



ISSN: 2301-5373
E-ISSN: 2654-5101

Volume 10 • Number 4 • May 2022

JELIKU

Jurnal Elektronik Ilmu Komputer Udayana

Informatics Study Program

Faculty of Mathematics and Natural Sciences

Udayana University

Table of Contents

Implementation of Generalized Learning Vector Quantization (GLVQ) and Particle Swarm Optimization (PSO) for Breast Cancer Classification

I Made Satria Bimantara, I Wayan Supriana, Luh Arida Ayu Rahning Putri, I Wayan Santiyasa, Ngurah Agus Sanjaya ER, Anak Agung Istri Ngurah Eka Karyawati 307-318

Klasifikasi Berita Hoaks Covid-19 Menggunakan Kombinasi Metode K-Nearest Neighbor dan Information Gain

Marissa Audina, AAIN Eka Karyawati, I Wayan Supriana, I Ketut Gede Suhartana, I Gede Santi Astawa, I Wayan Santiyasa 319-327

Penerapan Steganography Untuk Perlindungan Hak Cipta Menggunakan Metode Least Significant Bit (LSB)

I Gusti Ngurah Bagus Pramana Putra, I Ketut Gede Suhartana, I Komang Ari Mogi, Cokorda Rai Adi Pramatha, I Putu Gede Hendra Suputra, I Gede Arta Wibawa 329-340

Identifikasi Ekspresi Idiomatik Menggunakan Distributional Semantic Based Approach dan Truth Discovery

Ni Made Yuli Cahyani, AAIN Eka Karyawati, Luh Arida Ayu Rahning Putri, Agus Muliantara, Ida Bagus Gede Dwidasmara, Luh Gede Astuti 341-350

Implementasi LSTM Pada Analisis Sentimen Review Film Menggunakan Adam Dan RMSprop Optimizer

Karlina Surya Witanto, Ngurah Agus Sanjaya ER, AAIN Eka Karyawati, I Gusti Agung Gede Arya Kadyanan, I Ketut Gede Suhartana, Luh Gede Astuti 351-362

Klasifikasi Cerita Pendek Berbahasa Bali Berdasarkan Umur Pembaca dengan Metode Naive Bayes

Luh Ristiari, AAIN Eka Karyawati, I Putu Gede Hendra Suputra, Agus Muliantara, I Dewa Made Bayu Atmaja Darmawan, I Made Widiartha 363-370

SUSUNAN DEWAN REDAKSI
JURNAL ELEKTRONIK ILMU KOMPUTER UDAYANA (JELIKU)

- Penanggung Jawab : Dra. Ni Luh Watiniasih M.Sc., Ph.D.
Dr. Ir. I Ketut Gede Suhartana, S.Kom., M.Kom
- Redaktur : Gst Ayu Vida Mastrika Giri, S.Kom., M.Cs
- Penyunting/Editor : Dr. A A Istri Ngurah Eka Karyawati, S.Si., M.Eng
Cokorda Rai Adi Pramatha, S.T., M.M., Ph.D
Dr. Ngurah Agus Sanjaya ER, S.Kom., M.Kom
Agus Muliantara, S.Kom., M.Kom
I Made Widiartha, S.Si., M.Kom
I Gusti Agung Gede Arya K., S.Kom., M.Kom
Drs. I Wayan Santiyasa, M.Si.
- Desain Grafis : I Komang Ari Mogi, S.Kom., M.Kom
Ida Bagus Made Mahendra, S.Kom., M.Kom
Luh Arida Ayu Rahning Putri, S.Kom., M.Cs
I Gede Santi Astawa, ST., M.Cs
- Fotografer : Ida Bagus Gede Dwidasmara, S.Kom., M.Cs
I Dewa Made Bayu Atmaja Darmawan, S.Kom., M.Cs
I Putu Gede Hendra Suputra, S.Kom., M.Kom
Dra. Luh Gede Astuti, M.Kom
- Sekretariat : I Wayan Supriana, S.Si., M.Cs
Made Agung Raharja, S.Si., M.Cs
I Gusti Anom Cahyadi Putra, ST., M.Cs
I Gede Arta Wibawa, ST., M.Cs

Implementasi *Generalized Learning Vector Quantization (GLVQ)* dan *Particle Swarm Optimization (PSO)* Untuk Klasifikasi Kanker Payudara

I Made Satria Bimantara^{a1}, I Wayan Supriana^{a2}, Luh Arida Ayu Rahning Putri^{a3}, I Wayan Santiyasa^{a4}, Ngurah Agus Sanjaya ER^{a5}, Anak Agung Istri Ngurah Eka Karyawati^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia

¹satriabimantara@student.unud.ac.id

²wayan.supriana@unud.ac.id

³rahningputri@unud.ac.id

⁴santiyasa@unud.ac.id

⁵agus_sanjaya@unud.ac.id

⁶eka.karyawati@unud.ac.id

Abstrak

Kematian terbesar akibat kanker setiap tahun disebabkan oleh kanker payudara (KP). Salah satu penyebab tingginya angka kejadian KP adalah deteksi dini yang terhambat. *Machine learning* telah banyak dimanfaatkan untuk deteksi dini secara otomatis serta mengklasifikasikan jenis kanker. Metode klasifikasi yang dapat digunakan untuk mengklasifikasikan KP ke dalam KP jinak atau ganas adalah GLVQ. Kepekaan inisialisasi vektor bobot awal secara acak pada GLVQ berpengaruh pada hasil tingkat akurasi. Optimasi vektor bobot awal pada GLVQ dapat menggunakan metode optimasi seperti PSO. Data *Breast Cancer Wisconsin Diagnostic Data Set* digunakan dengan beberapa tahapan pengolahan data, yaitu penanganan pencilaan dengan metode *Winsorizing*, normalisasi *z-score*, dan reduksi dimensi dengan PCA. Hasil optimasi vektor bobot yang ditunjukkan melalui nilai rata-rata *fitness* yang dihasilkan pada PSO dipengaruhi oleh perubahan parameter φ_1 , φ_2 , dan ω . Nilai rata-rata *fitness* tertinggi sebesar 0,91868 dihasilkan melalui kombinasi parameter $\varphi_1 = 2,4$, $\varphi_2 = 2,1$, dan $\omega = 0,6$. Tingkat akurasi dan tingkat kesalahan hasil klasifikasi kanker payudara yang dihasilkan metode GLVQ dipengaruhi oleh perubahan parameter α dan n_w . Kombinasi $\alpha = 0,1$, $n_w = 5$, epoch maksimum sebesar 100, dan toleransi kesalahan minimum sebesar 10^{-6} menghasilkan nilai rata-rata akurasi tertinggi sebesar 0,956044. Performa PSO-GLVQ memberikan nilai akurasi, *recall*, dan *F2-Score* yang lebih tinggi dibandingkan GLVQ.

Kata Kunci: *Generalized Learning Vector Quantization, Particle Swarm Optimization, optimasi vektor bobot, klasifikasi kanker payudara*

1. Pendahuluan

Kanker payudara merupakan tumor ganas yang menyerang jaringan payudara, yaitu kelenjar susu, saluran kelenjar susu, dan jaringan penunjang lainnya [1]. Kanker ini biasanya terjadi pada perempuan dan menjadi kanker paling umum nomor dua di dunia [1]. Delapan sampai sembilan persen wanita terkena kanker payudara menurut hasil survei yang dilakukan *World Health Organization (WHO)* [2]. Kanker payudara juga dapat diderita oleh pria, namun kemungkinan kejadian yang lebih rendah dibandingkan wanita [3].

Salah satu penyebab tingginya angka kejadian kanker payudara yaitu terhambatnya upaya deteksi dini kanker payudara [3]. Padahal, kemungkinan pasien sembuh menjadi lebih tinggi apabila jenis kanker diketahui sejak dini [4]. Deteksi dini terhadap tingkat keganasan kanker dapat menurunkan tingkat kematian pasien [5]. Keterlambatan deteksi dini terhadap pasien kanker payudara sebelum kanker mulai menjadi ganas dapat berujung pada kematian [6].

Machine learning sebagai hasil perkembangan teknologi dan ilmu pengetahuan dapat dimanfaatkan untuk melakukan pendeteksian dini secara otomatis [2]. Bidang kesehatan dan pengobatan yang membantu dokter dan ahli dalam mengklasifikasikan jenis kanker telah memanfaatkan klasifikasi berbasis *machine learning* [7]. Metode GLVQ merupakan salah satu metode dalam *machine learning*

yang bisa dimanfaatkan untuk melakukan klasifikasi. Metode ini memanfaatkan vektor bobot sebagai basis di dalam melakukan klasifikasi.

Sato dan Yamada memperkenalkan GLVQ pada tahun 1996 [8] sebagai variasi dan penyempurnaan dari algoritma *Learning Vector Quantization* (LVQ) khususnya pada LVQ2.1. Penelitian [8] menggunakan metode GLVQ untuk penerjemahan bahasa isyarat pada tahun 2020. Hasil penelitiannya menunjukkan bahwa GLVQ memberikan akurasi sebesar 71,37% pada nilai *learning rate* sebesar 0,9 dan lebih tinggi dibandingkan penelitian serupa yang dilakukan Hermawan [8] dengan menggunakan metode LVQ yang memperoleh nilai akurasi hanya 61,54%. Penelitian [9] menyatakan bahwa masalah konvergensi dan ketidakstabilan masih dimiliki metode LVQ yang dikemukakan oleh Kohonen. GLVQ sebagai pembelajaran LVQ modern dapat mencapai konvergen lebih cepat apabila dibandingkan dengan LVQ.

Kepekaan terhadap inialisasi vektor bobot menjadi salah satu masalah utama dalam pembelajaran GLVQ [10]. Sejumlah vektor masukan yang diwakilkan oleh data latih dipilih secara langsung sebagai vektor bobot awal. Cara ini masih memiliki kelemahan karena data latih yang dipilih secara acak dan tidak tepat untuk dijadikan sebagai vektor bobot masukan menyebabkan hasil tingkat akurasi yang buruk [11]. Inialisasi vektor bobot awal perlu dipilih yang optimal karena vektor bobot menjadi acuan di dalam proses klasifikasi [12] dan mempengaruhi hasil klasifikasi [13]. Oleh karena itu, perlu dilakukan optimasi vektor bobot pada GLVQ untuk mendapatkan vektor bobot awal yang optimal sehingga diharapkan mendapatkan nilai akurasi yang lebih tinggi di dalam melakukan klasifikasi. Algoritma optimasi seperti *Particle Swarm Optimization* (PSO) bisa digunakan untuk hal tersebut [14].

Algoritma PSO memiliki beberapa keunggulan dibandingkan algoritma optimasi yang lainnya. Menurut Ridwansyah et al [15], permasalahan optimasi dapat diselesaikan menggunakan Algoritma PSO. Algoritma PSO memiliki keunggulan dari segi efisiensi apabila dibandingkan dengan metode optimasi yang lain [13]. Asriningtias mengungkapkan bahwa proses pelatihan *neural network* memiliki waktu lebih cepat apabila menggunakan Algoritma PSO daripada Algoritma Genetika. Kombinasi *Neural Network* dan Algoritma PSO terbukti dapat memberikan nilai akurasi yang lebih besar apabila dibandingkan hanya menggunakan *neural network* saja. Akurasinya bertambah sebesar 7,78% [15].

Penelitian ini memanfaatkan Algoritma PSO untuk mengoptimasi vektor bobot awal pada GLVQ sebelum diinisialisasi, sehingga tahap pelatihan pada GLVQ menggunakan vektor bobot awal yang optimal. Pengujian parameter terbaik dari GLVQ dan PSO dilakukan untuk mendapatkan kombinasi parameter terbaik dari keduanya dalam mengklasifikasikan kanker payudara. Pengujian untuk membandingkan hasil akurasi yang dihasilkan metode GLVQ dengan dan tanpa optimasi PSO dilakukan menggunakan sejumlah metrik performansi.

2. Metode Penelitian

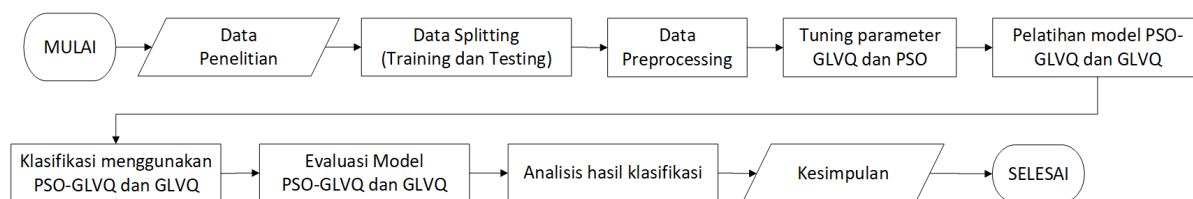
2.1. Data Penelitian

Data *Breast Cancer Wisconsin (Diagnostic) Data Set* yang didapat dari *University of California Irvine (UCI) Machine Learning Repository* digunakan sebagai data sekunder. Beberapa penelitian sebelumnya telah menggunakan data ini dengan teknik pengolahan data dan metode klasifikasi yang berbeda [16]. Data ini memiliki 569 baris data dengan 32 atribut. Rincian ke-32 atribut tersebut, yaitu *ID*, *diagnosis*, dan 30 atribut hasil komputasi sistem *Xcyt*. Atribut *ID* tidak memberikan informasi yang berarti karena hanya menunjukkan nomor unik setiap pasien yang menderita kanker payudara. Oleh karena itu, atribut ini tidak diikutkan ke dalam model [16]. Kondisi kanker payudara yang dialami pasien dari hasil pemeriksaan ditunjukkan melalui atribut *diagnosis*. Atribut ini menjadi label kelas pada data. Terdapat 357 baris data pasien yang mengidap kanker payudara jinak (*benign*) yang disimbolkan dengan "B" dan 212 baris data pasien yang mengidap kanker payudara ganas (*malignant*) yang disimbolkan dengan "M".

2.2. Desain Penelitian

Gambar 1 menunjukkan diagram alir penelitian. Penelitian diawali dengan membagi data penelitian menjadi data latih dan data uji menggunakan metode *Holdout*. Proporsi data latih dan data uji masing-masing sebesar 80% dan 20%. Data latih digunakan untuk: (i) menentukan kombinasi parameter GLVQ terbaik; (ii) menentukan kombinasi parameter PSO terbaik; (iii) mengoptimasi vektor bobot awal dengan PSO; dan (iv) proses pelatihan dengan PSO-GLVQ dan GLVQ, dengan menggunakan *5-fold cross-validation*. Perhitungan nilai akurasi dan tingkat kesalahan dari: (i) klasifikasi kanker payudara yang dihasilkan metode GLVQ menggunakan vektor bobot tanpa dioptimasi dengan PSO; dan (ii) klasifikasi

kanker payudara yang dihasilkan metode GLVQ menggunakan vektor bobot yang dioptimasi dengan PSO, menggunakan data uji.



Gambar 1. Diagram alir desain penelitian

Identifikasi dan penanganan data penciran adalah tahapan *preprocessing* yang pertama. Metode *Winsorizing* digunakan untuk mengidentifikasi dan menangani data penciran pada suatu atribut. Penelitian ini menggunakan *threshold* nilai $k=5$ untuk menentukan data penciran pada suatu atribut [17]. Semua nilai yang berada di bawah persentil ke-5 (P_5) diubah menjadi nilai pada P_5 , sedangkan semua nilai yang berada di atas persentil ke-95 (P_{95}) diubah menjadi nilai pada P_{95} [17]. Hal ini dilakukan untuk seluruh atribut yang ada.

Normalisasi data merupakan tahapan *preprocessing* yang kedua. Metode normalisasi *z-score* seperti persamaan (1) digunakan sebagai metode normalisasi data. Tahapan ini memiliki peran yang penting di dalam penambangan data, khususnya klasifikasi dan klusterisasi. Agar proses penambangan data tidak bias, nilai pada setiap atribut yang memiliki rentang berbeda perlu distandarisasi atau dinormalisasi [18].

$$x_i^1 = \frac{x_i - \bar{B}}{\sigma_B} \quad (1)$$

Reduksi atribut atau dimensi merupakan tahapan *preprocessing* yang ketiga. Proses pelatihan model klasifikasi dapat berjalan lebih cepat dan tetap memberikan hasil yang optimal dengan melakukan reduksi dimensi pada data. Metode *Principal Component Analysis* (PCA) diterapkan untuk melakukan reduksi dimensi pada penelitian ini. Metode ini memilih sejumlah k komponen utama (KU). Sejumlah k KU yang didapat selanjutnya digunakan untuk mereduksi atribut awal pada data penelitian ke dalam ranah baru. Kumulatif proporsi nilai variansi yang dihasilkan dari setiap komponen akan digunakan untuk memilih sejumlah k KU. Penelitian ini menggunakan *threshold* nilai minimum kumulatif proporsi variansi yang bisa dijelaskan sebesar 80% [19] dalam menentukan sejumlah k KU yang dipertahankan.

Data latih yang sudah di-*preprocessing* kemudian digunakan pada tahapan *tuning parameter* dari metode GLVQ. Tahapan ini menggunakan metode *Grid Search* dengan validasi *5-fold cross-validation*. Lima nilai akurasi diperoleh untuk setiap kombinasi parameter GLVQ yang dihasilkan dari lima eksperimen yang dilakukan. Eksperimen ke- i adalah melatih model GLVQ menggunakan seluruh data latih selain data pada *fold* ke- i ; kemudian mengevaluasi model tersebut menggunakan data pada *fold* ke- i . Kelima nilai akurasi ini kemudian digunakan untuk mendapatkan nilai rata-rata akurasi untuk setiap kombinasi parameter GLVQ. Kombinasi parameter optimal dari metode GLVQ yang menghasilkan nilai rata-rata akurasi tertinggi adalah luaran dari tahapan ini.

Data latih yang sudah di-*preprocessing* serta kombinasi parameter optimal dari metode GLVQ digunakan pada tahapan *tuning parameter* dari metode PSO. Tahapan ini menggunakan metode *Grid Search* dengan validasi *5-fold cross-validation*. Lima nilai *fitness* diperoleh untuk setiap kombinasi parameter PSO yang dihasilkan dari lima eksperimen yang dilakukan. Eksperimen ke- i adalah mengoptimasi vektor bobot awal dengan seluruh data latih selain data pada *fold* ke- i ; kemudian menghitung nilai *fitness* dari vektor bobot awal yang sudah dioptimasi menggunakan data pada *fold* ke- i . Kelima nilai *fitness* ini kemudian digunakan untuk mendapatkan nilai rata-rata *fitness* untuk setiap kombinasi parameter PSO. Kombinasi parameter optimal dari metode PSO yang menghasilkan nilai rata-rata *fitness* tertinggi adalah luaran dari tahapan ini.

Kombinasi parameter optimal dari GLVQ dan PSO yang telah didapat dari tahapan sebelumnya kemudian digunakan pada tahap pelatihan GLVQ. Terdapat dua cara inisialisasi vektor bobot pada tahap ini, yaitu menggunakan vektor bobot awal yang diinisialisasi secara acak dan menggunakan vektor bobot awal yang dioptimasi dengan Algoritma PSO. Fungsi pembangkit acak berdistribusi *uniform* dalam rentang -1 sampai 1 [9] digunakan untuk menginisialisasi vektor bobot awal secara acak, sedangkan Algoritma PSO digunakan untuk mengoptimasi vektor bobot awal. Setelah vektor bobot awal dipastikan sudah diinisialisasi, baik menggunakan PSO atau secara acak, proses dilanjutkan dengan pelatihan menggunakan algoritma pelatihan GLVQ. Luaran dari proses ini adalah dua model pengklasifikasi yang sudah dilatih, yaitu PSO-GLVQ dan GLVQ.

Tahap klasifikasi atau pengujian GLVQ dilakukan setelah dua model pengklasifikasi hasil pelatihan didapatkan. Tahap ini dilakukan untuk mengevaluasi performa dari kedua model pengklasifikasi tersebut menggunakan data uji. *Confusion matrix* digunakan untuk merepresentasikan hasil klasifikasi dari kedua model. Evaluasi model PSO-GLVQ dan GLVQ menggunakan sejumlah ukuran performa, seperti *accuracy*, *error rate*, *recall*, *precision*, dan *F2-Score* dihitung berdasarkan *confusion matrix* masing-masing.

2.3. Tahapan PSO Untuk Optimasi Vektor Bobot Pada GLVQ

Penelitian [13] digunakan sebagai acuan pada tahap optimasi vektor bobot menggunakan Algoritma PSO. Vektor bobot awal pada GLVQ dioptimasi menggunakan Algoritma PSO. Vektor bobot optimal diperoleh dari *Gbest* pada iterasi terakhir. Vektor bobot yang sudah dioptimasi selanjutnya digunakan pada proses pelatihan GLVQ. Adapun langkah-langkah di dalam Algoritma PSO [13]:

1. Tentukan data latih, jumlah partikel sebagai Np , maksimum iterasi sebagai E , laju belajar kecerdasan individu (*cognition*) sebagai φ_1 , laju belajar hubungan sosial antar individu sebagai φ_2 , dan dua bilangan acak (dalam interval $0 - 1$) masing-masing r_1 dan r_2 .
2. Lakukan langkah 2.a – 2.g sebagai tahap inialisasi awal.

2.a Tentukan batas minimum sebagai T_{min} dan batas maksimum sebagai T_{max} untuk setiap atribut.

2.b Tentukan kecepatan minimum sebagai V_{min} dan kecepatan maksimum sebagai V_{max} untuk setiap atribut. Nilai V_{min} sama dengan $-V_{max}$ dan nilai V_{max} dapat ditentukan oleh user [20]. Nilai v_{max} dapat ditentukan dengan memperhatikan rentang nilai dari setiap atribut yang ada [20], sehingga V_{max} dan V_{min} masing-masing dapat ditentukan menggunakan persamaan (2) dan (3).

$$V_{max(j)} = (T_{max(j)} - T_{min(j)}) \quad (2)$$

$$V_{min(j)} = -V_{max(j)} \quad (3)$$

2.c Inialisasi kecepatan awal seluruh partikel dengan nilai 0.

2.d Inialisasi posisi awal seluruh partikel dengan menggunakan persamaan (4).

$$x_{i,j}^{(0)} = T_{min(j)} + rand[0,1] \times (T_{max(j)} - T_{min(j)}) \quad (4)$$

2.e Inialisasi nilai *fitness* seluruh partikel dengan nilai 0.

2.f Inialisasi *Pbest* awal setiap partikel dengan masing-masing posisi awal setiap partikel yang telah diinisialisasi.

2.g Inialisasi *Gbest* awal dengan *Pbest* dari partikel dengan nilai *fitness* tertinggi.

3. Atur nilai $iter_0$ sama dengan nol.

4. Selama nilai $iter_t$ kurang dari E , lakukan langkah 4.a – 4.d.

4.a Untuk setiap partikel, lakukan:

(i) Hitung nilai *fitness* partikel menggunakan persamaan (10); dan

(ii) Perbarui nilai *Pbest* dengan memperhatikan nilai *fitness* pada iterasi ke- $iter_{t-1}$ dan nilai *fitness* pada iterasi ke- $iter_t$. Jika nilai *fitness* partikel pada iterasi ke- $iter_t$ lebih baik dibandingkan nilai *fitness* pada iterasi ke- $iter_{t-1}$, maka kondisi partikel pada iterasi ke- t akan menjadi *Pbest* yang baru dari partikel tersebut dan begitupun sebaliknya. Pembaruan nilai *Pbest* menggunakan persamaan (5) berikut.

$$Pbest_i^{t+1} = \begin{cases} Pbest_i^t, & fitness(x_i^{t+1}) \leq fitness(Pbest_i^t) \\ x_i^{t+1}, & fitness(x_i^{t+1}) > fitness(Pbest_i^t) \end{cases} \quad (5)$$

4.b Pilih partikel dengan nilai *fitness* paling maksimum dari semua partikel yang ada, kemudian jadikan sebagai *Gbest* sesuai dengan persamaan (6).

$$Gbest^{t+1} = \begin{cases} Gbest^t, & argmax(fitness(Pbest_i^{t+1})) \leq fitness(Gbest^t) \\ Pbest_i^{t+1}, & argmax(fitness(Pbest_i^{t+1})) > fitness(Gbest^t) \end{cases} \quad (6)$$

4.c Untuk setiap partikel, lakukan:

(i) Perbarui kecepatan menggunakan persamaan (7); dan

$$v_{i,j}^{t+1} = \omega \times v_{i,j}^t + \varphi_1 \times r_1 \times (Pbest_{i,j}^t - x_{i,j}^t) + \varphi_2 \times r_2 \times (Gbest_j^t - x_{i,j}^t) =$$

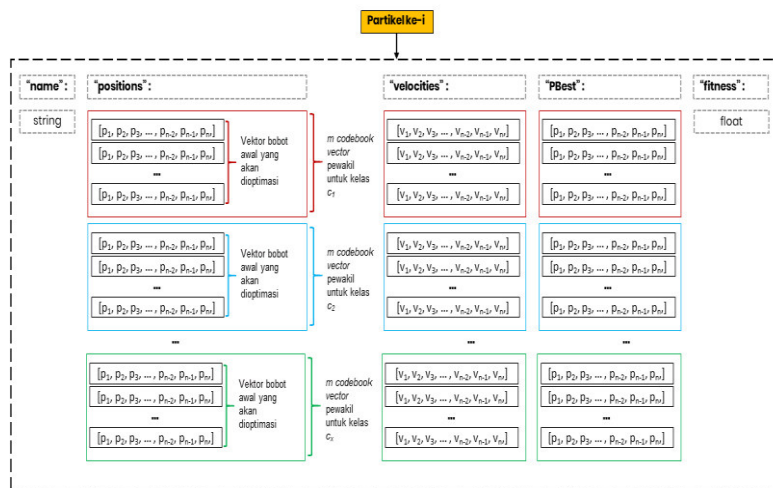
$$\begin{cases} v_{max(j)}, v_{i,j}^{t+1} \geq v_{max(j)} \\ v_{min(j)}, v_{i,j}^{t+1} \leq v_{min(j)} \\ v_{i,j}^{t+1}, v_{min(j)} < v_{i,j}^{t+1} < v_{max(j)} \end{cases} \quad (7)$$

(ii) Perbarui posisi menggunakan persamaan (8).

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} = \begin{cases} T_{max(j)}, x_{i,j}^{t+1} \geq T_{max(j)} \\ T_{min(j)}, x_{i,j}^{t+1} \leq T_{min(j)} \\ x_{i,j}^{t+1}, T_{min(j)} < x_{i,j}^{t+1} < T_{max(j)} \end{cases} \quad (8)$$

4.d Perbarui nilai $iter_{t+1}$ menggunakan persamaan (9).

$$iter_{t+1} = iter_t + 1 \quad (9)$$



Gambar 2. Ilustrasi satu partikel yang diinisialisasi untuk proses optimasi vektor bobot awal pada GLVQ

2.4. Perhitungan Nilai *Fitness* Partikel

Data latih digunakan pada proses optimasi vektor bobot awal pada PSO, sehingga fungsi objektif pada kasus optimasi PSO ini adalah memaksimalkan jumlah data latih yang terklasifikasi benar oleh suatu vektor bobot (yang diwakili satu partikel). Fungsi *fitness* sesuai persamaan (10) digunakan untuk menghitung nilai *fitness* partikel berdasarkan fungsi objektif yang telah ditetapkan. Perhitungan nilai *fitness* setiap partikel mengikuti tahapan-tahapan sebagai berikut [12]:

1. Masukkan data latih beserta dengan label kelasnya. Misalkan label kelas hasil klasifikasi dari data latih ke- i yaitu C_i .
2. Lakukan perhitungan jarak antara data latih dengan setiap vektor bobot dalam satu partikel menggunakan persamaan (12). Misalkan D_j^i merupakan jarak antara vektor bobot ke- j dengan data latih ke- i .
3. Label kelas hasil klasifikasi untuk data latih ke- i dapat ditentukan dari nilai terkecil yang diperoleh pada hasil perhitungan jarak nomor 2. Misalkan T_i merupakan label kelas hasil klasifikasi untuk data latih ke- i .
4. Nilai *fault* diperbarui dengan syarat: (a) Jika $T_i \neq C_i$, maka nilai *fault* bertambah satu dari nilai sebelumnya; dan (b) Jika $T_i = C_i$, maka nilai *fault* tidak berubah dari nilai sebelumnya.
5. Hitung nilai *fitness* partikel dengan persamaan (10). F_i merupakan fitness partikel ke- i , N_{latih} merupakan jumlah seluruh baris data latih, dan *fault* merupakan jumlah seluruh baris data latih yang terklasifikasi salah dari kelas target aslinya.

$$F_i = \frac{N_{latih} - fault}{N_{latih}} \quad (10)$$

2.5. Tahapan Pelatihan Menggunakan GLVQ

Suatu data masukan dapat dikenali melalui vektor bobot yang sudah dilatih dengan metode GLVQ, sehingga bisa diklasifikasikan ke luaran yang tepat. Tahapan pelatihan menggunakan GLVQ adalah sebagai berikut [9].

1. Inisialisasi sampel data latih $V = \{v_i \in \mathbb{R}^n, i = 1, \dots, m\}$.
2. Masukkan nilai maksimum epoch sebagai T dan nilai laju pembelajaran sebagai α_0 .
3. Inisialisasi vektor bobot awal sebagai $W = \{w_j \in \mathbb{R}^n, j = 1, \dots, M\}$ secara acak. Apabila dilakukan optimasi terhadap vektor bobot dengan PSO, maka gunakan vektor bobot hasil optimasi PSO sebagai vektor bobot awal dalam pelatihan GLVQ.
4. Inisialisasi nilai $t = 0$.
5. Selama kondisi berhenti belum tercapai, lakukan langkah 5.a sampai 5.b.

