

Klasifikasi Biji Jagung Berdasarkan Tekstur Dan Warna Menggunakan Metode *Backpropagation* Berbasis Web

M. Syafiih^{a1}, Nadiyah^{a2}

^{a1}Teknologi Informasi, ^{a2}Teknik Informatika, Fakultas Teknik Universitas Nurul Jadid
Probolinggo, Indonesia
1m.syafii@unuja.ac.id
2nadiyah@unuja.ac.id

Abstract

Indonesia is an agrarian country because most of its people depend on the agricultural sector. Paiton sub-district, Probolinggo district, East Java is one of the areas where the majority of people work as farmers who still cultivate corn crops. Corn is widely consumed by the surrounding community because it is rich in nutrients, corn can also be used as food and livestock. In previous studies there were less than optimal results in determining the quality of corn. Therefore, the quality of corn quality must be maintained in such a way because corn production is decreasing in productivity every year due to reduced planting land. Gray Level Co-occurrence Matric (GLCM), RGB (Red, Green, Blue) and Backpropagation methods. So that researchers will use these methods to classify the quality of corn kernels. It is hoped that this utilization can solve the problem of middlemen so as not to lose money when buying corn from farmers. The result of this research is the process of determining the quality of corn kernels based on color and texture features using the GLCM, RGB, and Backpropagation methods with a total of 150 images consisting of 120 training data and 30 test data. The classification system test results obtained an accuracy value of 75%. So that the backpropagation method can determine the quality of corn kernels based on images using a computer system so that it can be implemented.

Keywords: Digital Image, RGB, Maize Seed quality classification, Backpropagation algorithm.

1. Pendahuluan

Indonesia merupakan negara yang mempunyai beragam bahan pangan lokal selain beras yang mengandung karbohidrat tinggi. [1] Jagung merupakan salah satu bahan pangan yang mudah ditemukan. Jagung biasa dikonsumsi oleh sebagian masyarakat sebagai makanan pokok. [2] Namun disayangkan, rendahnya tingkat konsumsi jagung dalam keluarga membuat sebagian besar masyarakat lebih memilih untuk mengonsumsi beras sebagai makanan pokok. [3] Belakangan tingkat konsumsi konsumsi beras semakin tinggi akibat jumlah penduduk yang semakin meningkat. [4] Parahnya mereka memiliki pemahaman yang salah bahwa makan jagung dikaitkan dengan kemiskinan. Sebagai dampaknya jagung semakin tidak termasuk dalam menu makanan keluarga. [5] Faktanya jagung memiliki kandungan nutrisi yang diperlukan untuk kesehatan. Kandungan karbohidrat dalam biji jagung sangat banyak, mencapai 80% dari keseluruhan bahan biji kering. Karbohidrat jagung lebih rendah dari beras. Untuk setiap 100 g jagung (dikupas, digiling) mengandung 361 hingga 366 kalori. Jagung memiliki kandungan serat dan karoten yang lebih tinggi dibandingkan beras [6]. Karoten merupakan zat penting dalam pembentukan vitamin A jagung dengan biji berwarna kuning mengandung lebih banyak protein dan vitamin A. Kecamatan Paiton Kabupaten Probolinggo Jawa Timur, sebagai salah satu wilayah yang mayoritas masyarakatnya bertani, bahkan sampai saat ini masih membudidayakan tanaman padi, kedelai, dan jagung. Jagung sendiri banyak dikonsumsi masyarakat sekitar karena memiliki kandungan gizi yang tinggi, jagung juga dapat digunakan sebagai bahan pangan dan ternak. Sehingga kualitas mutu jagung harus dijaga agar tetap baik karena produksi jagung semakin menurun produktivitasnya setiap tahun akibat berkurangnya lahan tanam. Masalah yang sering timbul di daerah sekitar ialah ketika petani menjual hasil panen jagung dengan menggunakan poster atau gambar yang diberikan kepada tengkulak. Ketika tengkulak melakukan pengecekan barang di lokasi panen jagung. Kondisi jagung berbeda dengan poster yang diberikan oleh petani, sehingga tengkulak perlu mengklasifikasikan kualitas biji jagung untuk menghindari ketidakstabilan harga. Tengkulak melakukan klasifikasi kualitas biji jagung untuk menghindari ketidakstabilan harga jagung. Hal ini menyebabkan pihak ketiga berperan dalam menentukan harga jagung. Kementerian Perdagangan telah membuat kebijakan harga untuk menjaga

stabilitas harga jagung bagi produsen dan konsumen. [7] Aplikasi ini dibuat untuk membantu dalam proses seleksi dan pengolahan biji jagung yang memiliki peran penting dalam penentuan kualitas dari jagung. Metode klasifikasi yang digunakan adalah Backpropagation yang merupakan teknik pemrosesan data yang terbukti efektif.

