengannya dan sangat cocok untuk digunakan dalam klasifikasi jenis obat berdasarkan gejala pasien [1]. Algoritma KNN adalah algoritma yang relatif sederhana dan mudah digunakan, tetapi memiliki beberapa kelebihan yang membuatnya sangat populer[2]. Salah satu kelebihan utama adalah kemampuan KNN untuk tidak memerlukan asumsi tentang distribusi data, yang membuatnya sangat efektif dalam menghadapi data yang tidak berdistribusi normal[3]. Selain itu, KNN juga dapat digunakan untuk mengklasifikasikan data yang memiliki atribut numerik dan kategorikal, membuatnya sangat fleksibel dalam aplikasi kesehatan yang beragam[4]. Jenis obat akan melalui proses klasifikasi berdasarkan data-data pasien seperti umur pasien, kandungan kolesterol, tekanan darah dan rasio natrium. Data-data tersebut akan dikumpulkan dan akan diolah melalui beberapa tahap yaitu tahap pemahaman data, preprocessing data, *Exploratory Data Analysis (EDA)*, *preparation data*, dan evaluasi dengan algoritma KNN. Penelitian ini menggunakan python dan librarynya untuk mengolah data.

## 2. Metode Penelitian

Studi penelitian ini memanfaatkan algoritma pembelajaran mesin, yaitu metode klasifikasi yang menggunakan algoritma *K-Nearest Neighbors* (KNN). Diagram alir pada Gambar 1 menunjukkan beberapa tahapan penelitian ini.





Gambar 1. Diagram Alir

### 2.1. Pemahaman data

Studi ini menggunakan data sekunder dari situs *Kaggle*, yang datanya berjumlah 200 baris dan 6 kolom yang berisi informasi tentang umur, jenis kelamin, kolesterol, tekanan darah, rasio natrium, dan jenis obat. Tabel 1 menunjukkan data yang terdapat pada dataset.

Tabel 1. Tabel Dataset

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY
...	...	...	...	...	...	...

Pada tahap ini, program menggunakan library python (*pandas*, *numpy*, *seaborn* dan *matplotlib*) untuk memuat statistik distribusi jumlah data yang terdapat pada kolom-kolom dataset, mengecek tipe data dan statistik rata rata jumlah kolom, serta memvisualisasikan data dengan histogram.

### 2.2. Preprocessing data

Sebelum melanjutkan ke tahap berikutnya, dataset harus melalui tahap preprocessing data untuk menghilangkan data yang salah atau tidak sesuai. Tahapan preprocessing data termasuk tahapan berikut:

- a. Pengecekan *missing values*  
 Dataset harus diperiksa terlebih dahulu apakah terdapat nilai-nilai data yang hilang pada dataset. Pada dataset yang digunakan saat ini tidak terdapat nilai-nilai data yang hilang
- b. Pengecekan duplikasi  
 Dataset yang digunakan juga harus diperiksa apakah terdapat data yang duplikasi atau tidak. Pada dataset yang digunakan saat ini juga tidak terdapat data yang duplikasi
- c. *Encoding*  
 Karena beberapa fitur dalam dataset berupa data kategorikal, seperti pada kolom 'Sex', 'BP', 'Cholesterol', dan 'Drug', harus dilakukan proses encoding menggunakan *LabelEncoder* untuk mengkonversi nilai kategorikal menjadi numerik.
- d. Pemisahan fitur dan target  
 Fitur (X) dan target (y) dipisahkan untuk diproses lebih lanjut. Fitur 'Drug' dihapus dari dataset X karena merupakan target data.

### 2.3. Exploratory Data Analysis (EDA)

EDA merupakan metode yang digunakan untuk memahami dan menelaah data sebelum memulai analisis yang lebih mendalam[5]. Pada tahap ini dataset yang digunakan akan dieksplorasi dengan visualisasi dan statistik untuk memahami hubungan dataset. Ringkasan statistik dan visualisasi distribusi digunakan untuk memahami variabel numerik, sementara plot hubungan antar variabel membantu mengidentifikasi korelasi. Variabel kategorikal divisualisasikan untuk melihat distribusinya. Tahap ini juga mencakup penanganan missing values dan pemahaman tentang target jika ada.

### 2.4. Preparation data

Pada tahap *preparation* data, data dimodifikasi agar sesuai dengan kebutuhan analisis atau pemodelan. Langkah-langkah melibatkan pemisahan fitur dan target, *encoding* variabel kategorikal, penanganan nilai yang hilang, dan penyusunan data *training* dan *testing*. Untuk memastikan bahwa skala data konsisten, standarisasi atau normalisasi juga dapat dilakukan. Persiapan data yang baik akan dapat membuat data menjadi lebih siap digunakan dan akan memacu hasil yang akurat dan relevan

### 2.5. Klasifikasi dengan Algoritma KNN

Algoritma ini memiliki kemampuan untuk melakukan klasifikasi dengan mencari jarak terdekat antara data dengan K-Nearest Neighbors [1]. Pada dataset ini, jenis obat akan diklasifikasi berdasarkan gejala pasien yang ada di dalamnya. Adapun rumus dari algoritma KNN ini:

$$Euclidean\ distance = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{1}$$

*Euclidean distance* = jarak ketetangaan

- $x_i$  = data train
- $y_i$  = data test
- $i$  = variable
- $n$  = dimensi data

## 3. Hasil dan Diskusi

### 3.1. Pemahaman data

Pada tahap awal ini, dilakukan pemeriksaan pada dataset untuk memahami struktur dataset. Dataset memiliki ukuran 200 baris dan 6 kolom yang mencakup umur, jenis kelamin, kolesterol, tekanan darah, rasio natrium dan jenis obat. Program selanjutnya akan memeriksa tipe data kolom, memeriksa statistik deskriptif yang menunjukkan rata-rata data per kolom, dan memvisualisasikan histogram untuk menunjukkan distribusi dan frekuensi data-data per kolom pada dataset.

```
memeriksa data
print(shape(*, df_data))
ukuran : (200, 6)

df_data[0]
<class 'pandas.core.frame.DataFrame'>
Integritas: 200 entri(s), 6 x 200
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Age         200 non-null    int64
 1   Sex         200 non-null    object
 2   BP          200 non-null    object
 3   Cholesterol 200 non-null    object
 4   NaToK      200 non-null    float64
 5   Drug       200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 8.5+ KB
```

Gambar 2. Program Proses Pemahaman Data

```
df.describe()

```

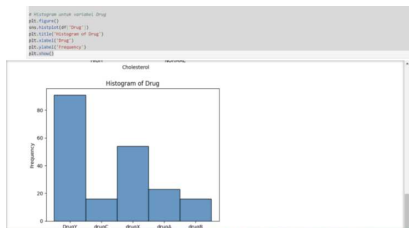
	Age	No. No. K
count	200.000000	200.000000
mean	48.375000	16.084451
std	16.544015	7.223956
min	15.000000	6.200000
25%	31.000000	10.440000
50%	40.000000	12.000000
75%	58.000000	19.160000
max	74.000000	38.247000

```
df.info()

```

	Drug	Cholesterol
Drug	object	object
Drug1	object	object
Drug2	object	object
Drug3	object	object
Drug4	object	object
Drug5	object	object
dtypes: object 6		

Gambar 3. Program Proses Pemahaman Data



Gambar 4. Program Proses Pemahaman Data

### 3.2. Data Cleaning

Tahap selanjutnya adalah data cleaning yang bertujuan untuk penanganan nilai yang hilang serta pengecekan duplikat data. Pemeriksaan yang dilakukan menunjukkan bahwa tidak ada nilai yang hilang di seluruh kolom dataset dan data yang duplikat juga tidak ditemukan. Oleh karena itu, dataset ini bersifat konsisten dan siap untuk melanjutkan tahap analisis EDA.

```
df.isnull().sum()

```

	Age	Sex	BP	Cholesterol	No. No. K	Drug	dtypes
df.isnull().sum()	0	0	0	0	0	0	0

```
df.duplicated().sum()

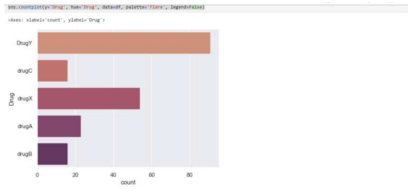
```

0

Gambar 5. Program Proses Data Cleaning

### 3.3. Exploratory Data Analysis (EDA)

EDA dilakukan untuk lebih dalam memahami struktur dan karakteristik yang terdapat pada dataset. Grafik-grafik yang dibuat memperlihatkan distribusi jumlah data pada kolom yang ditampilkan. Gambar 6 menunjukkan sebaran jumlah data dari kategori jenis obat. Informasi ini membantu pemahaman tentang dataset yang digunakan.



Gambar 6. Program Proses EDA



Gambar 7. Program Proses EDA

### 3.4. Preparation data

Dataset akan melalui proses encoding variabel kategorikal pada kolom 'Sex', 'BP', 'Cholesterol', dan 'Drug' dikarenakan data-data pada kolom yang disebutkan masih bertipe *string* dan perlu diubah menjadi tipe *integer* agar bisa diproses. Kemudian, dataset akan dibagi menjadi fitur (X) dan target (y), dan dibagi lagi menjadi *data training* dan *data testing* dengan rasio 80:20, dimana 80% untuk *training* dan 20% untuk *testing*. Setelah pembagian data, output ukuran pembagian *data training* dan *data testing* akan dicetak seperti pada Gambar 8.

```
df.head()
  Age  Sex  BP  Cholesterol  Na.to.X  Drug
0   23  0  0      0         25.355  0
1   47  1  1      0         13.093  3
2   47  1  1      0         10.114  3
3   26  0  2      0          7.798  4
4   61  0  1      0         18.043  0

X = df.drop(columns = ['Drug'])
y = df['Drug']

print("X : ", X.shape)
print("y : ", y.shape)

X = (2000, 13)
y = (2000, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

print(f"Data x_train : {X_train.shape}")
print(f"Data y_train : {y_train.shape}")
print(f"Data x_test : {X_test.shape}")
print(f"Data y_test : {y_test.shape}")

Data x_train : (1600, 13)
Data y_train : (1600, 1)
Data x_test : (400, 13)
Data y_test : (400, 1)
```

Gambar 8. Program Proses Preparation Data

### 3.5. Modeling

Tahap terakhir adalah modeling dengan menggunakan algoritma KNN sebagai model. Model KNN diinisialisasi dengan jumlah tetangga yang dipakai ada (3), (2), (1).

- a. KNN dengan k=3  
Hasil akurasi KNN jika menggunakan k=3 sudah cukup bagus dikarenakan berhasil memperoleh akurasi sebesar 77.50% yang menandakan bahwa k yang digunakan cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.76, *recall* sebesar 0.69 dan *f1-score* sebesar 0.66.
- b. KNN dengan k=2  
Hasil akurasi KNN jika menggunakan k=2, mengalami penurunan menjadi 65.00% yang menandakan bahwa k yang digunakan masih belum cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.50, *recall* sebesar 0.59 dan *f1-score* sebesar 0.49 yang menandakan penurunan dari k = 3.
- c. KNN dengan k=1  
Hasil akurasi KNN jika menggunakan k=1, mengalami kenaikan sedikit hanya menjadi 67.50% yang menandakan bahwa k yang digunakan juga masih belum cukup optimal. Model KNN juga menunjukkan hasil akurasi rata rata dari *precision* sebesar 0.58, *recall* sebesar 0.60 dan *f1-score* sebesar 0.58 yang menandakan sedikit kenaikan dari k=2.

```
knm = KNeighborsClassifier(n_neighbors=1)
knm.fit(x_train, y_train)

y_pred = knm.predict(x_test)
knn_acc = accuracy_score(y_test, y_pred)

print(classification_report(y_test, y_pred))
print('Akurasi KNN = {}'.format(knn_acc*100))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	15
1	0.50	0.63	0.57	6
2	0.50	0.67	0.57	3
3	1.00	0.20	0.33	5
4	0.73	0.73	0.73	11

accuracy                    0.78        40  
macro avg                0.76        40  
weighted avg            0.62        40  
Akurasi KNN = 77.50%

Gambar 10. Program Modeling

```
testing = {'Age': [30],  
          'Sex': [1],  
          'BP': [2],  
          'Cholesterol': [8],  
          'Na_to_K': [4.5]}  
  
testing = pd.DataFrame(testing)  
testing
```

Age	Sex	BP	Cholesterol	Na_to_K	
0	35	1	2	0	4.5