5.a Hitung nilai laju pembelajaran dengan persamaan (11).

$$\alpha = \alpha_0 \left(1 - \frac{t}{T}\right) \quad (11)$$

5.b Untuk setiap sampel data latih v_i dan W , lakukan langkah 5.b.1 sampai 5.b.4.

5.b.1 Hitung jarak v_i dengan setiap w_j pada W menggunakan persamaan (12) dan tentukan w^+ dan w^- .

$$\|v_i - w_j\| = \sum_{k=1}^n (v_{i,k} - w_{j,k})^2 \quad (12)$$

5.b.2 Tentukan $d^+(v_i)$ dan $d^-(v_i)$.

5.b.3 Hitung perbedaan jarak relatif $\mu(v_i)$ sesuai dengan persamaan (13).

$$\mu(v_i) = \frac{d^+(v_i) - d^-(v_i)}{d^+(v_i) + d^-(v_i)} \in [-1, 1] \quad (13)$$

5.b.4 Lakukan perbaikan w^+ dan w^- masing-masing menggunakan persamaan (14) dan (15).

$$w^+ \leftarrow w^+ + \alpha \frac{\delta f}{\delta \mu} \frac{d^-(v_i)}{(d^+(v_i) + d^-(v_i))^2} (x - w^+) \quad (14)$$

$$w^- \leftarrow w^- - \alpha \frac{\delta f}{\delta \mu} \frac{d^+(v_i)}{(d^+(v_i) + d^-(v_i))^2} (x - w^-) \quad (15)$$

6. Luaran proses pelatihan adalah $W = \{w_j \in \mathbb{R}^n, j = 1, \dots, M\}$ sebagai vektor bobot hasil pelatihan.

Kondisi berhenti pada pembelajaran vektor bobot sesuai algoritma di atas yaitu apabila nilai maksimum *epoch* telah tercapai [21] atau perubahan nilai vektor bobot iterasi ke- t dibandingkan hasil iterasi ke- $(t - 1)$ kurang dari toleransi perubahan [22].

2.6. Tahapan Klasifikasi Menggunakan GLVQ

Vektor bobot hasil pelatihan menggunakan metode PSO-GLVQ dan GLVQ selanjutnya akan masuk ke tahap klasifikasi dengan menggunakan GLVQ. Tahapan klasifikasi menggunakan GLVQ adalah sebagai berikut [9].

1. Masukkan himpunan vektor bobot sebagai W dan himpunan data uji sebagai V .
2. Untuk setiap $v_i \in V$ lakukan langkah 2.a sampai 2.c.
 - 2.a Hitung $d(v_i, w_j)$ sesuai persamaan (12) untuk setiap $w_j \in W$.
 - 2.b Cari indeks dari w_j yang memiliki jarak paling minimum dengan v_i sesuai persamaan (16).
$$s(v_i) = \operatorname{argmin}_{j=1, \dots, M} d(v_i, w_j) \quad (16)$$
 - 2.c Kelas dari v_i dapat ditentukan oleh $c(w_{s(v_i)})$.

Luaran proses ini adalah hasil klasifikasi dari setiap $v_i \in V$ dengan metode GLVQ.

2.7. Evaluasi dan Seleksi Model Klasifikasi

Pemilihan kombinasi parameter optimal dari metode GLVQ dan PSO menggunakan data latih yang sudah di-*preprocessing*. Proses tersebut menggunakan metode *Grid Search* dengan validasi *5-fold cross-validation*. Pelatihan model PSO-GLVQ dan GLVQ dilakukan dengan keseluruhan data latih yang sama menggunakan kombinasi parameter optimal dari kedua metode ini. Evaluasi PSO-GLVQ dan GLVQ dilakukan dengan menggunakan data uji yang sama yang sudah di-*preprocessing*. *Confusion matrix* digunakan untuk merepresentasikan hasil klasifikasi kedua model pengklasifikasi tersebut. *Confusion matrix* ini selanjutnya digunakan sebagai acuan dalam menghitung sejumlah ukuran evaluasi model pengklasifikasi. Tabel 1 merupakan ilustrasi *confusion matrix* yang digunakan pada penelitian ini.

Tabel 1. *Confusion matrix* untuk mengevaluasi model pengklasifikasi PSO-GLVQ dan GLVQ

		Kelas hasil klasifikasi		Jumlah
		Malignant	Benign	
Kelas Aktual	Malignant	TP	FN	P
	Benign	FP	TN	N
Jumlah		P'	N'	

3. Hasil dan Pembahasan

3.1. Hasil Implementasi Tahap Pengolahan Data

Data latih dan data uji dibagi dari data penelitian pada awal tahap tahap pengolahan data. Terdapat 455 baris data untuk data latih dan 114 baris data untuk data uji. Terdapat 281 baris data teridentifikasi sebagai data pencilan untuk ke-30 atribut yang diidentifikasi dari 455 baris data pada data latih. Data latih yang sudah ditangani data pencilannya kemudian dinormalisasi dengan metode *z-score*. Metode PCA kemudian digunakan untuk mereduksi atribut pada data latih yang sudah dinormalisasi. Berdasarkan hasil perhitungan *variance*, proporsi *variance*, dan kumulatif proporsi *variance* yang bisa dijelaskan dari ke-30 komponen yang terbentuk dari data latih, sejumlah empat *principal component* (PC) pertama dapat dipilih untuk mereduksi atribut. Empat PC dapat menjelaskan *variance* dari data latih sebesar 81,0492%.

3.2. Pengaruh Perubahan Parameter Pada GLVQ Terhadap Hasil Akurasi

Pengujian pengaruh perubahan parameter-parameter pada GLVQ, yaitu α (laju pembelajaran) dan n_w (jumlah vektor bobot per kelas) menggunakan data latih yang sudah di-*preprocessing*. Pengujian ini menggunakan metode *Grid search* dengan validasi *5-fold cross-validation*. Himpunan parameter α {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} [8] dan himpunan parameter n_w {1, 2, 3, 4, 5} [9] digunakan pada pengujian ini. Nilai *epoch* maksimum (*maxepoch*) ditetapkan sebesar 100 dan toleransi kesalahan minimum (*minerror*) ditetapkan sebesar 10^{-6} . Nilai *maxepoch* sebesar 100 dapat menghasilkan akurasi tertinggi dengan waktu komputasi yang lebih singkat dibandingkan nilai *epoch* maksimum yang berada dalam rentang 100 sampai 1000 [23]. Nilai *minerror* sebesar 10^{-6} digunakan berdasarkan penelitian [21] yang menyatakan bahwa nilai akurasi yang dihasilkan tidak dipengaruhi oleh toleransi kesalahan minimum. Dari keseluruhan anggota himpunan untuk setiap parameter pada GLVQ, sejumlah 45 kombinasi parameter dihasilkan berdasarkan skema *Grid Search*. Kombinasi parameter pada GLVQ yang menghasilkan nilai rata-rata akurasi tertinggi dapat ditentukan dari keseluruhan hasil kombinasi parameter beserta nilai akurasi pada setiap *fold*.

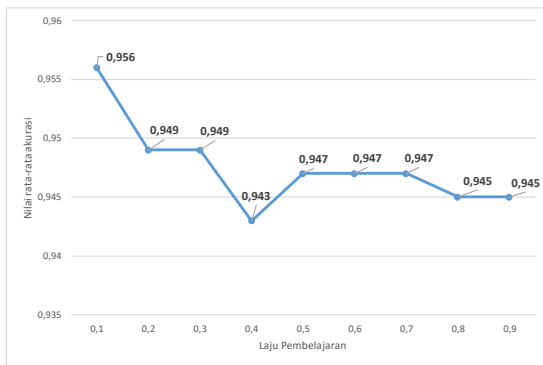
Tabel 2. Cuplikan hasil kombinasi parameter pada GLVQ beserta nilai akurasi yang dihasilkan menggunakan metode *Grid Search* dengan validasi *5-fold cross-validation*

n_w	α	Akurasi Fold-1	Akurasi Fold-2	Akurasi Fold-3	Akurasi Fold-4	Akurasi Fold-5	Rata-rata akurasi
1	0,1	0,9560	0,9341	0,9121	0,9341	0,9341	0,9341
1	0,2	0,9560	0,9341	0,9231	0,9341	0,9451	0,9385
1	0,3	0,9560	0,9341	0,9231	0,9341	0,9451	0,9385
1	0,4	0,9560	0,9341	0,9231	0,9341	0,9451	0,9385
1	0,5	0,9560	0,9341	0,9231	0,9341	0,9451	0,9385
...
5	0,6	0,9890	0,9341	0,9121	0,9451	0,9560	0,9473
5	0,7	0,9780	0,9341	0,9121	0,9451	0,9670	0,9473
5	0,8	0,9890	0,9341	0,9121	0,9451	0,9451	0,9451

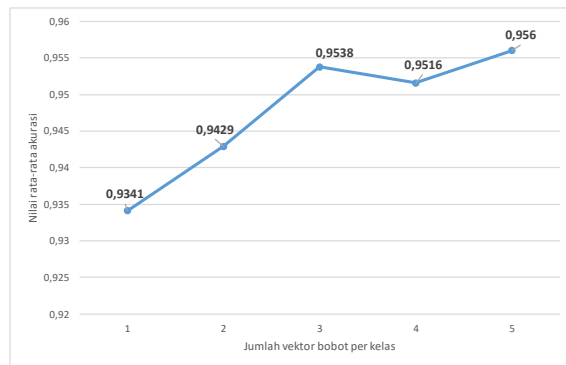
Klasifikasi Kanker Payudara Menggunakan *Generalized Learning Vector Quantization (GLVQ)* dan *Particle Swarm Optimization (PSO)*

5 0,9 0,9780 0,9341 0,9121 0,9451 0,9560 0,9451

Kombinasi parameter pada GLVQ, yaitu $\alpha=0,1$, $n_w=5$, $maxepoch=100$, dan $minerror=10^{-6}$ adalah kombinasi parameter optimal dari GLVQ berdasarkan keseluruhan hasil pengujian. Nilai rata-rata akurasi sebesar 0,956043 dihasilkan dengan kombinasi ini. Nilai rata-rata tingkat akurasi pada hasil klasifikasi kanker payudara memiliki kecenderungan menurun apabila parameter α semakin besar yang dijelaskan melalui Gambar 3. Semakin kecil parameter α , maka vektor bobot semakin cepat konvergen pada tahap pelatihan yang berpengaruh terhadap hasil klasifikasi, begitupun sebaliknya [21]. Nilai rata-rata tingkat akurasi pada hasil klasifikasi kanker payudara memiliki kecenderungan meningkat apabila parameter n_w semakin besar yang dijelaskan melalui Gambar 4. Semakin besar parameter n_w artinya semakin banyak jumlah vektor bobot perwakilan di setiap kelas yang ada. Jumlah vektor bobot perwakilan di setiap kelas yang semakin banyak dapat memberikan ruang solusi yang lebih besar ketika vektor bobot berlatih dengan data latih. Nilai parameter n_w dibuat meningkat pada penelitian ini karena kesalahan generalisasi dari model GLVQ dapat diminimalkan dengan konfigurasi parameter n_w yang besar [9]. Kemampuan generalisasi dan ketangguhan dari model GLVQ meningkat seiring parameter n_w membesar [24].



Gambar 3. Grafik pengaruh perubahan parameter α terhadap nilai rata-rata tingkat akurasi hasil klasifikasi kanker payudara



Gambar 4. Grafik pengaruh perubahan parameter n_w terhadap nilai rata-rata tingkat akurasi hasil klasifikasi kanker payudara

3.3. Pengaruh Perubahan Parameter Pada PSO Terhadap Hasil Optimasi Vektor Bobot

Pengujian pengaruh perubahan parameter-parameter pada PSO yaitu φ_1 , φ_2 dan ω menggunakan data latih yang sudah di-*preprocessing* dan kombinasi parameter optimal GLVQ dari pengujian sebelumnya. Metode *Grid Search* dengan validasi *5-fold cross-validation* digunakan pada pengujian ini untuk melihat pengaruh perubahan parameter-parameter pada PSO terhadap hasil optimasi vektor bobot (yang ditunjukkan melalui nilai rata-rata *fitness*) untuk klasifikasi kanker payudara. Himpunan parameter φ_1 dan φ_2 yang digunakan masing-masing yaitu {2.1, 2.2, 2.3, 2.4, 2.5}. Himpunan parameter ω yang digunakan yaitu {0.5, 0.6, 0.7, 0.8, 0.9, 1.0} [14]. Parameter lain pada PSO seperti jumlah partikel (N_p) dan jumlah iterasi maksimum (E_{max}) nilainya dibuat tetap masing-masing sebesar 30 [20] dan 100 [12]. Dari keseluruhan anggota himpunan untuk setiap parameter pada PSO, sejumlah 150 kombinasi parameter dihasilkan berdasarkan skema *Grid Search*. Kombinasi parameter pada PSO yang menghasilkan nilai rata-rata *fitness* tertinggi dapat ditentukan dari keseluruhan hasil kombinasi parameter beserta nilai *fitness* pada setiap *fold*.

Tabel 3. Cuplikan hasil kombinasi parameter pada PSO beserta nilai *fitness* yang dihasilkannya dengan metode validasi *5-fold cross-validation*

φ_1	φ_2	ω	Fitness Fold-1	Fitness Fold-2	Fitness Fold-3	Fitness Fold-4	Fitness Fold-5	Rata-rata fitness
2,1	2,1	0,5	0,8681	0,8791	0,8132	0,9121	0,8681	0,8681
2,1	2,1	0,6	0,8462	0,8571	0,8352	0,8462	0,7802	0,8330
2,1	2,1	0,7	0,8462	0,9231	0,7473	0,8681	0,9011	0,8571
2,1	2,1	0,8	0,8681	0,8132	0,8791	0,8132	0,8681	0,8484
2,1	2,1	0,9	0,8462	0,8132	0,9341	0,7363	0,8462	0,8352
...
2,5	2,5	0,7	0,8022	0,8901	0,8901	0,8132	0,9231	0,8637
2,5	2,5	0,8	0,9231	0,9121	0,8242	0,8352	0,7253	0,8440
2,5	2,5	0,9	0,8352	0,8681	0,8352	0,9121	0,8352	0,8571
2,5	2,5	1	0,7692	0,9341	0,7912	0,8571	0,8681	0,8440

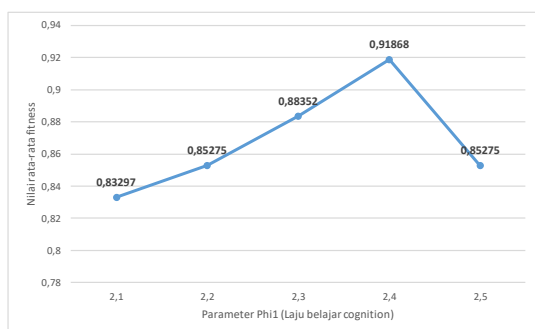
Kombinasi parameter pada PSO, yaitu $\varphi_1=2,4$, $\varphi_2=2,1$, $\omega=0,6$, $E_{max}=100$, dan $N_p=30$ adalah kombinasi parameter optimal dari PSO berdasarkan keseluruhan hasil pengujian. Nilai rata-rata *fitness* tertinggi sebesar 0,91868 dihasilkan dengan kombinasi ini. Kombinasi parameter optimal GLVQ dari hasil pengujian sebelumnya, yaitu n_w sebesar 5 digunakan pada tahap pengujian parameter PSO untuk membentuk struktur partikel yang diinisialisasi.

Ketika parameter φ_1 berubah dalam interval 2,1 sampai 2,4, maka nilai rata-rata *fitness* memiliki kecenderungan meningkat, walaupun terjadi penurunan nilai rata-rata *fitness* ketika parameter φ_1 lebih dari 2,4. Hal ini dijelaskan melalui grafik pada Gambar 5. Partikel sulit menemukan solusi vektor bobot yang optimal ketika laju belajar komponen *cognition* yang semakin kecil. Hal ini dapat terbukti dengan nilai rata-rata *fitness* terendah saat parameter φ_1 sama dengan 2,1. Laju belajar komponen *cognition* sebesar 2,4 menjadi batas optimal pada penelitian ini. Nilai rata-rata *fitness* yang dihasilkan menurun apabila nilai parameter φ_1 lebih dari 2,4.

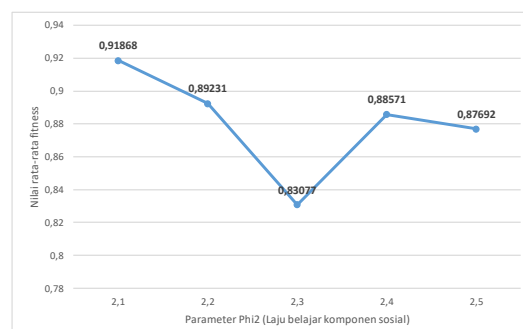
Ketika parameter φ_2 berubah dalam interval 2,1 sampai 2,5, maka nilai rata-rata *fitness* memiliki kecenderungan menurun, walaupun terjadi kenaikan nilai rata-rata *fitness* ketika parameter φ_2 berubah dari 2,3 menjadi 2,4. Hal ini dijelaskan melalui grafik pada Gambar 6. Partikel cenderung menemukan solusi vektor bobot yang optimal ketika laju belajar komponen sosial yang semakin kecil, begitupun sebaliknya. Hal ini dapat terbukti dengan nilai rata-rata *fitness* tertinggi saat parameter φ_2 sama dengan 2,1. Hasil optimasi vektor bobot yang optimal dihasilkan dari kombinasi parameter laju belajar komponen *cognition* yang lebih besar dari laju belajar komponen sosial pada penelitian ini.

Parameter ω sebesar 1,0 memberikan nilai rata-rata *fitness* terendah. Ketika parameter ω semakin besar, maka nilai rata-rata *fitness* cenderung menurun, walaupun terjadi kenaikan nilai rata-rata *fitness* ketika parameter ω berubah dari 0,5 menjadi 0,6. Kecepatan partikel diperbarui menggunakan persamaan (17) ketika parameter ω bernilai 1,0, sehingga tidak ada faktor pengontrol kelembaman partikel. Ketika kecepatan partikel diperbarui menggunakan persamaan (17), maka sekumpulan partikel cenderung terjebak pada optimum lokal [20]. Gambar 7 menampilkan hasil yang senada. Solusi vektor bobot lain yang berpotensi memberikan nilai *fitness* yang lebih baik dari 0,77802 tidak berhasil ditemukan oleh sekumpulan partikel.

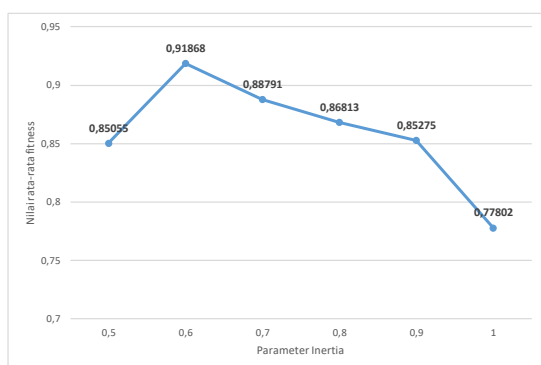
$$v_{i,j}^{t+1} = v_{i,j}^t + \varphi_1 \times r_1 \times (Pbest_{i,j}^t - x_{i,j}^t) + \varphi_2 \times r_2 \times (Gbest_j^t - x_{i,j}^t) \quad (17)$$



Gambar 5. Grafik pengaruh perubahan parameter φ_1 terhadap hasil optimasi vektor bobot menggunakan PSO



Gambar 6. Grafik pengaruh perubahan parameter φ_2 terhadap hasil optimasi vektor bobot menggunakan PSO



Gambar 7. Grafik pengaruh perubahan parameter ω terhadap hasil optimasi vektor bobot menggunakan PSO

3.4. Perbandingan Hasil Klasifikasi PSO-GLVQ dan GLVQ

Hasil perbandingan metode PSO-GLVQ dan GLVQ menggunakan sejumlah ukuran evaluasi model klasifikasi disajikan seperti pada Tabel 4. Algoritma PSO yang digunakan untuk mengoptimasi vektor bobot awal pada GLVQ menghasilkan kinerja yang lebih baik jika dibandingkan dengan vektor bobot yang diinisialisasi secara acak. Tingkat akurasi pada data uji yang dihasilkan PSO-GLVQ yaitu 0,938596. Nilai ini lebih tinggi sebesar 0,008771 dari GLVQ yang memberikan tingkat akurasi sebesar 0,929825. Tingkat kesalahan pada data uji yang dihasilkan PSO-GLVQ yaitu 0,061404. Nilai ini lebih rendah apabila dibandingkan hasil yang dihasilkan metode GLVQ, yaitu sebesar 0,070175.

Tabel 4. Hasil perbandingan ukuran evaluasi antara metode GLVQ dan PSO-GLVQ untuk klasifikasi kanker payudara

Model	TP	TN	FN	FP	Accuracy	Error rate	Recall	Precision	F2-Score
PSO-GLVQ	35	72	0	7	0,938596	0,061404	1	0,833333	0,961538
GLVQ	36	70	2	6	0,929825	0,070175	0,947368	0,857143	0,927835

Kualitas dari metode PSO-GLVQ dan GLVQ dalam mengklasifikasikan data uji kanker payudara perlu diukur dengan ukuran selain *accuracy* dan *error rate*. Model pengklasifikasi dapat memberikan hasil yang bias ketika model dilatih menggunakan data yang *imbalanced class* dan diukur hanya dengan ukuran *accuracy*. Hal tersebut dikarenakan model pengklasifikasi memberikan keberhasilan yang rendah dalam mengklasifikasikan kelas minoritas, namun memberikan keberhasilan yang tinggi dalam mengklasifikasikan kelas mayoritas [25]. Ukuran *recall*, *precision* [10], dan *F2-score* [25] dapat digunakan untuk mengetahui kinerja model ketika dilatih menggunakan data dengan *imbalanced class* pada permasalahan klasifikasi biner.

Data dari kelas minoritas mengandung lebih banyak informasi yang berguna, sehingga keberhasilan model pengklasifikasi mengidentifikasi kelas minoritas lebih penting apabila dibandingkan dengan kelas mayoritas [25]. Oleh karena itu, pada kasus klasifikasi data medis ukuran evaluasi *recall* lebih diperhatikan [25]. Nilai FN berusaha diminimumkan oleh model pengklasifikasi kanker payudara yang dikembangkan. Sangat fatal apabila pasien yang seharusnya mengidap kanker payudara ganas diklasifikasikan sebagai pasien dengan kanker payudara jinak oleh model pengklasifikasi (nilai FN semakin besar). Pasien dengan kanker payudara jinak yang diklasifikasikan oleh model pengklasifikasi sebagai pasien dengan kanker payudara ganas lebih bisa ditoleransi apabila dibandingkan dengan kasus sebelumnya, walaupun tetap model pengklasifikasi sebisa mungkin meminimumkan kedua tipe kesalahan ini.

Metode PSO-GLVQ menghasilkan *recall* yang sempurna, yaitu sebesar satu (100%), sedangkan metode GLVQ menghasilkan 0,947368. Selisih *recall* keduanya sebesar 0,052632. PSO-GLVQ berhasil memberikan label positif pada seluruh baris data uji yang memang berlabel positif secara sempurna. Metode GLVQ menghasilkan *precision* sebesar 0,857143, sedangkan metode PSO-GLVQ menghasilkan *precision* sebesar 0,833333. *F2-score* dari metode PSO-GLVQ dan GLVQ dihitung dengan memperhatikan nilai *recall* dan *precision* yang dihasilkan masing-masing metode. Metode PSO-GLVQ menghasilkan *F2-score* sebesar 0,961538. Nilai ini lebih tinggi apabila dibandingkan dengan metode GLVQ yang menghasilkan *F2-score* sebesar 0,927835. Selisih *F2-score* keduanya sebesar 0,033703.

4. Kesimpulan

Kombinasi parameter optimal dari PSO $\varphi_1 = 2,4$, $\varphi_2 = 2,1$, dan $\omega = 0,6$ menghasilkan nilai rata-rata *fitness* tertinggi sebesar 0,91868 dengan validasi *5-fold cross-validation*. Kombinasi parameter optimal dari GLVQ $\alpha = 0,1$, $n_w = 5$, *epoch* maksimum sebesar 100, dan toleransi kesalahan minimum sebesar 10^{-6} menghasilkan nilai rata-rata tingkat akurasi tertinggi sebesar 0,956044 dengan validasi *5-fold cross-validation*. Kedua kombinasi parameter optimal dari PSO dan GLVQ digunakan membentuk model PSO-GLVQ dan GLVQ. Performa *accuracy*, *error rate*, *recall*, dan *F2-score* model PSO-GLVQ lebih baik jika dibandingkan metode GLVQ pada 20% data uji yang telah ditetapkan. Model PSO-GLVQ memberikan *accuracy*, *error rate*, *recall*, dan *F2-score* masing-masing sebesar 0,938596, 0,061404, 1, 0,961538. Model GLVQ memberikan *accuracy*, *error rate*, *recall*, dan *F2-score* masing-masing sebesar 0,929825, 0,070175, 0,947368, 0,927835.

References

- [1] N. P. D. Rakasiwi, G. B. Setiawan, and I. G. N. W. Aryana, "Karakteristik Kanker Payudara Dengan Metastasis Tulang Tahun 2015-2017 Di RSUP Sanglah Denpasar," *JURNAL MEDIKA UDAYANA*, vol. 9, no. 1, pp. 17–22, Jan. 2020, doi: 10.24843.MU.2020.V9.i1.P04.
- [2] F. S. Nugraha, M. J. Shidiq, and S. Rahayu, "Analisis Algoritma Klasifikasi Neural Network Untuk Diagnosis Penyakit Kanker Payudara," *Jurnal Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 149–156, Aug. 2019, doi: 10.33480/pilar.v15i2.601.
- [3] C. Song, S. Sugiharto, and O. D. Wahyuni, "Edukasi Kanker Payudara Dan Deteksi Dinipada Kader Wanita Kelurahan Tomang," *Jurnal Bakti Masyarakat Indonesia*, vol. 4, no. 2, pp. 351–359, Aug. 2021.
- [4] E. Susilowati, A. T. Hapsari, M. Efendi, and P. E. Kresnha, "Diagnosa Penyakit Kanker Payudaramenggunakan Metode K-Means Clustering," *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol. 10, no. 1, pp. 27–32, 2019, [Online]. Available: <https://jurnal.umj.ac.id/index.php/just-it>
- [5] H. Wijaya, "Optimization of Application of Genetic Algorithm Using C4.5 Method to Predict Breast Cancer Disease," *bit-Tech*, vol. 2, no. 1, pp. 1–9, 2019, [Online]. Available: <http://jurnal.kdi.or.id/index.php/bt>
- [6] C. Aroef, Y. Rivan, and Z. Rustam, "Comparing random forest and support vector machines for breast cancer classification," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 815–821, Apr. 2020, doi: 10.12928/TELKOMNIKA.V18I2.14785.
- [7] H. Oktavianto and R. P. Handri, "Analisis Klasifikasi Kanker Payudara Menggunakan Algoritma Naive Bayes," *Informatics Journal*, vol. 8, no. 2, pp. 45–54, 2019, [Online]. Available: <https://archive.ics.uci.edu/ml/>.
- [8] D. Gustiar, S. H. Sitorus, and D. M. Midyanti, "Penerjemahan Bahasa Isyarat Menggunakan Metode Generalized Learning Vector Quantization (GLVQ)" *Coding : Jurnal Komputer dan Aplikasi*, vol. 8, no. 03, pp. 1–8, 2020.
- [9] C. Diao, D. Kleyko, J. M. Rabaey, and B. A. Olshausen, "Generalized Learning Vector Quantization for Classification in Randomized Neural Networks and Hyperdimensional Computing," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jun. 2021, pp. 1–9. doi: 10.1109/IJCNN52387.2021.9533316.
- [10] T. Villmann, A. Bohnsack, and M. Kaden, "Can learning vector quantization be an alternative to SVM and deep learning? - Recent trends and advanced variants of learning vector quantization for classification learning," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 1, pp. 65–81, 2017, doi: 10.1515/jaiscr-2017-0005.
- [11] R. Arniantya, B. D. Setiawan, and P. P. Adikara, "Optimasi Vektor Bobot Pada Learning Vector Quantization Menggunakan Algoritme Genetika Untuk Identifikasi Jenis Attention Deficit Hyperactivity Disorder Pada Anak," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, pp. 679–687, Feb. 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] W. A. Setyowati and W. F. Mahmudy, "Optimasi Vektor Bobot Pada Learning Vector Quantization Menggunakan Particle Swarm Optimization Untuk Klasifikasi Jenis Attention Deficit Hyperactivity Disorder (ADHD) Pada Anak Usia Dini," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 11, pp. 4428–4437, Nov. 2018.
- [13] I. Romadhona, I. Cholissodin, and Marji, "Penerapan Algoritme Particle Swarm Optimization-Learning Vector Quantization(PSO-LVQ) Pada Klasifikasi Data Iris," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 6418–6428, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [14] K. D. Prebiana, I. G. S. Astawa, and I. W. Supriana, "Optimasi Pembobotan Jaringan Syaraf Tiruan Pada Klasifikasi Kanker Payudara," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 9, no. 1, pp. 151–159, Aug. 2020.
- [15] E. Purwaningsih, "Penerapan Particle Swarm Optimization pada Metode Neural Network untuk Perawatan Penyakit Kulitmelalui Immunotherapy," *JUSTIN: Jurnal Sistem dan Teknologi Informasi*, vol. 8, no. 2, pp. 207–211, 2020, doi: 10.26418/justin.v8i2.39869.
- [16] Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *International Journal of Informatics and Information System*, vol. 4, no. 1, pp. 13–20, Mar. 2021, [Online]. Available: <http://archive.ics.uci.edu/ml>.
- [17] C. Leys, M. Delacre, Y. L. Mora, D. Lakens, and C. Ley, "How to classify, detect, and manage univariate and multivariate outliers, with emphasis on pre-registration," *International Review of Social Psychology*, vol. 32, no. 1, pp. 1–10, 2019, doi: 10.5334/irsp.289.