Proses evaluasi kualitas biji jagung untuk klasifikasi kualitas masih dilakukan secara manual melalui pengamatan visual. Pengamatan kualitas dengan cara tersebut memiliki beberapa kelemahan, antara lain membutuhkan waktu yang lama dan menghasilkan produk dengan kualitas yang tidak konsisten. Perbedaan ini disebabkan oleh keterbatasan visual manusia, kelelahan menyebabkan persepsi kualitas oleh setiap pengamat. [8] Klasifikasi ini untuk menentukan kualitas biji jagung berdasarkan tekstur dan warna berdasarkan pola citra digital dengan menggunakan metode *Backpropagation* untuk mendapatkan hasil yang akurat. kualitas biji jagung dengan memanfaatkan perbedaan indeks warna RGB dari berbagai kualitas biji jagung. Tujuan dari penelitian ini adalah untuk menghasilkan suatu sistem perangkat lunak berbasis backpropagation untuk mengklasifikasikan biji jagung ke dalam dua jenis kualitas biji jagung, yaitu biji jagung yang baik dan biji jagung yang tidak baik.

2. Metode Penelitian

Metode penelitian dapat mempermudah proses penelitian untuk memperoleh hasil penelitian dengan tahapan yang sistematis. Beberapa tahapan awal penelitian adalah studi literatur, identifikasi masalah, pengumpulan dataset, preproccesing, ekstraksi fitur, klasifikasi *backpropagation*, [9] implementasi sistem, pengujian, kesimpulan. Selanjutnya digambarkan pada gambar berikut :



Gambar 1. Kerangka Penelitian [10]

2.1. Identifikasi Masalah

Adapun permasalahan yang sering terjadi ialah penjual sering mengatakan jagung ini super. Atau kualitasnya bagus, bijinya tongkolnya berisi, klasifikasi ini sudah banyak dilakukan sebelumnya sehingga akan dilakukan pengembangan dari beberapa permasalahan yang terjadi. Dari hasil identifikasi masalah tersebut menjadi rujukan atau gambaran dalam pengembangan sistem ini, akan dibuat sebuah user interface pengklasifikasian jenis jagung berdasarkan fitur warna dan tekstur dengan menggunakan metode Backpropagation. Hal ini dapat mempermudah dan mempercepat dalam mengetahui kualitas baik atau buruk.

2.2. Studi Literatur

Merupakan bahan untuk mempelajari teori-teori dan pengetahuan yang berhubungan dengan masalah penelitian. Sebagai pembanding terhadap penelitian sebelumnya dan acuan dalam memperbaiki atau melengkapi penelitian yang digunakan sebagai solusi dari permasalahan yang telah diidentifikasi sebelumnya. Studi literatur berasal dari buku, jurnal dan penelitian terdahulu yaitu mengklasifikasikan jenis jagung berdasarkan fitur warna dan tekstur. Hal ini membutuhkan dataset berupa citra jagung yang berjumlah sekitar 75 untuk setiap jenisnya.

2.3. Pengumpulan Dataset

Pada tahap ini tempat yang menjadi objek adalah sawah yang ditanami jagung di daerah Paiton, desa Pandean. Pada saat mengajukan pertanyaan mengenai seputar jagung, jagung dipanen pada usia 99 - 106 hari. Dataset berisi sekumpulan citra biji jagung yang telah diambil dari petani sebanyak 150 data yang terdiri dari dua jenis yaitu 75 citra biji jagung yang jelek dan 75 citra biji jagung yang bagus. Adapun beberapa spesifikasi hardware yang digunakan dalam penelitian ini antara lain handphone vivo y22, *Random Acces Memory* (RAM) 4GB.