```
pred_coba = knm.predict(testing)  
print('Hasil Prediksi Pasien Baru = {}'.format(pred_coba))  
Hasil Prediksi Pasien Baru =  
[4]
```

Gambar 11. Program Modeling

## 4. Kesimpulan

Berdasarkan hasil studi penelitian yang sudah dilakukan, dapat diketahui bahwa k = 3 adalah k yang paling cocok dan optimal untuk digunakan dengan akurasi 77,50%, sedangkan k = 1 dan k = 2 lebih buruk dengan akurasi hanya 65,00% dan 67,50%, masing-masing. Karena jumlah data yang terbatas dalam dataset yang digunakan, akurasi tidak dapat mencapai 100%. Oleh karena itu, algoritma KNN dapat dengan mudah dan optimal mengklasifikasikan obat berdasarkan gejala

pasien. Oleh karena itu, peneliti menyarankan untuk menggunakan dataset dengan jumlah data yang lebih besar untuk mendapatkan hasil yang lebih akurat dan menghasilkan model terbaik.

#### Daftar Pustaka

- [1] Hasran. (2020). *Indonesian Journal of Data and Science Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor*. 1(1), 6–10. <http://bit.ly/datasetcardio>.
- [2] Novitasari, H. B., Hadianto, N., Sfenrianto, Rahmawati, A., Prasetyo, R., Miharja, J., & Gata, W. (2019). K-nearest neighbor analysis to predict the accuracy of product delivery using administration of raw material model in the cosmetic industry (PT Cedefindo). *Journal of Physics: Conference Series*, 1367(1). <https://doi.org/10.1088/1742-6596/1367/1/012008>
- [3] Azis, A., Zy, A. T., & Sunge, A. S. (2024). Prediksi Penjualan Obat Dan Alat Kesehatan Terlaris Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(1), 117–124. <https://doi.org/10.47233/jteksis.v6i1.1078>
- [4] andiani, L., Ilmu Komputer, J., & Palupi Rini, D. (2019). Analisis Penyakit Jantung Menggunakan Metode KNN Dan Random Forest. In *Prosiding Annual Research Seminar* (Vol. 5, Issue 1).
- [5] Sagala, N. T. M., & Aryatama, F. Y. (2022). *SISTEMASI: Jurnal Sistem Informasi Exploratory Data Analysis (EDA): A Study of Olympic Medallist Analisis Data Eksplorasi (EDA): Studi Peraih Medali Olimpiade* (Vol. 11, Issue 3). <http://sistemasi.ftik.unisi.ac.id>

Halaman ini sengaja dibiarkan kosong

# Perlindungan Transkrip Akademik Mahasiswa dengan Kombinasi Algoritma Rijndael dan SHA-3

Amsal Hamonangan Butarbutar<sup>a1</sup>, Ida Bagus Gede Dwidasmara<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>butarbutar.2208561134@student.unud.ac.id  
<sup>2</sup>dwidasmara@unud.ac.id

## Abstract

*This paper addresses the issue of securing academic transcripts for students by utilizing a combination of Rijndael encryption algorithm and Secure Hashing Algorithm-3 (SHA-3). The primary purpose is to enhance the protection and integrity of academic records against unauthorized access and tampering. SHA-3 is selected for its advanced cryptographic capabilities, surpassing its predecessors in security and efficiency. When combined with the Rijndael algorithm, which serves as the foundation for the Advanced Encryption Standard (AES), the system achieves robust encryption and data integrity verification. Through comprehensive design, implementation, and testing phases, the study demonstrates that the proposed method is effective in safeguarding academic transcripts. The results indicate that the system not only ensures data security but also operates efficiently. This research provides a valuable reference for further development and implementation of secure academic data management systems.*

**Keywords:** Rijndael, SHA-3, Academic Transcript, Cryptography

## 1. Pendahuluan

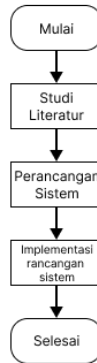
Di era digital ini, keamanan data menjadi salah satu aspek yang sangat penting dalam berbagai bidang, termasuk pendidikan tinggi. Dalam konteks akademik, transkrip mahasiswa adalah dokumen yang menyimpan jejak akademis seseorang, yang mencakup detail kursus yang diambil, nilai yang diperoleh, dan prestasi lainnya. Keandalan dan keamanan transkrip akademik menjadi krusial dalam memastikan integritas dan kepercayaan terhadap proses pendidikan. Namun, tantangan muncul ketika mempertimbangkan perlindungan data ini dalam lingkungan digital yang rentan terhadap berbagai serangan siber yang dapat dengan mudah memodifikasi data menyebabkan hilangnya integritas data pada transkrip akademik tersebut. Dalam rangka memperkuat dan mencegah hilangnya integritas dan keamanan transkrip akademik mahasiswa, penelitian ini memperkenalkan sebuah pendekatan yang menggabungkan dua teknologi kriptografi yang kuat. Kriptografi merupakan sebuah ilmu menjaga kerahasiaan pesan dengan cara menyandikannya dalam bentuk yang tidak dapat dipahami lagi[1]. Adapun teknik kriptografi yaitu dengan Algoritma Rijndael dan Secure Hash Algorithm-3 (SHA-3). Algoritma Rijndael lebih efisien dibandingkan dengan algoritma Camellia berdasarkan teori kompleksitas waktu sebagai parameter efisiensi. Algoritma Camellia lebih efektif dibandingkan algoritma Rijndael berdasarkan nilai Avalanche Effect sebagai parameter efektivitas. Algoritma Rijndael memiliki kualitas lebih tinggi dibandingkan algoritma Camellia berdasarkan perhitungan deviasi maksimum, koefisien korelasi, deviasi ketidakteraturan, dan PSNR sebagai parameter kualitas. Penelitian ini dilakukan pada file berbentuk citra[2]. Sedangkan SHA-3 merupakan teknik hashing yang memiliki kinerja dan ketahanan yang lebih baik daripada algoritma SHA-1 yang menjadi alasan menggunakan algoritma SHA-3 karena merupakan algoritma SHA yang paling baru dan kinerja yang lebih baik daripada varian lainnya[3]. Dengan menerapkan pendekatan tersebut diharapkan dapat memberi perlindungan yang lebih kuat terhadap data sensitif dari ancaman yang beragam termasuk serangan terhadap keutuhan data dan upaya modifikasi data informasi pada transkrip akademik tersebut.



## 2. Metode Penelitian

### 2.1 Alur Penelitian

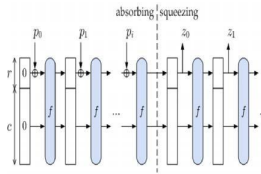
Pada penelitian ini, metode pengumpulan data yang dilakukan adalah secara kualitatif dengan melakukan studi literatur yang bertujuan untuk mengumpulkan semua metode yang digunakan dan dibutuhkan dalam penelitian ini sebagai sumber informasi yang didapat dari berbagai jurnal dan riset ilmiah yang sudah pernah dilakukan agar dapat mendukung gagasan yang akan dibuat pada penelitian ini. Dan selanjutnya akan mengimplementasikan rancangan tersebut langsung ke dalam sebuah sistem untuk mengujinya.



Gambar 1. Alur Penelitian

### 2.2 Algoritma SHA-3 (KECCAK)

SHA-3 merupakan salah satu bentuk dari sebuah fungsi hash yang merupakan komplementar dari fungsi SHA-1 dan SHA-2. SHA-3 memiliki sistem kerja dengan konstruksi *sponge* yang panjang dari *message digest* dapat disesuaikan[4]. Pada proses hash suatu *message* diawali dengan proses *padding message* yang secara otomatis menyesuaikan dengan panjang outputnya sesuai yang diinginkan. Selanjutnya membagi *message* yang sudah di-*padding* menjadi ukuran sesuai  $r$ -bit[5]. Perancangan SHA-3 menggunakan konstruksi spons yaitu data diserap ke dalam spons. Pada fase ini blok pesan XOR menjadi bagian dari *state* yang kemudian diubah secara keseluruhan menggunakan fungsi permutasi  $f$ , kemudian hasilnya diperas. Pada fase ini, blok keluaran dibaca dari *subset* keadaan yang sama. dengan fungsi transformasi  $f$ .



Gambar 2. Konstruksi Sponge

Berikut langkah-langkah contoh pembuatan pesan menggunakan hash SHA-3, secara umum adalah sebagai berikut:

- Menambah Bit Buffer
- Menambah Nilai Panjang Pesan Asli
- Menginisialisasi MD Buffer
- Memproses Pesan dalam Blok dengan ukuran 512 bit

Tabel 1. SHA-3 Padding

Type	Output Length	Rate (r)	Capacity (c)
SHA3-224	224	1152	448
SHA3-256	256	1088	512
SHA3-384	384	832	768
SHA3-512	512	576	1024

### 2.3 Algoritma Rijndael

Algoritma Rijndael menggunakan block berukuran 128-bit input dan memiliki 3 ukuran *key*, yaitu 128-bit, 192-bit, dan 256-bit. Input dan output dari algoritma Rijndael terdiri dari urutan data yang berukuran 128-bit[6]. Urutan data setiap satu kelompok 128-bit input disebut sebagai blok data atau *plaintext* yang kemudian akan dienkripsi menjadi *ciphertext*. Pengelompokan jenis Rijndael ini didasarkan pada panjang kunci yang digunakan. Angka-angka yang terdapat di belakang kata Rijndael menunjukkan panjang kunci yang digunakan dalam satuan bit. Berdasarkan ukuran bloknya, Rijndael bekerja pada matriks dengan ukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Selain itu, yang menjadi perbedaan tipe Rijndael ini adalah banyaknya round yang dipakai. Rijndael-128 menggunakan 10 round, Rijndael-192 menggunakan 12 round, sedangkan Rijndael-256 menggunakan 14 round.

Adapun tahapan yang dilakukan dalam algoritma Rijndael adalah sebagai berikut ini:

Tahap 1 : AddRoundKey, di mana dilakukan operasi XOR antara state dengan *cipher key*. Tahap pertama ini juga dikenal sebagai *initial round*.

Tahap 2 : Putaran sebanyak  $(Nr - 1)$  kali. Setiap putaran mencakup:

- SubBytes: Mengganti isi matriks dengan matriks substitusi (S-Box).
- Shift Rows: Menggeser setiap baris dalam state, baris pertama tidak digeser, baris kedua digeser satu posisi, dan seterusnya.
- Mix Column: Mengalikan setiap elemen block cipher dengan *Mix-Column Matrix* dan memasukkan hasilnya ke dalam *block cipher* yang baru.

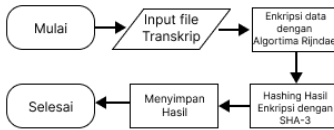
- *AddRoundKey*: Sama dengan tahap pertama, tetapi dilakukan pada setiap putaran dengan meng-XORkan *state* saat ini dengan *round key*.

Tahap 3 : Final round, yang mencakup *SubBytes*, *Shift Rows*, dan *AddRoundKey* untuk putaran terakhir.

### 3. Hasil dan Pembahasan

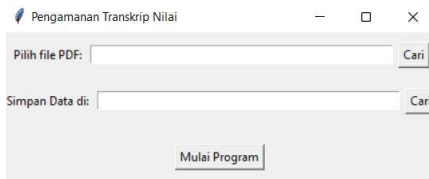
#### 3.1. Rancangan Desain Sistem dan Pengujian

Dalam penelitian ini, rancangan desain sistem yang diharapkan adalah saat user mengunduh transkrip akademi langsung dari website pembelajaran perkuliahan, maka file transkrip akademik tersebut akan otomatis ter-hashing dan terenkripsi, namun karena adanya keterbatasan dalam hal tersebut yang disebabkan oleh beberapa alasan lainnya juga maka Adapun rancangan dan alur sistem yang diharapkan adalah sebagai berikut ini.



Gambar 3. Flowchart Sistem

Dimana pada tahap pertama yaitu user akan diminta untuk menginputkan file yaitu file transkrip nilai yang selanjutnya sistem akan membaca isi data dari file transkrip nilai tersebut lalu akan mengenkripsi seluruh data yang ada dalam file transkrip tersebut dengan menggunakan algoritma Rijndael.



Gambar 4. GUI Sederhana Masukan Program

Kemudian akan dilanjut dengan menciptakan sebuah nilai hash dengan SHA-3 dari hasil enkripsi tersebut agar integrasi file transkrip tersebut dapat terjamin dan mengidentifikasi jika adanya perubahan nilai atau data dari file transkrip yang sudah di enkripsi tersebut, selanjutnya data yang dihasilkan dari tahapan-tahapan tadi akan disimpan dalam sebuah database, tetapi dalam pengujian ini akan menyimpannya dalam sebuah file berbentuk *.txt*, penyimpanan nilai meliputi data yang sudah di enkripsi dengan algoritma Rijndael, kunci enkripsi dan dekripsi, dan nilai hash dari nilai data yang sudah di enkripsii tadi. Tujuan dari penyimpanan nilai-nilai tadi agar dapat dengan mudah memverifikasi kembali transkrip nilai tersebut untuk mencegah terjadinya perubahan isi file transkrip nilai.

### 3.2. Implementasi Algoritma

#### a. Implementasi Algoritma Rijndael

Algoritma Rijndael diimplementasikan untuk mengenkripsi isi dari file transkrip agar tidak terbaca dan dimengerti oleh orang lain terutama yang tidak mengenal teori kriptografi yang menciptakan isi data dari file transkrip tersebut akan sangat sulit untuk diubah karena perlu untuk mendekripsi kembali ke bentuk yang dapat dibaca.

Tabel 2. Penggalan Program Enkripsi Menggunakan Algoritma Rijndael

```
def encrypt_data(data, key):  
    cipher = AES.new(key, AES.MODE_EAX)  
    nonce = cipher.nonce  
    ciphertext,tag= cipher.encrypt_and_digest(data)  
    return nonce, ciphertext, tag
```

Fungsi *encrypt\_data()* akan meminta variable yang akan menjadi data dan *key* yang mana data merupakan isi dari file yang sudah diinputkan dan *key* merupakan kunci yang sudah diciptakan dengan angka acak yang sudah diatur sedemikian rupa agar menghasilkan kunci yang rumit dan sulit ditebak karena kunci merupakan salah satu hal yang penting untuk diperhatikan. Pengenkripsian menggunakan fungsi dari Tabel 2 dibantu dengan salah satu *library python* yaitu *Crypto.Cipher* untuk meringkas sintaks dan logika yang panjang dalam pengenkripsian menggunakan algoritma rijndael.

#### b. Implementasi Secure Hash Algorithm-3 (SHA-3)

Algoritma SHA-3 diimplementasikan menggunakan Bahasa pemrograman Python yang bertujuan untuk menciptakan seuntai nilai hash unik dari setiap file yang akan diinputkan, setelah itu nilai hash dari tiap file akan disimpan untuk dibandingkan Kembali pada saat user ingin memeriksa integritas dari transkrip akademik mereka.

Tabel 3. Penggalan Program Menghitung SHA-3

```
def hash_sha3(pdf_file):  
    with open(pdf_file, 'rb') as f:  
        pdf_data = f.read()  
        hasher = hashlib.sha3_512()  
        hasher.update(pdf_data)  
        return hasher.hexdigest()
```

Dari penggalan sintaks diatas, fungsi *hash\_sha3()* akan meminta dan menerima inputan file pdf dari user setelah itu akan membuka file tersebut dan membaca isi filenya lalu dengan sintaks yang sudah dipersingkat menggunakan *library Python* akan menciptakan sebuah nilai hash yang disimpan dalam sebuah variable bernama *hasher*. *Dibawah* ini merupakan bukti pengujian penerapan SHA-3 dalam sistem ini