- [18] Suyanto, *Data Mining Untuk Klasifikasi dan Klasterisasi Data*, Revisi. Bandung: Informatika Bandung, 2018.
- [19] M. S. N. van Delsen, A. Z. Wattimena, and S. D. Saputri, "Penggunaan Metode Analisis Komponen Utama Untuk Mereduksi Faktor-Faktor Inflasi Di Kota Ambon," *Jurnal Ilmu Matematika dan Terapan*, vol. 11, no. 2, pp. 109–118, Dec. 2017, Accessed: Mar. 05, 2022. [Online]. Available: <https://ojs3.unpatti.ac.id/index.php/barekeng/article/view/352>
- [20] Suyanto, *Swarm Intelligence Komputasi Modern Untuk Optimasi dan Big Data Mining*. Bandung: Informatika Bandung, 2017.
- [21] S. Ramzini, D. E. Ratnawati, and S. Anam, "Penerapan Metode Learning Vector Quantization (LVQ) untuk Klasifikasi Fungsi Senyawa Aktif Menggunakan Notasi Simplified Molecular Input Line System (SMILES)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 6160–6168, Dec. 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [22] L. A. A. R. Putri and S. Hartati, "Klasifikasi Genre Musik Menggunakan Learning Vector Quantization dan Self Organizing Map," *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana*, vol. 9, no. 1, pp. 14–22, Apr. 2016.
- [23] M. D. Ariyawan, I. G. A. Wibawa, and L. A. A. R. Putri, "Diagnosis of Heart Disease Using Generalized Learning Vector Quantization (GLVQ) and Genetic Algorithms Methods," *Jurnal Ilmu Komputer*, vol. 13, no. 1, pp. 56–64, 2020.
- [24] S. Saralajew, L. Holdijk, M. Rees, and T. Villmann, "Robustness of Generalized Learning Vector Quantization Models against Adversarial Attacks," in *International Workshop on Self-Organizing Maps*, Feb. 2019, pp. 189–199. doi: 10.1007/978-3-030-19642-4_19.
- [25] D. Devarriya, C. Gulati, V. Mansharamani, A. Sakalle, and A. Bhardwaj, "Unbalanced breast cancer data classification using novel fitness functions in genetic programming," *Expert Systems with Applications*, vol. 140, Feb. 2020, doi: 10.1016/j.eswa.2019.112866.

Klasifikasi Berita Hoaks Covid-19 Menggunakan Kombinasi Metode *K-Nearest Neighbor* dan *Information Gain*

Marissa Audina^{a1}, AAIN Eka Karyawati^{a2}, I Wayan Supriana^{a3}, I Ketut Gede Suhartana^{a4}, I Gede Santi Astawa^{a5}, I Wayan Santiyasa^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Bali, Indonesia

¹marissaaudina@gmail.com

²eka.karyawati@unud.ac.id

³wayan.supriana@unud.ac.id

⁴ikg.suhartana@unud.ac.id

⁵santi.astawa@unud.ac.id

⁶santiyasa@unud.ac.id

Abstract

News is one of information resources that is being used by the public. However, not all news circulating in digital media are facts. Some people take the opportunity to share unfounded and irresponsible news. Since the Covid-19 pandemic hit Indonesia, hoax news about the pandemic has increasingly circulated in digital media. In this study, the author builds a model that can classify hoax news using the K-Nearest Neighbor method combined with the Information Gain feature selection. The data used are factual news data and hoax news data in Indonesian language. Evaluation is done by measuring the performance of the K-Nearest Neighbor model without feature selection and model performance by implementing Information Gain feature selection. The K-Nearest Neighbor model without feature selection with a value of $k=5$ obtained precision, recall, F1-Score, and accuracy performance of 87.5%, 96.5%, 91.8%, and 91.6%, respectively. While the K-Nearest Neighbor model with a combination of 0.5% Information Gain threshold feature selection with a value of $k=3$ obtained precision, recall, F1-Score, and accuracy performance of 93.3%, 96.6%, 95%, and 95%, respectively.

Keywords: *K-Nearest Neighbor*, *Information Gain*, TF-IDF, Klasifikasi Teks, Berita Hoaks

1. Pendahuluan

Berita digunakan oleh masyarakat sebagai salah satu sumber informasi. Tidak semua berita yang beredar di media digital adalah fakta. Beberapa individu atau kelompok mengambil kesempatan untuk menyebarkan berita atau informasi yang tidak dapat dipertanggungjawabkan kebenarannya dan terdapat indikasi *hoax*. [1] Data dari laman resmi kominfo.go.id menyatakan bahwa sebanyak 800.000 situs terindikasi sebagai situs penyebaran hoaks di Indonesia. Menurut Kamus Besar Bahasa Indonesia, hoaks (bahasa Inggris: *hoax*) memiliki makna informasi bohong. Sejak pandemi Covid-19 melanda Indonesia, berita hoaks mengenai pandemi tersebut semakin banyak beredar di media digital. Data terbaru dari Kementerian Komunikasi dan Informatika, sebanyak 5457 sebaran hoaks Covid-19 sudah ditindaklanjuti sejak 23 Januari 2020 hingga 18 Maret 2022 [2].

Berita-berita yang didapatkan dari media dapat diklasifikasikan menjadi berita hoaks dan berita fakta. Pengklasifikasian berita tersebut membutuhkan suatu metode atau algoritma agar tidak menggunakan cara manual dan menghabiskan waktu yang lama. Peranan informatika dibutuhkan dalam hal ini untuk membangun suatu model klasifikasi yang dapat mengkategorikan dua jenis berita tersebut. Penelitian mengenai klasifikasi berita hoaks telah dilakukan oleh beberapa peneliti seperti penelitian klasifikasi berita *clickbait* menggunakan *K-Nearest Neighbor* yang menghasilkan akurasi terbaik 71% dengan parameter nilai $k=11$ pada skenario 80% data latih dan 20% data uji [3]. Kemudian penelitian mengenai identifikasi hoaks berbasis *text mining* menggunakan *K-Nearest Neighbor* menghasilkan akurasi sebesar 75.4% pada nilai k optimal

bernilai 4 [4]. Penelitian selanjutnya adalah analisis sentimen terhadap ulasan pengguna MRT Jakarta menggunakan *Information Gain* dan *Modified K-Nearest Neighbor* dengan peningkatan akurasi 4-5% setelah menggunakan seleksi fitur *Information Gain* [5].

Berdasarkan penelitian yang dilakukan sebelumnya, pada penelitian ini penulis melakukan klasifikasi berita hoaks menggunakan metode *K-Nearest Neighbor* yang dikombinasikan dengan seleksi fitur *Information Gain*. Penulis berharap bahwa dengan menggunakan kombinasi metode ini dapat menghasilkan performa *precision*, *recall*, *f1-score*, dan akurasi yang lebih baik dibandingkan penelitian sebelumnya.

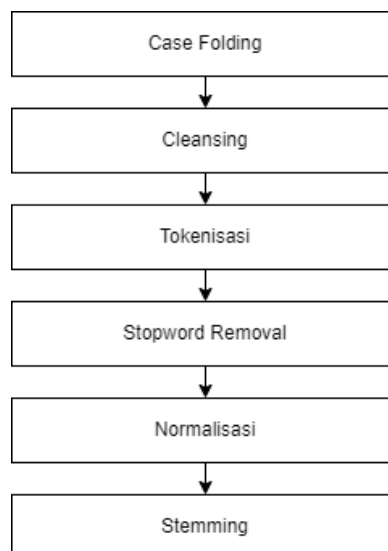
2. Metode Penelitian

2.1 Dataset

Jenis data sekunder digunakan pada penelitian ini. Dataset diperoleh dalam bentuk berita yang berkaitan dengan Covid-19. Data berita hoaks bersumber dari <https://cekfakta.com>, sedangkan data berita fakta bersumber dari <https://detik.com>. Bagian berita yang digunakan adalah isi berita. Data berjumlah 300 dengan format *file *.csv* yang meliputi 150 berita hoaks dan 150 berita fakta yang bersumber dari media internet. Seluruh data sudah dilabeli oleh lembaga media internet tersebut. Data berita kemudian dibagi dengan presentase data latih sebesar 80% dan data uji sebesar 20%. Data latih tersebut kemudian dibagi lagi menjadi data latih dan data validasi untuk digunakan dalam proses pelatihan model dengan menggunakan *N-Fold Cross Validation* dengan nilai $N = 10$.

2.2 Preprocessing

Sebelum melakukan tahap pembobotan, data terlebih dahulu melalui tahapan *preprocessing*. *Preprocessing* adalah proses yang dilakukan untuk mengolah data ulasan yang belum terstruktur menjadi terstruktur sehingga data dapat dilanjutkan ke proses klasifikasi. Adapun alur *preprocessing* seperti pada Gambar 1.



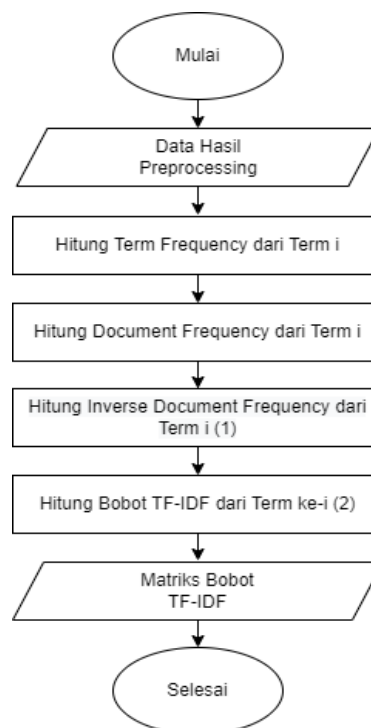
Gambar 1. Alur *Preprocessing*

Data dokumen berita akan melalui proses *case folding* yaitu proses untuk membuat bentuk data yang sama yaitu hanya berisi huruf kecil. *Case folding* dilakukan agar data yang ada menjadi sama rata [6]. Kemudian proses *cleansing* untuk menghapus seluruh karakter yang berupa HTML ataupun web yang tidak memiliki makna atau kaitan terhadap analisis sentimen. Pada proses ini juga dilakukan proses penghapusan *punctuation* atau tanda baca. *Tokenization* adalah proses pemisahan kata dalam suatu paragraf atau kalimat sehingga terbagi menjadi token-token tertentu. *Stopword removal* merupakan proses penghapusan kata atau fitur yang tidak berpengaruh dan tidak penting terhadap klasifikasi. Penghapusan ini dilakukan untuk membuat proses klasifikasi berjalan efisien [6]. Kemudian proses normalisasi, yaitu mengubah dan

mengembalikan bentuk penulisan tidak baku ke bentuk penulisan yang sesuai dengan KBBI. Pada proses ini digunakan korpus yang berisi kumpulan kata tidak baku dan bentuk baku dari kata tersebut. Proses terakhir adalah *stemming* yang berfungsi agar kata-kata berimbuhan (awalan dan akhiran) dapat diekstraksi ke bentuk akarnya atau dapat dikatakan sebagai kata dasar. *Stemming* dilakukan untuk menyamakan data yang berbeda penulisannya [6].

2.3 Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF merupakan metode pembobotan kata untuk menentukan keterhubungan kata pada suatu dokumen [4]. Data yang diproses akan diubah menjadi data numerik dengan metode pembobotan TF-IDF, yang merupakan penggabungan dua konsep yaitu TF dan IDF. *Term Frequency* (TF) merupakan frekuensi dari kemunculan kata dalam sebuah dokumen, sedangkan *Inverse Document Frequency* (IDF) adalah perhitungan dari distribusi kata secara luas pada koleksi dokumen. Kata atau *term* yang muncul di dalam sebagian besar dokumen akan mempunyai nilai IDF mendekati nol. Adapun tahapan dari TF-IDF ditunjukkan oleh Gambar 2.



Gambar 2. TF-IDF

- Menghitung jumlah kemunculan *term i* dalam dokumen *j* ($tf_{i,j}$).
- Menghitung jumlah dokumen yang mengandung *term i* (df_i)
- Menghitung nilai bobot *inverse document frequency* (*idf*) dengan menggunakan persamaan :

$$idf_i = \log \left(\frac{N}{df_i} \right) \quad (1)$$

Keterangan :

N = jumlah dokumen secara keseluruhan

- Menghitung nilai bobot TF-IDF dengan menggunakan persamaan :

$$w_{i,j} = tf_{i,j} \times idf_i \quad (2)$$

Keterangan :

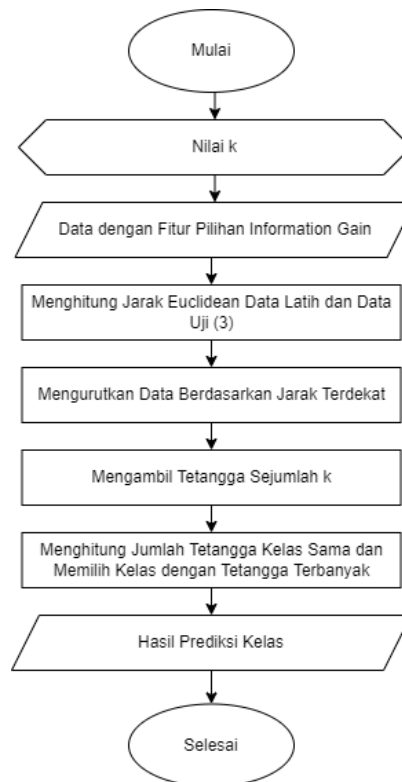
$w_{i,j}$ = bobot *term* i terhadap dokumen j

$tf_{i,j}$ = frekuensi *term* i pada dokumen j

idf_i = nilai bobot IDF *term* i

2.4 *K-Nearest Neighbor* (KNN)

Metode KNN sering diterapkan pada *data mining* dan *text mining*. KNN merupakan metode pengklasifikasian objek berdasarkan tetangga yang paling dekat dengannya. KNN memberikan keanggotaan kelas ke data berdasarkan mayoritas tetangganya, dengan objek yang ditetapkan ke kelas yang paling umum di antara k tetangga terdekatnya (k adalah bilangan bulat positif bernilai kecil). Pada penelitian ini, fitur kata yang sudah melalui proses pembobotan TF-IDF akan menghasilkan suatu matriks yang berisikan bobot nilai TF-IDF dengan dokumen sebagai baris dan fitur kata sebagai kolom. Setiap vektor dokumen dengan nilai bobot fitur pada data latih akan dihitung jaraknya dengan vektor pada data uji. Adapun tahapan metode ditunjukkan oleh Gambar 3.



Gambar 3. Klasifikasi *K-Nearest Neighbor*

Tahapan dari KNN adalah sebagai berikut :

- Menentukan jumlah tetangga k yang akan digunakan.
- Melakukan perhitungan jarak antara data latih dan data uji menggunakan rumus persamaan *Euclidean distance* di bawah [7]:

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (3)$$

Keterangan :

d = jarak

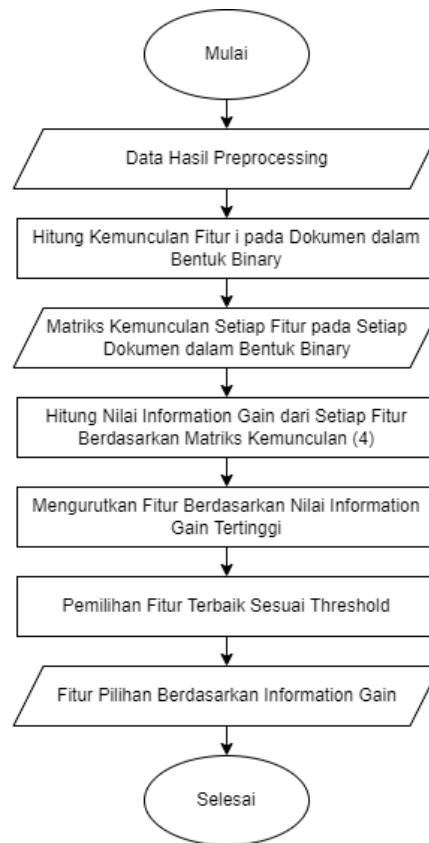
x = data uji (*data testing*)

y = data latih (*data training*)

c. Mendapatkan hasil pengklasifikasian

2.5 Information Gain

Information gain banyak digunakan pada klasifikasi data tekstual sebagai metode seleksi fitur. *Information gain* melakukan perhitungan mengenai pengaruh suatu fitur terhadap keseragaman kelas pada data. Seleksi fitur ini menghitung hadir tidaknya suatu kata yang berkontribusi pada pengambilan keputusan klasifikasi yang benar di kelas apapun [8]. Data tersebut dipecah menjadi sub data dengan nilai fitur tertentu. Jika suatu fitur memperoleh nilai *information gain* yang tinggi, maka fitur tersebut dikatakan memiliki pengaruh pada proses klasifikasi. Adapun tahapan *Information Gain* ditunjukkan oleh Gambar 4.



Gambar 4. *Information Gain*

Langkah-langkah dari *Information Gain* adalah sebagai berikut :

- Isi data dan label data sebagai input.
- Melakukan perhitungan nilai *Information Gain* dari setiap fitur dengan rumus berikut [9] :

$$\text{Information Gain } I(t_j) \text{ of } t_j = -\sum_{r=1}^k \frac{n_r}{n} \log \left(\frac{n_r}{n} \right) - E(t_j) \quad (4)$$

Dimana $E(t_j)$ adalah *entropy* bersyarat yang dihitung dengan persamaan :

$$E(t_j) = -\sum_{r=1}^k \left\{ \left[\frac{n(t_j)}{n} \right] P(c_r | t_j) \cdot \log [P(c_r | t_j)] + \left[\frac{n-n(t_j)}{n} \right] P(c_r | \neg t_j) \cdot \log [P(c_r | \neg t_j)] \right\} \quad (5)$$

Keterangan :

n_r : jumlah total dokumen dengan kelas r .

- n : jumlah total dokumen.
- n(tj) : banyaknya dokumen yang mengandung term tj dari korpus berukuran $n \geq n(tj)$.
- $P(c_r|t_j)$: peluang *term* tj terdapat pada kelas r dalam dokumen.
- $P(c_r|\neg t_j)$: peluang *term* tj tidak terdapat pada kelas r dalam dokumen.
- c. Mengurutkan fitur-fitur berdasarkan nilai *Information Gain* tertinggi.
- d. Memilih fitur terbaik sesuai dengan *threshold* yang diberikan.

2.6 Evaluasi

Pada tahap evaluasi, *confusion matrix* digunakan untuk menghitung akurasi, *recall*, *precision*, dan *error rate*. *Confusion matrix* dapat digunakan untuk mengevaluasi kualitas *classifier*. Pada *confusion matrix* dua kelas, matriks menunjukkan *true positives*, *true negatives*, *false positives*, dan *false negatives*. *Confusion matrix* untuk dua kelas ditunjukkan pada Tabel 1 [10].

Tabel 1. Confusion Matrix

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

Keterangan :

- TP = *True Positive* (total prediksi benar dari data positif)
- FN = *False Negative* (total prediksi salah dari data positif)
- TN = *True Negative* (total prediksi benar dari data negatif)
- FP = *False Positive* (total prediksi salah dari data negatif)

Adapun rumus untuk menghitung *precision*, *recall*, *F1-Score*, dan akurasi adalah sebagai berikut:

$$Precision = \frac{TP}{(TP+FP)} \tag{6}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{7}$$

$$F1-Score = \frac{2 \times recall \times precision}{(recall+precision)} \tag{8}$$

$$Akurasi = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{9}$$

3. Hasil dan Pembahasan

Sebanyak 80% dari total data digunakan pada tahap pelatihan dan validasi. Beberapa eksperimen dilakukan yaitu model KNN tanpa seleksi fitur, model KNN dengan eksperimen beberapa nilai *threshold Information Gain*, dan pengujian kedua model terbaik dengan menggunakan data baru yaitu 20% data uji yang sudah disiapkan sebelumnya. Perubahan nilai k dilakukan pada metode *K-Nearest Neighbor*. Perubahan nilai k pada eksperimen adalah k=3, k=5, k=7, k=9, dan k=11. Pengujian terhadap *threshold* dilakukan pada seleksi fitur *Information Gain*. *Threshold* adalah persentase jumlah fitur yang terseleksi dari seluruh fitur yang telah diurutkan berdasarkan nilai *Information Gain* tertinggi. *Threshold* yang digunakan adalah 50%, 25%, 20%, 10%, 5%, 2%, 1%, 0.5%, 0.2%, dan 0.1%. Pada setiap iterasi *10-Fold Cross Validation*, akan dihitung rata-rata performa *F1-Score* dan akurasi dengan menggunakan persamaan (8) dan (9). Nilai k yang menghasilkan performa *F1-Score* tertinggi dipilih sebagai model terbaik yang kemudian digunakan pada proses *testing* data baru. Nilai k yang

menghasilkan *F1-Score* tertinggi memiliki makna bahwa hasil prediksi berita hoaks akan lebih akurat kebenarannya.

Setelah dilakukan proses pelatihan dan validasi terhadap model *K-Nearest Neighbor* menggunakan *10-Fold Cross Validation*, didapatkan nilai k dengan performa *F1-Score* terbaik yaitu k = 5 dengan nilai *F1-Score* 93.9% serta akurasi 94.2%. Sehingga, nilai k = 5 dipilih sebagai model terbaik dan akan digunakan pada proses pengujian data uji dengan menggunakan data baru yang belum pernah melalui proses pelatihan dan validasi.

Tabel 2. Hasil Evaluasi Pengujian *K-Nearest Neighbor* tanpa Seleksi Fitur

Nilai k	Ukuran Evaluasi (Rata-Rata Fold)	
	F1-Score	Akurasi
3	0.908	0.908
5	0.939	0.942
7	0.927	0.929
9	0.934	0.938
11	0.935	0.938

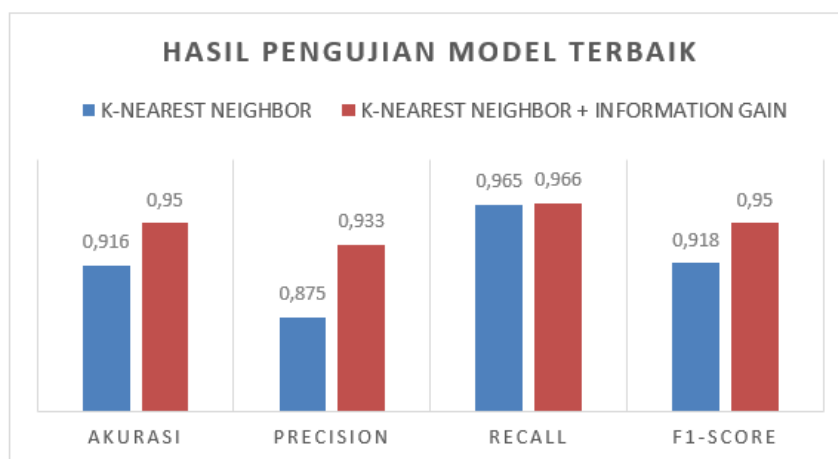
Jumlah keseluruhan fitur adalah 4934 fitur. Pada setiap eksperimen perubahan nilai *threshold*, dilakukan juga perubahan nilai k sehingga didapatkan kombinasi *threshold* dan parameter nilai k yang menghasilkan performa terbaik. Kombinasi *threshold* 0.5% dengan parameter nilai k=3 menghasilkan performa terbaik pada proses pelatihan dan validasi yaitu *F1-Score* sebesar 97.3%, serta akurasi sebesar 97.5%, sehingga kombinasi ini dipilih menjadi model terbaik. *Threshold* ini menyeleksi sekitar 25 fitur dengan nilai *Information Gain* tertinggi. Fitur-fitur tersebut terdiri dari beberapa fitur yang dominan pada dokumen kelas hoaks, dan beberapa fitur lainnya dominan pada dokumen kelas fakta. Fitur-fitur tersebut menjadi ciri khas dari kedua kelas berita sehingga model dapat mengklasifikasikan berita dengan baik, ditunjukkan oleh performa akurasi yang dihasilkan.

Tabel 3. Hasil Evaluasi Pengujian *K-Nearest Neighbor* dengan Seleksi Fitur

Threshold	Ukuran Evaluasi (Rata-Rata Fold)	
	F1-Score	Akurasi
50%	0.255	0.575
25%	0.271	0.579
20%	0.266	0.58
10%	0.869	0.875
5%	0.912	0.913
2%	0.952	0.954
1%	0.958	0.958
0.5%	0.973	0.975
0.2%	0.937	0.942
0.1%	0.925	0.929

Dua model terbaik yang dipilih adalah model yang menghasilkan *F1-Score* terbaik, yaitu model KNN tanpa seleksi fitur dengan nilai k = 5, dan model KNN dengan kombinasi seleksi fitur *Information Gain threshold* 0.5% dengan nilai k=3. Kedua model tersebut kemudian diuji kembali menggunakan data baru yang belum pernah melewati tahap pelatihan dan validasi sebelumnya.

Setelah melakukan pengujian terhadap kedua model dengan menggunakan data baru, hasil performa model ditunjukkan pada Gambar 5.



Gambar 5. Hasil Pengujian Model Terbaik

Pada Gambar 5 ditunjukkan bahwa kombinasi metode KNN dan *Information Gain* menghasilkan performa yang lebih baik dalam klasifikasi berita hoaks. Terdapat peningkatan pada setiap performa evaluasi model. Model KNN tanpa seleksi fitur dengan nilai $k=5$ menghasilkan performa *precision*, *recall*, *F1-Score*, dan akurasi secara berturut-turut yaitu 87.5%, 96.5%, 91.8%, dan 91.6%. Sedangkan model KNN dengan kombinasi seleksi fitur *Information Gain threshold* 0.5% dengan nilai $k=3$ menghasilkan performa *precision*, *recall*, *F1-Score*, dan akurasi secara berturut-turut yaitu 93.3%, 96.6%, 95%, dan 95%. Pada nilai *recall* kedua model, tidak terdapat perbedaan yang signifikan. *Recall* menghitung presentase prediksi kelas hoaks benar terhadap seluruh data yang kelas sebenarnya adalah hoaks. Hal ini menandakan bahwa kedua model terbaik yang diuji sama-sama dapat dengan baik mengklasifikasi berita hoaks ke dalam kelas hoaks dengan sedikit kesalahan pada klasifikasi berita kelas hoaks ke dalam kelas fakta.

4. Kesimpulan

Setelah dilakukan validasi model dengan *10-Fold Cross Validation*, nilai $k=5$ dipilih menjadi model terbaik pada eksperimen metode *K-Nearest Neighbor*. Pada pengujian data baru, model *K-Nearest Neighbor* tanpa seleksi fitur dengan nilai $k=5$ menghasilkan performa *precision*, *recall*, *F1-Score*, dan akurasi secara berturut-turut yaitu 87.5%, 96.5%, 91.8%, dan 91.6%. Pada eksperimen seleksi fitur *Information Gain*, kombinasi *threshold* 0.5% dengan parameter nilai $k=3$ adalah kombinasi yang dipilih menjadi model terbaik. *Threshold* ini menyeleksi sekitar 25 fitur dengan nilai *Information Gain* tertinggi. Pada pengujian data baru, model *K-Nearest Neighbor* dengan seleksi fitur *Information Gain* menghasilkan performa *precision*, *recall*, *F1-Score*, dan akurasi secara berturut-turut yaitu 93.3%, 96.6%, 95%, dan 95%. Sehingga, dapat disimpulkan bahwa pada klasifikasi berita hoaks dengan data yang digunakan pada penelitian ini, kombinasi metode *K-Nearest Neighbor* dan *Information Gain* menghasilkan performa *precision*, *recall*, *F1-Score*, dan akurasi yang lebih tinggi dibandingkan dengan metode *K-Nearest Neighbor* tanpa seleksi fitur.

Daftar Pustaka

- [1] C. Juditha, "Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya," *J. Pekommas*, vol. 3, no. 1, pp. 31–44, 2018, [Online]. Available: <https://jurnal.kominfo.go.id/index.php/pekommas/article/view/2030104>.
- [2] Kominfo.go.id, "Penanganan Sebaran Konten Hoaks Covid-19 Jumat (18/02/2022)," 2022. <https://kominfo.go.id/content/detail/40067/penanganan-sebaran-konten-hoaks->

- covid-19-jumat-18022022/0/infografis (accessed Mar. 26, 2022).
- [3] R. Sagita, U. Enri, and A. Primajaya, "Klasifikasi Berita Clickbait Menggunakan K-Nearest Neighbor (KNN)," *JOINS (Journal Inf. Syst.*, vol. 5, no. 2, pp. 230–239, 2020, doi: 10.33633/joins.v5i2.3705.
 - [4] I. W. Santiyasa, G. P. A. Brahmantha, I. W. Supriana, I. G. G. A. Kadyanan, I. K. G. Suhartana, and I. B. M. Mahendra, "Identification of Hoax Based on Text Mining Using K-Nearest Neighbor Method," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 10, no. 2, pp. 217–226, 2021, doi: 10.24843/jlk.2021.v10.i02.p04.
 - [5] A. A. Paramitha, Indriati, and Y. A. Sari, "Analisis Sentimen Terhadap Ulasan Pengguna MRT Jakarta Menggunakan Information Gain dan Modified K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 4, pp. 1125–1132, 2020.
 - [6] K. D. Yonatha Wijaya and A. A. I. N. E. Karyawati, "The Effects of Different Kernels in SVM Sentiment Analysis on Mass Social Distancing," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 9, no. 2, p. 161, 2020, doi: 10.24843/jlk.2020.v09.i02.p01.
 - [7] H. P. Hadi and T. S. Sukamto, "Klasifikasi Jenis Laporan Masyarakat Dengan K-Nearest Neighbor Algorithm," *JOINS (Journal Inf. Syst.*, vol. 5, no. 1, pp. 77–85, 2020, doi: 10.33633/joins.v5i1.3355.
 - [8] A. B. P. Negara, H. Muhandi, and I. M. Putri, "Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes dan Seleksi Fitur Information Gain," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 3, pp. 599–606, 2020, doi: 10.25126/jtiik.2020711947.
 - [9] C. C. Aggarwal and C. C. Aggarwal, *Machine Learning for Text: An Introduction*. 2018.
 - [10] M. A. Imron and B. Prasetyo, "Improving Algorithm Accuracy K-Nearest Neighbor Using Z-Score Normalization and Particle Swarm Optimization to Predict Customer Churn," *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 56–62, 2020, [Online]. Available: <https://shmpublisher.com/index.php/joscecx/article/view/7%0Ahttps://shmpublisher.com/index.php/joscecx/index>.