2.4. Preprocessing

Pre-processing yaitu bagian dari pengolahan data citra mentah yang telah didapatkan pada tahap pengumpulan data dengan tujuan untuk mendapatkan kualitas citra yang lebih baik untuk digunakan pada tahap selanjutnya. Tahapan yang dilakukan adalah *remove background* yang dilakukan secara manual, kemudian *resize* (pengecilan ukuran resolusi citra). Tahap *pre-processing* berguna untuk dapat meningkatkan kualitas citra yang diperoleh sehingga mudah untuk diolah pada tahap selanjutnya.

2.5. Ekstraksi Fitur

Pada tahap pertama ekstraksi fitur dalam penelitian ini adalah tahap ekstraksi fitur tekstur. Penelitian ini menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*) untuk menganalisa tekstur pada citra grayscale. Fitur yang digunakan pada penelitian ini adalah energi, kontras, homogenitas, dan korelasi. Langkah selanjutnya setelah mengekstraksi fitur tekstur dengan metode GLCM adalah tahap mengekstraksi fitur warna dengan menggunakan metode RGB (*Red, Green, Blue*) di dalam metode RGB dalam mengekstraksi gambar berwarna terdapat tiga komponen, yaitu:



Gambar 2 Ekraksi Fitur [11]

2.6. Klasifikasi *Backpropagation*

Pada proses klasifikasi, data yang telah memiliki label kelas digunakan dalam proses klasifikasi. Proses ini dilakukan untuk membentuk yang digunakan untuk mengklasifikasikan data baru, model yang digunakan untuk data testing adalah *Backpropagation*. Tujuan dari algoritma *Backpropagation* adalah untuk memodifikasi bobot untuk melatih jaringan syaraf agar dapat memetakan input yang berubah-ubah menjadi output dengan benar.

2.7. Implementasi Sisetem

Setelah tahap klasifikasi dengan algoritma *Backpropagation*, langkah selanjutnya adalah mengimplementasikan sistem dalam bentuk aplikasi web dengan menggunakan *web framework flask*. *Flask* merupakan *web framework* yang populer untuk mengembangkan aplikasi web dengan menggunakan bahasa *python*.

2.8. Pengujian

Tahap pengujian merupakan tahap yang sangat penting karena untuk mengetahui seberapa tepat keunggulan algoritma *Backpropagation* dalam memprediksi data baru. Pada pengujian ini menggunakan 20 data yang dibagi menjadi dua kelas, yaitu baik dan buruk Untuk menguji tingkat akurasi pada data testing dapat dilakukan dengan persamaan sebagai berikut:

$$\text{Akurasi} = \frac{\sum \text{Data Benar}}{\sum \text{Data Uji}} \times 100 \quad (16)$$

Keterangan :

\sum Data Benar = Banyak Data Benar

\sum Data Uji = Banyak Data Uji

2.9. Kesimpulan

Hasil klasifikasi berupa informasi tentang kategori biji jagung misalnya kategori atau jenis tertentu dan tingkat akurasi model dalam prediksi. Tahapan penelitian yang dilakukan setelah uji coba selesai dilakukan terhadap sistem kemudian dianalisa mengenai keakuratan metode Backpropagation dalam mengklasifikasikan kualitas biji jagung. akhirnya didapatkan kesimpulan mengenai penelitian tentang klasifikasi kualitas biji jagung dengan pengolahan citra dari metode *Backpropagation*.

3. Hasil dan Pembahasan

3.1. Dataset

Dataset yang digunakan pada penelitian ini adalah citra biji jagung yang dikumpulkan sebanyak 150 biji jagung, dimana citra biji jagung tersebut merupakan salah satu jenis jagung yaitu jenis Bisi, masing-masing kelas memiliki 75 dataset yang baik dan 75 dataset yang buruk. Dari masing-masing kualitas biji jagung diambil 15 citra untuk dijadikan data testing. Jadi totalnya ada 30 sebagai data testing dan 120 sebagai data training. Pengambilan dataset menggunakan kamera handphone dengan ukuran 2.84 MB dengan tempat, kondisi dan waktu yang sama.



Biji Jagung Kualitas Bagus



Biji Jagung Kualitas Buruk

Gambar.3 Gambar Contoh Dataset

Pada gambar diatas merupakan gambar biji jagung yang belum melewati tahap pre-pocessing. Ketika proses pengambilan gambar biji jagung harus jelas, karena mempengaruhi ketika proses.