```
Enter the path to the academic transcript file: Tran  
skrip Sementara 2208561134.pdf  
SHA-3 hash of the file: 71674f95a42af1096c8fe275cc9e  
917b8449e9f9e4dd0ed13165efc27001c26b6ab6a65b2d47bf77  
a915d3814fad4ebc87c5014dfb773e5e6ae95d925f8111ba1
```

Gambar 5. Hasil Hashing File Transkrip Akademik

Pada sistem yang sudah dirancang, implementasi dari *Secure Hash Algorithm-3* (SHA-3), digunakan sebagai acuan untuk memastikan keutuhan data transkrip karena perubahan sekecil apapun akan mengubah seluruh nilai hash yang sudah disimpan. Ketika transkrip diakses atau dimodifikasi, nilai hash akan dihitung ulang dan dibandingkan dengan nilai hash yang tersimpan, jika nilai hash tidak cocok maka berarti transkrip telah diubah dan dirusak. Meskipun seseorang mendapatkan Salinan transkrip, mereka tidak dapat mengubahnya tanpa terdeteksi.

**Tabel 4.** Pengujian Hashing Berbagai File

Nama File	Nilai Hash
Transkrip Sementara 2208561134.pdf	71674f95a42af1096c8fe275cc9e917b8449e9fbc4dd0e d13165efc27001c26b6ab6a65b2df7bf77a915d3814fad 4ebc87c501dfb773e5e6ae95d925f8111ba1
Sertifikat Anton.pdf	1cf05dc493b50d47c94e6c10c087e70d24e2f6027ed7f 335232fbdedb36d31d36409793c3269052360cd3f4a5e 78f45dea755b47dda6c28e5ec0dc37c3c77602
Transkrip Nilai 2208561022.pdf	67e8406c5ea2fdac3e96584397a1ea43ddd0e58c40bf5 aa801e399d44276c9ed176e82d68920cbf92e321a16 5605480efa1a665513bf32fc6a69cfe0967c1b7

#### 4. Kesimpulan

Kombinasi antara algoritma Rijndael dan SHA-3 dalam penelitian ini terbukti efektif dalam meningkatkan keamanan transkrip akademik mahasiswa. Algoritma Rijndael, yang merupakan dasar dari Advanced Encryption Standard (AES), memberikan enkripsi yang kuat dan cepat, sementara SHA-3 memastikan integritas data dengan fungsi hash yang aman dan tahan terhadap serangan kriptografi modern. Namun demikian, ada beberapa aspek yang dapat ditingkatkan untuk penelitian selanjutnya. Pertama, implementasi sistem ini dapat diuji dalam skala yang lebih besar dan dalam lingkungan yang lebih beragam untuk menguji ketahanannya terhadap berbagai jenis serangan. Kedua, penggunaan teknik tambahan seperti mekanisme autentikasi multifaktor dapat diteliti lebih lanjut untuk meningkatkan keamanan sistem secara keseluruhan. Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi algoritma Rijndael dan SHA-3 adalah solusi yang efektif dan efisien untuk perlindungan transkrip akademik mahasiswa. Implementasi metode ini diharapkan dapat menjadi referensi bagi pengembangan sistem keamanan data akademik lainnya di masa depan

#### Daftar Pustaka

- [1] M. Azhari, D. I. Mulyana, F. J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 01, pp. 163–171, 2022.
- [2] B. S. W. Poetro and R. Wardoyo, "Perbandingan Efisiensi, Efektifitas dan Kualitas Algoritma Rijndael dengan Algoritma Camellia pada Citra Digital," *BIMIPA*, vol. 24, no. 3, pp. 281–291, 2014.
- [3] M. P. Sari, "Analisis Algoritma SHA-3 Keamanan pada Data Pribadi," *JURNAL Tecnoscienza*, vol. 5, no. 2, pp. 231–242, 2021.
- [4] R. Munir, "SHA-3 (Keccak)," *Diakses pada*, vol. 20, pp. 2022–2023, 2023.
- [5] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 803–812, 2017.
- [6] N. W. Hidayatulloh, M. Tahir, H. Amalia, N. A. Basyar, A. F. Prianggara, and M. Yasin, "Mengenal Advance Encryption Standard (AES) sebagai Algoritma Kriptografi dalam Mengamankan Data," *Digital Transformation Technology*, vol. 3, no. 1, pp. 1–10, 2023.

## Klasifikasi Kematangan Sayuran Pare dengan Metode KNN

I Gede Yogananda Adi Baskara<sup>a1</sup>, Luh Arida Ayu Rahning Putri<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>baskara.2208561061@student.unud.ac.id  
<sup>2</sup>rahningputri@unud.ac.id

### Abstract

*The bitter melon plant (*Momordica charantia* L) is a vegetable commodity that has commercial potential if cultivated on an agribusiness scale. The bitter melon plant products currently have quite a lot of consumers and have even entered supermarkets. However, the selection of bitter melon vegetables still uses human eye assessment which has the weakness of being subjective and inconsistent, so the level of accuracy is low. Based on these problems, researchers will create a system that is able to classify the level of maturity of bitter melon vegetables using HSV feature extraction with the KNN method at the classification stage and with the help of the Python programming language. In this research, 160 datasets will be used which are divided into 3 types of classes, namely category cooked bitter melon vegetables and raw bitter melon vegetables. The dataset is divided into two ories, namely 128 training data and 32 test data. The next stage is testing the data using the K-Nearest Neighbor method using the value  $k=3$ . From the test results, an accuracy rate of 88% was obtained.*

**Keywords:** Python, Kematangan, Pare, KNN, HSV

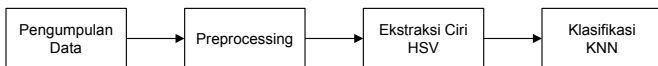
### 1. Pendahuluan

Tanaman pare merupakan sayuran yang berpotensi menguntungkan secara besar-besaran untuk tujuan bisnis. Awalnya, tanaman merambat asal Asia ini hanya ditanam secara sederhana karena rendahnya permintaan pasar dan konsumen. Namun, seiring dengan banyaknya penelitian tentang tanaman pare dan munculnya varietas-varietas unggul dalam hal cita rasa dan penampilan, produk dari tanaman pare kini sudah cukup laku dan bahkan dijual di supermarket besar [1]. Biasanya pare diolah dengan cara digoreng, direbus atau dikukus, diasinkan, dijadikan jus dan dikeringkan menjadi teh. Pare mengandung senyawa fenolik yang berpotensi sebagai antioksidan. Kualitas produk olahan pare sangat dipengaruhi oleh tingkat kematangan pare itu sendiri. Pare berbentuk seperti mentimun dan berwarna hijau ketika masih muda, namun berubah menjadi kuning saat sudah matang. Masyarakat cenderung mengonsumsi pare yang belum matang. Menilai kematangan pare hanya dari penampilan luarnya tidaklah mudah, namun bisa dilakukan dengan melihat warna keseluruhan buah pare tersebut [2]. Namun, pemilihan sayuran pare masih menggunakan penilaian mata manusia memiliki kelemahan yaitu bersifat subyektif dan tidak konsisten sehingga tingkat akurasinya rendah. Sehingga, dibutuhkan metode otomatis yang dapat meningkatkan akurasi dengan penilaian yang konsisten dalam klasifikasi tingkat kematangan sayuran pare berdasarkan fitur warna. Salah satu teknik pengolahan warna citra untuk mengklasifikasi kematangan suatu objek dengan menggunakan nilai HVS (*Hue, Saturation, Value*). Nilai HVS didapatkan dari mengkonversi nilai RGB citra. Sistem HVS dipandang lebih dekat daripada sistem RBG dalam mendeskripsikan sensasi warna oleh mata manusia sehingga perbedaan warna akan semakin terlihat dan meningkatkan akurasi dari metode klasifikasi. Salah satu metode yang dapat digunakan dalam proses klasifikasi adalah k-nearest neighbor (K-NN). Metode K-NN merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan posisi suatu data dengan data-data lainnya, artinya jarak terdekat antara data yang dievaluasi dengan k tetangga terdekatnya pada data pelatihan [3]. Penelitian dengan menggunakan metode klasifikasi K-NN sudah dilakukan

sebelumnya. Objek penelitian berupa buah pinang dengan mengklasifikasi menjadi tiga kelas yaitu mentah, matang, dan tua. Penelitian ini menggunakan metode K-Nearest Neighbor (K-NN) dengan data sebanyak 600 menggunakan parameter RGB. Didapatkan akurasi terbaik dengan akurasi 98,33% dengan K terbaik adalah  $K = 7$  [4]. Penelitian lain yang terkait juga pernah dilakukan untuk mendeteksi tingkat kematangan buah pisang berdasarkan ciri warna kulit pisangnya. Data yang digunakan berupa citra/gambar pisang dengan mengambil fitur warna yaitu nilai RGB lalu dikonversi ke Hue, Saturation, dan Intensitas (HSI) dengan tingkat akurasi mencapai 85% [5]. Penelitian serupa juga pernah dilakukan dengan menggunakan cabai sebagai objek, yang bertujuan untuk mengklasifikasikan tingkat kematangan cabai. Penelitian ini menggunakan ekstraksi ciri nilai HSV, sedangkan untuk proses klasifikasinya menggunakan metode K-Nearest Neighbor yang dikombinasikan dengan logika fuzzy. Pengujian dilakukan terhadap 60 data cabai rawit, terdiri dari 15 cabai matang, 15 cabai mentah, 15 cabai setengah matang, dan 7 cabai busuk. Hasil penelitian ini memperoleh akurasi sebesar 86,66% [6]. Dari pemaparan studi literatur sebelumnya pengolahan citra mampu memberikan solusi dari permasalahan klasifikasi kematangan buah dengan memanfaatkan pengolahan citra digital dan kecerdasan buatan dengan adanya penelitian ini diharapkan dapat membantu petani pare pasca panen mendapatkan kualitas pare terbaik berdasarkan kematangannya.

## 2. Metode Penelitian

Pada sistem klasifikasi kematangan sayuran pare akan diklasifikasikan kedalam dua kelas yaitu mentah dan matang. Data citra pare yang diambil dari website roboflow milik Kubo Data terdiri sebanyak 160 data citra pare akan dibagi kedalam dua label yang terdiri dari 80 citra pare mentah dan 80 citra pare matang. Penelitian ini menggunakan 80% dari dataset sebagai data latih dan 20% dari dataset sebagai data uji. Dibawah ini adalah gambar diagram alir dari sistem klasifikasi sayuran pare.



Gambar 1. Diagram Alir Sistem

### 2.1 Pengumpulan Data

Pada tahapan awal penelitian ini memperoleh dataset citra sayuran pare sebanyak 1100 data citra sayuran pare tetapi hanya akan diambil sebanyak 160 citra pare yang terdiri dari 80 buah citra pare mentah dan 80 buah citra pare matang. Dataset sayuran pare didapatkan dari dataset publik yang dipublikasi oleh KUBO DATA melalui website Roboflow.

### 2.2 Preprocessing

Pada tahapan preprocessing diawali dengan memotong citra sayuran pare dengan mengambil titik tengah citra lalu membuat titik penentuan yang membentuk persegi sehingga nanti akan menghasilkan citra berukuran 50x50 piksel. Pemotongan citra ini menggunakan matriks dengan memanfaatkan *library Numpy*.

### 2.3 Ekstraksi Ciri HSV

Pada tahapan ekstraksi fitur akan dilakukan proses transformasi ruang warna dari RGB menjadi HSV menggunakan fungsi *cv2.cvtColor* pada *library OpenCV*. Berikut parafrasa dari penjelasan tentang HSV (Hue, Saturation, Value). HSV (Hue, Saturation, Value) merupakan representasi dari warna dasar dan distribusi dari panjang gelombang cahaya yang menentukan panjang gelombang dominan dari warna dasar tersebut. Dalam penentuan Hue digunakan satuan derajat, dengan ketetapan derajat pada warna dasar yaitu biru 240 derajat, hijau 120 derajat, dan merah 0 derajat. Saturation (S) menunjukkan intensitas warna, dimana nilai semakin tinggi membuat warna semakin cerah. Value (V) merupakan nilai kerapatan warna, pada warna-warna gelap nilai

kerapatannya rendah, sedangkan pada warna-warna cerah nilai kerapatannya tinggi [5]. Proses ini bertujuan untuk mendapatkan perbedaan warna yang lebih akurat dengan mendapatkan kanal warna *Hue*, *Saturation*, dan *Value* yang nanti nilai dari ketiga kanal warna tersebut akan dijadikan sebagai parameter untuk melakukan klasifikasi.

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right)$$
$$S = 1 - \frac{\min(R,G,B)}{V}$$
$$V = \frac{R+G+B}{3}$$

Gambar 2. Rumus Mengubah HSV

#### 2.4 Klasifikasi KNN

KNN (K-Nearest Neighbor) merupakan salah satu metode untuk melakukan pengelompokan atau klasifikasi dengan mengacu pada ciri-ciri dari tetangga terdekatnya. Kelebihan dari KNN adalah kemampuannya untuk mengeksekusi data pelatihan dalam jumlah besar. Metode klasifikasi ini cukup sederhana namun memberikan hasil yang cukup akurat [7]. Pada penelitian ini memanfaatkan algoritma KNN pada *library Scikit*. Nilai rerata H, S, V akan dijadikan sebagai data inputan. KNN menggunakan pendekatan *euclidean* mendapatkan hasil yang lebih baik, berikut adalah rumus *euclidean*:

$$d(xy) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

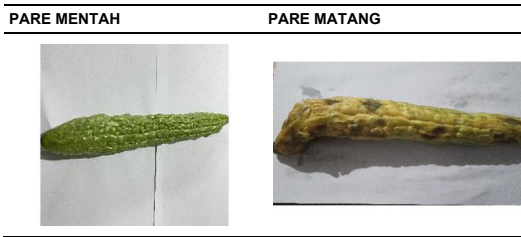
Keterangan:

- d = jarak kedekatan
- x = data training
- y = data testing
- n = jumlah atribut antara 1 s.d n
- i = atribut individu antara 1 s.d n

#### 3. Hasil dan Diskusi

Pada Sistem klasifikasi kematangan sayuran pare, penelitian ini menggunakan algoritma K-Nearest Neighbor untuk mengklasifikasi data dan menggunakan bahasa pemrograman python beserta *library seperti cv2, numpy, pyplot, pandas*, dan *scikit-learn* sebagai pendukung untuk mempermudah dalam penelitian kali ini. Tingkat kematangan sayuran pare yang digunakan adalah pare mentah dan pare matang. Berikut ini merupakan contoh tingkat kematangan sayuran pare yang digunakan dalam penelitian kali ini.





Gambar 3. Tingkat Kematangan Sayuran Pare

### 3.1 Dataset Sayuran Pare

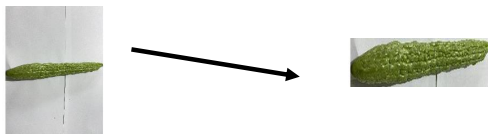
Penelitian ini memakai 160 data citra sayuran pare yang diambil dari Kubo Data melalui website Roboflow. Dari 160 data tersebut telah dikelompokkan kedalam dua label yaitu mentah dan matang dengan jumlah data yang seimbang. 80% dari dataset akan dipakai sebagai data latih dan 20% dari dataset akan digunakan sebagai data uji.



Gambar 4. Dataset Sayuran Pare

### 3.2 Tahap Preprocessing

Data citra yang telah dikumpulkan akan dilakukan cropping untuk menghilangkan bagian yang kurang penting pada image setelah itu dilakukan resize untuk mengubah rasio ukuran gambar sehingga mempermudah analisa gambar dan mengurangi ukuran penyimpanan yang digunakan. Berikut ini merupakan hasil dari cropping dataset.



**Gambar 5.** Hasil Proses Cropping

Setelah proses cropping dilakukan proses resize citra dari citra awal hasil cropping diubah dalam citra yang berukuran 50x50 pixel dengan menggunakan library OpenCV dalam melakukan resize citra.



**Gambar 6.** Hasil Proses Resize

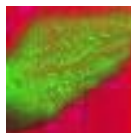
### 3.3 Ekstraksi Ciri HSV

Pada tahapan ini gambar akan diubah kedalam hsv untuk mendapatkan ekstraksi nilai objek gambar. Diawali dengan mengambil nilai gambar rgb yang diubah kedalam hsv lalu nilai dari setiap parameter yaitu hue, saturation, value dipisahkan untuk mendapatkan nilai dari setiap parameter dan dimasukkan kedalam tabel berbentuk csv.



**Gambar 7.** Gambar data dalam RGB

Dari gambar rgb tersebut akan diubah menjadi gambar hsv untuk mendapatkan informasi fitur dari data latih dan mendapatkan informasi apakah buah pare tersebut sudah matang atau masih mentah. Berikut ini merupakan gambar hasil hsv dari gambar diatas.



**Gambar 8.** Gambar data dalam HSV

Hasil dari ekstraksi gambar diatas maka diperoleh nilai dari objek gambar berupa hue, saturation, value yang termasuk kedalam kelompok data latih seperti tabel berikut ini seperti tabel berikut.

**Tabel 1.** Nilai Hue, Saturation, Value

H	S	V
414.184	960.504	1.433.056

Berikut ini merupakan 15 data ekstraksi ciri dari 32 data uji dimana data ini menampilkan nilai hue, saturation, value serta label yang bernilai 1 untuk pare yang matang dan bernilai 0 untuk pare yang mentah. Setelah itu data akan diolah menggunakan metode selanjutnya yaitu K-

Nearest Neighbor agar diketahui akurasi sistem klasifikasi ini menggunakan metode K-Nearest Neighbor.

**Tabel 2.** Contoh Ekstraksi Tabel Uji

H	S	V	Label
59.263.175	64.235.875	1.247.065	1
47.831.875	63.885.725	1.264.319	1
5.440.765	583.777	123.001.125	1
536.868	62.521.475	12.265.665	1
5.506.665	29.139.925	1.672.659	1
48.185.825	210.978	200.690.325	1
60.706.725	36.410.775	1.858.143	1
32.841.125	35.307.225	175.629	1
7.891.975	46.457.375	1.347.658	1
392.883	563.666	136.901.475	0
73.900.325	595.007	1.412.405	0
64.666	528.793	12.942.055	0
8.546.995	48.753.675	14.175.405	0
6.773.985	478.426	12.980.355	0
89.883.225	5.114.305	133.840.275	0
4.243.255	5.603.525	14.593.875	0

### 3.4 Klasifikasi KNN

Pada metode klasifikasi ini akan menggunakan 32 data uji dengan 16 data pare matang dan 16 pare mentah. Akurasi yang didapatkan dari masing-masing nilai k mempunyai rentang nilai yang sangat beragam. Berikut ini merupakan akurasi yang didapatkan berdasarkan masing-masing nilai k yang digunakan.

**Table 3.** Hasil Uji Coba Akurasi

Nilai K	Data yang Sesuai	Data Keseluruhan	Akurasi
3	28	32	88%
4	26	32	82%
5	25	32	78%
6	26	32	82%
7	23	32	72%
8	22	32	69%
9	22	32	69%
10	20	32	63%

Akurasi yang didapatkan dari hasil uji coba tersebut menunjukkan bahwa nilai tertinggi yang dapat diperoleh adalah 88% dengan menggunakan nilai  $k = 3$  sehingga acuan dalam melakukan klasifikasi sistem dengan metode K-Nearest Neighbor adalah  $k$  dengan nilai terbaik yaitu 3.

### 3.5 Hasil Akhir

Setelah diketahui bahwa nilai  $k = 3$  merupakan nilai terbaik dari hasil uji coba akurasi. Berikut akan ditampilkan hasil data uji sebanyak 32 data yang terdiri dari 16 data pare matang dan 16 data pare mentah dengan menggunakan metode K-Nearest Neighbor memakai nilai  $k = 3$ .

Tabel 4. Hasil 32 Data Uji

Data Tes	Kelas Asli	Hasil KNN	Informasi
2	Mentah	Mentah	Tepat
6	Mentah	Mentah	Tepat
8	Mentah	Mentah	Tepat
10	Mentah	Mentah	Tepat
11	Mentah	Mentah	Tepat
17	Matang	Matang	Tepat
19	Matang	Matang	Tepat
25	Matang	Matang	Tepat
27	Mentah	Matang	Tidak Akurat
29	Mentah	Mentah	Tepat
30	Mentah	Mentah	Tepat
33	Matang	Matang	Tepat
35	Matang	Matang	Tepat
36	Matang	Matang	Tepat
40	Matang	Matang	Tepat
43	Matang	Matang	Tepat
45	Matang	Matang	Tepat
59	Mentah	Mentah	Tepat
60	Mentah	Mentah	Tepat
68	Mentah	Mentah	Tepat
69	Mentah	Mentah	Tepat
85	Matang	Mentah	Tidak Akurat
88	Matang	Matang	Tepat
97	Mentah	Mentah	Tepat
101	Mentah	Mentah	Tepat
102	Mentah	Mentah	Tepat
103	Mentah	Mentah	Tepat
104	Matang	Mentah	Tidak Akurat

Data Tes	Kelas Asli	Hasil KNN	Informasi
111	Matang	Matang	Tepat
115	Matang	Matang	Tepat
125	Matang	Matang	Tepat
127	Matang	Mentah	Tidak Akurat

Setelah itu akan dilakukan perhitungan manual untuk menentukan akurasi dari hasil 32 data uji yang menggunakan k=3 dengan confusion matrix euclidean distance pada sebagai berikut.

Tabel 5. Confusion Matrix k = 3

Kelas Asli	Kelas Prediksi	
	Matang	Mentah
Matang	13	3
Mentah	1	15

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP}{Data} \times 100\% \\
 &= \frac{(13 + 15)}{32} \times 100\% = 88\%
 \end{aligned}$$

Jadi akurasi yang didapatkan dari perhitungan confusion matrix k=3 adalah sebesar 88%