This page is intentionally left blank.

Penerapan Steganography Untuk Perlindungan Hak Cipta Menggunakan Metode Least Significant Bit (LSB)

I Gusti Ngurah Bagus Pramana Putra^{a1}, I Ketut Gede Suhartana^{a2}, I Komang Ari Mogi^{a3}, Cokorda Rai Adi Pramatha^{a4}, I Putu Gede Hendra Suputra^{a5}, I Gede Arta Wibawa^{a6},

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Bali, Indonesia

¹ngurahpramana170@gmail.com

²ikg.suhartana@unud.ac.id

³arimogi@unud.ac.id

⁴cokorda@unud.ac.id

⁵hendra.suputra@unud.ac.id

⁶gede.arta@unud.ac.id

Abstract

Lontar as one of the manuscripts (ancient manuscripts) which is the cultural heritage of the ancestors which is unique in recording and transferring traditional knowledge which is done through writing in the Lontar media. However, the slow pace of ejection made the manuscript very vulnerable for various reasons. The damage resulted in an increase in the information contained in Lontar, then the media was converted to digital. If these various types of letters are digitized, what will be the proof of authenticity that these lontar scripts are His? In addition, this digitization can also be used to develop access for triplets and the general public to their knowledge. From these problems, this research was conducted to build a desktop-based application that implements steganography by hiding secret messages into the PNG extension by utilizing the Least Significant Bit (LSB) method as a means of copyright protection in the Balinese Lontar Manuscript. This research succeeded in doing the insertion only with the last 3-2-3 bit formation in the RGB channel, so between the cover image and the stegoimage there will be no significant difference to the human sense of sight even though the inserted message is 100% of the maximum capacity of the image. The test results prove that there is an increase in security and the imperceptibility value is maintained. This is evidenced by the results of the average MSE value of 0.01856775 dB and PSNR 97.22405 dB. Then the user who first encrypts the file has definitive proof of copyright ownership and data security.

Keywords: *Steganography, Least Significant Bit (LSB), Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR)*

1. Pendahuluan

Lontar sebagai salah satu contoh manuskrip (naskah kuno) yang merupakan warisan kebudayaan dari leluhur yang memiliki Keunikan cara perekaman dan transfer pengetahuan tradisional dengan menulis di media daun lontar. Meski agak berbeda dengan perlakuan terhadap naskah-naskah Lontar kuno yang semakin ditinggalkan oleh peradaban modern di belahan dunia lain, naskah-naskah yang ditulis di atas daun lontar merupakan bagian aktif dari budaya literasi masyarakat Bali modern. Seiring dengan praktik budaya dan dukungan sumber daya alam, bagi orang Bali sendiri, Lontar adalah kitab suci yang dipelajari tidak hanya untuk disucikan, tetapi juga untuk digunakan sebagai pedoman dalam kehidupan sehari-hari (suluh nikang prabha). Namun lambat laun usia lontar tersebut membuat material naskah lontar sangat rentan terhadap kerusakan dari berbagai penyebab seperti jamur, kelembaban, invansi serangga, dan kontak tangan manusia sehingga lontar-lontar tua sangat rentan terhadap pelapukan. Adanya pelapukan tersebut berdampak pada hilangnya informasi yang terdapat pada lontar. Dengan adanya faktor tersebut, perlu dilakukan pelestarian seperti melakukan alih media ke digital. Selain itu, digitalisasi ini dapat digunakan untuk memberikan akses pengetahuan kepada pengunjung dan masyarakat umum.

Sementara itu, kebutuhan akan keamanan dan kerahasiaan data atau informasi semakin meningkat, terlebih lagi banyak jenis naskah lontar yang akan digitalisasi. Jika manuskrip lontar dari berbagai jenis didigitalkan, apa bukti definitif bahwa manuskrip lontar ini ada di tangan mereka? Oleh karena itu, media diperlukan untuk melindungi kepemilikan hak cipta atas manuskrip digital dengan cara menyisipkan identitas kepemilikan ke dalam manuskrip lontar bali yang telah di digitalisasi. Steganografi adalah seni dan ilmu menyembunyikan pesan dalam suatu media sehingga tidak ada orang lain selain pengirim dan penerima yang mengetahui atau mengetahui bahwa pesan rahasia itu benar-benar ada. Dalam steganografi, ia berkembang dengan menyembunyikan informasi pada file media digital, yang dapat berupa media gambar, audio, atau video, khususnya pada naskah lontar yang akan digunakan yaitu dengan format PDF[1]. Terdapat penelitian dengan mengimplementasikan metode Least Significant Bit (LSB). Seperti penelitian [2], pada penelitian ini mengimplementasikan metode (LSB) dan metode (EOF) untuk menyisipkan pesan teks kedalam file video. Penelitian proses *embedding* teks LSB memerlukan waktu lebih sedikit dibandingkan EOF dan sebaliknya proses ekstraksi EOF memerlukan waktu lebih sedikit dibandingkan LSB. Kemudian terdapat penelitian [3], pada penelitian ini menghasilkan kombinasi Teknik steganografi dan kriptografi dengan metode LSB-RSA. Dalam eksperimen membuktikan adanya peningkatan keamanan serta nilai imperceptibility yang tetap terjaga. Hal ini membuktikan dengan hasil PSNR 57.2258dB, MSE 0.1232dB, metode ini juga tahan terhadap serangan salt and papper.

Berdasarkan penjelasan tersebut, penulis ingin melakukan penelitian yang akan digunakan untuk membuat aplikasi desktop yang menerapkan metode least significant bit (LSB) untuk menyimpan pesan pada gambar. Least Significant Bit (LSB) menambahkan bit-bit data yang akan disembunyikan (message)[4], metode ini melakukan penyimpanan data dengan mengganti bit-bit yang tidak signifikan (paling sedikit piksel) pada file yang berisi (image) dengan bit file untuk penyimpanan. Untuk meningkatkan keamanan, digunakan kombinasi steganografi dan kriptografi, di mana pesan rahasia dienkripsi terlebih dahulu menggunakan algoritma Advanced Encryption Standard (AES) untuk memungkinkan penerapan yang efisien, lebih efektif dalam perangkat lunak dan dokumentasi.

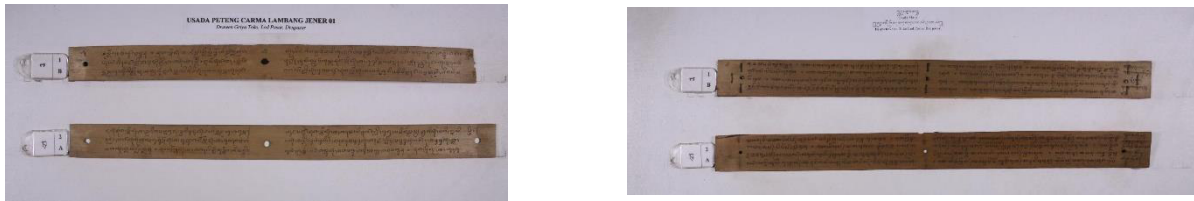
2. Metode Penelitian

Pada penelitian yang dilakukan penulis terdiri dari beberapa tahapan, yaitu data yang digunakan berupa file dokumen dalam bentuk PDF yang telah dikumpulkan dimasukkan ke dalam sistem, setelah itu akan dilakukannya enkripsi/penyisipan file PDF ke dalam *cover-image* RGB dalam bentuk .PNG, enkripsi file pdf ini menggunakan metode AES dan menyisipkan file PDF terenkripsi ke dalam gambar dengan metode LSB. Setelah itu untuk poses dekripsi/ekstraksi dimulai dengan mengambil gambar yang sudah terenkripsi sebelumnya dan melakukan ekstraksi stego-image dengan metode LSB yang dimana user akan mendapatkan file dokumen yang disisipkan dalam gambar. Pada penelitian ini, *cover-image* yang digunakan berupa gambar RPG (24 bit), berbeda dengan penelitian yang telah disebutkan sebelumnya [2][3] yang menggunakan *cover-image* gambar *grayscale* (8 bit) dan juga dalam bentuk video. Tahapan enkripsi yang dilakukan pada penelitian ini terdapat 3 pilihan yaitu untuk menyisipkan pesan dalam bentuk teks, dokumen, atau gambar sedangkan pada penelitian sebelumnya hanya menyisipkan pesan dalam bentuk teks. Pada sistem yang dibangun oleh penulis, terdapat juga penambahan fitur yang tersedia dalam 2 jenis untuk memilih jumlah piksel gambar, hal ini memungkinkan pengguna untuk memilih antara kualitas gambar yang lebih baik atau kapasitas penyimpanan yang lebih baik.

2.1. Data

Jenis data yang digunakan pada penelitian ini yaitu data primer. Data primer yang berasal dari berbagai sumber yang dikumpulkan dengan menggunakan Teknik observasi. Berupa dokumentasi manuskrip lontar. Pengambilan data gambar perlu dilakukan alih media ke digital dengan cara scanner atau dokumentasi dan disimpan dalam format file .PNG. Data file PDF dikumpulkan sendiri dengan cara membuat beberapa file PDF yang bersumber dari <https://archive.org/details/Bali>. Data yang diambil sebanyak 50 data, permasing-masing file PDF memiliki halaman dan ukuran yang menyesuaikan dengan banyaknya isi file PDF tersebut. Pesan teks dengan minimal 1 dan maksimal 16 karakter sebanyak 50 data. Seluruh data akan digunakan sebagai evaluasi (testing)[5].

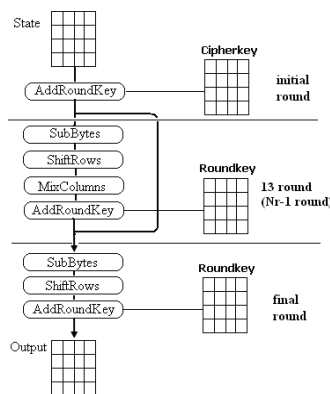
Tiap bagian lontar tersebut memiliki bahasan yang berbeda-beda seperti lontar usada yang membahas pengobatan tradisional bali contohnya seperti :



Gambar 1. Manuskrip Lontar Bali Usada.
 (Sumber; <https://archive.org/details/Bali>)

2.2. Enkripsi AES

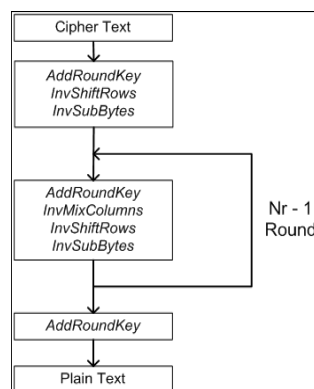
Sebelum melakukan proses penyisipan pesan kedalam gambar manuskrip lontar bali, data terlebih dahulu melalui tahapan enkripsi AES. Ketika enkripsi sangat penting dalam kriptografi, maka keamanan untuk menjaga kerahasiaan data yang dikirim[6]. Pesan aslinya adalah teks biasa yang diterjemahkan ke dalam kode yang tidak dapat dipahami. Enkripsi dapat diartikan sebagai enkripsi atau kode.



Gambar 2. Ilustrasi Proses Enkripsi

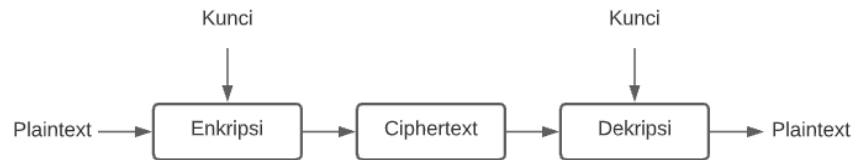
2.3. Dekripsi AES

Dekripsi adalah kebalikan dari enkripsi. Pesan terenkripsi dikembalikan ke bentuk aslinya (teks asli) yang disebut dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi[7].



Gambar 3. Ilustrasi Proses Dekripsi

Dengan membalikkan transformasi enkripsi dan mengimplementasikannya ke arah yang berlawanan, Anda dapat menghasilkan enkripsi balik yang mudah dipahami dari algoritme AES[8]. Konversi byte yang digunakan dalam enkripsi terbalik adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey. Gambar 4 menunjukkan proses umum dekripsi AES.



Gambar 4. Proses Enkripsi dan Dekripsi secara sederhana

Gambaran enkripsi dan dekripsi secara matematis dapat di notasikan sebagai berikut:

$C = ciphertext$

$P = plaintext$

Fungsi enkripsi E memetakan P ke C

$E(P) = C$

Fungsi dekripsi D memetakan C ke P

$D(C) = P$

$D(E(P)) = P$

Enkripsi bertujuan untuk memberikan layanan keamanan seperti :

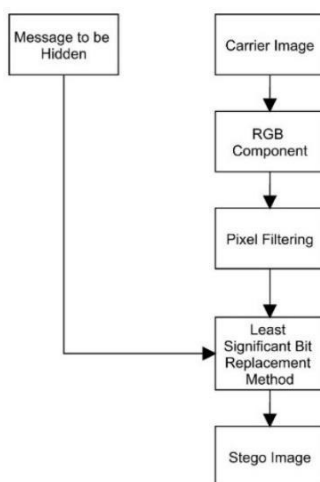
1. Kerahasiaan. Tujuannya adalah untuk mencegah orang yang tidak berwenang membaca pesan tersebut.
2. Integritas data. Kami menjamin semua bagian pesan tidak akan berubah sejak pengirim membuat/mengirim data sampai penerima data membuka data.
3. Otentikasi. Berkaitan dengan identifikasi, ini mengidentifikasi kebenaran pihak yang berkomunikasi dan kebenaran pengirim pesan.
4. Non-penyangkalan. Ini memberikan cara untuk membuktikan bahwa dokumen tersebut berasal dari orang tertentu. Ini akan terbukti benar berdasarkan pengakuan orang tersebut jika seseorang mencoba mengakui kepemilikan dokumen tersebut.

2.4. LSB

Metode ini paling sederhana, tetapi paling tidak tahan terhadap semua proses yang dapat mengubah nilai intensitas gambar. Citra adalah representasi (gambar) atau kemiripan dari suatu objek. Citra digital adalah citra yang dapat diproses oleh komputer[9]. Sebuah citra digital dapat merepresentasikan sebuah matriks yang terdiri dari M kolom dan N baris. Di sini, perpotongan kolom dan baris disebut piksel dan merupakan elemen terkecil dari gambar. Piksel memiliki dua parameter: koordinat dan intensitas atau warna.

Dalam prosedur ini, sistem mengubah nilai LSB (least significant bit) bit warna untuk menyembunyikan bit yang sesuai dengan bit label. Proses ini hanya mengubah nilai bit terakhir dari data, membuat gambar yang direkonstruksi terlihat sangat mirip dengan gambar aslinya.

Metode LSB menyembunyikan data rahasia dari piksel paling tidak signifikan di file sampul. Perubahan kecil pada nilai piksel selalu mempengaruhi file container, tetapi perubahan yang terjadi sangat kecil sehingga tidak terdeteksi oleh indera manusia[3]. Fakta ini pada akhirnya digunakan sebagai teknik untuk menyembunyikan data dan pesan.



Gambar 5. Alur Proses Algoritma LSB

Untuk mengilustrasikan bagaimana data disimpan menggunakan metode LSB, misalnya piksel kontainer berikut:

01001101	00101110	10101110	10001010	10101111	10100010	00101011	10101010
----------	----------	----------	----------	----------	----------	----------	----------

Digunakan untuk menyimpan huruf "H" (01001000), yang mengubah piksel wadah menjadi:

01001100	00101111	10101110	10001010	10101111	10100010	00101010	10101010
----------	----------	----------	----------	----------	----------	----------	----------

2.5. MSE dan PSNR

Saat mengembangkan dan mengimplementasikan rekonstruksi citra, citra yang direkonstruksi harus dibandingkan dengan citra aslinya. Metrik umum yang digunakan untuk tujuan ini adalah rasio signal-to-noise (PSNR) puncak yang tinggi. Artinya, hasil rekonstruksi sangat mirip dengan gambar aslinya. PNSR didefinisikan sebagai:

$$PSNR = 10 \log_{10} \left(\frac{C_{Max}^2}{MSE} \right) \quad (1)$$

Di sini, MSE dinyatakan sebagai kesalahan kuadrat rata-rata yang didefinisikan sebagai:

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

Dimana x dan y adalah koordinat gambar, M dan N adalah dimensi gambar, S_{xy} adalah gambar Stego, dan C_{xy} adalah gambar sampel. C_{Max}² adalah nilai maksimum untuk gambar. PSNR sering dinyatakan dalam desibel (dB) pada skala logaritmik[10]. 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.

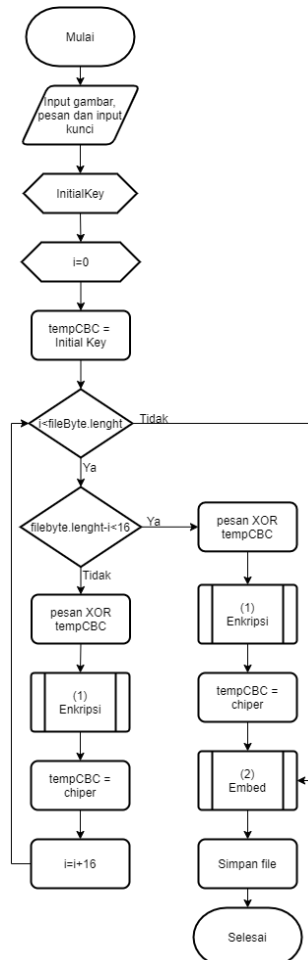
2.6. Analisis Kebutuhan

Tahap ini merupakan analisis kebutuhan sistem. Pengumpulan data pada tahap ini dilakukan untuk menemukan kondisi dan fitur yang dibutuhkan sistem untuk memenuhi kebutuhan dan keinginan penggunaanya.

1. Aplikasi ini dapat dijalankan di komputer dengan sistem operasi windows.
2. Aplikasi ini dapat melakukan proses enkripsi file PDF.
3. Aplikasi ini dapat menyisipkan file PDF terenkripsi ke dalam gambar.
4. Aplikasi ini dapat mengekstrak gambar yang dimasukkan ke dalam file PDF terenkripsi
5. Aplikasi ini dapat mendekripsi file PDF terenkripsi.

2.7. Desain Sistem

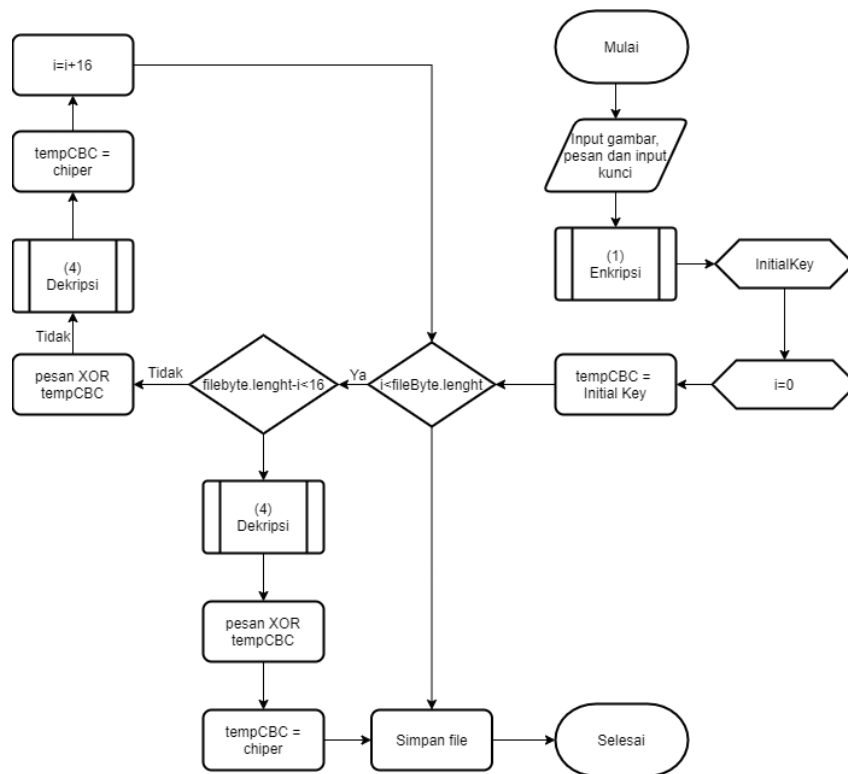
Perancangan aplikasi steganografi yang dibangun memiliki dua proses yaitu enkripsi embed dan dekripsi ekstraksi. Dalam proses enkripsi, pengguna memasukkan kunci, pesan rahasia dalam bentuk dokumen, dan gambar dengan ekstensi *.png. Setelah itu, hasil enkripsi akan dimasukkan menggunakan metode Least Significant Bit (LSB) ke dalam citra gambar yang sudah dimasukkan oleh pengguna. Hasil penyisipan akan dikirim ke pengguna untuk di unduh.



Gambar 6. Flowchart Enkripsi dan Embed

Flowchart pada Gambar 6 merupakan alur secara umum dari proses Enkripsi dan Embed, dimulai dengan menginputkan gambar, pesan dan kunci. Melakukan inialisasi InitialKey dan i sama dengan 0 dan dilanjutkan melakukan proses mengambil nilai InitialKey dan menyimpan pada tempCBC.

Terdapat kondisi jika i lebih kecil dari fileByte.length maka masuk ke kondisi berikutnya yaitu jika fileByte.length dikurangi i lebih kecil dari 16 maka masuk pada proses XOR antara pesan dengan tempCBC kemudian masuk pada proses enkripsi dimana sebelumnya pesan dilakukan padding agar genap 16byte dan hasil chipper-nya akan menjadi tempCBC dan dilanjutkan melakukan proses Embed. Dan Jika fileByte.length dikurangi i tidak lebih kecil dari 16 sama akan masuk pada proses XOR antara pesan dengan tempCBC kemudian masuk pada proses enkripsi dan hasil chipper-nya akan menjadi tempCBC yang akan digunakan pada perulangan berikutnya[11]. Dan jika kondisi i tidak lebih kecil dari fileByte.length maka akan dilanjutkan dengan proses Embed, kemudian menyimpan file dan selesai.



Gambar 7. Flowchart Ekstraksi dan Dekripsi

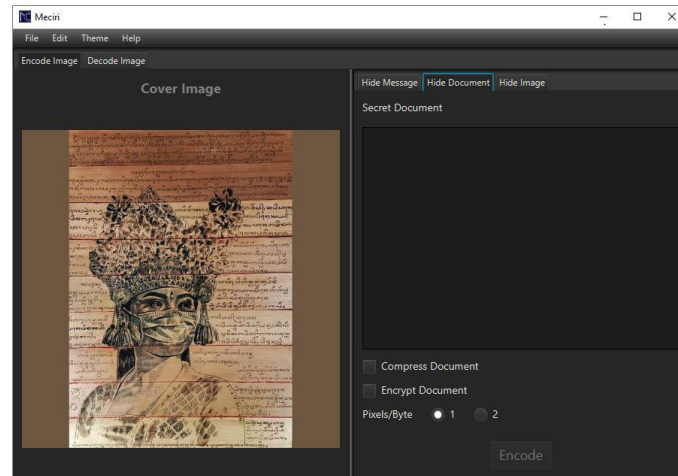
Flowchart secara umum dari proses Ekstraksi dan Dekripsi system ditunjukkan oleh Gambar 7 Proses dimulai dengan menginputkan gambar, pesan dan kunci. Setelah itu dilakukan proses Ekstraksi pesan pada gambar, dilanjutkan dengan melakukan inialisasi initialkey dan i sama dengan 0 serta melakukan proses mengambil nilai initialkey dan menyimpan pada tempCBC.

Terdapat kondisi jika i lebih kecil dari fileByte.length maka masuk ke kondisi berikutnya yaitu jika fileByte.length dikurangi i lebih kecil dari 16 maka masuk pada proses XOR antara pesan dengan tempCBC kemudian masuk pada proses enkripsi dimana sebelumnya pesan dilakukan padding agar genap 16 byte dan hasil chipper-nya akan menjadi tempCBC dan dilanjutkan menyimpan file. Dan jika fileByte.length dikurangi i tidak lebih kecil dari 16 maka sama akan masuk pada proses Dekripsi dan dilanjutkan dengan proses XOR antara pesan dengan tempCBC dan hasil dekripsinya akan menjadi tempCBC yang akan digunakan pada perulangan berikutnya. Dan jika i tidak lebih kecil dari fileByte.length maka akan dilanjutkan dengan menyimpan file dan selesai.

3. Hasil dan Pembahasan

3.1. Antarmuka Aplikasi

Antarmuka dari sistem yang dibuat untuk mengimplementasikan penelitian ini dibuat dengan bahasa pemrograman java. Pada antarmuka ini, terdapat 2 bagian yaitu bagian encode image, dan decode image.



Gambar 8. Tampilan sistem untuk Enkripsi dan penyisipan

Gambar 8. menunjukkan implementasi tampilan aplikasi yang dibangun. Tampilan ini merupakan tampilan untuk melakukan enkripsi dan penyisipan (*embed*). Terdapat tiga pilihan tab atas yaitu untuk menyisipkan pesan dalam bentuk text, dokumen, dan gambar. Setelah gambar cover di-*input*-kan maka gambar *cover* akan ditampilkan pada aplikasi. Terdapat juga kotak centang di tab pesan dan dokumen untuk menyembunyikan atau mengenkripsi *file*. Kotak enkripsi akan membuka jendela yang memungkinkan *user* memvalidasi kata sandi tau memilih gambar sebagai kata sandi.

Tombol pilihan tersedia dalam 2 jenis untuk memilih jumlah piksel per byte (atau piksel per piksel). Dalam hal gambar, ini memungkinkan user untuk memilih antara kualitas yang lebih baik atau kapasitas penyimpanan yang lebih baik. Kemudian, tombol Encode memungkinkan user memilih tujuan gambar yang disandikan. Jika operasi berhasil, sebuah jendela akan memberi tahu user bahwa proses telah berhasil.

3.2. Aplikasi Pengujian


Pengujian dari sistem pengaman file pdf dengan menggunakan dua metode keamanan yaitu dengan algoritma AES-128 dan steganografi *Least Significant Bit* (LSB) pada citra digital ini bertujuan untuk memastikan apakah sistem yang dikembangkan sudah tepat sesuai dengan kebutuhan didapatkan.

a. MSE dan PSNR

Pengujian ini dilakukan untuk mengetahui kemiripan *stegoimage* dengan *cover-image* menggunakan metode steganografi *Least Significant Bit* (LSB) dengan menghitung nilai PSNR. Pada pengujian ini dilakukan penyisipan file manuskrip lontar bali dengan format .pdf ke dalam *cover image* dengan format .png berukuran 1404x936 piksel. Dimana gambar tersebut memiliki perbedaan objek gambar dengan komposisi warna dalam piksel-pikselnya. Setiap gambar disisipkan sebanyak 1 kali dengan file pdf.

Tabel 1. *Cover-image* untuk pengujian

No	Tampilan Gambar	Nama Gambar
1		usada-peteng-carma-lambang-jener.png

2		usada-mata.png
---	---	----------------

Pada tabel 1 adalah contoh *cover-image* yang akan digunakan untuk wadah disisipkan pesan dokumen dalam bentuk .pdf

Tabel 2. Hasil pengujian nilai MSE

No	Gambar (.png)	File PDF	MSE(Red)	MSE(Green)	MSE(Blue)	MSE(Total)
1	usada-peteng-carma-lambang-jener.png	Data1	0.0221604	0.00571322	0.0235827	0.0171521
2	usada-mata.png	Data2	0.0257689	0.00572464	0.0284565	0.0199834

pada tabel 2 hasil pengujian *file* menggunakan MSE, didapatkan nilai MSE lebih kecil dari 0.15. sehingga dapat dikatakan akurasi dari pengujian citra digital dapat diterima dengan rata-rata 0.01856775 dB.