3.2. Analisis Data

Berdasarkan hasil penelitian yang dilakukan, didapat hasil pada setiap tahapan yang ada, Adapun hasil dari tahapan – tahapan yang mudah dilakukan pada penelitian ini, terdiri dari hasil pengumpulan dataset, pre-ppsessing, ekstrak fitur warna dan tekstur, normalisasi dan klasifikasi, berikut penjelasan yang lebih terperinci.

3.3. Pre`-processing

Selanjutnya data yang diperoleh diolah terlebih dahulu untuk mendapatkan kualitas citra yang baik sehingga memudahkan proses penentuan kualitas biji jagung berdasarkan fitur warna dan tekstur. Pada proses ini dilakukan app paint pada citra jagung dengan kualitas baik dan buruk yang diperoleh dari tahap pengambilan citra, dimana 120 citra biji jagung digunakan untuk data latih dan 30 citra untuk data uji. Tahap pra-pemrosesan dilakukan untuk background dengan warna putih secara manual dan mengubah gambar berwarna menjadi hitam dan putih menjadi abu-abu.

Pengubahan gambar asli menjadi grayscale bertujuan untuk mempermudah proses implementasi pada gambar grayscale. Adapun langkah gambar grayscale sebagai berikut:

a. *Remove background*

Remove Background dilakukan secara manual menggunakan google remove to remove, tujuannya agar lebih mudah saat melakukan preprocessing.

b. Proses Gambar *Grayscale*

Proses *Grayscale* Gambar menjadi hitam, putih dan abu-abu. Setiap gambar akan dikonversi menggunakan skrip python dari gambar berwarna menjadi abu-abu dengan format save *.jpg.

```
rootPath = 'D:\Skripsi\DATASETFIX'
folders = [os.path.join(rootPath, x) for x in (rootPath)]
all_images = [img for folder in folders for img in (folder)]
targetPath = "D:\Skripsi\ori2grayscale"
# D:\Skripsi\thedata\ori2grayscale
for currentPath in os.listdir(rootPath):
    subPath = os.path.join(rootPath, currentPath)
    subTargetPath = os.path.join(targetPath, currentPath)
    if not os.path.isdir(subTargetPath):
        os.mkdir(subTargetPath)
    idx = 0
    for filename in os.listdir(subPath):
        filepath = os.path.join(subPath, filename)
        img = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
        target_path = os.path.join(subTargetPath, "%s-%03d.jpg" % (currentPath, idx+1))
        print(target_path)
        cv2.imwrite(target_path, img)
        idx += 1
```

Gambar 1.4 Segmen Program Grayscale



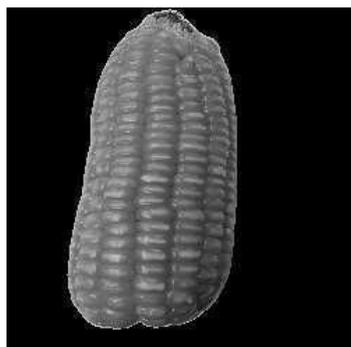
Gambar Asli



Gambar Hasil GrayScale

Gambar .5 Gambar asli dan grayscale

Gambar di atas merupakan dari gambar asli dan warna asli berubah menjadi menjadi warna abu-abu *setelah* dilakukan proses grayscale. Untuk contoh hasil grayscale dari setiap class terdapat pada gambar dibawah ini



Gambar .6 Hasil Grayscale Per Class

Selanjutnya melakukan Resize gambar dari ukuran 980 x 1280 pixel menjadi ukuran 250 x 250 pixel

```
class Glcm:
    def __init__(self):
        self.result = []
    def forOder0(self, a, b):
        size = 0
        for i in a[0]:
            size=size+1
        for i in range(len(a)):
            for j in range(len(a[i])-1):
                p=a[i][j]
                q=a[i][j+1]
                b[p][q]=b[p][q]+1
        matriksIterasi1=list(map(list,zip(*b)))
        for i in range(len(b)):
            for j in range(len(b)):
                b[i][j]=b[i][j]+matriksIterasi1[i][j]

        count=0
        for i in b:
            for j in i:
                count=count+j
        return self.normalisasi(b, count)
```