#### 4. Kesimpulan

Berdasarkan hasil pengujian yang sudah dilakukan untuk mengklasifikasi kematangan sayuran pare dengan menggunakan metode K-Nearest Neighbor mendapatkan akurasi terbaik yaitu 88% untuk nilai k=3 dengan menggunakan data latih sebanyak 128 data dan data uji sebanyak 32 data dengan masing-masing kelas yakni kelas matang dan kelas mentah. Jadi berdasarkan ekstraksi ciri HSV dan metode K-Nearest Neighbor dapat diterapkan untuk mengklasifikasi sayuran pare dari tingkat kematangannya berdasarkan nilai dari Hue, Saturation, Value.

#### Daftar Pustaka

- [1] W. Gunadi, K. Sulandjari, and M. Syafii, "Respon Pertumbuhan dan Hasil Tanaman Pare (Momordica charantia L) Varietas Lipa F1 pada Berbagai Taraf Ketebalan Mulsa Jerami Padi," *Jurnal Agrotek Indonesia (Indonesian Journal of Agrotech)*, vol. 6, no. 2, pp. 62-69, 2021.
- [2] T. Saras, Pare: Sayuran Ajaib untuk Kesehatan Anda. Tiram Media, 2023.
- [3] F. Liantoni and F. N. Annisa, "Fuzzy K-Nearest Neighbor Pada Klasifikasi Kematangan Cabai Berdasarkan Fitur Hsv Citra," *JlPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 3, no. 2, 2018.
- [4] D. R. Febrinamas, R. Hidayati, and I. Nirmala, "Klasifikasi Buah Pinang Berdasarkan Data Sensor Menggunakan Metode K-Nearest Neighbor Berbasis Web," *Journal of Computer System and Informatics (JoSYC)*, vol. 4, no. 4, pp. 1046-1055, 2023.
- [5] M. Ismail, I. Indrabayu, and I. S. Areni, "Support Vector Machine Method to Reduce the Execution Time of Vehicle Plate Recognition System," *EPI International Journal of Engineering*, vol. 1, no. 1, pp. 69-75, 2018.
- [6] F. M. Sarimole and A. Rosiana, "Classification of Maturity Levels in Areca Fruit Based on HSV Image Using the KNN Method," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 4, no. 1, pp. 64-73, 2022.
- [7] E. Arisawati, "Penerapan K-Nearest Neighbor Berbasis Genetic Algorithm Untuk Penentuan Pemberian Kredit," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 1, no. 1, pp. 1-11, 2017.

# Analisis Sentimen Ulasan Aplikasi RedBus Menggunakan Metode SVM dan AdaBoost

Shelomita Putrinda Culio<sup>a1</sup>, Luh Gede Astuti<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>shelomitapc@gmail.com

<sup>2</sup>g.astuti@unud.ac.id (Corresponding Author)

## Abstract

*Bus transportation is a mode of transportation relied upon by the public due to its accessibility and affordable ticket prices. Sentiment analysis of RedBus app reviews on the Google Play Store can provide insight into user sentiment toward the app. The aims of this study are to analyze sentiment in reviews of the RedBus app using two approaches: the Random Forest model and a combination of Random Forest with AdaBoost. The analysis classifies user opinions as positive or negative. The study uses the TF-IDF method for feature extraction, and the evaluation methods include K-Fold Cross Validation and Confusion Matrix. The findings indicate that the combination of Random Forest with AdaBoost significantly enhances performance compared to the standard Random Forest model. Using a combination of Random Forest with AdaBoost results in an average accuracy of 89.0%, while the standard Random Forest model attains an average accuracy of 85.0%.*

**Keywords:** Sentiment Analysis, RedBus, Random Forest, AdaBoost, TF-IDF

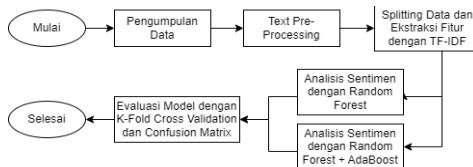
## 1. Pendahuluan

Transportasi publik memiliki peran yang penting dalam kehidupan masyarakat sehari-hari. Sebagai salah satu moda transportasi utama, bus cukup menjadi andalan masyarakat dalam berpergian, baik dalam kota maupun antarkota, khususnya karena alasan aksesibilitas yang luas dan harga tiket yang cukup terjangkau. Perkembangan era digital dan dorongan kebutuhan masyarakat terhadap layanan yang memberikan kemudahan dalam pemesanan tiket bus, menyebabkan mulai munculnya beberapa aplikasi pemesanan tiket bus dan *shuttle online*. Salah satu aplikasi pemesanan tiket bus *online* yang cukup populer di Indonesia adalah RedBus. Aplikasi RedBus menawarkan akses pada informasi lengkap mengenai jadwal, rute, harga tiket, dan pilihan pembayaran yang fleksibel. Pada layanan Google Play Store, aplikasi RedBus memiliki total unduhan yang cukup tinggi dengan lebih dari 50 juta unduhan dan 2 juta ulasan. Pandangan dan tingkat kepuasan pengguna dicurahkan dalam ulasan yang beragam. Ulasan pengguna Aplikasi RedBus pada Google Play Store tersebut dapat dianalisis dengan menggunakan teknik khusus, yaitu analisis sentimen, untuk mengklasifikasikan bagaimana sentimen masyarakat terhadap aplikasi. Analisis sentimen, juga dikenal sebagai *opinion mining*, merupakan studi komputasi yang menganalisis perilaku, pandangan, dan emosi yang terkandung dalam suatu teks [1]. Teknik ini bertujuan untuk mengkategorikan teks berdasarkan opini positif, negatif, atau netral. Pendekatan yang dapat digunakan untuk proses analisis sentimen antara lain *machine learning-based*, *lexicon-based*, dan *hybrid* [2]. Beberapa teknik atau metode *machine learning* yang cukup populer pada *text mining* adalah *Support Vector Machine*, *Random Forest*, *Decision Tree*, dan *Naive Bayes*. *Random Forest* merupakan metode dalam *machine learning* yang menyatukan banyak *decision tree* yang dilatih secara paralel untuk membuat suatu prediksi [3]. Permasalahan yang dapat muncul pada metode *machine learning* adalah memiliki tingkat akurasi yang relatif rendah [4]. Salah satu langkah yang dapat diambil untuk meningkatkan keakuratan hasil pengujian adalah menggunakan metode *ensemble*, khususnya algoritma *boosting*, yaitu metode yang menggabungkan beberapa model pembelajaran yang lebih lemah untuk menciptakan model yang lebih kuat [5]. Contoh dari algoritma *boosting* tersebut adalah

*AdaBoost* (*Adaptive Boost*). Penggunaan *AdaBoost* untuk meningkatkan akurasi algoritma analisis sentimen sudah dibuktikan pada penelitian terdahulu, seperti pada penelitian mengenai perbandingan algoritma *Multinomial Naive Bayes* dan *Decision Tree* pada analisis sentimen aplikasi PeduliLindungi [6]. Pengaplikasian *AdaBoost* mampu meningkatkan rata-rata akurasi *Multinomial Naive Bayes* dari 83,7% menjadi 88,8% dan akurasi rata-rata *Decision Tree* dari 82,8% menjadi 84,1%. Selain itu, pada penelitian lain mengenai klasifikasi data *mining* pada dataset diabetes yang *imbalance* menggunakan algoritma *K-Nearest Neighbor* [7], *AdaBoost* mampu meningkatkan akurasi tertinggi dari 92,90% menjadi 95,40%. Oleh karena itu, penelitian ini akan menerapkan algoritma *AdaBoost* bersama dengan metode *machine learning* lainnya. Penelitian berupa analisis sentimen ulasan aplikasi RedBus menggunakan dua pendekatan, yaitu metode *Random Forest* dan kombinasi *Random Forest* dengan *AdaBoost*. Perbandingan antara kedua pendekatan ini akan memberikan gambaran yang jelas mengenai dampak penggunaan *AdaBoost* pada kinerja algoritma *Random Forest*, terutama pada akurasi analisis sentimen ulasan.

## 2. Metode Penelitian

### 2.1. Tahapan Penelitian



Gambar 1. Tahapan Penelitian

Penelitian diawali dengan mengumpulkan data teks berlabel yang berisi ulasan atau komentar pengguna aplikasi RedBus. Data teks yang telah dikumpulkan kemudian diolah melalui *text preprocessing* dalam beberapa tahap, seperti *cleaning text*, *case folding*, *stopword removal*, dan lainnya. Data tersebut kemudian dibagi menjadi data *training* dan data *testing*. Data diekstraksi fitur menggunakan teknik TF-IDF. Kemudian, dilakukan pembangunan dua model analisis sentimen, yaitu model *Random Forest* dan model kombinasi *Random Forest* dengan *AdaBoost*. *Grid Search* digunakan untuk melakukan pencarian parameter terbaik untuk kedua model. Tahap terakhir adalah melakukan evaluasi terhadap kedua model dengan dengan *K-Fold Cross Validation*, *Confusion Matrix*, serta perhitungan *accuracy*, *precision*, *recall*, dan F-1 score untuk setiap *fold*. Seluruh tahapan penelitian ini akan diimplementasikan menggunakan bahasa pemrograman *Python* dan platform *Google Colab*. Tahapan keseluruhan penelitian ditunjukkan pada Gambar 1.

### 2.2. Pengumpulan Data

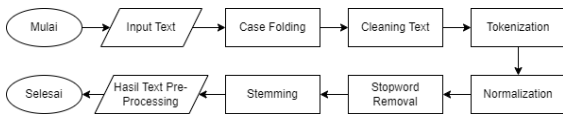
Data ulasan teks pengguna tentang aplikasi RedBus dalam penelitian ini merupakan data sekunder yang diperoleh dari platform Kaggle. Dataset tersebut berasal dari ulasan pengguna pada aplikasi RedBus di Google Play Store. Jumlah ulasan yang digunakan adalah sebanyak 500 ulasan yang terdiri dari 250 ulasan berlabel positif dan 250 ulasan berlabel negatif. Contoh bentuk dataset untuk penelitian ini terdapat pada Tabel 1.

Tabel 1. Contoh Bentuk Dataset

no	content	label
348	Bahaya aplikasi ini jauh dari realnya	Negatif
37	Ga sesuai pemberangkatan, Gajelas	Negatif
355	Sangat membantu dalam mencari tiket Bus Positif	

### 2.3. Text Preprocessing

Text preprocessing merupakan proses membersihkan dan memformat teks sehingga siap untuk digunakan atau dianalisis pada tahap selanjutnya [8]. Tahapan text preprocessing yang diaplikasikan dalam penelitian ini terlihat pada Gambar 2.



Gambar 2. Tahapan Preprocessing

Text preprocessing dimulai dengan *case folding*, yaitu teks diubah menjadi huruf kecil. Setelah itu, dilakukan *cleaning text* untuk menghilangkan karakter khusus, tanda baca, dan simbol-simbol yang tidak relevan. Teks kemudian dibagi menjadi token-token terpisah pada proses *tokenization*. Tahap selanjutnya adalah *normalization* untuk mengubah kata yang tidak sesuai menjadi ke dalam bentuk standar, seperti mengganti kata *slang* menjadi kata formal. *Stopword removal* merupakan proses menghilangkan kata-kata umum yang tidak memberikan informasi penting bagi analisis sentimen. Tahap terakhir adalah *stemming*, yaitu proses mengubah kata ke bentuk dasarnya dengan menghapus imbuhan. Teks yang telah melalui semua proses tersebut akan menjadi data yang lebih bersih dan terstruktur, sehingga siap untuk proses analisis sentimen.

### 2.4. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF merupakan sebuah pendekatan untuk membobotkan kata-kata dalam dokumen berdasarkan seberapa sering kata tersebut muncul dalam korpus teks. Konsep utama TF-IDF terdiri dari dua bagian, yaitu *Term Frequency* (TF) yang menilai frekuensi kemunculan sebuah kata dalam dokumen tertentu, dan *Inverse Document Frequency* (IDF) yang menilai kelangkaan kata tersebut di seluruh kumpulan dokumen (korpus) [9], seperti yang digambarkan pada persamaan (1).

$$TF - IDF(t, d, D) = \frac{jml \text{ muncul kata } t \text{ dalam dokumen } d}{jml \text{ total kata dalam dokumen } d} \times \log \left( \frac{\text{Total dokumen dalam } D}{jml \text{ dokumen yang mengandung } t} \right) \quad (1)$$

Dimana *t* adalah kata yang ingin dihitung frekuensi kemunculannya, *d* adalah dokumen tempat kata *t* muncul, dan *D* adalah jumlah total dokumen. Pada analisis sentimen, kata-kata penting dan unik dalam dokumen akan memiliki bobot yang besar, sedangkan kata-kata umum dan sering muncul akan memiliki bobot yang lebih kecil. Proses ekstraksi fitur menggunakan TF-IDF dapat membantu untuk fokus pada kata-kata yang relevan dan mengabaikan kata yang tidak relevan dalam analisis sentimen.

### 2.5. Random Forest

Random Forest merupakan metode *machine learning* yang menyatukan banyak *decision tree* untuk menciptakan prediksi yang lebih akurat. Algoritma ini melibatkan pengambilan sampel acak dengan penggantian (*bootstrap*) dari data *training* untuk membangun *decision tree* secara terpisah. Setiap *tree* dilatih menggunakan subset acak dari fitur yang tersedia pada setiap simpul.



Prediksi akhir dihitung sebagai rata-rata (untuk regresi) atau voting mayoritas (untuk klasifikasi) dari semua *tree* [3]. Metode ini menggunakan indeks Gini untuk menentukan pemisahan terbaik pada setiap simpul pohon keputusan, yaitu dengan mengukur tingkat ketidakseimbangan distribusi kelas pada data seperti pada persamaan (2). Pemisahan yang optimal adalah yang meminimalkan nilai indeks tersebut [10].

$$Gini = 1 - \sum_{i=1}^k (P_i)^2 \quad (2)$$

Keterangan:

$k$  = jumlah kelas,  $P_i$  = proporsi kelas  $i$  dalam data

Dalam penelitian ini, *Random Forest* digunakan untuk mengklasifikasikan teks berdasarkan sentimen positif dan negatif dengan melakukan pembelajaran pada data training berlabel, yaitu ulasan teks, dan membuat prediksi pada data baru. Beberapa parameter dapat disesuaikan untuk mengoptimalkan performa model algoritma ini, seperti  $n\_estimators$  yang merupakan jumlah *decision tree* yang akan dibangun,  $max\_depth$  yang merupakan kedalaman maksimum dari setiap *tree*, dan  $random\_state$  yang merupakan pengontrol proses pengacakan dalam pengambilan sampel.

## 2.6. AdaBoost (Adaptive Boost)

*AdaBoost (Adaptive Boost)* merupakan algoritma *ensemble* yang umum digunakan sebagai algoritma *boosting*. Algoritma ini meningkatkan kinerja model klasifikasi dengan cara menggabungkan beberapa model yang lemah (*weak classifier*) menjadi model yang lebih kuat (*strong classifier*). Konsep kerja algoritma ini melibatkan pembobotan data *training* berdasarkan kesalahan klasifikasi sebelumnya, yaitu data yang salah diklasifikasi akan mendapatkan bobot lebih tinggi sehingga pada pelatihan model baru selanjutnya ditekankan pada data yang sulit diklasifikasikan [11]. Proses ini akan dilakukan hingga jumlah iterasi tertentu, menyesuaikan dengan parameter  $n\_estimators$  yang digunakan. Dalam analisis sentimen, *AdaBoost* dapat bermanfaat untuk mengurangi risiko *overfitting* dan meningkatkan akurasi prediksi dengan mengatasi kesulitan dalam data teks yang kompleks atau tidak seimbang.

## 2.7. K-Fold Cross Validation

*K-Fold Cross Validation* merupakan prosedur pengambilan sampel ulang yang digunakan untuk melakukan evaluasi model *machine learning* menggunakan sampel data yang terbatas. Metode ini bekerja dengan membagi data menjadi  $k$  *fold*, tiap *fold* berfungsi sebagai data *training* dan *testing* secara bergantian. Maka, setiap data akan memiliki kesempatan untuk digunakan dalam set *testing* sekali dan dalam set *training*  $k-1$  kali. *K-Fold Cross Validation* berguna untuk mengurangi risiko bias dan memberikan evaluasi model yang lebih andal [12]. Nilai  $k$  yang sering digunakan adalah  $k=5$  dan  $k=10$ . Dalam penelitian ini, penggunaan *K-Fold Cross Validation* dilakukan untuk memperoleh hasil evaluasi analisis sentimen yang lebih objektif dan stabil.

## 2.8. Confusion Matrix

*Confusion Matrix* merupakan metode evaluasi yang diterapkan untuk menggambarkan performa suatu metode klasifikasi. Matriks ini memperlihatkan hasil prediksi model dibandingkan dengan kondisi aktual dari data yang diamati [13]. *Confusion Matrix* memiliki 4 elemen utama seperti pada Tabel 2.

**Tabel 2.** Confusion Matrix

		Prediction	
		True Positive (TP)	False Negative (FN)
Actual	True Positive (TP)		
	False Positive (FP)		True Negative (TN)

Berdasarkan 4 elemen tersebut, beberapa metrik evaluasi dapat dihitung, yaitu *accuracy* yang mengindikasikan sejauh mana data telah diklasifikasikan dengan tepat dari seluruh data yang digunakan seperti pada persamaan (3), *precision* yang mengukur proporsi prediksi positif yang benar seperti pada persamaan (4), *recall* yang mengukur kemampuan model untuk mendeteksi kasus positif seperti pada persamaan (5), dan *F-1 score* yang memperlihatkan gambaran kinerja yang seimbang antara *precision* dan *recall* seperti pada persamaan (6) [13].

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \times 100\% \quad (3)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \quad (5)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (6)$$

### 3. Hasil dan Diskusi

#### 3.1. Text Preprocessing

Tahap *text preprocessing* yang diterapkan pada 500 data ulasan (250 label positif dan 250 label negatif) dilakukan menggunakan bahasa pemrograman *Python*. Dataset dinormalisasi pada tahap *normalization* menggunakan file *colloquial-indonesian-lexicon.csv* yang diperoleh dari GitHub untuk menyelaraskan format dan konten data. Proses *stopword removal* dilakukan dengan menggunakan *library* NLTK dan proses *stemming* menggunakan *library* Sastrawi. Selanjutnya dilakukan perhitungan persentase tanda baca (*punct*) dan perhitungan panjang teks (*review\_len*) untuk setiap data ulasan. Hasil dari tahap *text preprocessing* tersaji dalam Tabel 3.

**Tabel 3.** Hasil *Text Preprocessing*

Tahap Pre-Processing Content	
Initial data	Mantap, banyak potongan harga tiket, banyak pilihan buss
Case Folding	mantap, banyak potongan harga tiket,banyak pilihan buss
Cleaning Text	mantap banyak potongan harga tiket banyak pilihan buss
Tokenization	['mantap', 'banyak', 'potongan', 'harga', 'tiket', 'banyak', 'pilihan', 'buss']
Normalization	['mantap', 'banyak', 'potongan', 'harga', 'tiket', 'banyak', 'pilihan', 'bus']
Stopword Removal	['mantap', 'potongan', 'harga', 'tiket', 'pilihan', 'bus']
Stemming	['mantap', 'potong', 'harga', 'tiket', 'pilih', 'bus']

Jumlah data yang dihasilkan setelah melalui *text preprocessing* adalah tetap sejumlah 500 ulasan. Selanjutnya, label data diubah dari bentuk teks menjadi data nominal dengan proses *mapping*, yaitu label negatif diubah menjadi 0 dan label positif diubah menjadi 1. Hasil perubahan bentuk label ulasan terlihat pada Tabel 4.

**Tabel 4.** Tabel Hasil Label Mapping

content	label
['aplikasi', 'bantu', 'cari', 'beli', 'tiket', 'bus']	1
['aplikasi', 'sampah', 'jam', 'berangkat', 'tidak', 'sesuai']	0
['bayar', 'loading', 'ulang', 'gagal']	0

**3.2. Ekstraksi Fitur**

Setelah melalui tahap *text preprocessing*, data di *split* untuk dialokasikan sebagai data *training* sebanyak 80% dan data *testing* sebanyak 20%. Data ulasan berlabel positif terbagi menjadi 200 data *training* dan 50 data *testing*, begitu pula dengan data ulasan berlabel negatif terbagi menjadi 200 data *training* dan 50 data *testing*.

Setelah itu, dilanjutkan ekstraksi fitur terhadap data ulasan menggunakan TF-IDF. Fitur yang diekstraksi menggunakan TF-IDF akan digunakan sebagai masukan untuk model analisis sentimen. Tahap ekstraksi fitur ini menggunakan bahasa pemrograman *Python* dengan *library* TF-IDF dari *Scikit-Learn* berupa modul *TfidfVectorizer*. Total jumlah fitur yang telah diekstrak adalah sebanyak 400 fitur dari total 200 dokumen. Hasil tahap ekstraksi fitur diperlihatkan pada Gambar 3 dan 4.