Tabel 3. Hasil pengujian nilai PSNR

No	Gambar (.png)	File PDF	PSNR (Red)	PSNR (Green)	PSNR (Blue)	PSNR (Total)
1	usada-peteng-carma-lambang-jener.png	Data1	93.5071	106.984	92.8851	97.7919
2	usada-mata.png	Data2	91.9985	106.964	91.0064	96.6562

pada tabel 3 hasil pengujian *file* menggunakan PSNR dengan mengambil sampel *file* citra digital, didapatkan hasil yang sangat baik, yaitu lebih besar dari 37 desibel (dB), didapatkan bahwa nilai PSNR dari total *channel red* nilai rata-rata PSNR untuk penyisipan pdf dari daya tampung maksimal gambar adalah 92.83349, nilai rata-rata PSNR dari total *channel green* untuk penyisipan pdf dari daya tampung maksimal adalah 106.7154, dan nilai rata-rata PSNR dari total *channel blue* untuk penyisipan pdf dari daya tampung maksimal gambar adalah 92.54417. dengan rata-rata 97.22405 dB.

b. Perbandingan tampilan *cover image* dengan *stegoimage*

Tabel 4. Contoh Perbandingan tampilan *cover-image* dengan *Stegoimage*

No	<i>Cover image</i>	<i>Stegoimage</i>
----	--------------------	-------------------



Pada tabel 4 adalah contoh perbandingan tampilan *cover-image* dengan *stegoimage* tidak akan terlihat perbedaan yang significant oleh indra pengelihat manusia walaupun pesan yang disisipkan sebesar 100% dari daya tamping maksimal gambar.

c. Pengujian kesamaan file dengan *Hamming Distance*

Tabel 5. Hasil pengujian kesamaan file dengan *Hamming Distance*

No	File input		File Output		Hamming Distance	Persentase Kesamaan file pdf (%)
	Nama (pdf)	Ukuran (kb)	Nama (pdf)	Ukuran (kb)		
1	Data1	7.04	Dekrip1	7.04	0	100
2	Data2	6.99	Dekrip2	6.99	0	100

Pada tabel 5 didapatkan bahwa semua file yang diuji kesamaannya menunjukkan *hamming* distance sebanyak 0, hal tersebut berarti antara bit-bit file pdf asli dengan bit bit file pdf setelah dienkripsi-dekripsi tidak ada yang terbalik (sama). Jadi, dari 10 kali pengujian, didapatkan bahwa file pdf asli dapat dikembalikan dengan presentase kesamaan 100%.

4. Kesimpulan

Penerapan steganografi menggunakan metode penyisipan least significant bit (LSB) yang dibangun pada aplikasi ini berhasil dalam menyisipkan chipper text terenkripsi pada gambar manuskrip lontar bali dengan format .png dan juga berhasil dalam mengekstraksi gambar dan mengembalikan berkas (file) pdf terenkripsi. Metode kriptografi AES (Advanced Encryption Standard) yang dibangun pada aplikasi ini juga berhasil melakukan enkripsi maupun dekripsi terhadap (file) pdf manuskrip lontar bali sehingga pesan dapat disandikan dan dikembalikan seperti semula dengan persentase keberhasilan 100%. Dilihat dari nilai MSE dan PSNR didapatkan bahwa besarnya pesan tidak berpengaruh pada tingkat akurasi karena pesan akan dimaksimalkan menjadi 16 karakter dan fungsi pad pada library AES 128 bit, sehingga penyisipan citra digital dikatakan sangat baik, karena tidak terdapat perbedaan yang significant melalui sinyal. Hasil pengujian membuktikan adanya peningkatan keamanan serta nilai imperceptibility yang tetap terjaga. Hal ini dibuktikan dengan hasil rata-rata nilai MSE 0.01856775 dB dan PSNR 97.22405 dB. Maka pengguna yang pertama kali mengenkripsi file memiliki bukti definitif atas kepemilikan hak cipta dan keamanan data.

Daftar Pustaka

- [1] J. I. Logika, "Penelitian ini mengembangkan sebuah metoda pengamanan pesan yang dinamakan Steganografi dan dibangun pula sebuah Aplikasi Steganografi dengan algoritma Least Significant Bit . Steganografi adalah seni dan ilmu menulis pesan tersembunyi atau menyembunyikan," vol. I, no. 2, pp. 35–38, 2019.
- [2] U. A. Anti, A. H. Kridalaksana, and D. M. Khairina, "Steganografi Pada Video Menggunakan Metode Least Significant Bit (LSB) Dan End Of File (EOF)," *Inform. Mulawarman J. Ilim. Ilmu Komput.*, vol. 12, no. 2, p. 104, 2017, doi: 10.30872/jim.v12i2.658.
- [3] J. I. Sari, H. T. Sihotang, and T. Informatika, "Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma Hill Cipher Dan Metode Least Significant Bit (LSB)," *J. Mantik Penusa*, vol. 1, no. 2, pp. 1–8, 2017, [Online]. Available: <http://ejournal.pelitanusantara.ac.id/index.php/mantik/article/view/253>.
- [4] S. Nur'aini, "Steganografi Pada Digital Image Menggunakan Metode Least Significant Bit Insertion," *Walisongo J. Inf. Technol.*, vol. 1, no. 1, p. 73, 2019, doi: 10.21580/wjit.2019.1.1.4025.
- [5] Ketut Gura Arta Laras, "DIGITISASI LONTAR MUSEUM NASKAH LONTAR DESA ADAT DUKUH PENABAN, KECAMATAN KARANGASEM, KABUPATEN KARANGASEM, BALI," vol. 26, no. 1, p. 6, 2021.
- [6] A. Hafiz, "Steganografi Berbasis Citra Digital Untuk Menyembunyikan Data Menggunakan Metode Least Significant Bit (Lsb)," *J. Cendikia*, vol. 17, no. 1, pp. 194–198, 2019.
- [7] I. D. FADHILAH, "RANCANG BANGUN SISTEM KEAMANAN DATA DENGAN METODE STEGANOGRAFI LSB BERBASIS WEBSITE." p. 85, 2019.
- [8] D. Darwis, R. Prabowo, and N. Hotimah, "Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman untuk Meningkatkan Keamanan Data," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, p. 389, 2018, doi: 10.25126/jtiik.201854727.
- [9] N. Anwar, "Perancangan Steganografi Hidden Message Dengan Metode Least Significant Bit Insertion (Lsb) Berbasis Matlab," *J. Algoritm. Log. dan Komputasi*, vol. 1, no. 1, pp. 25–30, 2018, doi: 10.30813/j-alu.v1i1.1107.
- [10] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *J. Comput. Commun.*, vol. 07, no. 03, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
- [11] G. Wibisono, T. Waluyo, and E. I. H. Ujjianto, "Kajian Metode Metode Steganografi Pada Domain Spasial," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 5, no. 2, pp. 259–264, 2020, doi: 10.33480/jitk.v5i2.1212.

This page is intentionally left blank.

Identifikasi Ekspresi Idiomatik Menggunakan *Distributional Semantic Based Approach* dan *Truth Discovery*

Ni Made Yuli Cahyani^{a1}, AAIN Eka Karyawati^{a2}, Luh Arida Ayu Rahning Putri^{a3}, Agus Muliantara^{a4}, Ida Bagus Gede Dwidasmar^{a5}, Luh Gede Astuti^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia

¹yulicahyani1101@gmail.com

²eka.karyawati@unud.ac.id

³rahningputri@unud.ac.id

⁴muliantara@unud.ac.id

⁵dwidasmar@unud.ac.id

⁶lg.astuti@unud.ac.id

Abstract

Idiomatic expressions are phrases that consist of a sequence of two or more words that have a meaning that cannot be predicted from the meaning of the individual words that compose it. Idiomatic expressions exist in almost all languages but are difficult to extract because there is no algorithm that can precisely decipher the structure of idiomatic expressions, so most rule-based machine translation systems generally translate idiomatic expressions by translating word for word their constituents, but the translation results do not produce the true meaning of the idiomatic expression. Based on this problem, the author tries to do research on the identification of the use of idiomatic expressions in Indonesian sentences. First, the author conducts the sentence classification process using BERT to find out whether the sentence contains idiomatic expressions or not. Furthermore, idiomatic expressions are identified based on distributional semantic based approach and then validated automatically using the Truth Discovery method. From the research conducted, the identification of idiomatic expressions in Indonesian sentences using Distributional Semantic Based Approach and Truth Discovery obtained an accuracy of 0.82; precision 1.0; recall 0.64 and f1-score 0.78.

Keywords: *Idiomatic Expressions, BERT, Truth Discovery, Validation, Distribution Semantic*

1. Pendahuluan

Bahasa memegang peranan penting yaitu sebagai alat komunikasi dalam kehidupan sosial masyarakat. Dalam berbahasa, suatu makna tidak hanya dilambangkan dalam satu bentuk bahasa, tetapi juga dapat diungkapkan dalam berbagai bentuk. Bentuk adalah ekspresi makna, sehingga bentuk itu sendiri dapat merangsang penafsiran lebih dari satu makna, salah satu contohnya yaitu dapat dilihat dalam penggunaan idiom. Idiom biasa digunakan dalam kegiatan berkomunikasi sehari-hari yaitu untuk mengungkapkan suatu maksud agar penyampaiannya menjadi lebih menarik atau lebih sopan. Penggunaan idiom itu sendiri sering ditemukan dalam puisi, novel, lirik lagu, surat kabar, majalah atau artikel [1]. Ekspresi idiomatik adalah frasa yang terdiri dari urutan dua kata atau lebih yang memiliki makna yang tidak dapat diprediksi dari makna kata-kata individu penyusunnya. Ekspresi idiomatik ada di hampir semua bahasa dan sulit untuk diekstrak karena tidak ada algoritma yang dapat secara tepat menguraikan struktur ekspresi idiomatik. Identifikasi ekspresi idiomatik adalah masalah yang menantang dengan penerapan yang luas. Mengidentifikasi ekspresi idiomatik sangat penting untuk aplikasi pemrosesan bahasa alami seperti *machine translation*, *information retrieval* dan sebagainya [2]. Sebagian besar sistem mesin terjemahan berbasis aturan umumnya menerjemahkan ekspresi idiomatik dengan cara menerjemahkan kata demi kata penyusun ekspresi idiomatik, sehingga hasil terjemahan tidak menghasilkan makna yang sebenarnya dari ekspresi idiomatik tersebut.

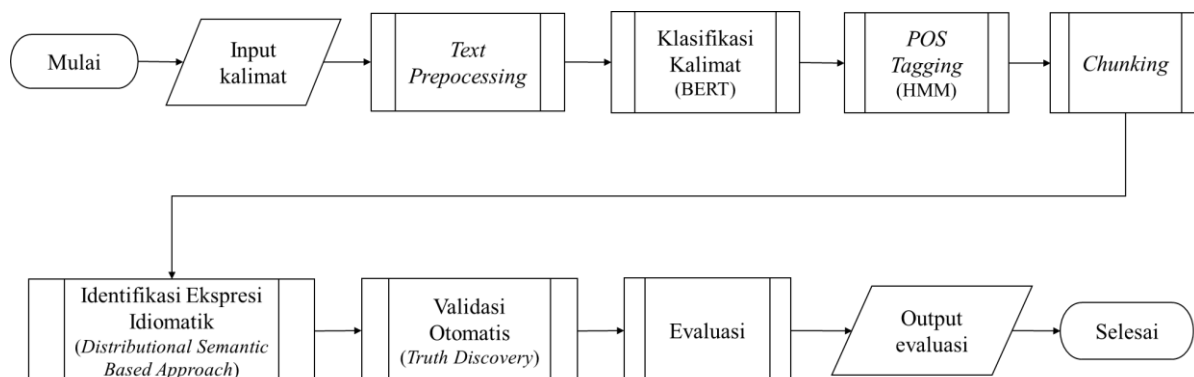
Penelitian tentang identifikasi ekspresi idiomatik bahasa Indonesia belum pernah dilakukan sebelumnya, namun terdapat beberapa penelitian yang serupa dalam bahasa lain. Seperti penelitian [3], pada penelitian ini memperkenalkan pendekatan *semi-supervised* yang menggunakan

representasi terdistribusi dari makna kata untuk menangkap metaforitas. Peneliti menggunakan model *word embedding* untuk mengukur kemiripan semantik antara frasa kandidat dan kumpulan metafora yang telah ditentukan sebelumnya. Penelitian ini memperoleh nilai *precision* 0,5945, *recall* 0,756, *F-score* 0,6657 dan *accuracy* 0,6290. Kemudian terdapat penelitian [4], pada penelitian ini memperkenalkan metode identifikasi metafora pertama yang mengintegrasikan representasi makna yang dipelajari dari data linguistik dan visual dengan menerapkan metode *embedding* kata atau frasa untuk tugas identifikasi metafora. Pada penelitian ini memperoleh nilai *precision* yaitu 0,73, *recall* yaitu 0,80 dan *F-score* 0,76 untuk metode *WordCos* menggunakan *linguistic embeddings*.

Berdasarkan permasalahan di atas, penulis mencoba melakukan penelitian mengenai identifikasi ekspresi idiomatik pada kalimat berbahasa Indonesia. Pertama-tama penulis akan melakukan proses klasifikasi kalimat menggunakan BERT untuk mengetahui apakah kalimat mengandung ekspresi idiomatik atau tidak. Selanjutnya ekspresi idiomatik diidentifikasi berdasarkan pendekatan berbasis semantik distribusi (*Distributional Semantic Based Approach*) yang merupakan kombinasi dari pendekatan pada penelitian sebelumnya yang telah dipaparkan di atas. Metode ini mengidentifikasi ekspresi idiomatik pada tingkat frasa dilakukan dengan menjumlahkan kemiripan semantik antara kandidat dan kumpulan contoh ekspresi idiomatik dengan kemiripan semantik antara kata penyusun frasa kandidat. Kemudian melakukan validasi secara otomatis menggunakan kombinasi algoritma *Sums* dan *Average-Log* yang merupakan algoritma dari metode *Truth Discovery* dengan sumbernya merupakan berbagai macam website yang membahas mengenai ekspresi idiomatik bahasa Indonesia. Diharapkan dengan dilakukannya penelitian ini dapat membantu dalam mengidentifikasi penggunaan ekspresi idiomatik dalam suatu kalimat secara otomatis yang nantinya juga dapat dimanfaatkan untuk membantu mengoptimalkan aplikasi *machine translation* dalam menerjemahkan kalimat yang mengandung ekspresi idiomatik.

2. Metode Penelitian

Penelitian yang dilakukan penulis terdiri dari beberapa tahapan, yaitu data berupa kumpulan kalimat berpola dasar berbahasa Indonesia yang tidak ataupun mengandung ekspresi idiomatik yang telah dikumpulkan dimasukkan ke dalam sistem, setelah itu akan dilakukan *text preprocessing*. Kemudian masuk ke tahap klasifikasi kalimat menggunakan BERT. Selanjutnya kalimat yang diklasifikasikan sebagai 'kalimat_idiom' akan masuk ke tahap *POS Tagging* dan *chunking*. Setelah mendapatkan hasil *chunking* berupa frasa, selanjutnya akan masuk ke tahap identifikasi ekspresi idiomatik menggunakan *Distributional Semantic Based Approach*. Setelah itu frasa yang telah diidentifikasi sebagai ekspresi idiomatik akan divalidasi secara otomatis menggunakan metode *Truth Discovery* yaitu kombinasi algoritma *Sums* dan *Average-Log*. Hasil identifikasi yang sudah divalidasi secara otomatis tersebut akan menjadi output akhir dari penelitian ini yang selanjutnya akan masuk ke tahap evaluasi. Secara umum, alur penelitian dapat dilihat pada gambar berikut.



Gambar 1. Alur Umum Penelitian

2.1 Ekspresi Idiomatik Bahasa Indonesia

Ekspresi idiomatik merupakan ungkapan yang maknanya tidak sesuai dengan prinsip komposisionalitas, dan tidak terkait dengan makna bagian [5]. Makna idiomatik adalah makna sebuah satuan bahasa yang menyimpang dari makna leksikal atau makna unsur-unsur pembentuknya. Hal ini berarti suatu idiom tidak dapat diterjemahkan kata per kata tetapi harus dilihat secara utuh dari unsur-unsur pembentuknya. Ekspresi idiomatik yang akan diidentifikasi dalam penelitian ini yaitu idiom yang

memiliki kategori frasa nomina, frasa verba dan frasa adjektiva yang disusun oleh dua kata. Contoh dari ekspresi idiomatik tersebut adalah sebagai berikut:

Tabel 1. Contoh Ekspresi Idiomatik Bahasa Indonesia

	Kategori	Contoh
Frasa Nomina	kata benda + kata benda	Kutu buku
	kata benda + kata sifat	Kuda hitam
	kata benda + kata kerja	Bunga tidur
	kata bilangan + kata benda	Empat mata
Frasa Verba	kata kerja + kata benda	Adu mulut
	kata kerja + kata sifat	Naik pitam
	kata kerja + kata kerja	Jatuh bangun
	kata kerja + kata bilangan	Bermuka dua
Frasa Adjektiva	kata sifat + kata benda	Ringan tangan
	kata sifat + kata sifat	Panjang lebar

2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder yaitu data yang sudah tersedia sebelum peneliti memulai penelitian [6]. Terdapat tiga data yang digunakan, dimana data ini bersumber dari media internet, buku ataupun publikasi. Data pertama berupa *Indonesian Manually Tagged Corpus*, yaitu kumpulan kalimat bahasa Indonesia yang telah diberikan tag secara manual. Data ini digunakan pada tahap POS *Tagging*. Contoh data dapat dilihat pada Tabel 2.

Tabel 2. *Indonesian Manually Tagged Corpus*

Contoh Kalimat dan Kelas Katanya
Binatang/NN ini/PR tidak/NEG bisa/MD dibunuh/VB karena/SC masyarakat/NN India/NNP menganggap/VB mereka/PRP suci/JJ . /Z
Jumlah/NN dan/CC harga/NN barang/NN itu/PR masih/MD dirundingkan/VB . /Z
Buaya-buaya/NN itu/PR berukuran/VB panjang/NN antara/IN 40/CD -/Z 50/CD centimeter/NN . /Z

Data kedua berupa kumpulan kalimat berpola dasar berbahasa Indonesia yang mengandung sebuah ekspresi idiomatik dan kalimat berpola dasar berbahasa Indonesia yang tidak mengandung ekspresi idiomatik. Data ini berjumlah 2000 kalimat yang telah dilabeli sebagai kalimat biasa dan kalimat idiom secara manual oleh pakar dan berdasarkan pada buku kamus idiom bahasa Indonesia, dengan jumlah kalimat pada setiap label yaitu 1000 kalimat. Data ini digunakan pada tahap klasifikasi kalimat dan tahap identifikasi ekspresi idiomatik. Contoh data dapat dilihat pada Tabel 3.

Tabel 3. Contoh Kalimat Bahasa Indonesia

indeks	kalimat	kategori	frasa_idiom	validasi
1	Orang tua itu rela membanting tulang demi menyekolahkan ketiga anaknya.	kalimat_idiom	membanting tulang	idiom
2	Ayah adalah tangan kanan Pak Camat.	kalimat_idiom	tangan kanan	idiom
3	Huda suka menjadikan Andik sebagai kambing hitam.	kalimat_idiom	kambing hitam	idiom
4	Gadis itu sedang membaca sebuah buku novel.	kalimat_biasa	none	bukan_idiom
5	Tangan kanan Roni terkena minyak panas saat menggoreng ikan.	kalimat_biasa	none	bukan_idiom
6	Ayah membeli kambing hitam.	kalimat_biasa	none	bukan_idiom

Data berikutnya yaitu data sumber web berupa situs-situs web yang membahas mengenai ekspresi idiomatik bahasa Indonesia yang berjumlah 104 halaman web dengan total jumlah klaim 4358 klaim. Data ini akan digunakan pada tahap validasi *Truth Discovery*. Contoh data ditunjukkan pada Tabel 4.

Tabel 4. Contoh Data Situs Web

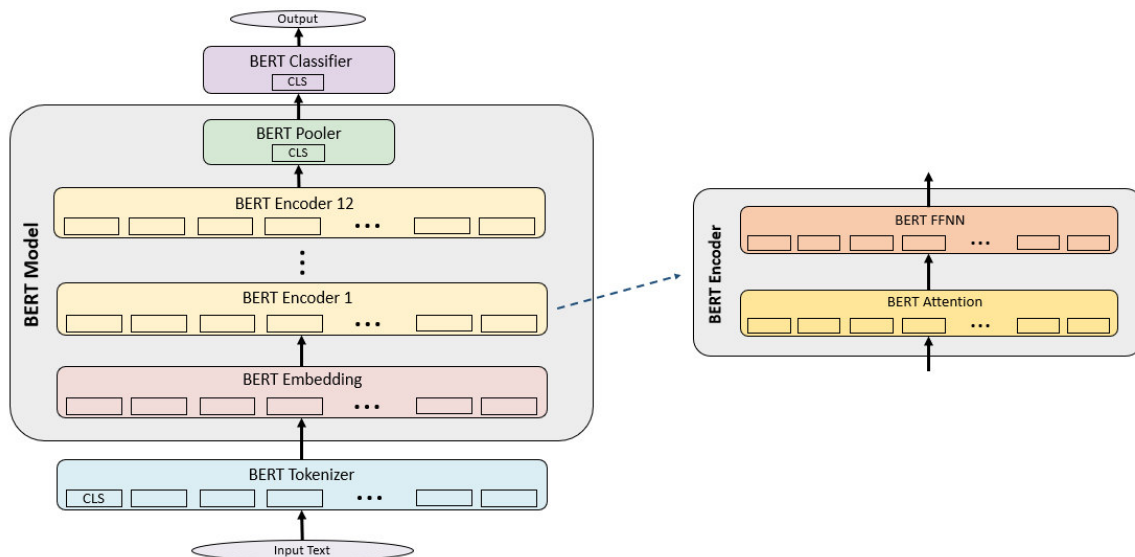
indeks	source_web	source_url	idiom_claims
1	salamadian.com	https://salamadian.com/contoh-ungkapan-bahasa-indonesia-dan-artinya-idiom/	Adu Mulut
2	salamadian.com	https://salamadian.com/contoh-ungkapan-bahasa-indonesia-dan-artinya-idiom/	Akal Bulus
3	salamadian.com	https://salamadian.com/contoh-ungkapan-bahasa-indonesia-dan-artinya-idiom/	Angkat Bicara
4	salamadian.com	https://salamadian.com/contoh-ungkapan-bahasa-indonesia-dan-artinya-idiom/	Angkat Kaki

2.3 Text Preprocessing

Pada penelitian ini, data kumpulan kalimat bahasa Indonesia akan dilakukan *text preprocessing* terlebih dahulu agar data tersebut siap dan dapat diolah untuk tahap selanjutnya. *Text preprocessing* digunakan untuk menyajikan data berupa teks dalam format yang sesuai. Adapun langkah-langkah yang dilakukan untuk *text preprocessing* pada penelitian ini adalah *punctuation removal* untuk menghilangkan simbol-simbol yang tidak diperlukan pada kalimat [7], *tokenization untuk memecah kalimat menjadi token yang dalam hal ini berupa kata*, dan *case conversion* untuk mengkonversi bentuk huruf dalam teks menjadi seragam yaitu menjadi huruf kecil [8].

2.4 Bidirectional Encoder Representations from Transformers (BERT)

Pada tahap ini dilakukan klasifikasi terhadap kalimat inputan ke dalam dua kategori yaitu 'kalimat_biasa' jika kalimat tidak mengandung ekspresi idiomatik dan 'kalimat_idiom' jika kalimat mengandung ekspresi idiomatik. Tahap ini membutuhkan model klasifikasi yang dibangun menggunakan BERT. *Bidirectional Encoder Representations from Transformers* atau disingkat BERT adalah model representasi bahasa terlatih yang dikembangkan oleh Devlin et al. (2019) [9]. BERT merupakan metode *state-of-the-art* dalam pembangunan *language model* dengan pendekatan *deep learning*. Arsitektur BERT dapat dilihat pada gambar berikut [10].



Gambar 2. Arsitektur BERT

2.5 POS Tagging

Pada penelitian ini, *POS Tagging* digunakan untuk memberikan label pada setiap kata dalam kalimat dengan kelas kata yang sesuai untuk kata tersebut, seperti kata benda, kata kerja, kata sifat, dan lain-lain. Pada tahap *POS Tagging* ini membutuhkan model yang dibangun menggunakan algoritma *Hidden Markov Model* (HMM). Adapun persamaan HMM ditunjukkan pada persamaan (1) [11]:

$$t_1^n = \prod_{i=1}^n \overbrace{P(w_i|t_i)}^{\text{emisi}} \overbrace{P(t_i|t_{i-1})}^{\text{transisi}} \quad (1)$$

Dengan $P(t_i|t_{i-1})$ merupakan probabilitas transisi yang mewakili probabilitas sebuah tag jika diketahui tag sebelumnya, yang dapat dihitung dengan persamaan (2):

$$P(t_i|t_{i-1}) = \frac{\text{Count}(t_{i-1}, t_i)}{\text{Count}(t_{i-1})} \quad (2)$$

Dan $P(w_i|t_i)$ merupakan probabilitas emisi yaitu probabilitas sebuah kata yang dilabeli *tag* tertentu, yang dihitung dengan persamaan (3).

$$P(w_i|t_i) = \frac{\text{Count}(t_i, w_i)}{\text{Count}(t_i)} \quad (3)$$

Keterangan:

t_1^n = kelas kata yang dicari

w_i = kata yang dicari kelas katanya

t_i = kelas kata dari w_i yang ada di *corpus*

t_{i-1} = kelas kata sebelum kelas kata dari w_i yang ada di *corpus*

Part-of-Speech Tagging pada penelitian ini dilakukan untuk memudahkan tahap selanjutnya yaitu *chunking* dimana kata-kata akan dikelompokkan ke dalam bentuk frasa yang ditentukan berdasarkan label kelas kata dari kata-kata tersebut.

2.6 Chunking

Pada penelitian ini, metode *chunking* digunakan untuk menemukan frasa nomina, frasa verba ataupun frasa adjektiva yang akan dikategorikan sebagai frasa kandidat dimana frasa tersebut memiliki konstruksi sintaksis pembentuk ekspresi idiomatik. Untuk menemukan frasa kandidat tersebut pada suatu kalimat, penulis akan mendefinisikan tata bahasa *chunk* (*chunk grammar*) menggunakan *tag* dari *part-of-speech tagging*, yang terdiri dari aturan *Regular Expressions* yang menunjukkan bagaimana kalimat harus dipotong.

2.7 Distributional Semantic Based Approach

Pada tahap ini akan dilakukan identifikasi terhadap frasa-frasa kandidat yang dihasilkan dari proses *chunking* apakah frasa tersebut merupakan ekspresi idiomatik atau tidak menggunakan pendekatan berbasis semantik distribusi (*Distributional Semantic Based Approach*) [4] [3] dengan menjumlahkan kemiripan semantik antara kandidat dan kumpulan contoh ekspresi idiomatik dengan kemiripan semantik antara kata penyusun frasa kandidat yang dapat dilihat pada persamaan (4):

$$sim = \sum_{i=1}^n sim(phrase, idiom\ example_i) + sim(w_1, w_2) \quad (4)$$

Keterangan:

sim = *similarity*

$phrase$ = frasa kandidat idiom

$idiom\ example$ = contoh idiom

w_1, w_2 = kata-kata penyusun frasa kandidat idiom

Pada tahap ini, perhitungan nilai *similarity* dilakukan dengan menggunakan *cosine similarity* yang menerima inputan berupa representasi vektor dari kata-kata dan juga frasa yang dihasilkan dari BERT *Embedding*. Setelah mendapatkan hasil nilai *similarity* antar frasa kandidat dengan contoh idiom dan nilai *similarity* antar kata penyusun frasa kandidat, selanjutnya kedua nilai *similarity* tersebut dijumlahkan berdasarkan persamaan (4), kemudian akan dipilih dari *frasa-frasa* kandidat tersebut yang diidentifikasi sebagai frasa idiom berdasarkan frasa yang memiliki nilai *similarity* lebih besar dari batas yaitu 0,5.

2.8 Truth Discovery

Truth Discovery atau bisa disebut pengecekan fakta bertugas untuk menemukan pernyataan yang benar diantara banyaknya pernyataan yang diklaim yang diajukan banyak sumber untuk objek yang sama [12]. Pada penelitian ini, proses validasi otomatis frasa idiom menggunakan kombinasi algoritma *Sums* dan *Average-Log* yang persamaannya ditunjukkan oleh (5), (6) untuk algoritma *Sums* dan (7) untuk algoritma *Average-Log* [13].

$$T^i(s) = \sum_{c \in C_s} B^{i-1}(c) \quad (5)$$

$$B^i(c) = \sum_{s \in S_c} T^i(s) \quad (6)$$

$$T^i(s) = \log|C_s| \cdot \frac{\sum_{c \in C_s} B^{i-1}(c)}{|C_s|} \quad (7)$$

Keterangan:

S = kumpulan sumber

S_c = kumpulan sumber yang memberikan klaim c

C = kumpulan klaim

C_s = kumpulan klaim disediakan oleh sumber s

$T^i(s)$ = *trustworthiness score* dari sumber s

$B^i(c)$ = *belief score* dari klaim c

$B^{i-1}(c)$ = *belief score* sebelumnya dari klaim c

Algoritma *Sums* dan *Average-Log* akan menghasilkan *trustworthiness score* dari sumber *website* yang dihitung menggunakan kombinasi persamaan (5) dan (7) dan *belief score* dari setiap klaim yang diberikan oleh sumber yang dihitung menggunakan persamaan (7). Kemudian untuk memvalidasi suatu frasa merupakan frasa idiom akan menggunakan *belief score*, dimana suatu frasa dianggap benar atau valid sebagai frasa idiom jika mempunyai *belief score* di atas nilai parameter *threshold* (t) yaitu persentil 15 dari data *belief score* yang persamaanya ditunjukkan oleh (5).

$$t = \text{percentile}(\text{belief score}, 15) \quad (8)$$

Langkah-langkah melakukan validasi otomatis menggunakan *Truth Discovery* antara lain:

- Menginput data sumber web dan data frasa bahasa Indonesia sebagai dataset, *initial value* yang ditetapkan pada penelitian ini yaitu 0.5 dan *iteration value* adalah 3.
- Membuat tabel *One Hot Encoding* yang berisikan *claim value* c pada sumber s , dimana *claim value* c akan bernilai 1 jika klaim c tersedia pada sumber s dan bernilai 0 untuk sebaliknya.
- Menghitung nilai *claim source score* C_s yaitu banyaknya jumlah klaim yang diberikan oleh setiap sumber s .
- Melakukan inialisasi *claim value* = *claim value* \times *initial value*.
- Menghitung *trustworthiness score* sumber s dengan menggunakan persamaan (5) dan (7).
- Menghitung *belief score* klaim c dengan menggunakan persamaan (6)
- Mengulangi langkah e dan f sesuai *iteration value*.
- Menghitung nilai parameter *threshold* dengan menggunakan persamaan (8)
- Melakukan proses validasi dimana suatu frasa dianggap benar atau valid sebagai frasa idiom jika mempunyai *belief score* di atas nilai parameter *threshold*.