Gambar 7 Segmen Proses GLCM

menyiapkan kelas data yang akan disimpan ke file csv dari hasil ekstraksi GLCM. Dan fitur-fitur yang digunakan adalah fitur Entropi, Energi, Kontras, Korelasi, dan Homogenitas. Untuk source code lengkap nya dapat dilihat pada halaman lampiran.

```
!python object_detection_demo_flow-master\resize_images.py --raw-dir "D:\Skripsi\ori2grayscale\baik" --save-dir "D:\Skripsi\ori2grayscale2resize\baik"
75 files to resize from directory "D:\Skripsi\ori2grayscale\baik" to target size:(250, 250)
.....
Done resizing 75 files.
Saved to directory: "D:\Skripsi\ori2grayscale2resize\baik"

!python object_detection_demo_flow-master\resize_images.py --raw-dir "D:\Skripsi\ori2grayscale\buruk" --save-dir "D:\Skripsi\ori2grayscale2resize\buruk"
75 files to resize from directory "D:\Skripsi\ori2grayscale\buruk" to target size:(250, 250)
.....
Done resizing 75 files.
Saved to directory: "D:\Skripsi\ori2grayscale2resize\buruk"
```

Gambar .8 Hasil dari proses GLCM

Pada tahapan berikutnya ialah melakukan inialisasi untuk menyimpan fitur GLCM dimana hasil ekstraksi fitur akan otomatis tersimpan pada file csv dengan menjalankan fungsi insertoCsv() dengan nama filenya adalah 'FixGlcmFeatures.csv'.

```
class Data:
    def __init__(self, path):
        imageObject = Image.open(path)
        cropped = imageObject.crop((0,0,250,250))
        self.citra = np.array(cropped)
        self.citra = self.citra.transpose(2,0,1).reshape(3,-1)
        self.matriksIterasi = [[0 for i in range(256)] for j in range(256)]

    def insertoCsv(self,data):
        row = data;
        with open('FixGlcMFeatures.csv', 'a') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        print("succes")
        csvFile.close()
```

Gambar.9 Segmen kelas Data Hasil Output dari GLCM

Merupakan inisialisasi untuk menyimpan fitur GLCM dimana hasil ekstraksi fitur akan otomatis tersimpan pada file csv dengan menjalankan fungsi insertoCsv() dengan nama filenya adalah 'FixGlcMFeatures.csv'. Adapun tabel hasil hasil ekstrak fitur GLCM sebagai berikut

Tabel 1 Hasil Ekstrak Fitur

| Keterangan | Data 1 | Data 2 |
|--------------------|-----------------------|-----------------------|
| Energi 0° | 0.6132619368900873 | 0.6337320455609928 |
| Contras 0° | 0.6765349098000871 | 0.686295850062627 |
| Homogeneity 0° | 253.95759932159004 | 262.3981823709167 |
| Corelation 0° | 0.9588631444179109 | 0.9578719306985576 |
| Energi 45° | 0.9588631444179109 | 0.6337320455609295 |
| Contras 45° | 0.676534909800007 | 0.6862958500625519 |
| Homogeneity 45° | 253.95759932155875 | 262.39818237088673 |
| Corelation 45° | 0.9588631444179154 | 0.9578719306985594 |
| Energi 90° | 0.763077948320751 | 0.7786628834432208 |
| Contras 90° | 0.9999983559197761 | 0.9999984236819742 |
| Homogeneity 90° | 0.0012382125384342675 | 0.0012627954106108322 |

| | | |
|--------------------|--------------------|--------------------|
| Corelation 90° | 0.9999997240740344 | 0.9999997145212128 |
| Energi 135° | 0.6132631323488729 | 0.6337332022384019 |
| Corelation 135° | 0.6765374975283421 | 0.6862983597032368 |
| Homogenity 135° | 253.95556765442902 | 262.3960831787372 |

| | | |
|---------------------|--------------------|--------------------|
| Corelation 135° | 0.9588634084571864 | 0.9578721943478149 |
| Energi Means | 0.6132631323488729 | 0.6337332022384019 |
| Corelation Means | 0.6765374975283421 | 0.6862983597032368 |
| Homegenity Means | 253.95556765442902 | 262.3960831787372, |
| Corelation Means | 0.9588634084571864 | 0.9578721943478149 |
| Class | Baik | Buruk |