```

Hasil kata ekstraksi fitur dengan TF-IDF:
'ada', 'admin', 'adminnya', 'adu', 'agen', 'aju', 'akibat', 'akses', 'aktif',
'akun', 'akurat', 'alam', 'alas', 'alhamdulillah', 'aman', 'an', 'anak', 'apk',
'aplikasi', 'app', 'april', 'arah', 'armada', 'asih', 'asli', 'baca', 'bagus',
'baik', 'belas', 'banding', 'bandung', 'banget', 'bantu', 'banyak', 'batal', 'batas',
'bayar', 'bebas', 'beda', 'bekas', 'belah', 'beli', 'berangkat', 'bersih', 'besar',
'blar', 'bikin', 'bilang', 'bintang', 'biru', 'bis', 'bocor', 'bohong', 'booking',
'bookingnya', 'bukti', 'buruk', 'bus', 'busuk', 'butuh', 'cacat', 'call', 'cancel',
'cari', 'centen', 'cepat', 'cetak', 'cikampak', 'cirebon', 'class', 'coba', 'cocok',
'contoh', 'crew', 'cs', 'cuman', 'customer', 'daftar', 'dah', 'dana', 'dasar',
'deh', 'depan', 'dgan', 'dibalikin', 'dicancel', 'dikabarin', 'diskon', 'dll', 'doang',
'download', 'duduk', 'duit', 'eh', 'email', 'enak', 'enggak', 'eror', 'error',
'evaluasi', 'executive', 'fasilitas', 'fast', 'feedback', 'friendly', 'gagal', 'gampang',
'gara', 'gila', 'gilir', 'good', 'gue', 'guna', 'habis', 'handoyo', 'hangus',
'hapus', 'harap', 'harga', 'hasil', 'hati', 'hilang', 'hubung', 'iklan', 'indomaret',
'indonesia', 'info', 'informasi', 'install', 'jadi', 'jadwal', 'jakarta', 'jalan', 'jalan',
'jam', 'jamin', 'jangkau', 'jaya', 'jelas', 'jelek', 'jemput', 'jual', 'jurus',
'kabar', 'kacau', 'kadaluarsa', 'kadang', 'kait', 'kali', 'kalideres', 'kalo', 'kampung',
'kantor', 'kapok', 'karawang', 'kasih', 'kayak', 'kecewa', 'keluh', 'kemarin', 'kembali',
'kendala', 'kenen', 'kenja', 'kesini', 'ketik', 'kinin', 'klik', 'kode', 'kolom'
    
```

**Gambar 3.** Hasil Ekstraksi Fitur dengan TF-IDF

```

Review_len punct ada admin adminnya adu agen aju akibat \
374 113 2.7 0.0 0.000000 0.0 0.0 0.0 0.0 0.0
302 32 0.0 0.0 0.000000 0.0 0.0 0.0 0.0 0.0
362 155 0.6 0.0 0.304804 0.0 0.0 0.0 0.0 0.0
106 201 1.5 0.0 0.000000 0.0 0.0 0.0 0.0 0.0
364 75 1.3 0.0 0.000000 0.0 0.0 0.0 0.0 0.0

akses aktif akun akurat alam alas alhamdulillah aman an anak \
374 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
302 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
362 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
106 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
364 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