2.9 Evaluasi Sistem

Pada penelitian ini akan dilakukan pengukuran performa menggunakan *confusion matrix*. Parameter yang digunakan adalah *accuracy* (9), *precision* (10), *recall* (11) dan *f1-score* (12).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (9)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

$$F1 - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

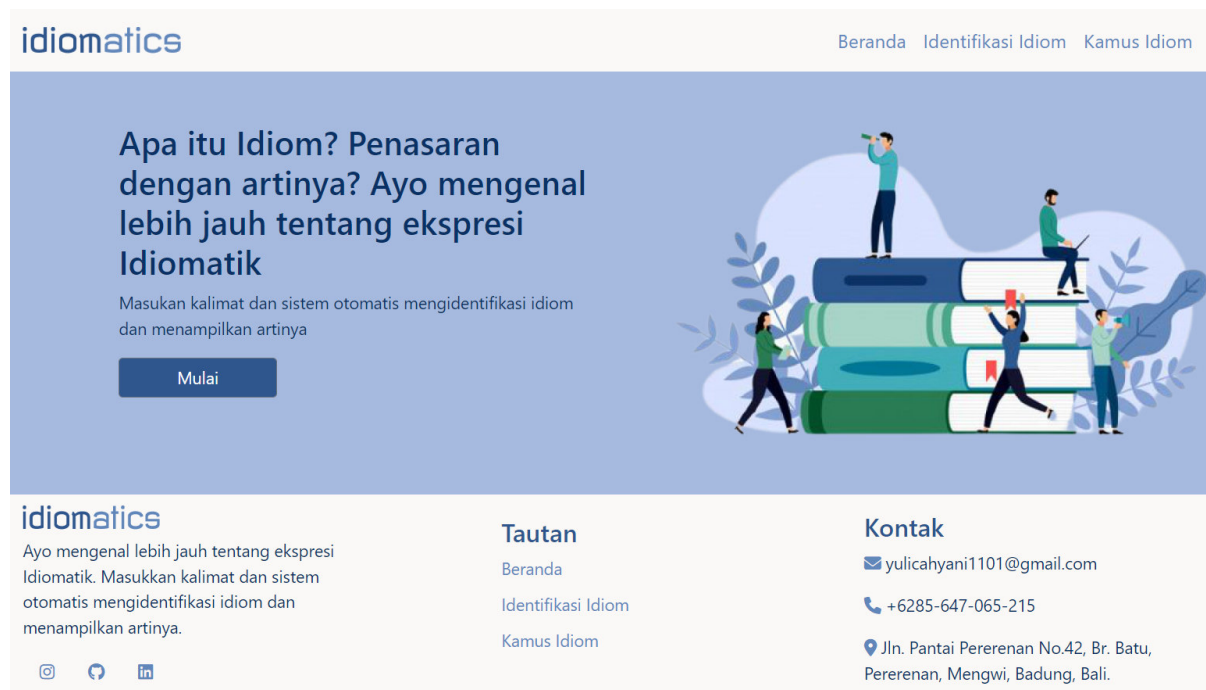
Keterangan:

- True Positive* (TP), yaitu total hasil dari prediksi kelas positif dan sesuai dengan kelas aslinya yang positif.
- True Negative* (TN), yaitu total hasil dari prediksi kelas negatif dan sesuai dengan kelas aslinya yang negatif.
- False Positive* (FP), yaitu total hasil dari prediksi kelas positif namun tidak sesuai dengan kelas aslinya yang negatif.
- False Negative* (FN), yaitu total hasil dari prediksi kelas negatif namun tidak sesuai dengan kelas aslinya yang positif.

3. Hasil dan Pembahasan

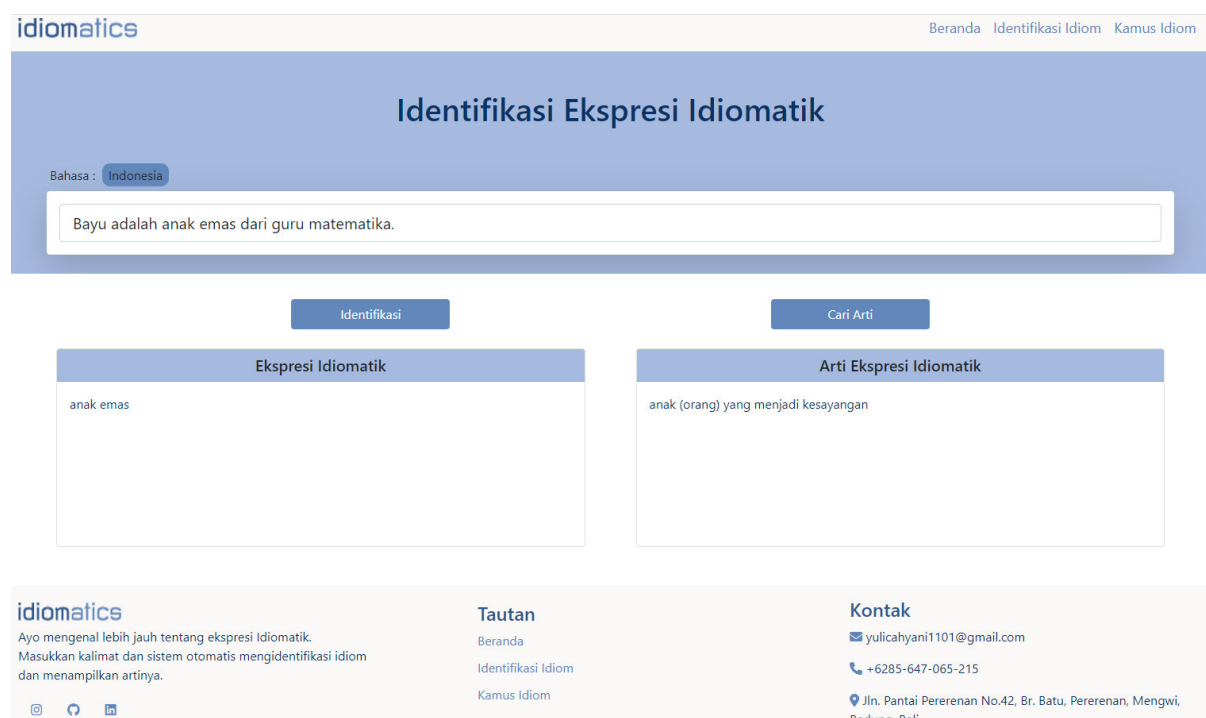
3.1. Tampilan Antarmuka Sistem Identifikasi Ekspresi Idiomatik

Tampilan antarmuka sistem terdiri dari tiga halaman, yaitu halaman beranda, halaman identifikasi idiom, dan halaman kamus idiom. Pada saat pertama kali membuka sistem, akan ditampilkan halaman beranda yang dapat dilihat pada Gambar 3. Pada halaman beranda menampilkan nama sistem serta deskripsi singkat dari sistem.



Gambar 3. Halaman Beranda

Halaman identifikasi yang dapat dilihat pada Gambar 4. merupakan halaman untuk melakukan identifikasi ekspresi idiomatik. Pada halaman ini terdapat *text box* untuk menerima inputan dari user yang berupa suatu kalimat berpola dasar berbahasa Indonesia. Kemudian terdapat tombol identifikasi untuk melakukan proses identifikasi ekspresi idiomatik pada kalimat inputan dan juga terdapat tombol cari arti untuk menampilkan arti dari ekspresi idiomatik yang telah diidentifikasi sebelumnya.



Gambar 4. Halaman Identifikasi Idiom

Selanjutnya halaman kamus idiom dapat dilihat pada Gambar 5. Pada halaman kamus idiom terdapat *text box* untuk menerima inputan dari user yang berupa huruf, kata, atau frasa berbahasa Indonesia. Kemudian terdapat tombol cari untuk melakukan proses pencarian berdasarkan inputan dan menampilkan hasil berupa ekspresi idiomatik beserta arti dan contoh penggunaanya dalam kalimat.



idiomatics Beranda Identifikasi Idiom Kamus Idiom

Kamus Ekspresi Idiomatik

Bahasa : Indonesia

Cari Idiom

Kata Pembentuk	Ekspresi Idiomatik	Arti	Contoh Penggunaan
anak	anak emas	anak (orang) yang menjadi kesayangan	dia merupakan anak emas kepala sekolah

idiomatics
Ayo mengenal lebih jauh tentang ekspresi Idiomatik. Masukkan kalimat dan sistem otomatis mengidentifikasi idiom dan menampilkan artinya.

Tautan
Beranda
Identifikasi Idiom
Kamus Idiom

Kontak
✉ yulicahyani1101@gmail.com
☎ +6285-647-065-215
📍 Jln. Pantai Pererenan No.42, Br. Batu, Pererenan, Mengwi, Badung, Bali.

Gambar 5. Halaman Kamus Idiom

3.2. Evaluasi Identifikasi Ekspresi Idiomatik

Pada penelitian ini, untuk mengetahui performa identifikasi ekspresi idiomatik menggunakan *Distributional Semantic Based Approach* dan *Truth Discovery* dilakukan pengujian atau evaluasi untuk mendapatkan nilai dari akurasi, presisi, *recall*, dan *f1-score*. Pada pengujian ini, seluruh data kalimat Bahasa Indonesia digunakan sebagai data uji, hal ini terjadi karena dalam identifikasi yang tidak memerlukan data latih. Pengujian akan dilakukan sebanyak 10 kali dengan jumlah data uji yang berbeda-beda dalam setiap pengujian. Selanjutnya hasil evaluasi didapatkan melalui rata-rata dari 10 kali pengujian tersebut. Hasil evaluasi identifikasi ekspresi idiomatik menggunakan *Distributional Semantic Based Approach* dan *Truth Discovery* dapat dilihat pada Tabel 5.

Tabel 5. Hasil Evaluasi Identifikasi Ekspresi Idiomatik

Pengujian	Jumlah Data	Akurasi	Presisi	<i>Recall</i>	<i>F1-Score</i>
1	200	0.83	1.0	0.67	0.80
2	400	0.82	1.0	0.62	0.76
3	600	0.82	1.0	0.66	0.79
4	800	0.82	1.0	0.63	0.77
5	1000	0.82	1.0	0.65	0.79
6	1200	0.82	1.0	0.64	0.78
7	1400	0.83	1.0	0.66	0.79
8	1600	0.82	1.0	0.64	0.78
9	1800	0.81	1.0	0.64	0.78
10	2000	0.82	1.0	0.64	0.78
Rata-rata		0.82	1.0	0.64	0.78

Pada Tabel 5. di atas dapat kita lihat nilai akurasi, presisi, *recall* dan *f1-score* yang diperoleh dari pengujian dengan jumlah data 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 dan 2000. Dari pengujian yang telah dilakukan, dapat dikatakan model identifikasi ekspresi idiomatik menggunakan *Distributional Semantic Based Approach* dan *Truth Discovery* memiliki performa cukup baik dengan rata-rata nilai akurasi, presisi, *recall* dan *f1-score* yang diperoleh yaitu 0,82; 1,0; 0,64 dan 0,78. Rata-rata akurasi, presisi dan *f1-score* memiliki nilai yang cukup baik sedangkan nilai *recall* yang dihasilkan

cukup rendah, yang berarti jumlah ekspresi idiomatik yang tidak berhasil diidentifikasi cukup banyak. Hal tersebut dapat disebabkan oleh beberapa faktor sebagai berikut:

- a. Pada tahap klasifikasi, BERT tidak berhasil mengklasifikasikan kalimat ke dalam kategori `kalimat_idiom`.
- b. Prediksi kelas kata yang dihasilkan pada tahap *Part-of-Speech Tagging* kurang optimal sehingga menyebabkan proses *Chunking* juga memberikan hasil yang kurang optimal dalam menemukan frasa kandidat yang memiliki konstruksi sintaksis pembentuk ekspresi idiomatik.
- c. Keterbatasan data sumber web, dimana klaim berupa ekspresi idiomatik yang disediakan oleh sumber web cenderung sedikit sehingga pada proses identifikasi terdapat beberapa frasa yang sebenarnya merupakan idiom tapi tidak dinyatakan sebagai idiom karena tidak ada klaim dari sumber web atas frasa tersebut.

4. Kesimpulan

Berdasarkan pada penelitian yang telah dilakukan serta hasil yang diperoleh dari 10 kali pengujian dengan jumlah data yang berbeda menunjukkan bahwa identifikasi ekspresi idiomatik menggunakan *Distributional Semantic Based Approach* dan *Truth Discovery* memiliki performa yang cukup baik dalam mengidentifikasi ekspresi idiomatik pada kalimat berbahasa Indonesia dengan rata-rata akurasi sebesar 0,82; presisi sebesar 1,0; *recall* sebesar 0,64 dan *f1-score* sebesar 0,78. Dari penelitian yang telah dilakukan serta hasil yang diperoleh, saran-saran yang dapat disampaikan untuk dapat dipertimbangkan dalam pengembangan dari penelitian selanjutnya yaitu Algoritma HMM untuk *Part-of-Speech Tagging* dapat diganti dengan algoritma lainnya seperti *Brill Tagger* yang menggunakan aturan leksikal dan kontekstual berdasarkan *Transformation Based Learning* untuk mendapatkan hasil prediksi kelas kata yang lebih optimal. Kemudian memperbanyak data sumber web yang digunakan dalam membangun model *Truth Discovery* sehingga dalam identifikasi ekspresi idiomatik memperoleh akurasi, presisi, *recall* dan *f1-score* yang lebih baik.

Daftar Pustaka

- [1] V. V. Virdaus, "Ekspresi Idiomatik dalam Lirik Nine Track Mind Charlie Puth Album 2016", *Media of Teaching Oriented and Children*, vol.4, no.1, 2020.
- [2] A. Barrera, R. Verma and R. Vincent, "SemQuest: University of Houston's Semantics-based Question Answering System" in *Text Analysis Conference (TAC)*, Houston, Nist.Gov, 2011.
- [3] O. Zayed, J. P. McCrae, and P. Buitelaar, "Phrase-level Metaphor Identification using Distributed Representations of Word Meaning," in *Proceedings of the Workshop on Figurative Language Processing*, New Orleans, Louisiana, 2018, pp. 81–90.
- [4] E. Shutova, D. Kiela, J. Maillard, "Black Holes and White Rabbits: Metaphor Identification with Visual Features" in *Proceedings of NAACL-HLT*, San Diego, California, 2016, pp.160–170.
- [5] A. Chaer, *Pengantar Semantik Bahasa Indonesia*, Jakarta: Rineka Cipta, 2013.
- [6] A. Anggito, and J. Setiawan, *Metodelogi Penelitian Kualitatif*, Jawa Barat: CV Jejak, 2018.
- [7] D. Sarkar, *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second ed.*, Bangalore, Karnataka, India: Appress Media, 2019.
- [8] A. E. Karyawati, P. A. Utomo, and I. G. A. Wibawa, "Comparison of SVM and LWC for Sentiment Analysis of SARA," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 16, no. 1, p. 45, 2022, doi: 10.22146/ijccs.69617.
- [9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding" in *NAACL HLT (Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies)*, Minneapolis, Minnesota, 2019, vol.1, pp. 4171–4186.
- [10] S. Kachuee and M. Sharifkhani, "TiltedBERT: Resource Adjustable Version of BERT," 2022, [Online]. Available: <http://arxiv.org/abs/2201.03327>.

- [11] D. Jurafsky, and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Third ed.*, Palo Alto: Stanford University, 2017.
- [12] N. A. Sanjaya, T. Abdessalem, M. L. Ba, and S. Bressan, "Harnessing Truth Discovery Algorithms on The Topic Labelling Problem", in *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services*, 2018, pp. 8-14.
- [13] J. Pasternack and D. Roth, "Knowing What to Believe (when you already know something)", in *Coling (International Conference on Computational Linguistics)*, 2010, vol. 2, pp. 877–885.

Implementasi LSTM pada Analisis Sentimen *Review* Film Menggunakan *Adam* dan *RMSprop Optimizer*

Karlina Surya Witanto^{a1}, Ngurah Agus Sanjaya ER^{a2}, AAIN Eka Karyawati^{a3}, I Gusti Agung Gede Arya Kadyanan^{a4}, I Ketut Gede Suhartana^{a5}, Luh Gede Astuti^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Bali, Indonesia

¹gabriella.linatan@gmail.com

²agus_sanjaya@unud.ac.id

³eka.karyawati@unud.ac.id

⁴gungde@unud.ac.id

⁵ikg.suhartana@unud.ac.id

⁶lg.astuti@unud.ac.id

Abstract

Movies are an entertainment that is in great demand by many groups from children, teenagers, adults, and parents. In the current digital era, various films can be watched on television to digital streaming services. Public opinion on the films watched can be in the form of positive opinions or negative opinions. Sentiment analysis is one of the fields of Natural Language Processing (NLP) which is able to build a system to recognize and extract opinions in the form of text, sentiment analysis is usually used to find out people's opinions or assessments of a products, services, politics, or other topics. Through sentiment analysis from the collection of reviews, the public can get various recommendations for films that can be watched. The method implemented to classify review data into positive reviews and negative reviews in this study is LSTM by comparing two different optimizers, namely Adam and RMSprop. This study succeeded in providing sentiment predictions with different optimizers with accuracy values for the LSTM application with Adam Optimizer reaching 77.11% and the LSTM application with RMSprop reaching 80.07%.

Keywords: Film, Review, Sentiment, NLP, LSTM, Adam, RMSprop

1. Pendahuluan

Film adalah sebuah hiburan yang banyak diminati oleh banyak golongan dari anak-anak, remaja, dewasa, dan orang tua. Pada era digital saat ini, berbagai film dapat ditonton melalui televisi hingga melalui layanan *digital streaming*. Berbagai layanan *digital streaming* seperti Netflix, Viu, Iflix, We TV, dan layanan *movie streaming* lainnya mampu menarik lebih dari dua ratus juta pelanggan hingga tahun 2020. Pada tahun 2020 jumlah penonton mengalami peningkatan yang sangat mencolok karena adanya keterkaitan dengan pandemi Covid-19 yang membuat minat penonton lebih tinggi untuk menonton berbagai macam film [1]. Opini masyarakat terhadap film yang ditonton dapat berupa opini positif maupun opini negatif. Opini-opini tersebut dapat ditemukan pada sebuah *review* dan dapat dianalisis dengan analisis sentimen. Analisis sentimen atau *opinion mining* merupakan pengolahan bahasa alami untuk melacak sikap, perasaan, atau penilaian masyarakat terhadap suatu topik tertentu, produk, atau jasa. Analisis sentimen merupakan salah satu bidang dari *Natural Language Processing* (NLP) yang mampu membangun suatu sistem untuk mengenali dan melakukan ekstraksi opini dalam bentuk teks, analisis sentimen biasanya digunakan untuk mengetahui opini atau pendapat masyarakat terhadap layanan atau jasa, produk, politik, ataupun topik-topik lain. Melalui analisis sentimen dari kumpulan *review* tersebut, masyarakat bisa mendapatkan berbagai rekomendasi film yang dapat ditonton. *Long Short Term Memory* (LSTM) merupakan pengembangan dari algoritma *Recurrent Neural Network* (RNN) dimana terdapat modifikasi pada RNN dengan menambahkan *memory cell* atau *memory unit* yang dapat menyimpan informasi yang dipelajari LSTM dalam jangka waktu yang panjang [2]. LSTM memberikan solusi untuk mengatasi terjadinya *vanishing gradient* pada RNN saat memproses data *sequential* yang panjang. Permasalahan *vanishing gradient* ini mengakibatkan RNN gagal dalam menangkap *long term dependencies* [3], sehingga mengurangi akurasi dari suatu prediksi pada RNN [4]. Terdapat penelitian dengan mengimplementasikan metode LSTM untuk melakukan analisis sentimen pada situs IMDB dengan menerapkan *word2vec* [5]. Penelitian tersebut memberikan

hasil akurasi yang cukup baik untuk memprediksi sentimen, yaitu dengan nilai akurasi sebesar 80%. Penelitian lain dengan C-LSTM dengan menerapkan *Adam Optimizer* pada Klasifikasi Berita Bahasa Indonesia mampu memberikan nilai akurasi sebesar 93,27% [6]. Pada penelitian ini akan dilakukan analisis sentimen terhadap kumpulan *review* film melalui IMDb Largest Review Dataset yang sudah didapatkan melalui website IEEE Dataport. Metode yang diimplementasikan untuk melakukan klasifikasi terhadap data *review* ke dalam *review* positif dan *review* negatif adalah LSTM. Penelitian ini juga akan membandingkan hasil performa dari perbandingan dua *optimizer* yaitu *Adam Optimizer* dan *RMSprop Optimizer* terhadap penerapan algoritma LSTM, dimana kedua *optimizer* tersebut mampu mengoptimalkan performa dari algoritma LSTM. Sehingga dengan penggunaan *optimizer* yang tepat, diharapkan *output* yang akan diberikan mampu memberikan rekomendasi film kepada *user* serta memberikan prediksi analisis sentimen *review* yang benar.

2. Metode Penelitian

Penelitian ini meliputi beberapa tahapan yaitu, pengumpulan data teks kumpulan *review* film melalui dataset *IMDB Largest Review* dalam Bahasa Inggris yang diambil melalui web IEEE Dataport [7], *preprocessing*, *word embedding*, pemodelan *Long Short Term Memory* (LSTM) dengan menerapkan *K-Fold Cross Validation* untuk klasifikasi dengan membandingkan dua *optimizer* yang berbeda yaitu *Adam* dan *RMSprop Optimizer*, dan tahap evaluasi.

2.1. Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data sekunder, dimana data tersebut sudah tersedia sebelum peneliti memulai penelitian, dan data tersebut berhubungan dengan penelitian yang akan dilakukan oleh peneliti [8]. Data yang diambil merupakan dokumen kumpulan *review* film melalui dataset *IMDB Largest Review* yang bersumber dari web IEEE Dataport [7]. Pengumpulan data teks kumpulan *review* film melalui dataset *IMDB Largest Review* dalam Bahasa Inggris yang diambil melalui web IEEE Dataport [7]. Data yang digunakan pada penelitian ini sebanyak 14.000 data *review* film dalam Bahasa Inggris, yang mengandung 7.000 *review* dengan label positif dan 7.000 *review* dengan label negatif dalam format *.csv. Data tersebut akan digunakan untuk melakukan proses pelatihan dan pengujian dari model yang akan dibangun menggunakan algoritma LSTM, serta untuk mengembangkan *deep learning model* terhadap *binary classification* berdasarkan *review* film.

2.2. Preprocessing

Setelah melakukan pengumpulan data, tahap selanjutnya yaitu *preprocessing*. Langkah ini perlu dilakukan untuk mempersiapkan teks yang menjadi sumber data agar dapat diproses ke tahap selanjutnya [9]. Hal-hal yang dilakukan pada langkah ini, yaitu:

- a. *Delete Null Value*
Proses ini dilakukan dengan menghapus kolom yang memiliki nilai *null* sehingga tidak menyebabkan *error* ketika proses pelatihan dan pengujian model [10].
- b. *Tokenizing*
Kata-kata yang terdapat dalam dokumen akan dipecah menjadi bagian yang lebih kecil berupa kata tunggal yang memiliki arti atau sering disebut token, misalnya berupa kata, frasa, atau kalimat [11].
- c. *Case Folding*
Tahapan ini dilakukan untuk mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar, yaitu huruf kecil atau *lowercase* [12].
- d. *Stopword removal*
Stopword removal merupakan tahap *preprocessing* dimana dilakukan penghapusan kata-kata yang sering muncul dan memiliki arti sedikit atau tidak penting yang disebut sebagai *stop words*. Penghapusan *stop words* dilakukan untuk mengurangi kata dalam sistem [13].
- e. *Punctuation removal*
Punctuation removal merupakan proses untuk menghilangkan simbol-simbol yang terdapat pada dokumen teks [14].
- f. *Text to Sequence*
Proses ini merupakan proses merepresentasikan kamus kata (senilai *num_words*) menjadi ke bentuk angka [15].
- g. *Split Data*
Proses pembagian data dengan persentase 90% data pelatihan dan 10% data pengujian dari total jumlah data [16]. Langkah ini merupakan langkah untuk melatih *neural network* agar dapat mengenali pola sehingga model algoritma yang diterapkan bisa mendapatkan akurasi yang baik.

2.3. Word Embedding

Proses *word embedding* dilakukan untuk merubah representasi dari kata-kata yang terdapat dalam *dataset* menjadi sebuah vektor. Proses ini diawali dengan membuat list dari kata-kata yang terdapat dalam teks. Pada penelitian ini peneliti menggunakan *embedding* dari *Keras*. Setelah dilakukan tahapan *preprocessing* dan menghasilkan ulasan dalam bentuk list kata yang sudah ditokenisasi, tahap selanjutnya yaitu memberikan indeks pada setiap kata pada *dataset*. Pada tahap ini diperlukan dua parameter. Parameter yang pertama yaitu parameter *jumlah_vocab* yang berfungsi untuk mengatur ukuran kamus kata yang ingin digunakan. Pada penelitian ini, nilai dari *jumlah_vocab* yang digunakan oleh peneliti sebesar 10.000 *vocab*, dimana 10.000 *vocab* yang disimpan memiliki persentase yang besar terhadap kemunculan suatu kata. Parameter selanjutnya adalah *output_dim* yang berguna untuk mengatur panjang urutan vektor (dimensi vektor).

```


Pseudocode Word Embedding

algorithm : create embedding
input : d = dataset
output : matrix  $W_{(10,000,300)}$  of one-hot vectors for each possible byte value (0-9999)
let f be a list of tuples(byte_value, frequency)
for i=0 to 9999 do
    freq  $\leftarrow$  0

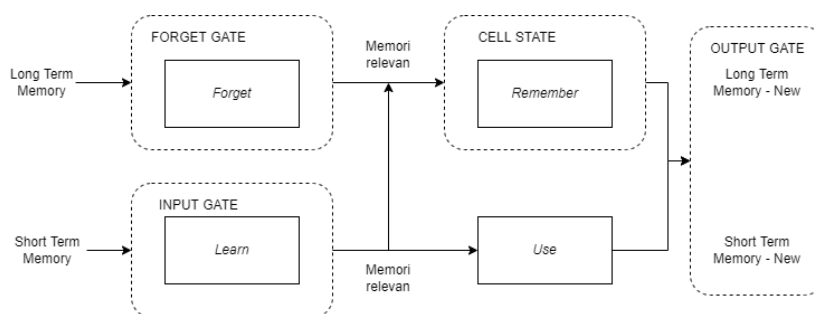
    for each item j in d do
        freq  $\leftarrow$  freq + frequencyOfOccurrence(i,j)
    end for
    append (i, freq) tuple of f
end for
f  $\leftarrow$  sort of based on frequencies
W  $\leftarrow$  embedding(f, 300)
return W
    
```

Gambar 1. Pseudocode Word Embedding

Pada proses *word embedding* seperti pada Gambar 1 terdapat metode yang digunakan yaitu *one hot encoder* atau *text to sequence* yang digunakan untuk mengubah kata menjadi angka atau bentuk numerik yang nantinya angka tersebut akan diproses pada tahap *word embedding* untuk diubah menjadi vektor. *Word embedding* menggunakan kumpulan kosakata dari data teks pelatihan sebagai input yaitu sebanyak 10.000 kata kemudian mempelajari representasi vektor dari kumpulan kata tersebut. *Word embedding* bekerja dengan menangkap informasi pada setiap kata atau *byte* yang memiliki skor kesamaan yang tinggi dan kata-kata yang memiliki tingkat kesamaan yang tinggi akan ditempatkan pada posisi yang berdekatan.

2.4. Long Short-Term Memory (LSTM)

LSTM merupakan bagian dari algoritma *Recurrent Neural Network* (RNN). LSTM memiliki koneksi secara berulang dan strukturnya seperti rantai yang dapat mempelajari ketergantungan jangka panjang (*Long-term Dependencies*) dalam kasus prediksi yang sebelumnya menjadi kelemahan algoritma RNN. Selain itu algoritma ini juga untuk menangani permasalahan pada *machine learning*, *speech recognition*, dan lain-lain [17]. LSTM menerapkan memori jangka panjang pada jaringan saraf untuk mengurangi masalah *vanishing gradient*.

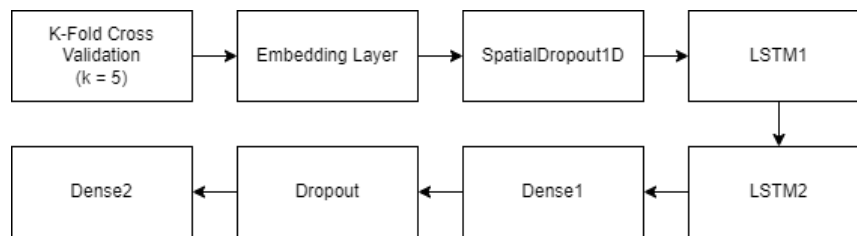


Gambar 2. Arsitektur LSTM secara Umum

Arsitektur LSTM pada Gambar 2 secara umum memiliki empat gerbang yaitu, *forget gate*, *input gate*, *cell state*, dan *output gate*. *Forget gate*, akan menerima informasi lama (*Long Term Memory*) dan akan dikalikan dengan aktivasi *sigmoid* yang bernilai antara 0 dan 1. Jika nilai yang dihasilkan mendekati 1 maka informasi tersebut dinyatakan relevan dan akan diproses, jika nilainya mendekati nilai 0 maka informasi tersebut akan dilupakan. *Input gate*, akan menerima atau mempelajari informasi baru (*Short*

Term Memory) dan informasi tersebut akan disimpan ke dalam *cell state* dan akan digunakan untuk memproses *output*. *Cell state*, informasi yang tidak dilupakan dan informasi dari *input gate* akan disimpan dalam *cell state*. Setiap informasi yang disimpan dalam setiap *gate* tersebut akan menghasilkan *Long Term Memory New* dan *Short Term Memory New* yang akan dijadikan sebagai *output* dari LSTM yang memberikan prediksi serta memori atau informasi yang paling relevan. Gambaran arsitektur LSTM secara umum tersebut menunjukkan bahwa LSTM mampu menyimpan memori dalam jangka panjang dan dapat mengurangi masalah *vanishing gradient* karena LSTM selalu memperbaharui memori yang dibutuhkan.

Seperti pada Gambar 3, pembuatan model LSTM pada penelitian ini mengimplementasikan metode *K-Fold Cross Validation* serta *sequence model* yang terdiri dari tujuh layer yang diantaranya menerapkan *Embedding layer*, *SpatialDropout1D layer*, *LSTM1 layer*, *LSTM2 layer*, *Dense1 layer*, *Dropout layer*, serta *Dense2 layer*.



Gambar 3. Layer Model LSTM

Pada penelitian ini menerapkan *K-Fold Cross Validation* dengan jumlah $k=5$ yang digunakan untuk membagi data ke dalam tiap ruang *fold*. Total jumlah data *review* sebanyak 14.000 data, yaitu 7.000 data *review* positif dan 7.000 data *review* negatif. Pada proses *split data*, data yang akan dilatih pada model sebanyak 90% dari total jumlah data. Sehingga 12.600 data akan dilatih dengan model yang akan dibangun. Data tersebut akan dibagi kedalam lima *fold* sama rata dan selanjutnya akan dilatih ke dalam model LSTM.

Embedding layer terdapat parameter ukuran *vocab* dan ukuran vektor *embedding*. Pada penelitian ini, peneliti menerapkan nilai sebesar 10.000 untuk ukuran *vocab*, dimana jumlah tersebut sudah memberikan representasi kata yang paling besar atau yang sering muncul. Lalu untuk ukuran vektor *embedding*, penulis menerapkan nilai sebesar 300. *SpatialDropout1D layer* memiliki fungsi untuk mencegah terjadinya *overfitting* dan *underfitting* pada data tekstual. *SpatialDropout1D* bekerja dengan mempertahankan fitur dalam data yang memiliki korelasi tinggi satu sama lain, sehingga fitur yang memiliki korelasi yang tinggi satu sama lain tidak akan masuk ke dalam proses regulasi dan fitur yang berkorelasi tinggi tersebut dapat digunakan.

LSTM1 layer yaitu LSTM layer pertama memiliki nilai *memory unit* yaitu jumlah unit LSTM dan menerapkan atribut *return_sequences=True*. Parameter *memory unit* nilainya harus ditentukan secara eksplisit dengan rentang angka yang sesuai dengan data yang penulis miliki untuk memberikan akurasi yang maksimal dari model tersebut dan dilakukan secara repetitif hingga ditemukan *hyperparameter* terbaik. *LSTM2 layer* yaitu LSTM layer kedua memiliki nilai *memory unit* yaitu jumlah unit LSTM, *kernel_regularizer(L2)*, dan *recurrent_regularizer(L2)*. Parameter *memory unit* nilainya harus ditentukan secara eksplisit dengan rentang angka yang sesuai dengan data yang penulis miliki untuk memberikan akurasi yang maksimal dari model tersebut dan dilakukan secara repetitif hingga ditemukan *hyperparameter* terbaik. *Kernel_regularizer* dan *recurrent_regularizer* merupakan parameter yang digunakan untuk mencegah terjadinya *overfitting* serta mempercepat proses *learning*, nilai parameter acuan dari fungsi regulasi yaitu 0 hingga tak terhingga [18]. Versi L2 pada *regularizer* bekerja dengan memperkirakan rata-rata data untuk menghindari *overfitting*.

Dense1 layer yaitu *dense layer* pertama yang berfungsi untuk menambahkan layer yang *fully connected*, artinya setiap neuron menerima input dari neuron lainnya sehingga saling terhubung. Jumlah unit yang digunakan pada *dense1 layer* sebanyak delapan neuron diikuti dengan fungsi aktivasi *relu*, dimana fungsi aktivasi *relu* berguna untuk mengoptimalkan fungsi aktivasi *sigmoid* yang akan diterapkan pada *Dense2 layer*. Selanjutnya terdapat *dropout layer* yang berfungsi untuk mencegah terjadinya *overfitting* dan mempercepat proses *learning*. Nilai yang digunakan berkisar dari 0 sampai 1, dimana semakin kecil nilai *dropout* maka data *overfit*. Sebaliknya, data akan menjadi *underfit*. *Dense2 layer* yaitu *dense layer* kedua yang berfungsi untuk menambahkan layer yang *fully connected* dan

disesuaikan dengan jumlah *class* yang ditentukan, artinya setiap neuron menerima input dari semua neuron lainnya sehingga saling terhubung. Jumlah unit yang digunakan pada *dense2 layer* sebanyak dua neuron karena terdapat dua kelas yang telah ditentukan, yaitu kelas positif dan kelas negatif. Penelitian berupa *binary-classification*, maka digunakan *loss function = binary_crossentropy*, fungsi optimasi, dan aktivasi *sigmoid*. Fungsi aktivasi *sigmoid* digunakan untuk memilih probabilitas kelas terbesar pada klasifikasi biner.

Pada penelitian ini arsitektur *deep learning* yang sudah dibangun akan dibandingkan dengan beberapa nilai *epoch* dan nilai *batch size*. Nilai *epoch* dapat menentukan berapa kali suatu model bekerja untuk mengolah data latih, satu *epoch* berarti setiap sampel dalam data latih memiliki kesempatan untuk memperbarui parameter model internal. Sedangkan untuk nilai *batch size* merupakan jumlah berapa banyak sampel dalam data latih yang akan digunakan dalam satu iterasi. Penulis akan melakukan penelitian dengan dua fungsi optimasi yang berbeda yaitu *Adam Optimizer* dan *RMSprop Optimizer*. Penulis akan melakukan percobaan secara repetitif hingga mendapatkan hasil yang maksimal dan menghasilkan perbandingan diantara kedua fungsi optimasi tersebut.

2.5. Adam Optimizer

Adaptive Moment Estimation (Adam) merupakan salah satu algoritma yang dapat menggantikan prosedur *stochastic gradient descent* klasik untuk memperbaharui *weight network* berdasarkan data *training* secara iteratif. Cara kerja *Adam* dapat digambarkan sebagai penggabungan sifat terbaik dari dua ekstensi *stochastic gradient descent* yaitu *adaptive gradient algorithm* dan *root mean square propagation* dengan penggabungan tersebut *Adam* mampu memberikan pengoptimalan suatu algoritma yang mampu menangani *sparse gradients* pada *noisy problem*. [19]. Dengan penggunaan teknik optimasi yang mampu menurunkan gradien, metode ini sangat efisien ketika bekerja pada data yang besar dan parameter yang besar. Sebelum menggunakan fungsi optimasi *Adam*, terdapat beberapa nilai yang harus didefinisikan terlebih dahulu, yaitu:

1. $m = 0$
2. $v = 0$
3. $\epsilon = 10^{-8}$
4. $t = 0$
5. $\alpha = 0.001$
6. $\beta_1 = 0.9$
7. $\beta_2 = 0.999$

Tahap-tahap yang dilakukan oleh *Adam Optimizer* yaitu:

1. Tambah t setiap iterasi
$$t = t + 1 \tag{1}$$

2. Menghitung *gradien*
$$g_t \leftarrow \nabla \theta f_t(\theta_t - 1) \tag{2}$$

3. Menghitung bias *first moment*
$$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3}$$

4. Menghitung bias *second moment*
$$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t \cdot g_t \tag{4}$$

5. Memperbaiki bias *first moment*
$$\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t} \tag{5}$$

6. Memperbaiki bias *second moment*
$$\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t} \tag{6}$$

7. Memperbaiki parameter
$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \tag{7}$$

Keterangan:

g = *gradient*

m = *first moment*

v = *second moment*

β_1, β_2 = *Exponential decay rates*

α = *Step size* atau *learning rate*

θ = parameter bobot yang akan diperbaiki

Ketujuh tahapan dilakukan secara berulang sebanyak jumlah *dataset* yang diambil secara acak hingga semua *epoch* selesai. Perbedaan antara *Adam* dengan *RMSProp* terletak pada perubahan *learning*

rate dimana *Adam* melakukan *bias correction* pada perhitungannya seperti pada tujuh langkah diatas [20].

2.6. *RMSprop Optimizer*

RMSprop memiliki kemiripan dengan *Adaprop*, yang merupakan bentuk improvisasi dari *Adagrad*. Gradien fungsi yang sangat kompleks seperti jaringan saraf memiliki kecenderungan untuk menghilang atau terjadinya masalah *vanishing gradient*. *RMSprop* merupakan fungsi optimasi yang memanfaatkan besarnya gradien terbaru untuk menormalkan gradien, fungsi ini mampu menjaga rata-rata bergerak di atas gradien *root mean square* sehingga disebut RMS. *RMSprop* merupakan salah satu fungsi optimasi yang mempertahankan rata-rata dari kuadrat gradien untuk setiap bobot, yang dapat dilihat seperti pada formula 8.

$$MeanSquare(w, t) = \rho * MeanSquare(w, t - 1) + 0.1 \left(\frac{\partial E}{\partial w}(t) \right)^2 \quad (8)$$

Keterangan:

w = bobot

t = *timestep*

$\rho = 0.9$

$\frac{\partial E}{\partial w}$ = *gradient*

2.7. Evaluasi

Evaluasi merupakan tahapan yang digunakan untuk mengukur kinerja dari suatu model dari metode yang diusulkan. Evaluasi dilakukan dengan menghitung nilai *precision*, *recall*, *f-1 score*, dan akurasi.

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (10)$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (11)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (12)$$

Keterangan:

TP = *True Positive* (total prediksi benar dari data positif)

TN = *True Negative* (total prediksi benar dari data negatif)

FP = *False Positive* (total prediksi salah dari data negatif)

FN = *False Negative* (total prediksi salah dari data positif)

3. Hasil dan Pembahasan

Penulis melakukan *hyperparameter tuning* untuk mendapatkan parameter terbaik yang nilainya akan diterapkan pada model *deep learning* yang akan dibangun. Sebelum mengimplementasikan model *Long Short-Term Memory* (LSTM) penulis terlebih dahulu membagi 90% data *training* ke dalam tiap *fold* dengan metode validasi, yaitu *K-Fold Cross Validation* dengan nilai $k=5$, artinya dari total 90% data *training* akan dibagi menjadi 5. Terdapat dua model *deep learning* yang akan dikembangkan oleh peneliti, yaitu model yang menerapkan fungsi optimasi *Adam* dan model yang menerapkan fungsi optimasi *RMSprop*.

3.1. Analisa Parameter

Pada penelitian ini akan dilakukan pengujian *hyperparameter* yaitu *batch size* dan *epoch* dengan menggunakan metode *K-fold Cross Validation* untuk metode validasi dengan menerapkan nilai *fold* sebanyak lima, tujuannya untuk mendapatkan *hyperparameter* dengan performa terbaik. *Hyperparameter* yang akan diujikan pada proses *training* model untuk klasifikasi *sentiment* dari *review* film dapat dilihat pada Tabel 1. Nilai *epoch* dapat diatur ke nilai integer antara satu dan tak terhingga [21]. Sedangkan nilai *batch size* akan dimulai dari nilai 32, 64, 128, dan 126 [22], dengan memperhatikan pada kasus yang dikerjakan dengan melakukan percobaan.

Tabel 1. *Hyperparameter* untuk Skenario Pengujian

No	Parameter	Ukuran
1	<i>Batch size</i>	[32, 64, 128, 256]
3	<i>Epoch</i>	[3, 5, 10, 15, 20]

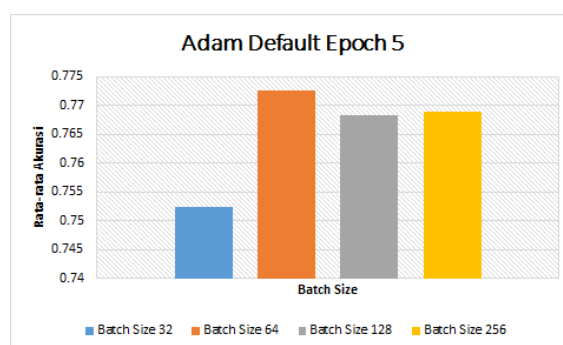
Pada proses ini peneliti menerapkan beberapa parameter yang akan diujikan untuk membandingkan fungsi optimasi *Adam* dan *RMSprop*. *Hyperparameter* yang akan diujikan yaitu nilai *batch size* dan *epoch* yang berbeda-beda. Pengujian yang akan dilakukan yaitu dengan mencari nilai *batch size* yang terbaik terlebih dahulu, nilai *batch size* yang akan diujikan yaitu 32, 64, 128, dan 256. Melalui *5-fold Cross Validation* penentuan nilai terbaik *batch size* diambil dari rata-rata akurasi dari data validasi. Setelah mendapatkan nilai *batch size* yang terbaik, nilai tersebut akan digunakan sebagai nilai *batch size* pada pengujian untuk mencari nilai *epoch* yang terbaik. Nilai *epoch* yang akan diujikan yaitu, 3, 5, 10, 15, dan 20. Melalui *5-fold Cross Validation* penentuan nilai terbaik *epoch* diambil dari rata-rata akurasi dari data validasi, lalu selanjutnya model tersebut akan diujikan dengan data *testing* yang sebelumnya belum pernah dikenali oleh sistem. Secara lebih jelas proses tuning *hyperparameter* dipaparkan pada materi di bawah ini.

a. Pengujian pada *Adam Optimizer*

Pengujian yang dilakukan dengan *Adam Optimizer* akan dilakukan dengan parameter-parameter *batch size* dan *epoch*. Hal yang pertama dilakukan adalah mencari nilai *batch size* yang terbaik, untuk mendapatkan nilai tersebut peneliti melakukan pengujian dengan membandingkan nilai *batch size* 32, 64, 128, dan 256 pada *default epoch* sebesar 5 *epoch*. Pengujian dengan *Adam Optimizer* untuk mendapatkan nilai *batch size* yang terbaik dapat dilihat pada Tabel 2 dan Gambar 4.

Tabel 2. Pengujian *Adam Optimizer* Untuk Mencari Nilai *Batch Size* Terbaik

Batch Size	Epoch	Fold					Mean
		1	2	3	4	5	
32	5	0.77381	0.756349	0.72619	0.751587	0.753968	0.752381
64	5	0.786111	0.776587	0.775794	0.763889	0.761111	0.772698
128	5	0.752381	0.775	0.769444	0.761508	0.782937	0.768254
256	5	0.78254	0.767857	0.759127	0.763889	0.771032	0.768889



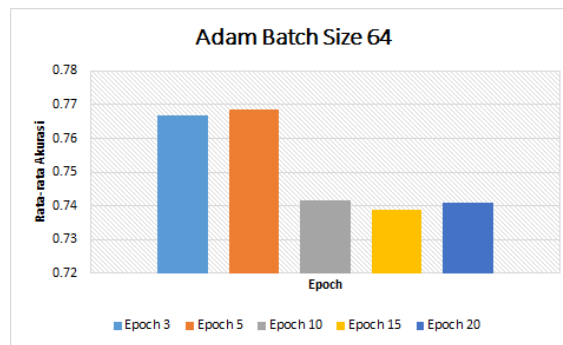
Gambar 4. Grafik Hasil Uji *Adam Optimizer* Dengan *Default Epoch*

Berdasarkan hasil pengujian pada Tabel 2 dan Gambar 4, dengan menerapkan *5-fold Cross Validation* nilai *batch size* 64 memberikan nilai rata-rata akurasi yang terbaik dibandingkan nilai *batch size* yang lainnya, yaitu sebesar 0.772698.

Nilai *batch size* 64 akan dijadikan nilai *default batch size* untuk diujikan pada nilai *epoch* yang berbeda-beda yaitu 3, 5, 10, 15, 20. Pengujian dengan *Adam Optimizer* terhadap nilai *epoch* yang berbeda-beda dapat dilihat pada Tabel 3 dan Gambar 5.

Tabel 3. Pengujian *Adam Optimizer* Dengan Nilai *Epoch*

Batch Size	Epoch	Fold					Mean
		1	2	3	4	5	
64	3	0.793651	0.753968	0.764683	0.774206	0.756746	0.766667
64	5	0.759524	0.768651	0.768651	0.75873	0.777778	0.768651
64	10	0.722619	0.757143	0.736905	0.743254	0.74881	0.741746
64	15	0.738889	0.740079	0.741667	0.721825	0.751984	0.738889
64	20	0.742063	0.734921	0.736905	0.739286	0.751587	0.740952



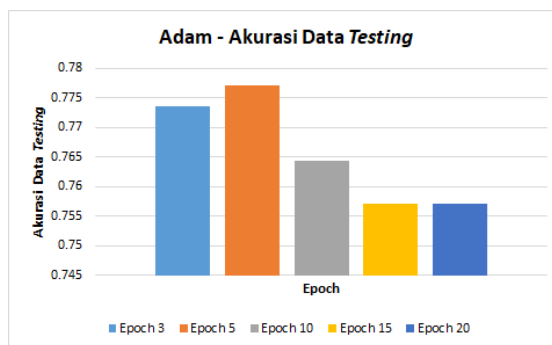
Gambar 5. Grafik Hasil Uji *Adam Optimizer* Dengan *Batch Size* Terbaik

Berdasarkan hasil pengujian pada Tabel 3 dan Gambar 5, dengan menerapkan *5-fold Cross Validation* nilai *batch size* 64 dan *epoch* 5 memberikan nilai rata-rata akurasi yang terbaik dibandingkan nilai yang lainnya, yaitu sebesar 0.768651.

Berdasarkan hasil pengujian, model dari *Adam Optimizer* yang mampu memberikan nilai rata-rata akurasi yang terbaik akan diujikan pada *data testing* seperti pada Tabel 4 dan Gambar 6.

Tabel 4. Pengujian Parameter Terbaik Pada Data *Testing Adam Optimizer*

Optimizer	Batch Size	Epoch	Acc Test
Adam	64	3	0.7736
	64	5	0.7771
	64	10	0.7643
	64	15	0.7571
	64	20	0.7571



Gambar 6. Grafik Hasil Uji *Adam Optimizer* Dengan Data *Testing*

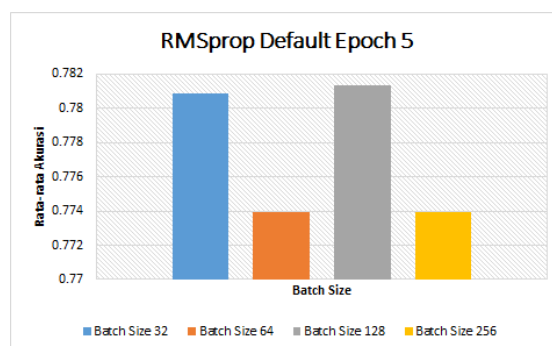
Berdasarkan hasil pada Tabel 4 dan Gambar 6, *Adam Optimizer* mampu memberikan nilai optimal pada *batch size* 64 dan *epoch* 5 dengan nilai akurasi pada data *testing* sebesar 0.7771.

b. Pengujian pada *RMSprop Optimizer*

Pengujian yang dilakukan dengan *RMSprop Optimizer* akan dilakukan dengan parameter-parameter *batch size* dan *epoch*. Hal yang pertama dilakukan adalah mencari nilai *batch size* yang terbaik, untuk mendapatkan nilai tersebut peneliti melakukan pengujian dengan membandingkan nilai *batch size* 32, 64, 128, dan 256 pada *default epoch* sebesar 5 *epoch*. Pengujian dengan *RMSprop Optimizer* untuk mendapatkan nilai *batch size* yang terbaik dapat dilihat pada Tabel 5 dan Gambar 7.

Tabel 5. Pengujian *RMSprop Optimizer* Untuk Mencari Nilai *Batch Size* Terbaik

Batch Size	Epoch	Fold					Mean
		1	2	3	4	5	
32	5	0.794444	0.780952	0.772222	0.778175	0.778571	0.780873
64	5	0.79127	0.765873	0.756746	0.779365	0.776587	0.773968
128	5	0.803968	0.788889	0.762698	0.773016	0.778175	0.781349
256	5	0.782937	0.772222	0.754762	0.778571	0.781349	0.773968



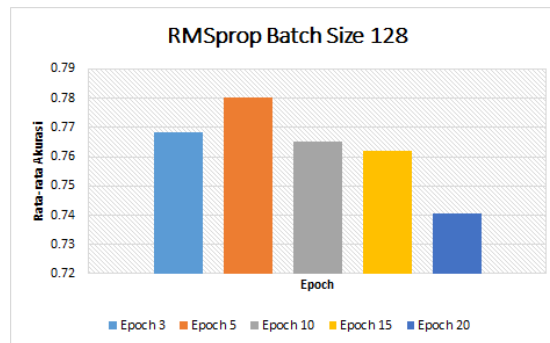
Gambar 7. Grafik Hasil Uji *RMSprop Optimizer* Dengan *Default Epoch*

Berdasarkan hasil pengujian pada Tabel 5 dan Gambar 7, dengan menerapkan 5-fold *Cross Validation* nilai *batch size* 128 memberikan nilai rata-rata akurasi yang terbaik dibandingkan nilai *batch size* yang lainnya, yaitu sebesar 0.781349.

Nilai *batch size* 128 akan dijadikan nilai *default batch size* untuk diujikan pada nilai *epoch* yang berbeda-beda yaitu 3, 5, 10, 15, 20. Pengujian dengan *RMSprop Optimizer* terhadap nilai *epoch* yang berbeda-beda dapat dilihat pada Tabel 6 dan Gambar 8.

Tabel 6. Pengujian *RMSprop* Optimizer Dengan Nilai *Epoch*

Batch Size	Epoch	Fold					Mean
		1	2	3	4	5	
128	3	0.786905	0.786508	0.755159	0.76627	0.747222	0.768413
128	5	0.784921	0.778968	0.780556	0.776984	0.780556	0.780397
128	10	0.778968	0.755159	0.755159	0.757937	0.778968	0.765238
128	15	0.766667	0.771429	0.756746	0.745238	0.770635	0.762143
128	20	0.727381	0.756349	0.738095	0.721429	0.759921	0.740635



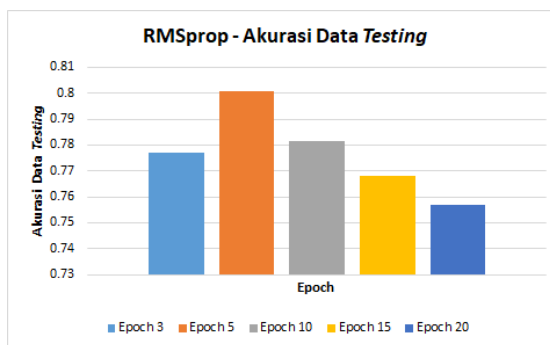
Gambar 8. Grafik Hasil Uji *RMSprop* Optimizer Dengan *Batch Size* Terbaik

Berdasarkan hasil pengujian Tabel 6 dan Gambar 8, dengan menerapkan *5-fold Cross Validation* nilai *batch size* 128 dan *epoch* 5 memberikan nilai rata-rata akurasi yang paling terbaik dibandingkan nilai yang lainnya, yaitu sebesar 0.780397.

Berdasarkan hasil pengujian, model dari *RMSprop* Optimizer yang mampu memberikan nilai rata-rata akurasi yang terbaik akan diujikan pada *data testing* seperti pada Tabel 7 dan Gambar 9.

Tabel 7. Pengujian Parameter Terbaik Pada Data *Testing* *RMSprop* Optimizer

Optimizer	Batch Size	Epoch	Acc Test
RMSprop	128	3	0.7771
	128	5	0.8007
	128	10	0.7814
	128	15	0.7679
	128	20	0.7571



Gambar 9. Grafik Hasil Uji *RMSprop Optimizer* Dengan *Data Testing*

Berdasarkan hasil pengujian pada Tabel 7 dan Gambar 9, *RMSprop Optimizer* mampu memberikan nilai optimal pada *batch size* 128 dan *epoch* 5 dengan nilai akurasi pada *data testing* sebesar 0.8007.

3.2. Evaluasi

Berikut merupakan hasil evaluasi pengujian dari model yang terbaik pada evaluasi LSTM menggunakan *Adam Optimizer* terhadap *data testing* dengan mendapatkan nilai akurasi, *precision*, *recall*, dan *f-1 score* yang dapat dilihat pada Tabel 8.

Tabel 8. Hasil *Confusion Matrix* Model *Adam Optimizer*

	Precision	Recall	F1-score
Negatif	0.76	0.79	0.78
Positif	0.79	0.76	0.78
Akurasi			0.78

Berikut merupakan hasil evaluasi pengujian dari model yang terbaik pada evaluasi LSTM menggunakan *RMSprop Optimizer* terhadap *data testing* dengan mendapatkan nilai akurasi, *precision*, *recall*, dan *f-1 score* yang dapat dilihat pada Tabel 9.

Tabel 9. Hasil *Confusion Matrix* Model *RMSprop Optimizer*

	Precision	Recall	F1-score
Negatif	0.77	0.85	0.81
Positif	0.84	0.75	0.79
Akurasi			0.80

4. Kesimpulan

Hasil tuning *hyperparameter* pada model LSTM untuk analisis sentimen *review* film menunjukkan bahwa model LSTM dengan *Adam Optimizer* dengan nilai *batch size* 64 dan nilai *epoch* 5 mampu memberikan performa yang terbaik dengan nilai akurasi sebesar 77,11%. Sedangkan model LSTM dengan *RMSprop Optimizer* dengan nilai *batch size* 128 dan nilai *epoch* 5 mampu memberikan performa yang terbaik dengan nilai akurasi sebesar 80,07%. Model LSTM dengan *Adam Optimizer* menunjukkan hasil yang baik yaitu dengan nilai *precision* pada sentimen negatif sebesar 76% dan pada sentimen positif sebesar 79%, nilai *recall* pada sentimen negatif sebesar 79% dan pada sentimen positif

sebesar 76%, nilai *f-1 score* pada sentimen negatif sebesar 78% dan pada sentimen positif sebesar 78%, serta nilai akurasi sebesar 77,11%. Sedangkan model LSTM dengan *RMSprop Optimizer* memberikan hasil yang lebih baik, dengan nilai *precision* pada sentimen negatif sebesar 77% dan pada sentimen positif sebesar 84%, nilai *recall* pada sentimen negatif sebesar 85% dan pada sentimen positif sebesar 75%, nilai *f-1 score* pada sentimen negatif sebesar 81% dan pada sentimen positif sebesar 79%, serta nilai akurasi sebesar 80,07%. Penerapan LSTM dengan membandingkan dua *optimizer* yaitu *Adam* dan *RMSprop* mampu dikembangkan berupa *website* analisis sentimen dimana *user* dapat memilih *optimizer* yang diinginkan serta *user* juga dapat mencari *review* film berdasarkan judulnya yang dimana *review* tersebut sudah diproses nilai prediksi sentimennya dengan model yang terbaik yaitu dengan penerapan *RMSprop Optimizer*.