apk aplikasi app april arah armada asih asli baca bagus baik \
374 0.0 0.264281 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
302 0.0 0.337140 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
362 0.0 0.114752 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
106 0.0 0.358108 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
364 0.0 0.000000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    
```

**Gambar 4.** Contoh Hasil Vektorisasi Fitur Beberapa Data

### 3.3. Pemodelan Analisis Sentimen *Random Forest* dan *AdaBoost*

Pada tahap analisis sentimen, dilakukan pembuatan 2 model, yaitu model *Random Forest* dan model *Random Forest* dengan penggunaan *AdaBoost*. Model *Random Forest* dibangun menggunakan bahasa pemrograman *Python* dengan menggunakan modul *RandomForestClassifier* dari library *Scikit-learn*. Evaluasi *training* model dilakukan dengan teknik *K-Fold Cross Validation* dan menggunakan jumlah *fold*  $k=5$  untuk memastikan kinerja model konsisten. Pada model ini, *Grid search* dengan modul *GridSearchCV* digunakan untuk mendapatkan parameter terbaik yang akan digunakan. Parameter terbaik yang digunakan dalam model *Random Forest* meliputi jumlah *tree* ( $n\_estimators=100$ ), kedalaman maksimum *tree* ( $max\_depth=None$ ), dan  $random\_state=0$ . Hasil evaluasi 80% data *training* untuk setiap *fold* dari model *Random Forest* ditampilkan pada Tabel 5. Akurasi terbaik yang dihasilkan pada model *Random Forest* adalah sebesar 90,0% dengan hasil akurasi rata-rata sebesar 87,0%, sedangkan nilai rata-rata *precision* sebesar 87,1%, *recall* sebesar 87,5%, dan *F1-score* 87,0%.

Tabel 5. Hasil Evaluasi *K-Fold Cross Validation* Model *Random Forest*

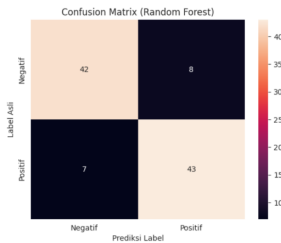
N-Fold	Accuracy	Precision	Recall	F1-Score
1	0.8875	0.897436	0.875	0.886076
2	0.8750	0.857143	0.900	0.878049
3	0.8500	0.850000	0.850	0.850000
4	0.8375	0.846154	0.825	0.835443
5	0.9000	0.880952	0.925	0.902439
Average	0.870	0.871	0.875	0.870

Sementara itu, hasil *testing* model *Random Forest* dengan menggunakan parameter terbaik tersebut pada 20% data *testing* mendapatkan hasil seperti pada tabel 6. Akurasi rata-rata yang didapatkan adalah sebesar 85%, dengan nilai rata-rata *precision* sebesar 85%, *recall* sebesar 85%, dan *F1-score* 85%.

Tabel 6. Hasil *Testing* Model *Random Forest*

	Accuracy	Precision	Recall	F1-Score
Negatif	0.85	0.86	0.84	0.85
Positif	0.85	0.84	0.86	0.85
Average	0.85	0.85	0.85	0.85

Kemudian, pada hasil evaluasi menggunakan *Confusion Matrix* didapatkan hasil bahwa model berhasil memprediksi dengan benar 43 ulasan positif dan salah menebak 7 ulasan positif menjadi negatif. Kemudian model berhasil memprediksi dengan benar 42 ulasan negatif dan salah menebak 8 ulasan negatif menjadi positif. Hasil *Confusion Matrix* ditampilkan pada Gambar 5.



**Gambar 5.** Hasil Evaluasi Confusion Matrix Model Random Forest

Pada model kedua, yaitu kombinasi *Random Forest* dengan algoritma *AdaBoost*, menggunakan modul *AdaBoostClassifier* dan *RandomForestClassifier* dari *library* Scikit-learn. Evaluasi model dilakukan dengan teknik *K-Fold Cross Validation* dengan jumlah *fold* yang digunakan adalah  $k=5$ . Pada model ini, *Grid search* dengan modul *GridSearchCV* digunakan untuk mendapatkan parameter terbaik yang akan digunakan. Model menggunakan *base estimator Random Forest* dengan parameter jumlah estimator ( $n\_estimator=50$ ), kedalaman maksimum *tree* ( $max\_depth=5$ ), dan *random state=0*. Kemudian, model *AdaBoost* dibangun dengan *base estimator Random Forest*, parameter tingkat pembelajaran ( $learning\_rate=0.1$ ), dan jumlah estimator ( $n\_estimator=20$ ). Hasil evaluasi 80% data *training* untuk setiap *fold* dari model *Random Forest* dengan *AdaBoost* ditampilkan pada Tabel 7. Akurasi terbaik yang dihasilkan pada model *Random Forest* dengan *AdaBoost* adalah sebesar 93,75% dengan hasil akurasi rata-rata sebesar 90,75%, sedangkan nilai rata-rata *precision* sebesar 91,22%, *recall* sebesar 90,0%, dan *F1-score* 90,64%.

**Tabel 7.** Hasil Evaluasi *K-Fold Cross Validation* Model *Random Forest* dengan *AdaBoost*

N-Fold	Accuracy	Precision	Recall	F1-Score
1	0.9250	0.972222	0.875	0.921053
2	0.9250	0.947368	0.900	0.923077
3	0.9000	0.900000	0.900	0.900000
4	0.8500	0.850000	0.850	0.850000
5	0.9375	0.926829	0.950	0.938272
<b>Average</b>	<b>0.9075</b>	<b>0.9122</b>	<b>0.90</b>	<b>0.9064</b>

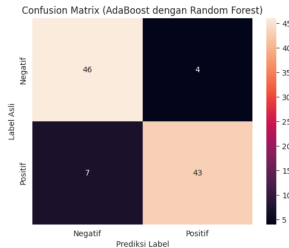
Sementara itu, hasil *testing* model *Random Forest* dengan *AdaBoost* menggunakan parameter terbaik tersebut pada 20% data *testing* mendapatkan hasil seperti pada tabel 8. Akurasi rata-rata yang didapatkan adalah sebesar 89%, dengan nilai rata-rata *precision* sebesar 89%, *recall* sebesar 89%, dan *F1-score* 89%.

**Tabel 8.** Hasil *Testing* Model *Random Forest*

	Accuracy	Precision	Recall	F1-Score
Negatif	0.89	0.87	0.92	0.89
Positif	0.89	0.91	0.86	0.89
<b>Average</b>	<b>0.89</b>	<b>0.89</b>	<b>0.89</b>	<b>0.89</b>

Selanjutnya, pada hasil evaluasi menggunakan *Confusion Matrix* didapatkan hasil bahwa model berhasil memprediksi dengan benar 43 ulasan positif dan salah menebak 7 ulasan positif menjadi

negatif. Kemudian model berhasil memprediksi dengan benar 46 ulasan negatif dan salah menebak 4 ulasan negatif menjadi positif. Hasil *Confusion Matrix* ditampilkan pada Gambar 6.



**Gambar 6.** Hasil Evaluasi *Confusion Matrix* Model *Random Forest* dengan *AdaBoost*

Berdasarkan hasil evaluasi yang didapatkan dari kedua model tersebut, didapatkan perbandingan nilai rata-rata *accuracy*, *recall*, *precision*, dan F1-score antara kedua model yang ditampilkan pada Tabel 9. Data pada tabel tersebut menunjukkan bahwa model kombinasi *Random Forest* dengan *AdaBoost* menghasilkan nilai rata-rata *accuracy*, *recall*, *precision*, dan F1-score yang lebih tinggi dibandingkan model *Random Forest* biasa.

**Tabel 9.** Perbandingan Kinerja Rata-Rata Model *Random Forest* dan *Random Forest + AdaBoost*

	Accuracy	Precision	Recall	F1-Score
<b>RF</b>	0.85	0.85	0.85	0.85
<b>RF + AB</b>	0.89	0.89	0.89	0.89

#### 4. Kesimpulan

Berdasarkan temuan dari penelitian yang telah dilakukan untuk membandingkan kinerja model *Random Forest* dan model kombinasi *Random Forest* dengan *AdaBoost* dalam analisis sentimen ulasan aplikasi RedBus, dapat disimpulkan bahwa pendekatan kombinasi *Random Forest* dengan *AdaBoost* memberikan peningkatan kinerja yang cukup signifikan. Model *Random Forest* dengan *AdaBoost* berhasil mencapai akurasi rata-rata sebesar 89,0%, sedangkan akurasi rata-rata model *Random Forest* biasa adalah 85,0%. Model kombinasi tersebut juga menunjukkan peningkatan kinerja pada nilai *precision* yang mencapai 89,0%, *recall* 89,0%, dan F1-score 89,0%. Saran untuk penelitian selanjutnya dapat mengeksplorasi model kombinasi lain, seperti menggunakan algoritma *boosting* yang berbeda atau model *ensemble* lain. Selain itu, juga dapat mempertimbangkan penggunaan metode *deep learning* untuk melakukan perbandingan kinerja dengan model yang diterapkan dalam penelitian ini.

#### Daftar Pustaka

- [1] B. Saberi and S. Saad, "Sentiment Analysis or Opinion Mining: A Review," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 5, pp. 1660-1666, 2017.
- [2] S. Thomas, Yuliana and N. P., "Studi Analisis Metode Analisis Sentimen pada YouTube," *JIFOTECH (Journal of Information Technology)*, vol. 1, no. 1, pp. 1-7., 2021.
- [3] S. Rigatti, "Random Forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31-39, 2017.

- [4] E. K. Adhitya, R. Satria and Subagyo, "Komparasi Metode Machine Learning dan Non-Machine Learning untuk Estimasi Usaha Perangkat Lunak," *Journal of Software Engineering*, vol. 1, no. 2, pp. 109-113, 2015.
- [5] M. B. Hamzah, "Classification of Movie Review Sentiment Analysis Using Chi-Square and Multinomial Naïve Bayes with Adaptive Boosting," *Journal of Advances in Information Systems and Technology*, vol. 3, no. 1, pp. 67-74, 2021.
- [6] C. M. S. Ramdani, A. N. Rachman and R. Setiawan, "Comparison of the Multinomial Naive Bayes Algorithm and Decision Tree with the Application of AdaBoost in Sentiment Analysis Reviews PeduliLindungi Application," *IJISTECH (International Journal of Information System and Technology)*, vol. 6, no. 4, pp. 419-430, 2022.
- [7] N. Novianti, M. Zarlis and P. Sihombing, "Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes," *Jurnal Media Informatika Budidarma*, vol. 6, no. 2, pp. 1200-1206, 2022.
- [8] A. I. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, no. 6, pp. 22-32, 2018.
- [9] S. Kalra, L. Li and H. R. Tizhoosh, "Automatic Classification of Pathology Reports using TF-IDF Features," *arXiv preprint arXiv:1903.07406*, pp. 1-9, 5 March 2019.
- [10] B. Warsito and A. Prahutama, "Sentiment Analysis on Tokopedia Product Online Reviews using Random Forest Method," in *the 5th International Conference on Energy, Environmental and Information System (ICENIS 2020)*, Semarang, 2020.
- [11] S. I. Gultom, "Implementasi Data Mining Menentukan Pola Hidup Sehat Bagi Pengguna KB Menggunakan Algoritma Adaboost (Studi Kasus: Dinas Serdang Bedagai)," *Informasi dan Teknologi Ilmiah (INTI)*, vol. 7, no. 3, pp. 298-304, 2020.
- [12] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," *Machine Learning Mastery*, 4 Oktober 2023. [Online]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/>. [Accessed 5 May 2024].
- [13] R. Arthana, "Mengenal Accuracy, Precision, Recall dan Specificity serta yang Diprioritaskan dalam Machine Learning," *Medium*, 5 April 2019. [Online]. Available: <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>. [Accessed 5 May 2024].

## Analisis Sentimen pada Ulasan Aplikasi myIM3 Menggunakan *Multinomial Naive Bayes* dengan TF-IDF

Ni Komang Ayu Juliana<sup>a1</sup>, Made Agung Raharja<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>juliana.2208561046@student.unud.ac.id  
<sup>2</sup>made.agung@unud.ac.id

### Abstract

*The digital service adoption in Indonesia has emerged as a primary trend to meet the needs of the millennial generation, seeking greater convenience and speed. Amidst this trend, self-service apps like MyIM3 by Indosat Ooredoo Hutchison have become a trusted solution for users to manage their services more efficiently. Sentiment analysis is crucial for understanding user responses to such apps. This study employs the Multinomial Naive Bayes algorithm with hyperparameter alpha 0.8 and TF-IDF to analyze sentiment towards user reviews on Google Play Store for MyIM3. The dataset, sourced from Kaggle, consists of 8475 reviews, pre-processed and labeled to 8212 reviews. Model evaluation with an 80:20 split reveals an overall accuracy of 89%, with a precision of 86% for negative (0) and 93% for positive (1) labels. The recall for negative is 95% and positive is 81%. Thus, this research contributes to understanding user perspectives on MyIM3 and provides a basis for enhancing the quality of app-based services.*

**Keywords:** *Multinomial Naive Bayes, myIM3, TF-IDF, Sentiment Analysis*

### 1. Pendahuluan

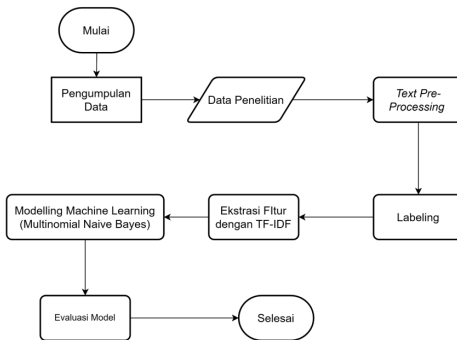
Perkembangan teknologi saat ini terus menawarkan kemudahan yang signifikan, termasuk dalam bidang telekomunikasi, informasi, dan transaksi pembelian. Fenomena ini tercermin dalam adopsi layanan digital di Indonesia, yang menjadi salah satu tren utama dalam memenuhi kebutuhan generasi milenial yang menginginkan segalanya menjadi lebih praktis dan cepat. Dukungan untuk tren ini terlihat dalam peningkatan jumlah pengguna internet, seperti yang dilaporkan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII). Berdasarkan survei terbaru mereka tahun 2023, total pengguna internet sekitar 215 juta orang yang menunjukkan peningkatan signifikan dari tahun sebelumnya sebesar 1,17% [1]. Di tengah dinamika ini, perusahaan penyedia layanan jaringan telekomunikasi, seperti Indosat Ooredoo Hutchison, terus berinovasi untuk memenuhi kebutuhan pelanggan [2]. Salah satu inovasi mereka adalah aplikasi self-service bernama MyIM3, yang telah menjadi solusi yang sangat diandalkan bagi pengguna untuk mengelola layanan mereka dengan lebih efisien. Dengan lebih dari 50 juta unduhan dan lebih dari 9 juta ulasan di *Google Play Store*, MyIM3 menjadi salah satu aplikasi layanan provider yang terkenal di pasar. Oleh karena itu, ulasan dan penilaian pengguna di platform seperti *Google Play Store* menjadi sangat penting, karena dapat mempengaruhi persepsi pengguna potensial tentang kualitas dan kegunaan suatu aplikasi. Dalam konteks ini, analisis sentimen menjadi alat yang efektif untuk memahami respons pengguna terhadap aplikasi MyIM3. Algoritma *Multinomial Naive Bayes* dipilih karena kecepatan dan kesederhanaannya dalam mengklasifikasikan data teks, serta kemampuannya untuk menangani distribusi frekuensi kata yang besar. Metode TF-IDF (*Term Frequency-Inverse Document Frequency*) digunakan sebagai metode pembobotan kata untuk mengurangi dampak kata-kata yang sering muncul namun kurang informatif, sambil memberikan penekanan pada kata-kata yang lebih jarang tetapi mungkin lebih penting dalam konteks sentimen. Dengan demikian, ini meningkatkan akurasi klasifikasi dengan memberikan bobot lebih pada kata-kata yang lebih signifikan dalam menentukan sentimen ulasan. Terdapat beberapa penelitian terdahulu yang membahas mengenai analisis sentimen menggunakan *Multinomial Naive Bayes* dan TF-IDF seperti misalnya pada penelitian mengenai Analisis



Sentimen terhadap Produk Skin *Game* di Forum *Review Female Daily* Menggunakan Metode *Multinomial Naive Bayes* dan TF-IDF yang mendapatkan hasil evaluasi sebesar 86,6%, *precision* 89,1%, dan *recall* 94,2% [3]. Selain itu terdapat Selain itu, terdapat penelitian lain mengimplementasikan algoritma *Multinomial Naive Bayes* dalam analisis sentimen opini pengguna aplikasi Brimo berbobot TF-IDF yang mendapatkan hasil evaluasi *confusion matrix* terbaik pada *fold-2* dengan nilai akurasi 98,02%, presisi 97,06%, *recall* 97,06%, dan *f1-score* 97,06% [4]. Terdapat juga penelitian yang berjudul Analisis Sentimen Opini Pengguna Twitter Pada Aplikasi Bibit Menggunakan *Multinomial Naive Bayes* yang memperoleh akurasi sebesar 91.50% dan tingkat akurasi validasi sebesar 85.35% [5]. Berdasarkan hasil studi sebelumnya, penelitian ini melakukan analisis sentimen pada ulasan pengguna di Google Play Store untuk aplikasi myIM3 menggunakan metode *Multinomial Naive Bayes*. Metode *Multinomial Naive Bayes* dengan pembobotan TF-IDF ini digunakan untuk memilah ulasan ke dalam kategori positif dan negatif dengan lebih presisi. Dengan mendapatkan informasi yang lebih akurat, diharapkan hasilnya dapat digunakan sebagai panduan untuk menjaga kualitas layanan, memperbaiki kekurangan, dan meningkatkan evaluasi ke depan untuk aplikasi myIM3.

## 2. Metode Penelitian

Dalam penelitian ini, beberapa langkah akan dilakukan untuk menganalisis sentimen dari ulasan di Google Play Store terhadap Aplikasi myIM3 menggunakan metode *Multinomial Naive Bayes* dan TF-IDF (*Term Frequency-Inverse Document Frequency*). Langkah-langkah tersebut dimulai dengan mengumpulkan data, melakukan *text pre-processing* pada data, melabelkan data yang sudah di *pre-processing*, mengekstraksi fitur menggunakan TF-IDF, kemudian memodelkan data menggunakan algoritma *Multinomial Naive Bayes*, dan terakhir mengevaluasi model yang dihasilkan menggunakan *confusion matrix*. Berikut adalah alur penelitian yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

### 2.1 Pengumpulan Data

Teknik pengumpulan data dalam penelitian dapat beragam, seperti menggunakan data yang tersedia secara umum di berbagai platform online, melakukan web scraping untuk mengambil data dari internet, dan menggabungkan beberapa dataset tertentu melalui data augmentation [6]. Dataset untuk penelitian ini diperoleh dari data publik yang berasal dari situs web Kaggle yang diunggah oleh pengguna dengan nama pengguna Ari Zidane. Kaggle adalah platform yang menyediakan berbagai dataset untuk keperluan penelitian dan pembelajaran. Dataset di Kaggle

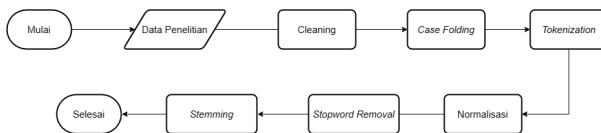
berisi data yang beragam yang telah dikumpulkan dan diproses oleh pengguna lain untuk berbagai tujuan penelitian. Dataset ini terdiri dari 8475 ulasan yang mencakup nama pengguna, skor atau rating, tanggal pembuatan ulasan, dan konten ulasan itu sendiri. Dataset ini terdiri dari 8475 ulasan yang berisi username, score atau rating, tanggal pembuatan ulasan, dan juga content atau ulasannya seperti pada Tabel 1.

Tabel 1. Dataset myIM3

userName	content	score	at
Nurul Aini	klo gk ada data nya gk bisa di buka	2	31/01/2024 23:56:00
Salsa Bila	bagus	5	31/01/2024 23:50:00
JebolGamingYT	Aku beli pulsa. Beli paket data. Di proses terus lama.. Sudah 1 jam. Biasaya cepet. Kok ini lama.. Segeraaa	1	31/01/2024 23:42:00
AREK BERBEK 1H	Apa'an sekarang im3 error bkin ribet.pas beli kartu perdana baru tapi disuruh beli pulsa.hbis beli pulsa harus ada registrasi dulu. Registrasi di WA harus ada kode otp dulu di SMS. Tapi g ada notifnya. Sumpah apaan. Pusing	1	31/01/2024 23:38:00
Denik Rahayu	bagus	5	31/01/2024 23:30:00

## 2.2 Text Pre-Processing

Tahap selanjutnya adalah pre-processing data adalah langkah awal di mana data disiapkan dan dibersihkan untuk melakukan analisis. Untuk tahapan dari pre-processing dapat dilihat pada Gambar 2.



Gambar 2. Alur Pre-Processing

Terdapat beberapa tahapan yang dilakukan pada *text pre-processing*, berikut adalah penjelasannya:

- Cleaning*, pada tahapan ini pembersihan data dari unsur-unsur yang tidak diperlukan atau berlebihan, seperti data kosong, karakter khusus, dan format yang tidak konsisten. Pada proses ini juga di cek apakah data yang didapatkan terdapat ulasan yang kosong atau tidak apabila ada, maka data tersebut akan dibuang.
- Case Folding* adalah proses dimana semua huruf dalam teks diubah menjadi huruf kecil untuk mencegah adanya duplikasi kata yang sama yang mungkin dianggap berbeda karena perbedaan kapitalisasi.

- c. *Tokenization* merupakan proses dimana teks atau kalimat dipisahkan menjadi unit yang lebih kecil, yaitu kata-kata. Proses tokenisasi membantu dalam mengidentifikasi elemen-elemen yang akan dianalisis dan mempermudah proses perhitungan frekuensi serta analisis konteks.
- d. Normalisasi, proses ini melibatkan perubahan kata-kata yang memiliki variasi bentuk seperti kata imbuhan dan slang menjadi bentuk dasar atau standar.
- e. *Stopword Removal* adalah tahap di mana kata-kata yang sering muncul tetapi tidak memiliki makna signifikan dihapus. Proses penghapusan stopwords bertujuan untuk mengeliminasi kata-kata tersebut agar analisis dapat terfokus pada kata-kata yang lebih informatif.
- f. *Stemming*, proses menghilangkan imbuhan dari kata-kata untuk kembali ke bentuk dasar atau akar kata.

Berikut adalah contoh dari hasil pre-processing data yang ditunjukkan pada Tabel 2.

**Tabel 2.** Hasil *Pre-Processing*

Data	Hasil <i>Pre-Processing</i>
tolong diperbaiki transferan pulsa nomor baru bisa langsung buat TF	['tolong', 'perbaiki', 'transfer', 'pulsa', 'nomor', 'baru', 'bisa', 'langsung', 'buat', 'transfer']
klo gk ada data nya gk bisa di buka	['kalau', 'tidak', 'ada', 'data', 'nya', 'tidak', 'bisa', 'di', 'buka']
Aplikasi yang bagus	['aplikasi', 'bagus']
mantap ??	['mantap']
Im3 sinyal busuk . waktunya ganti nih. Payah	['im3', 'sinyal', 'busuk', 'waktu', 'ganti', 'ini', 'payah']

### 2.3 Labeling

Dalam tahap labeling dataset, setiap data diberi label berdasarkan skor yang terkait dengan masing-masing data. Skor ini berasal dari rating yang diberikan oleh pengguna pada aplikasi myIM3. Prosedur ini bertujuan untuk mengkategorikan data ke dalam dua kelas yang berbeda berdasarkan tingkat kepuasan atau pendapat pengguna terhadap aplikasi. Dalam tahap labeling ini, data dengan skor 3 dianggap tidak dapat diinterpretasikan secara tegas, sehingga dianggap tidak cukup untuk dimasukkan ke dalam kelas sentimen netral. Selain itu, data dengan skor 3 juga tidak dapat dianggap sebagai sentimen negatif atau positif karena adanya ketidakpastian dalam interpretasi kata-kata pada ulasan dengan skor tersebut. Sehingga data dengan skor 3 dihapus untuk memudahkan proses analisis dan klasifikasi, menghindari ketidakpastian dalam interpretasi ulasan, serta dataset menjadi lebih terfokus pada sentimen positif dan negatif yang lebih jelas. Lalu untuk skor antara 1 hingga 2, data akan diberi label negatif atau 0, yang menandakan bahwa pengguna tidak puas dengan pengalaman mereka menggunakan aplikasi. Sementara untuk skor antara 4 hingga 5, data akan diberi label positif atau 1, yang menandakan bahwa pengguna puas atau sangat puas dengan pengalaman mereka.

### 2.4 Ekstraksi Fitur

Ekstraksi fitur adalah proses untuk mengubah data teks menjadi representasi numerik yang dapat digunakan oleh algoritma pembelajaran mesin. Dalam penelitian ini digunakan ekstraksi fitur dengan TF-IDF (*Term Frequency-Inverse Document Frequency*) untuk mengukur seberapa penting suatu kata dalam suatu dokumen atau korpus teks. TF-IDF menghitung nilai bobot untuk setiap kata berdasarkan frekuensi kata dalam dokumen tersebut dan kebalikannya dengan

frekuensi kata dalam seluruh korpus [7]. Nilai TF-IDF secara matematis direpresentasikan dalam Persamaan 1 berikut.

$$TF - IDF = w_{d,t} = tf_{d,t} \times \log\left(\frac{N}{df_t}\right) \quad (1)$$

Adapun nilai  $tf_{d,t}$  merupakan TF (*Term Frequency*) digunakan untuk mengevaluasi seberapa sering sebuah kata muncul dalam teks atau jumlah kemunculan istilah  $d$  dalam dokumen  $t$ . Sedangkan nilai  $df_{i,j}$  merupakan *Document Frequency* Document Frequency (DF) untuk menilai relevansi sebuah istilah dalam setiap kumpulan dokumen atau jumlah dokumen yang mencakup kemunculan istilah tersebut. Ini membantu dalam mengevaluasi istilah yang sering muncul dan yang jarang muncul dalam teks. Lalu nilai  $w_{d,t}$  merupakan nilai dari TF-IDF nya dan nilai  $N$  adalah jumlah dokumen. Dengan memanfaatkan metode TF-IDF, kita bisa memberikan nilai numerik kepada kata-kata dalam dokumen yang mencerminkan signifikansi kata-kata tersebut untuk dokumen tersebut dalam konteks keseluruhan kumpulan dokumen.

## 2.5 Multinomial Naive Bayes

*Multinomial Naive Bayes* adalah variasi dari algoritma Naive Bayes yang sangat cocok untuk klasifikasi teks, di mana fitur-fitur yang dihitung adalah frekuensi kata. Algoritma ini menerapkan prinsip distribusi multinomial digunakan untuk menggambarkan probabilitas dari kombinasi frekuensi kata-kata dalam dokumen. Ini berarti bahwa algoritma ini memperhitungkan frekuensi kemunculan kata dalam dokumen, bukan hanya keberadaan atau ketiadaan kata tersebut [8]. Algoritma Multinomial Naive Bayes dengan pembobotan TF-IDF ini dapat dihitung dengan Persamaan 2.

$$P(X_n | C) = \frac{\sum_{d \in C} (X_n, d \in C) + \alpha}{\sum_{d \in C} N_{d \in C} + V} \quad (2)$$

Dimana  $\sum_{d \in C} (X_n, d \in C)$  adalah jumlah pembobotan kata  $X_n$  dari seluruh data training pada kelas  $C$  dan  $\sum_{d \in C} N_{d \in C}$  adalah jumlah bobot seluruh term pada data training pada kelas  $C$ . Dalam penelitian ini, dataset yang telah melalui text preprocessing akan dibagi menjadi 80% data training dan sebanyak 20% data testing.

## 2.6 Evaluasi Model

Untuk tahapan evaluasi model, digunakan *confusion matrix*. *Confusion matrix* merupakan alat evaluasi kinerja yang penting dalam pengembangan model klasifikasi, termasuk dalam evaluasi model *Multinomial Naive Bayes*. *Confusion matrix* adalah tabel yang digunakan untuk membandingkan hasil prediksi dari model dengan label yang sebenarnya dari data uji. Pada tahapan ini, data tes sebanyak 20% akan akan diujikan dengan 80% data trainingnya. *Confusion matrix* memiliki empat sel atau komponen utama, yaitu:

- True Positives* (TP): Semua total data yang diprediksi benar sebagai positif atau berlabel 1 dan benar berlabel positif atau 1.
- False Positives* (FP): Semua total data yang diperkirakan sebagai positif atau berlabel 1, tetapi sebenarnya adalah negatif atau berlabel 0.
- True Negatives* (TN): Semua total data yang diperkirakan benar sebagai negatif atau berlabel 0 dan benar berlabel negatif atau 0.
- False Negatives* (FN): Jumlah data yang diprediksi sebagai negatif atau berlabel 0 tetapi sebenarnya adalah berlabel positif atau 1.

Dari matriks ini, kita dapat menghitung berbagai metrik evaluasi lainnya seperti precision, recall, dan F1-score. Berikut adalah rumus perhitungan precision yang ditunjukkan oleh Persamaan 3, perhitungan *recall* pada Persamaan 4, dan F1-score ditunjukkan oleh Persamaan 5:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

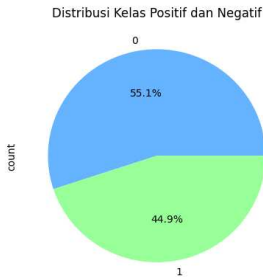
$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

*Precision* (presisi) mengukur kualitas positif yang diprediksi oleh model menggunakan nilai TP (*True Positives*) dan FP (*False Positives*), sedangkan *recall* untuk mengukur seberapa baik model mengidentifikasi semua positif yang sebenarnya menggunakan nilai TP (*True Positives*) dan FN (*False Negative*), dan *F1-Score* digunakan menggabungkan *precision* dan *recall* menjadi suatu nilai yang berguna untuk mengevaluasi model klasifikasi.

### 3. Hasil dan Pembahasan

#### 3.1 Hasil Analisis Data dan Pre-Processing

Dari hasil pengumpulan data melalui kaggle didapatkan data sebanyak 8470, lalu setelah dilakukan pre-processing dan labeling data yang digunakan adalah sebanyak 8420 karena data dengan skor 3 dihapus. Lalu untuk dilakukan penghitungan jumlah data yang telah diberi label 0 (negatif) dan label 1 (positif). Distribusi kelas positif dan negatif ini kemudian divisualisasikan dalam bentuk diagram pie untuk memberikan gambaran yang lebih jelas tentang proporsi atau perbandingan antara kedua kelas tersebut dalam dataset. Dengan visualisasi ini, dapat dilihat proporsi ulasan yang positif dan negatif dalam dataset. Adapun jumlah data setelah di hapuskan skor 3 adalah sebanyak 8212 dengan distribusi kelas positif atau berlabel 1 sebanyak 3690 atau sebesar 44.9% dan distribusi kelas negatif dengan label 0 sebanyak 4552 atau sebesar 55.1%, seperti pada Gambar 3.



Gambar 3. Distribusi Kelas Positif dan Negatif

#### 3.2 Hasil Ekstraksi Fitur TF - IDF

Pada tahap ekstraksi fitur ini, metode TF-IDF yang digunakan. Proses ekstraksi fitur melibatkan penggunaan *CountVectorizer* dan *TfidfTransformer* dari *library* scikit-learn yang dijalankan dengan bahasa pemrograman *Python* di *Google Colab*. Total vocabulary yang dihasilkan adalah sebanyak 8420 kata. Untuk hasil akhir dari matriks TF-IDF diperlihatkan melalui Gambar 4.