Referensi

- [1] CNN Indonesia, "Pandemi 2020 Buat Netflix Kebanjiran 36,6 Juta Pelanggan Baru," 2021. <https://www.cnnindonesia.com/hiburan/20210120132336-220-596129/pandemi-2020-buat-netflix-kebanjiran-366-juta-pelanggan-baru> (accessed Jun. 10, 2021).
- [2] N. K. Manaswi, *Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras.*, 1 penyunt. India: Apress, 2018.
- [3] A. Saxena and A. Sukumar, "Predicting bitcoin price using lstm And Compare its predictability with arima model," *Int. J. Pure Appl. Math.*, 2018.
- [4] J. Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, "LSTM network: a deep learning approach for short-term traffic forecast.," *IET Intell. Transp. Syst.*, pp. 11(2), 68-75., 2017.
- [5] A. Hassan and A. Mahmood, "Deep learning for sentence classification," *2017 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2017*, 2017.
- [6] Y. Widhiyana, T. Semiawan, I. G. A. Mudzakir, and M. R. Noor, "Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, pp. 354–361, 2021.
- [7] A. Barigidad and A. Mustafi, "IMDB Movie Reviews Dataset," *IEEE Dataport*, 2020. <https://doi.org/10.1109/ICCS49678.2020.9276893>.
- [8] A. Anggito and J. Setiawan, *Metodologi Penelitian Kualitatif*. Jawa Barat: CV Jejak, 2018.
- [9] K. D. Y. Wijaya and A. E. Karyawati, "The Effects of Different Kernels in SVM Sentiment Analysis on Mass Social Distancing," *JELIKU*, vol. 9, pp. 161–168, 2020.
- [10] A. Ranjan, "Data Cleaning in Natural Language Processing," 2020. <https://medium.com/analytics-vidhya/data-cleaning-in-natural-language-processing-1f77ec1f6406>.
- [11] I. G. C. P. Yasa, N. A. Sanjaya ER, and L. A. A. R. Putri, "Sentiment Analysis of SnackReview Using the Naïve Bayes Method," *JELIKU*, vol. 8, pp. 333–338, 2020.
- [12] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps*. 2019.
- [13] H. Manning, C., Raghavan, P. & Schütze, *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2009.
- [14] Harshith, "Text Preprocessing in Natural Language Processing," 2019. <https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python-6113ff5decd8>.
- [15] D. Dahman, "Natural Language Processing," 2021. <https://medium.com/sysinfo/natural-language-processing-nlp-b54d6506efe2>.
- [16] F. Galbusera, "A Deep Learning Model for the Accurate and Reliable Classification of Disc Degeneration Based on MRI Data," *Invest. Radiol.*, vol. 56, no. 2, pp. 78–85, 2021, [Online]. Available: <https://doi.org/10.1097/RLI.0000000000000709>.
- [17] J. Brownlee, "A Gentle Introduction to Long Short-Term Memory Networks by the Experts," 2017.
- [18] Keras, "Layer Weight Regularizers," 2022. <https://keras.io/api/layers/regularizers/>.
- [19] D. P. Kingma and J. L. Ba, *A Method For Stochastic Optimization*. ICLR, 2015.
- [20] J. Brownlee, "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning," 2017. <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.
- [21] J. Brownlee, "Keras: Difference Between a Batch and an Epoch," 2018. [https://lms.onnocenter.or.id/wiki/index.php/Keras:_Difference_Between_a_Batch_and_an_Epoch#:~:text=batch dan epoch.,Perbedaan Batch dan Epoch%3F,jumlah sampel dalam dataset training](https://lms.onnocenter.or.id/wiki/index.php/Keras:_Difference_Between_a_Batch_and_an_Epoch#:~:text=batch%20dan%20epoch.,Perbedaan%20Batch%20dan%20Epoch%3F,jumlah%20sampel%20dalam%20dataset%20training).
- [22] A. Khumaidi, "Penguujian Algoritma LSTM untuk Prediksi Kualitas Udara dan Suhu Kota Bandung," *Telematika*, vol. 15, 2021.

Klasifikasi Cerita Pendek Berbahasa Bali Berdasarkan Umur Pembaca dengan Metode *Naive Bayes*

Luh Ristiari¹, AAIN Eka Karyawati², I Putu Gede Hendra Suputra³, Agus Muliantara⁴,
I Dewa Made Bayu Atmaja Darmawan⁵, I Made Widiartha⁶

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Bali, Indonesia

¹luhris21@gmail.com

²eka.karyawati@unud.ac.id

³hendra.suputra@unud.ac.id

⁴muliantara@unud.ac.id

⁵dewabayu@unud.ac.id

⁶madewidiartha@unud.ac.id

Abstract

Short stories are short stories that tell an event that has happened in a short and clear way. Parents should be able to choose short stories that are suitable for their children because if the stories that parents bring to children are not in accordance with their age, it can affect the development of children. In this study, we will build a system that can classify text. The method used in this research is Nave Bayes with feature selection, namely Genetic Algorithm. This research was conducted to help parents so that their children do not read short stories that are not appropriate for their age so that they do not interfere with their child's development. The data used are children's short stories, youth short stories and adult short stories in Balinese. The best model performance is generated in the training and validation process using new data. The results of testing the Naive Bayes method without feature selection are 66% accuracy, 66% precision, 67% recall and 66% F1-score. While the Naive Bayes method uses feature selection, namely 72% accuracy, 72% precision, 78% recall and 73% F1-score.

Keywords: *Naive Bayes, Genetic Algorithm, Short Stories for Children, Short Stories for Teenagers and Short Stories for Adults*

1. Pendahuluan

Cerpen adalah cerita pendek yang menceritakan suatu kejadian yang pernah terjadi secara singkat dan jelas [7]. Sastra anak-anak berbeda dengan sastra orang dewasa, karena pada sastra anak-anak fokus pada gambaran kehidupan yang bermakna dan mudah dipahami oleh anak-anak. Orang tua harus bisa memilihkan cerpen yang sesuai untuk anaknya karena apabila cerita yang dibawakan orang tua kepada anak tidak sesuai dengan usianya, maka dapat mempengaruhi perkembangan anak. Objek yang belum memiliki label, dapat ditentukan labelnya dengan cara menemukan model yang bisa membedakan setiap label, proses ini disebut dengan klasifikasi. Dalam menyelesaikan proses klasifikasi dapat memanfaatkan peran teknologi sehingga waktu yang diperlukan akan berkurang dan menjadi lebih efisien [4].

Penelitian mengenai klasifikasi yang sudah dilakukan oleh peneliti sebelumnya seperti Seleksi Fitur Klasifikasi Kategori Cerita Pendek Menggunakan *Naive Bayes* dan Algoritma Genetika yang menghasilkan akurasi 84,29 [9]. Kemudian penelitian mengenai *Text Mining* Untuk Klasifikasi Kategori Cerita Pendek Menggunakan *Naive Bayes* menghasilkan akurasi 78,59% [8]. Lalu penelitian mengenai Klasifikasi Cerita Bahasa Indonesia Menggunakan Metode *Hybrid PSO-KNN (Modified Binary Particle Swarm Optimization* dengan *K-Nearest Neighbor*) menghasilkan akurasi 53% [6]. Selanjutnya penelitian mengenai Klasifikasi Berita Lokal Radar Malang menggunakan Metode *Naive Bayes* dengan Fitur N-Gram menghasilkan akurasi 78,66% [3] dan yang terakhir penelitian mengenai Klasifikasi Teks

Bahasa Bali dengan Metode *Supervised Learning Naïve Bayes Classifier* menghasilkan akurasi 92% [5].

Pada penelitian kali ini akan membangun sebuah sistem yang dapat mengklasifikasikan teks. Metode yang digunakan pada penelitian kali ini adalah *Naïve Bayes* dengan seleksi fitur yaitu Algoritma Genetika. Kategori yang digunakan adalah kategori anak-anak, kategori remaja dan kategori dewasa. Penelitian ini dilakukan untuk membantu orang tua agar anaknya tidak membaca cerpen yang tidak sesuai dengan usianya sehingga tidak mengganggu perkembangan anak.

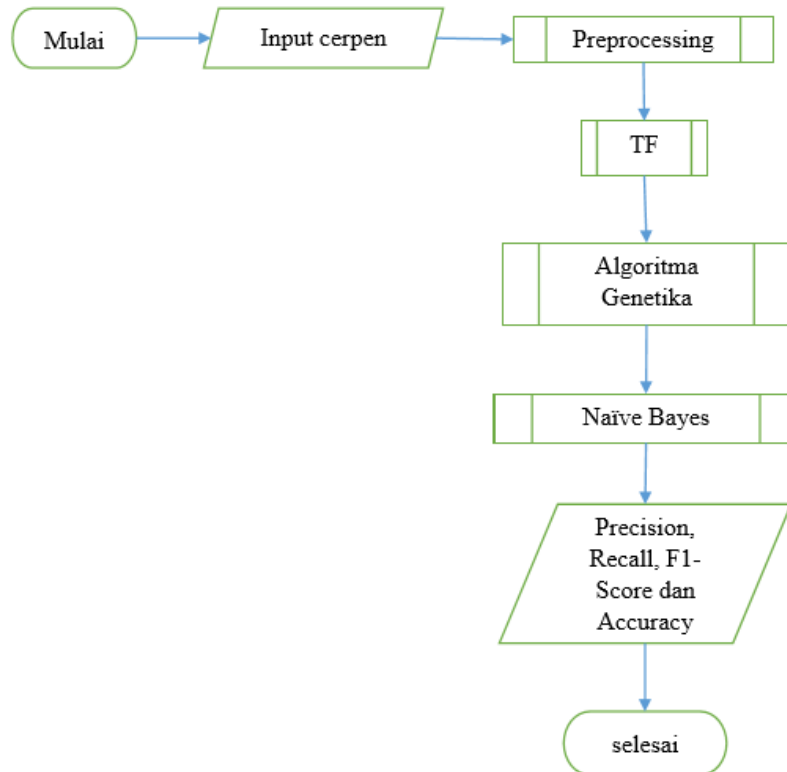
2. Metode Penelitian

2.1 Dataset

Data yang digunakan adalah 90 dokumen cerpen berbahasa Bali. Ada 3 kategori atau kelas yang digunakan yaitu kategori anak-anak, kategori remaja dan kategori dewasa. Data yang digunakan meliputi judul, isi dan label.

2.2 Alur Sistem

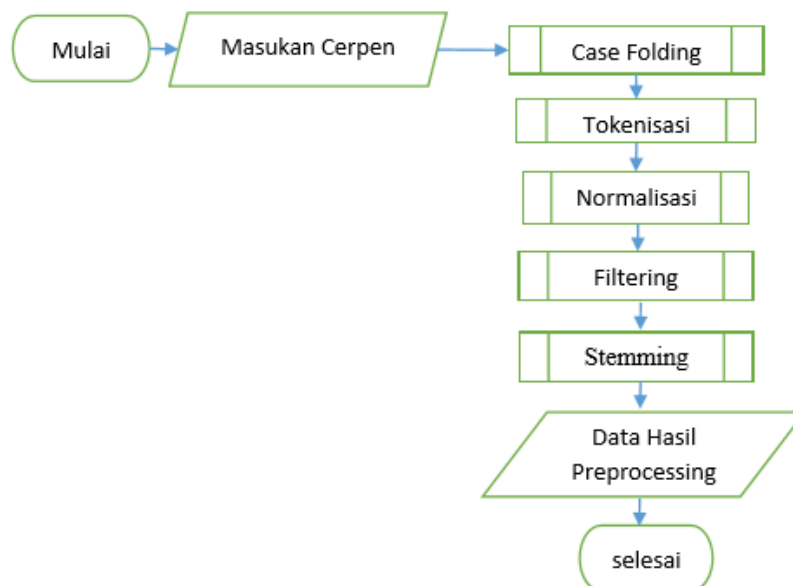
Tahap pertama yang dilakukan adalah pengumpulan data berupa cerpen berbahasa Bali dengan 3 kategori yaitu kategori anak-anak, kategori remaja dan kategori dewasa. Setelah data terkumpul maka tahap selanjutnya data dimasukkan ke dalam database lalu dilakukan proses split data. Karena data yang digunakan jumlahnya belum seimbang maka dilakukan proses *under sampling* agar jumlah data menjadi seimbang. Ketika data sudah seimbang, baru dilakukan proses *preprocessing*. Hasil dari *preprocessing* berupa *dictionary* lalu dibuat index sehingga menjadi *vocabolaries*. Setelah itu, *vocabolaries* melalui tahap pembobotan menggunakan *term frequency*. Selanjutnya, setelah tahap pembobotan dilakukan tahapan pemilihan fitur-fitur terbaik menggunakan Algoritma Genetika. Nilai fitness pada Algoritma Genetika didapatkan dari nilai *F1-score* pada proses *Naïve Bayes*.



Gambar 1. Alur Penelitian

Dalam *Naïve Bayes* tidak ada penentuan *hyper parameter* maka digunakan *K-Fold Cross Validation* untuk proses pengujian dan validasi. Pada proses *Naïve Bayes* menggunakan *K-Fold Cross Validation* menghasilkan *output* berupa *term* probabilitas yang ditetapkan sebagai model terbaik. Selanjutnya dilakukan pengujian model terbaik menggunakan *new data*, *output* dari pengujian model terbaik berupa hasil klasifikasi cerpen. Hasil akhir menghasilkan nilai rata-rata hasil evaluasi klasifikasi cerita pendek berbahasa Bali, yaitu rata-rata nilai *Precision*, *Recall*, *F1-Score* dan Akurasi. Alur penelitian bisa dilihat pada Gambar 1.

2.3 Preprocessing Data



Gambar 2. Alur Preprocessing

Dalam proses ini dokumen melalui tahap *case folding* yaitu merubah semua karakter menjadi huruf kecil. Lalu dilakukan proses tokenisasi yaitu memisahkan dokumen menjadi beberapa token. Setelah itu dilakukan proses normalisasi yaitu mengubah huruf é menjadi e. Selanjutnya dilakukan proses *filtering* yaitu menghilangkan token yang tidak penting. Setelah itu dilakukan proses *stemming* yaitu mengubah semua kata ke dalam bentuk kata dasar [1]. Hasil dari *preprocessing* berupa *dictionary* lalu dilakukan proses *indexing* sehingga menjadi *vocabolaries*. Alur proses *preprocessing* bisa dilihat pada Gambar 2.

2.4 Term Frequency

Setelah data melalui proses *preprocessing*, maka selanjutnya data yang berupa *vocabolaries* akan diberi bobot menggunakan Persamaan :

$$tf_{t,d} = \log f_{t,d} + 1 \quad (1)$$

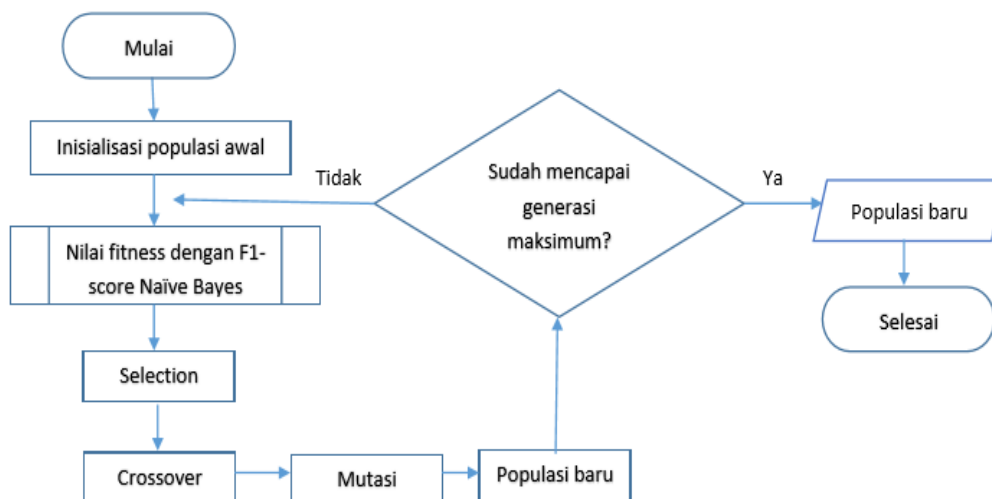
Keterangan:

$tf_{t,d}$ = term frequency

$f_{t,d}$ = jumlah kemunculan *term* t di dalam dokumen d

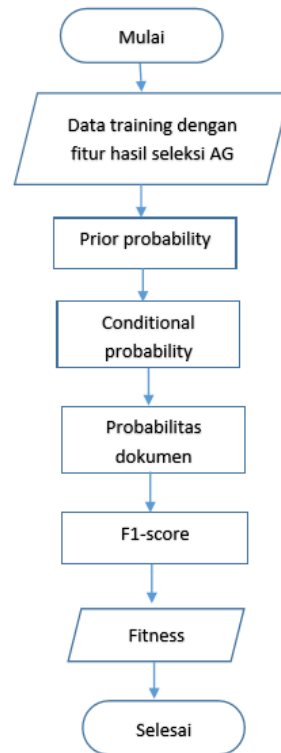
2.5 Algoritma Genetika

Tahap pertama yang dilakukan adalah inialisasi kromosom awal yaitu dibangkitkan bilangan acak yaitu 1 dan 0 sebanyak fitur pada proses *TF*. Setelah itu mencari nilai *fitness* yang didapatkan dari hasil evaluasi *F1-score* pada *Naïve Bayes*. Setelah itu, dilakukan tahapan seleksi. Pada tahapan seleksi ini menggunakan roulette wheel, langkah pertama kita mencari nilai *fitness* total, lalu menentukan peluang relatif setiap kromosom dan menentukan peluang kumulatif setiap kromosom. Setelah itu kita membangkitkan bilangan acak sejumlah kromosom yang ada. Jika bilangan acak lebih besar dari *PK* (*peluang kumulatif*) dan kurang dari *P* (*peluang kromosom*) maka kromosom tersebut terpilih untuk diseleksi. Setelah kromosom diseleksi, maka selanjutnya dilakukan inialisasi parameter *Pc*, dimana nilai awal yang ditetapkan adalah 0.5. Apabila bilangan acak kurang dari probabilitas *crossover* maka dilakukan proses *crossover*. Setelah itu, dilakukan inialisasi parameter *Pc*, dimana nilai awal yang ditetapkan adalah 0.5. Probabilitas mutasi mempengaruhi jumlah gen yang dimutasi. Proses mutasi menghasilkan kromosom baru. Ketika sudah mencapai generasi maksimum proses Algoritma Genetika berhenti dan menghasilkan populasi baru berupa fitur-fitur pilihan yang digunakan pada proses klasifikasi. Alur proses Algoritma Genetika bisa dilihat pada Gambar 3.



Gambar 3. Alur Proses Algoritma Genetika

Dalam menentukan nilai *fitness* dilakukan beberapa tahapan yaitu menginputkan *term unique* yang bernilai 1 sesuai bilangan acak yang dibangkitkan pada Algoritma Genetika. Kemudian menentukan *prior probability*, lalu menentukan *conditional probability*, lalu menentukan probabilitas dokumen. Setelah itu menentukan nilai *fitness* atau *F1-score* kategori anak, menentukan nilai *fitness* atau *F1-score* kategori remaja dan menentukan nilai *fitness* atau *F1-score* kategori dewasa. Setelah itu mencari rata-rata nilai *fitness* atau *F1-score* semua kategori. Sehingga didapatkan satu nilai *fitness* atau *F1-score*. Alur dalam menentukan nilai *fitness* bisa dilihat pada Gambar 4.



Gambar 4. Alur Proses Menentukan Nilai *Fitness* dengan *F1-score Naïve Bayes*

2.6 Naïve Bayes

Langkah pertama kita menginputkan fitur berdasarkan inialisasi bilangan acak yang bernilai 1 pada Algoritma Genetika untuk proses *Naïve Bayes* dengan seleksi *fitur*. Lalu untuk proses *Naïve Bayes* tanpa seleksi *fitur* inputannya berupa *term* hasil proses *TF*. Kemudian menentukan *prior probability* menggunakan Persamaan :

$$p(c) = \frac{n_c}{n} \quad (2)$$

Lalu menentukan *conditional probability* menggunakan Persamaan :

$$p(t_k|c) = \frac{n_k+1}{|v|+n} \quad (3)$$

Dan yang terakhir menentukan probabilitas dokumen menggunakan Persamaan :

$$p(c|d) \propto p(c) \prod_{1 \leq k \leq n_d} p(t_k|c) \quad (4)$$

Hasil dari proses *Naïve Bayes* menggunakan K-Fold Cross Validatin berupa *term* probabilitas atau model terbaik. Lalu hasil dari pengujian model terbaik menggunakan *new data* berupa hasil klasifikasi cerpen.

2.7 Evaluasi

Confusion matrix menampilkan prediksi klasifikasi dan klasifikasi yang aktual. *Precision* yaitu mengukur performa dokumen yang bersifat relevan dan bernilai positif diantara seluruh dokumen yang bersifat relevan. *Recall* yaitu mengukur performa dokumen yang bersifat relevan dan bernilai positif diantara seluruh dokumen yang bernilai benar. *F1-score* yaitu mengukur rata-rata *harmonic* dari *precision* dan

recall. Akurasi yaitu mengukur performa dokumen yang bernilai positif diantara seluruh dokumen yang ada [2]. Tabel *confusion matrix* bisa dilihat pada Tabel 1.

Tabel 1. *Confusion Matrik*

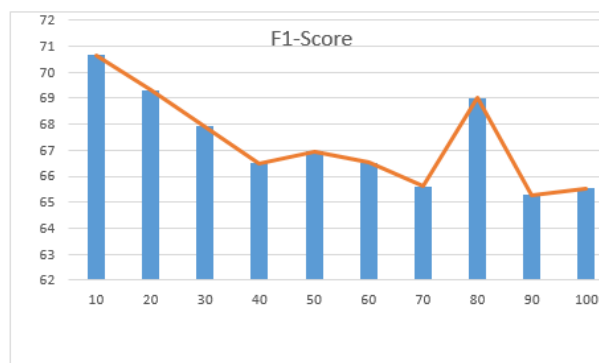
Aktual \ Prediksi	Anak	Remaja	Dewasa
Anak	TA	FAR	FAD
Remaja	FRA	TR	FRD
Dewasa	FDA	FDR	TD

3. Hasil dan Diskusi

Pada pengujian metode *Naïve Bayes* tanpa menggunakan seleksi fitur, untuk mendapatkan rata-rata hasil evaluasi digunakan *K-Fold Cross Validation* untuk proses validasi dan pelatihan, dengan $k = 3$. Didapatkan hasil evaluasi yaitu akurasi 65.278%, *precision* 65.278% *recall* 69.048% dan *F1-Score* 65.021%. Sedangkan pada pengujian metode *Naïve Bayes* menggunakan Algoritma Genetika, akan dilakukan perubahan pada parameter jumlah iterasi, jumlah kromosom, probabilitas *crossover* dan probabilitas mutasi. Dengan menetapkan kombinasi parameter awal yaitu jumlah iterasi 10, jumlah kromosom 2, probabilitas *crossover* 0.5 dan probabilitas mutasi 0.5.

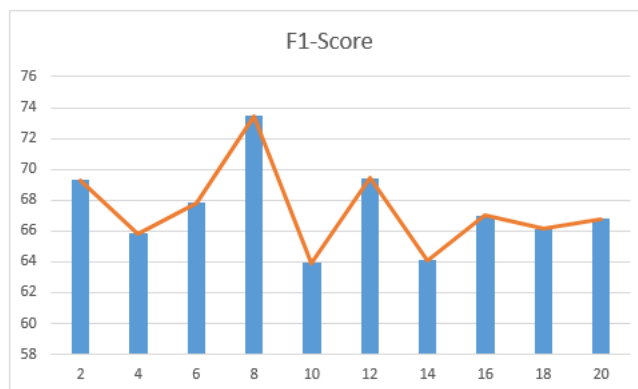
3.1 Pengujian Jumlah Iterasi

Pada pengujian ini akan dilakukan perubahan parameter jumlah iterasi yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Dapat dilihat bahwa hasil pengujian menunjukkan semakin kecil jumlah iterasi maka ukuran evaluasi cenderung meningkat, dan mencapai ukuran evaluasi terbaik pada iterasi ke-20. Dibandingkan iterasi ke-20, ukuran evaluasi iterasi ke-10 memang lebih besar namun performa ukuran evaluasi menggunakan *new data* terjadi *overfitting*. Rata-rata hasil evaluasi pada iterasi ke-20 yaitu akurasi 69.444%, *precision* 69.444% *recall* 75.026% dan *F1-Score* 69.311%. Pengaruh Jumlah Iterasi Terhadap *F1-Score* dapat dilihat pada Gambar 5.

Gambar 5. Pengaruh Jumlah Iterasi Terhadap *F1-Score*

3.2 Pengujian Jumlah Kromosom

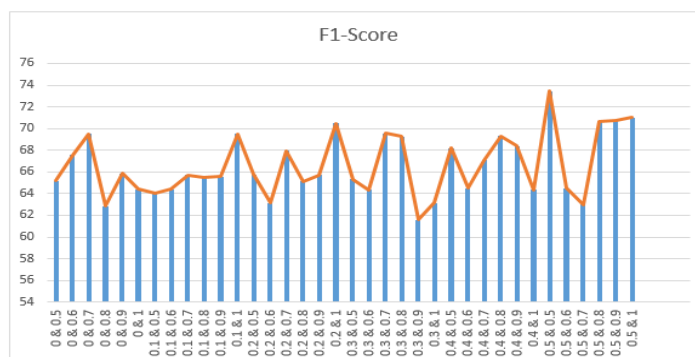
Pada pengujian ini, akan digunakan iterasi ke-20 dengan melakukan perubahan parameter jumlah kromosom yaitu 2, 4, 6, 8, 10, 12, 14, 16, 18, dan 20. Dapat dilihat bahwa hasil pengujian menunjukkan semakin besar jumlah kromosom semakin baik, tetapi setelah jumlah kromosom 8 ukuran evaluasi mengalami penurunan. Rata-rata hasil evaluasi pada jumlah kromosom 8 yaitu akurasi 73.611, *precision* 73.611% *recall* 79.349% dan *F1-Score* 73.465%. Pengaruh Jumlah Kromosom Terhadap *F1-Score* dapat dilihat pada Gambar 6.



Gambar 6. Pengaruh Jumlah Kromosom Terhadap *F1-Score*

3.3 Pengujian Pc dan Pm

Pada pengujian ini akan digunakan iterasi ke-20 dan jumlah kromosom 8 dengan melakukan perubahan parameter Pc 0.5, 0.6, 0.7, 0.8, 0.9, 1 dan Pm yaitu 0.1, 0.2, 0.3, 0.4, 0.5. Dapat dilihat bahwa hasil evaluasi pada pengujian ini meningkat walaupun hasilnya tidak stabil. Rata-rata hasil evaluasi pada Pc 0.5 dan Pm 0.5 yaitu akurasi 73.611, *precision* 73.611% *recall* 79.349% dan *F1-Score* 73.465%. Pengaruh Pm dan Pc Terhadap *F1-score* dapat dilihat pada Gambar 7.



Gambar 7. Pengaruh Pm dan Pc Terhadap *F1-score*

4. Kesimpulan

1. Pada pengujian seleksi fitur Algoritma Genetika, pengaruh perubahan parameter jumlah iterasi yaitu semakin kecil jumlah iterasi maka ukuran evaluasi cenderung meningkat, dan mencapai ukuran evaluasi terbaik pada iterasi ke-20. Dibandingkan iterasi ke-20, ukuran evaluasi iterasi ke-10 memang lebih besar namun performa ukuran evaluasi menggunakan *new data* terjadi overfitting. Pengaruh perubahan parameter jumlah kromosom yaitu semakin besar jumlah kromosom semakin baik, tetapi setelah jumlah kromosom 8 ukuran evaluasi mengalami penurunan. Ukuran evaluasi terbaik diperoleh ketika Pc = 0.5 dan Pm = 0.5. Sehingga menghasilkan kombinasi parameter terbaik yaitu jumlah iterasi 20, jumlah kromosom 8, Pc = 0.5 dan Pm = 0.5. Kombinasi parameter tersebut menyeleksi 3185 fitur dengan nilai fitness tertinggi.
2. Pada penelitian ini dilihat performa model terbaik yang dihasilkan pada proses pelatihan dan validasi. Hasil dari pengujian metode *Naïve Bayes* tanpa seleksi fitur yaitu akurasi 66%, *precision*

Klasifikasi Cerita Pendek Berbahasa Bali Berdasarkan Umur Pembaca dengan Metode *Naïve Bayes*

66%, *recall* 67% dan *F1-score* 66%. Sedangkan pada metode *Naïve Bayes* menggunakan seleksi fitur yaitu akurasi 72%, *precision* 72%, *recall* 78% dan *F1-score* 73%.

Daftar Pustaka

- [1] Abimanyu, C.G., Sanjaya ER, N.A dan Karyawati, A.A.I.N.E. 2020. Balinese Automatic Text Summarization Using Genetic Algorithm. *JITK*. 6(1).p. 13-20.
- [2] Arini, Wardhani, L.K., dan Octaviano, Dimas. 2020. Perbandingan Seleksi Fitur Term Frequency & Tri-Gram Character Menggunakan Algoritma *Naïve Bayes Classifier* (Nbc) Pada Tweet Hastag #2019gantipresiden. *KILAT*. 9(1).p.103-114.
- [3] Chandra, D.N., Indrawan, Gede., dan Sukajaya, I.N. 2019. Klasifikasi Berita Lokal Radar Malang menggunakan Metode *Naive Bayes* dengan Fitur N-Gram. *Jurnal Ilmu Komputer Indonesia (JIKI)*. 4(2). p.10-20.
- [4] Khadijah. 2016. *Pengembangan Kognitif Anak Usia Dini*. Perdana Publishing. Medan.
- [5] Putra, I.B.G.W., Sudarma, Made., dan Kumara, I.N.S. 2016. Klasifikasi Teks Bahasa Bali dengan Metode *Supervised Learning Naïve Bayes Classifier*. *Teknologi Elektro*. 15(2). p.81-86.
- [6] Rahayu, Anita. dan Rochmawati, Naim. 2019. Klasifikasi Cerita Bahasa Indonesia Menggunakan Metode Hybrid PSO-KNN (Modified Binary Particle Swarm Optimization dengan K-Nearest Neighbor). *JINACS*. 1(1). p.64-69.
- [7] Ruswati, S.O. 2020. *Bahasa Indonesia PAKET B Setara SMP/Mts Kelas IX*. Direktorat Pendidikan Masyarakat dan Pendidikan Khusus-Direktorat Jendral Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah-Kementrian Pendidikan dan Kebudayaan. Jakarta.
- [8] Somantri, Oman. 2017. Text Mining untuk Klasifikasi Kategori Cerita Pendek menggunakan *Naïve Bayes* (NB). *Jurnal Telematika*. 12(1). p.7-11 (1).
- [9] Somantri, Oman. dan Khambali, Mohammad. 2017. Seleksi Fitur Klasifikasi Kategori Cerita Pendek Menggunakan *Naïve Bayes* dan Algoritma Genetika. *JNTETI*. 6(3). p.301-306 (2).



ISSN



E-ISSN