```
dokumen = term_fit.transform(Ulasan)
tfidf_transformer = TfidfTransformer().fit(dokumen)
print(tfidf_transformer.idf_)

tfidf = tfidf_transformer.transform(term_frequency)
print(tfidf)

[9.32032636 9.32032636 9.32032636 ... 8.62717918 9.32032636 9.32032636]
(0, 7872) 0.17226885116600582
(0, 7481) 0.14068588532301415
(0, 6994) 0.4001940272639384
(0, 6518) 0.1569646540283102
(0, 6472) 0.26463947011826816
(0, 5937) 0.15138953877958244
(0, 4365) 0.36700462978223203
(0, 4105) 0.20288982922099022
(0, 3372) 0.24288886088196873
(0, 3252) 0.1358725903342128
(0, 1985) 0.12295473854258003
(0, 1915) 0.1881645464642987
(0, 1737) 0.24753531086175948
(0, 1391) 0.4001940272639384
(0, 1142) 0.31573411436871335
(0, 563) 0.21870321318939745
```

Gambar 4. Hasil TF-IDF

### 3.3 Hasil Modeling *Multinomial Naive Bayes*

Model *Multinomial Naive Bayes* dibangun dengan menggunakan bantuan *library* scikit-learn dalam bahasa pemrograman Python. Salah satu hyperparameter yang digunakan adalah alpha sebesar 0,8. Setelah itu model akan dilatih dengan data training sebanyak 80% atau 6569 data dan akan diuji menggunakan data test sebanyak 20% atau 1643 data yang ditunjukkan pada Gambar 5.

```
# Split data training dan data testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Membuat CountVectorizer untuk mengonversi data teks menjadi fitur numerik
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Melatih klasifikasi Naive Bayes Multinomial
mnb = MultinomialNB(alpha=0.8, fit_prior=True, force_alpha=True)
mnb.fit(X_train_vec, y_train)
```

```
MultinomialNB
MultinomialNB(alpha=0.8, force_alpha=True)
```

Gambar 5. Modeling dan *Split* Data

Setelah itu, model yang dibangun dengan model *Multinomial Naive Bayes* dan ekstraksi fitur TF-IDF akan diuji modelnya menggunakan *confusion matrix* untuk mengetahui nilai presisi, akurasi, dan F1 scorenya. Proses evaluasi atau pengujian model ini mendapatkan hasil akurasi keseluruhan sebesar 89%. Dengan precision untuk kelas negatif atau berlabel 0 adalah 86%, recall sebesar 95%, dan f1-score sebesar 91%. Sedangkan untuk kelas positif atau berlabel 1

didapatkan hasil precision sebesar 93%, recall sebesar 81%, dan f1-score sebesar 86%. Hasil dari confusion matrix dapat dilihat pada Gambar 6.

```
----- confusion matrix -----  
[[319 160]  
 [ 26 343]]  
----- classification report -----  
              precision    recall  f1-score   support  
  
     0           0.92       0.67       0.77         479  
     1           0.68       0.93       0.79         369  
  
 accuracy                   0.78         848  
 macro avg                   0.80         848  
 weighted avg                0.82         848
```

Gambar 6. Hasil Confusion Matrix

#### 4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa analisis sentimen pada aplikasi myIM3 menggunakan metode Multinomial Naive Bayes yang memanfaatkan *hyperparameter alpha* sebesar 0.8 dengan ekstraksi fitur TF-IDF menunjukkan nilai precision, recall, akurasi, dan F1-score untuk label negatif berturut-turut adalah 86%, 95%, dan 91%. Sementara untuk label positif, nilai precision, recall, akurasi, dan F1-score berturut-turut adalah 93%, 81%, dan 86%. Akurasi pengujian model menggunakan confusion matrix juga signifikan, mencapai 89%. Selain menggunakan ekstraksi fitur, peningkatan performa model juga dapat dilakukan dengan melakukan penyesuaian terhadap hyperparameter pada model.

#### Daftar Pustaka

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia, "Survey APJII 2023," 2023. [Daring]. Tersedia pada: <https://survei.apjii.or.id/>. Diakses pada 4 Mei 2024.
- [2] A. P. Pratama, A. R. Destriawan, E. K. Panggiarti, "Analisis Merger Pada PT Indosat Tbk (ISAT) Terhadap PT Hutchison 3 Indonesia (H3I)," *J. Akuntan Publik*, vol. 1, no. 4, pp 292-298, 2023.
- [3] Q. A. N. Prakoso, A. Muliawati, I. N. Isnainiyah, "Analisis Sentimen terhadap Produk Skin Game di Forum Review Female Daily Menggunakan Metode Multinomial Naive Bayes," *J. Informatik*, vol. 18, no. 3, pp. 198-207, 2022.
- [4] N. F. Arminda, N. Sulistyowati, T. N. Padilah, "Implementasi Algoritma Multinomial Naive Bayes Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Brimo," *J. Mahasiswa Teknik Informatika*, vol. 7, no. 3, pp. 1817-1822, 2023.
- [5] Z. G. Ngilo, Nuryulani, "Analisis Sentimen Opini Pengguna Twitter Pada Aplikasi Bibit Menggunakan Multinomial Naive Bayes," *J. Teknik dan Science*, vol. 2, no. 1, pp. 8-15, 2023.
- [6] P. A. N. Aryanti, I. B. M. Mahendra, "Analisis Sentimen Opini Berbahasa Indonesia Pada Sosial Media Menggunakan TF-IDF dan Support Vector Machine," *J. Elektronik Ilmu Komputer Udayana*, vol.12, no. 1, pp. 45-52, 2023.
- [7] I. Widyaningrum, D. Mustikasari, R. Arifin, S. L. Tsaqila, D. Fatmawati, "Algoritma Term Frequency – Inverse Document Frequency (TF-IDF) dan K-Means Clustering Untuk Menentukan Kategori Dokumen," *Prosiding Seminar Nasional Sistem Informasi dan Teknologi*, vol. 2022, no. 6, pp. 145-149, 2022.
- [8] M. N. Randhika, J. C. Young, A. Suryadibrata, H. Mandala, "Implementasi Algoritma Complement dan Multinomial Naive Bayes Classifier Pada Klasifikasi Kategori Berita Media Online," *J Teknik Informatika*, vol. 13, no. 1, pp. 19-25, 2021.

# Perlindungan pada Citra Motif Kain Songket dengan Teknik Watermarking Menggunakan RSA Encryption dan MSB Steganography

I Wayan Gede Gemuh Raharja RL<sup>a1</sup>, Nguhur Agus Sanjaya ER<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana

Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

<sup>1</sup>raharja.2208561004@student.unud.ac.id

<sup>2</sup>agus\_sanjaya@unud.ac.id

## Abstract

*This research develops a watermarking steganography technique using the MSB method to protect songket cloth motifs. In the MSB-based steganography method for embedding watermarks in images, the MSB transformation is used to replace data bits in image segments with secret data bits. The embedded watermark functions as an identification mark that is difficult to remove or change without destroying the authenticity of the original motif. Accuracy testing using PSNR and MSE produced an average PSNR of 75.177 dB and MSE of 0.0018, which shows that this technique is effective in maintaining the authenticity and integrity of Songket cloth motifs.*

**Keywords:** Image Processing, MSB, RSA, Watermarking, Songket

## 1. Pendahuluan

Karya seni di Indonesia terus mengalami perkembangan seiring dengan bertambahnya waktu. Karya-karya seni baru pun mulai menggunakan unsur-unsur modern yang abstrak sehingga tidak lagi terpaku pada aturan-aturan tradisional. Penggabungan elemen-elemen ini menghasilkan karya seni yang lebih modern, namun tidak mengurangi nilai estetika tradisionalnya [1]. Kain songket merupakan produk kerajinan tangan yang menjadi kekayaan budaya Indonesia. Salah satu ciri khas yang paling menonjol dari kain songket adalah motif ragam hiasnya [2]. Maraknya pembajakan dan pemalsuan akibat persaingan tidak sehat antara pengusaha atau pengrajin menyebabkan konflik dan kerugian bagi pihak yang lain [1]. Oleh karena itu, diperlukan upaya perlindungan yang efektif untuk memastikan keaslian dan integritas motif kain Songket. *Watermarking* adalah salah satu teknik yang bisa digunakan untuk melindungi motif kain Songket. Teknik ini melibatkan penyembunyian atau penyisipan informasi tertentu ke dalam media yang ingin dilindungi dari pencurian, pelanggaran hak cipta, pembajakan, dan tindakan serupa. *Watermarking* adalah cara untuk menyisipkan *watermark* atau tanda air ke dalam gambar guna melindungi hak ciptanya [3]. *Watermarking* pada penelitian ini diterapkan menggunakan metode enkripsi dan steganografi. Metode enkripsi yang digunakan adalah *RSA Encryption*, sementara metode steganografinya adalah MSB (*Most Significant Bit*). Teknik watermarking ini diterapkan pada motif kain Songket, sehingga informasi terkait motif tersebut dapat disisipkan ke dalamnya. Tanda pengenal yang disisipkan dengan metode *watermarking* ini tidak akan merusak motif kain songket. dengan . Tujuan penelitian ini adalah mengembangkan teknik watermarking khusus untuk melindungi motif kain Songket. Dengan demikian, motif kain Songket dapat dibedakan dari motif hasil pembajakan atau palsu secara akurat.



## 2. Metode Penelitian

### 2.1. Metode MSB (Most Significant Bit)

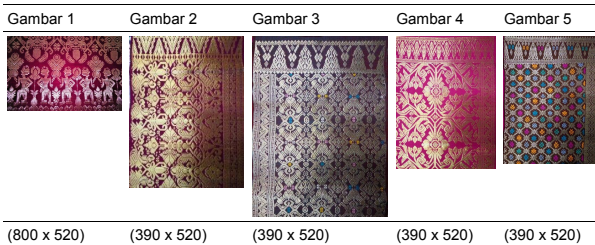
Metode MSB (Most Significant Bit) merupakan salah satu teknik dalam steganografi. Steganografi adalah cara untuk menyembunyikan informasi atau pesan dalam media lain seperti teks, citra digital, video atau suara, dengan tujuan menghindari kecurigaan [4]. Steganografi memiliki 2 tahapan, yaitu proses penyisipan dan pemulihan data data dari media penampung. Penyisipan dilakukan dengan cara mengganti bit-bit data pada media penampung dengan bit-bit data rahasia [5]. Pada metode ini data rahasia disembunyikan pada bit pertama [6]. Dalam data digital, digit paling kiri adalah digit tertinggi, sementara digit paling kanan adalah digit terendah.

### 2.2. RSA

Metode RSA adalah algoritma didasarkan pada pemfaktoran bilangan besar. Algoritma ini mengenkripsi dan mendekripsi data menggunakan kunci publik dan private. Kesulitannya dalam memfaktorkan bilangan besar menjadi faktor prima menjadi keunggulan algoritma ini [7]. Hingga kini belum ada metode atau perangkat yang dapat mendekripsi dengan cepat data terenkripsi algoritma RSA. Sehingga RSA dianggap sebagai algoritma dengan tingkat keamanan yang tinggi. Tingkat keamanan RSA ditentukan oleh panjang kunci yang digunakan. Semakin panjang bit kunci, semakin sulit untuk memecahkannya, karena memfaktorkan dua bilangan prima acak yang dipilih untuk menghasilkan kunci sangatlah sulit.

### 2.3. Data Citra

Data yang digunakan pada penelitian ini adalah 5 buah sampel data citra kain Songket. Data tersebut diperoleh dari internet (<https://id.pinterest.com/>). Data citra tersebut memiliki ukuran piksel yang bervariasi.



### 2.4. Analisis Kebutuhan

- Kebutuhan masukan
  - Media watermarking adalah data citra kain songket dengan ekstensi \*.jpg.
  - Data teks yang akan disisipkan.
- Kebutuhan keluaran
  - File hasil watermark berekstensi \*.jpg.
  - Plaintext atau watermark hasil ekstraksi dari stegano-image.
- Kebutuhan Proses

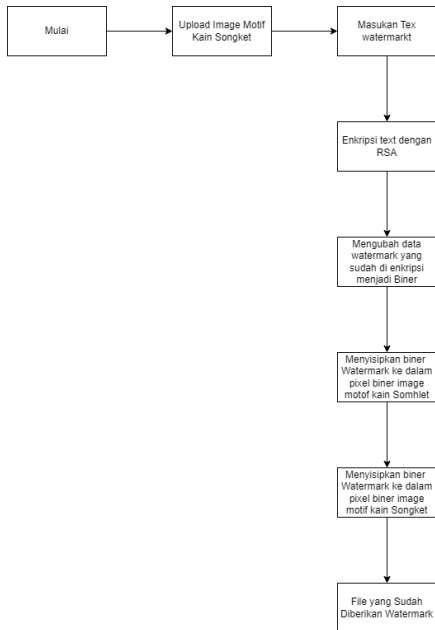
Pada sisi user plaintext akan di enkripsi menggunakan metode RSA *Encryption* untuk mendapatkan ciphertext kemudian ciphertext tersebut diubah kedalam bentuk biner dan disisipkan ke dalam citra digital dengan menggunakan metode MSB (*Most Significant Bit*)

## 2.5 Gambaran Umum Sistem

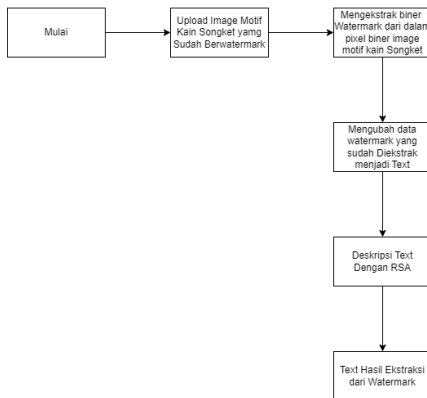
Implementasikan algoritma RSA *Encryption* dan MSB (*Most Significant Bit*) untuk menyisipkan watermark ke dalam citra motif kain songket dilakukan dengan cara user mengupload gambar Kartu Indonesia Sehat menggunakan format gambar yaitu \*.jpeg \*.jpg dan selanjutnya ada proses pembacaan watermark yaitu user akan mengupload stego-image citra songket dan system akan menampilkan hasil dari watermark.

## 2.6 Desain Sistem

Perancangan sistem dapat dilihat pada diagram alur di bawah ini. Terdapat dua proses utama dalam sistem ini, yaitu Proses *Embedding watermark* dan Proses Ekstraksi watermark. Tahapan kedua proses ini dapat dilihat pada gambar 1 dan gambar 2.



**Gambar 1** Diagram Alur Sistem saat Proses Embedding Watermark



Gambar 2 Diagram Alur Sistem Sistem saat Ekstraksi Watermark

### 2.7. Pengujian Akurasi Metode

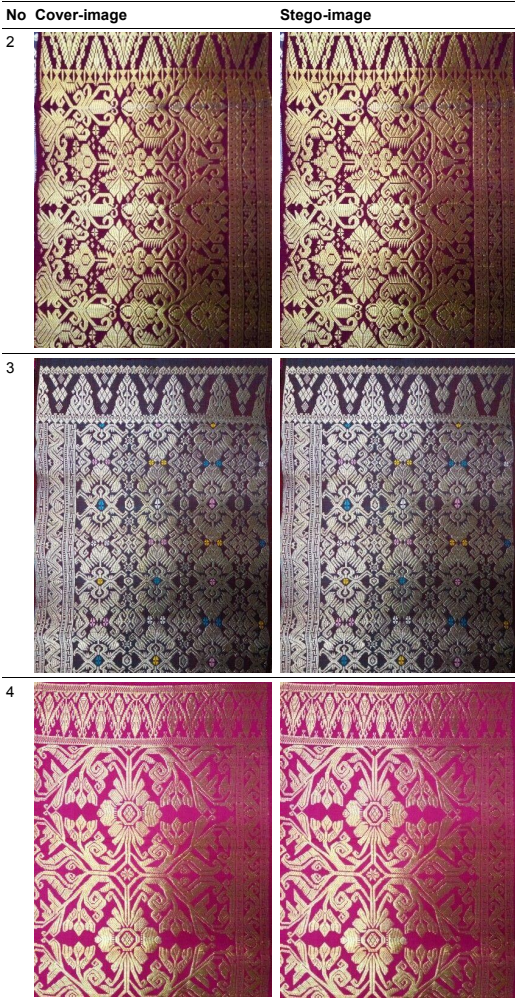
Pengujian akurasi metode dilakukan secara objektif. Pengujian metode steganografi pada sistem menggunakan referensi, *Peak Signal Noise to Ratio dan Mean Square Error*.

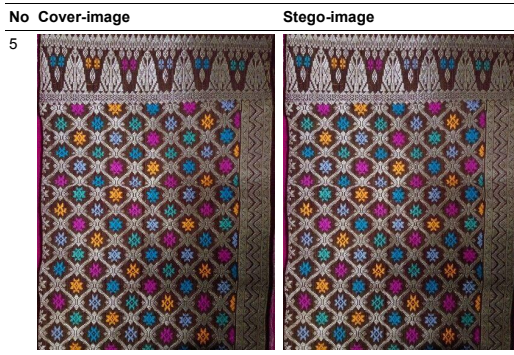
## 3. Hasil dan Pembahasan

### 3.1. Pengujian Gambar Hasil Watermarking

Tabel 1. Perbandingan Stego-image dan Cover-Image

No	Cover-image	Stego-image
1		





Berdasarkan tabel 1, secara kasat mata tidak terdapat perbedaan yang signifikan antara *over-image* dan *Stegano-image*. Sehingga tidak akan merusak motif kain Songket yang akan watermark.

### 3.2. Pengujian Performansi

#### a. PSNR

PSNR digunakan untuk mengukur kualitas citra. Nilai PSNR digunakan untuk mengukur kemiripan *cover-image* dengan *stego-image* hasil kompresi. Rumus PSNR adalah sebagai berikut:

$$PSNR = 20 \log_{10} \left( \frac{MAX}{\sqrt{MSE}} \right)$$

- MAX merupakan nilai maksimum piksel dalam gambar (misalnya 255 untuk gambar 8-bit).
- MSE merupakan Mean Squared Error antara *cover-image* dan *stego-image* hasil kompresi.

Semakin tinggi nilai PSNR yang diperoleh maka semakin mirip *cover-image* dengan *stego-image* hasil kompresi.

#### b. MSE

MSE digunakan untuk mengukur perbedaan kuadrat rata-rata antara nilai piksel dalam *cover-image* dan *stego-image*. Rumus MSE adalah sebagai berikut:

$$MSE = \frac{\sum_1^n (x - y)^2}{M + N}$$

- M dan N adalah dimensi gambar (lebar dan tinggi).
- x = Sinyal Asli, y = Sinyal Stegano,

Semakin rendah nilai PSNR yang diperoleh maka semakin mirip cover-image dengan stego-image hasil kompresi.

Tabel 2. Uji PSNR dan MSE

No	Σ Karakter Plaintext	Ukuran cover-image	Ukuran stego-image	PSNR	MSE
1	31	249.5 KB	286.2 KB	77.77	0.001
	42	249.5 KB	286.3 KB	77.70	0.001
2	31	82.5 KB	149.6 KB	74.63	0.002
	42	82.5 KB	149.6 KB	74.47	0.002
3	31	84.6 KB	155.1 KB	74.57	0.002
	42	84.6 KB	155 KB	74.41	0.002
4	31	75.3 KB	136.3 KB	74.52	0.002
	42	75.3 KB	136.3 KB	74.64	0.002
5	31	82.2 KB	150 KB	74.53	0.002
	42	82.2 KB	150 KB	74.53	0.002

Berdasar tabel 2, dapat dilihat bahwa ukuran stego-image serta nilai MSE dan PSNR dipengaruhi oleh jumlah karakter plaintext watermark. Ukuran stego-image serta nilai MSE dan PSNR berbanding lurus dengan jumlah plaintext yang disisipkan. Semakin banyak karakter plaintext yang disisipkan semakin besar ukuran stego-image yang dihasilkan. Dari lima data tersebut diperoleh rata-rata nilai MSE (0,0018) dan nilai PSNR (75,177 dB) untuk input 31-42 karakter plaintext. Stego-image yang dihasilkan dari metode MSB ini memiliki kualitas yang relatif baik karena menghasilkan nilai PSNR melebihi 50dB. Perbedaan stego-image dan cover-image yang dihasilkan juga sangat kecil karena menghasilkan nilai MSE yang sangat rendah.

#### 4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa Metode MSB (*Most Significant Bit*) dapat digunakan secara efektif untuk melindungi motif kain Songket. Watermark yang digunakan sebagai tanda pengenal dapat memastikan keaslian motif kain Songket sehingga dapat dibedakan dari motif kain hasil pemalsuan secara akurat. Penelitian ini menggunakan lima buah sampel data citra kain songket dengan variasi ukuran piksel. Pengujian akurasi metode dilakukan secara objektif menggunakan Peak Signal-to-Noise Ratio (PSNR) dan Mean Square Error (MSE). Nilai PSNR yang tinggi dan nilai MSE yang rendah pada hasil pengujian akurasi metode menunjukkan tingkat degradasi yang rendah pada kualitas citra hasil *watermarking*. Berdasarkan hasil pengujian, rata-rata PSNR adalah 75.177 dB dan MSE adalah 0.0018 untuk input plaintext dengan panjang 31-42 karakter. Dari data tersebut menunjukkan bahwa stegano-image yang dihasilkan memiliki kualitas yang baik. maka, Penelitian ini memberikan kontribusi dalam melindungi motif kain Songket dari tindakan pemalsuan motif yang merugikan pengrajin kain dengan memastikan integritas dan keaslian motif kain Songket.

#### Daftar Pustaka

- [1] A. A. I. Agung Maheswari, N. M. Gitariani, D. A. Y. Asmari, I. G. Suastika, "Perlindungan Hak Cipta Motif Tenun Endek dan Songket di Desa Telaga Tawang," MARTABE: Jurnal Pengabdian Masyarakat, vol. 3, no. 2, pp. 204-211, 2020. DOI: 10.31604/jpm.v3i2.204-211. p-ISSN: 2598-1218, e-ISSN: 2598-1226.
- [2] M. Pradana Adiputra and K. Nala Hari Wardana, "Pemberdayaan Pengrajin Songket Bali Utara," SULUH: Jurnal Abdimas, vol. 2, no. 1, pp. 33-40, Aug. 2020, doi: <https://doi.org/10.35814/suluh.v2i1.1549>.
- [3] Widiyono, Ari Putra Wibowo, Arief Soma Darmawan, "Watermarking Technique Using

- Least Significant Bit Method on Batik Motif Image," in Prosiding Seminar Edusainstech, FMIPA UNIMUS, 2020, pp. 571-576, ISBN: 978-602-5614-35-4.
- [4] Rosmala. Dewi, Angga Kusuma K. 2018. "Perbandingan Metode Most Significant Bit dan Least Significant Bit pada Steganografi untuk Keamanan Data Media Digital". MIND Journal. Vol. 3. No 2. Hal 36-46.
- [5] Wahyuni. Meri Sri. 2017. "Implementasi Steganografi Dalam Menyembunyikan Pesan Teks Dengan Metode Msb (Most Significant BIT). Jurnal Nasional Informatika dan Teknologi Jaringan. Vol 1. No 2.
- [6] Ratnasari. Anita Putri, Felix Andika Dwiyanto. 2020. "Metode Steganografi Citra Digital". Sains, Aplikasi, Komputasi dan Teknologi Informasi. Vol 2, No 2.
- [7] M. A. Zainuddin and D. I. Mulyana, "Penerapan Algoritma RSA untuk Keamanan Pesan Instan Pada Perangkat Android," J. CKI SPOT, vol. 9, no. 2, pp. 105–114, 2016.

p-ISSN

ISSN 2986-3929



9 772986 392004

e-ISSN

ISSN 3032-1948



9 773032 194009