Tabel diatas merupakan hasil dari ekstrak fitur Glcm dan hasil pengimputan data hasil ekstrak fitur. Hasil dari proses ekstrak fitur dari kualitas biji jagung.

```

for j in range(cols):
    k = image[i,j]
    if k[0] > k[1] and k[0] > k[2]:
        B_Color = B_Color + 1
        continue
    if k[1] > k[0] and k[1] > k[2]:
        G_Color = G_Color + 1
        continue
    if k[2] > k[0] and k[2] > k[1]:
        R_Color = R_Color + 1
        continue
    N_Color = N_Color + 1

totalImg = rows * cols

R.append(R_Color/totalImg)
G.append(G_Color/totalImg)
B.append(B_Color/totalImg)
print(" ")
print('Citra', filename, 'Blue:', B_Color/totalImg, 'Green:', G_Color / totalImg, 'Red:', R_Color / totalImg)
print(filepath, '\n')
simpanscv.append(filepath)

```

Gambar 10 Segmen Ekstrak fitur RGB

Adapun hasil dari proses tersebut seperti tabel di bawah ini

Tabel 2. Hasil Ekstrak fitur RGB

| | | |
|----------|-----------|----------|
| B | 0.010384, | 0.020016 |
|----------|-----------|----------|

| Class | Baik | Buruk |
|-------|------|-------|
|-------|------|-------|

3.4. Klasifikasi *Backpropagation*

Proses klasifikasi kualitas biji jagung berdasarkan fitur warna dan fitur tekstur menggunakan 120 data latih dan 30 data uji. Bertujuan untuk mentraining data pada data yang berguna untuk menentukan kualitas data yang akan diuji.

a. Split data training dan data testing

Berdasarkan data citra 150 citra biji jagung tersebut dibagi menjadi dua kelas, yaitu training 130 data dan 20 untuk data testing, sehingga dapat dilakukan pengujian performa model pada data tersebut. Pada pengujian ini, 20% data akan digunakan untuk data testing dan 80% untuk data training. Berikut adalah program data testing dan data training

```
x_train, x_val, y_train, y_val = train_test_split(X, y_train_encoded, test_size=0.2, random_state=0)
x_train.shape, x_val.shape, y_train.shape, y_val.shape
((120, 23), (30, 23), (120,), (30,))
```

Gambar 11. Segmen Split Data Training Dan Data Testing

b. Arsitektur Model

Pada proses ini terdapat beberapa lapisan dalam model. Pada baris pertama model disusun secara berurutan. Pada baris kedua, layer disusun dengan menggunakan 32 filter dan menggunakan relu aktivasi dan menentukan bentuk input sebanyak 23, pada baris terakhir, paramater output layer diisi sesuai dengan jumlah label yang tersedia.

```
}; import tensorflow as tf
}; import tensorflow as tf
model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation='relu', input_shape=[23,]),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, 'sigmoid')
])
```

Gambar 12. Segmen Arsitektur Model

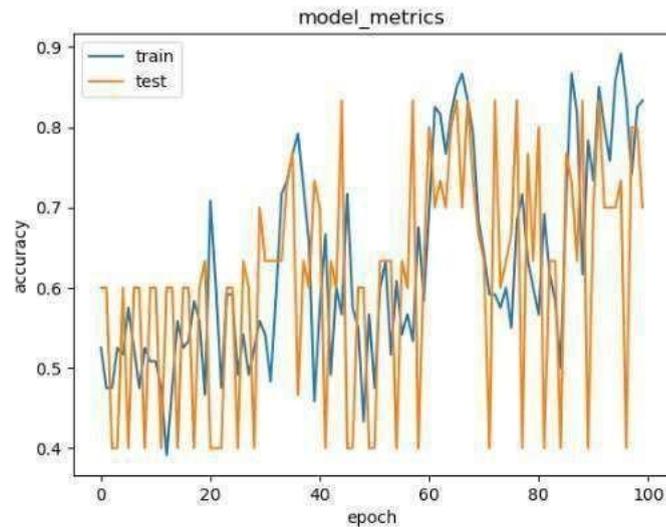
c. Training Algoritma

Pada tahapan pembelajaran model menggunakan data latih yakni `x_train`, `y_train` dengan fungsi `fit()`. Pembelajaran algoritma menggunakan 100 epochs atau iterasi dan validasi data juga diimplementasikan untuk memvalidasi proses pembelajaran terhadap data latih, dan terakhir callbacks yang akan memonitor proses pembelajaran model dimana akan memonitor accuracy dan validation accuracy berikut

```
results = model.fit(
    x_train, y_train,
    epochs=100,
    verbose=1,
    validation_data = (x_val, y_val),
    callbacks=[callbacks])
```

Gambar 13. Segmen Model

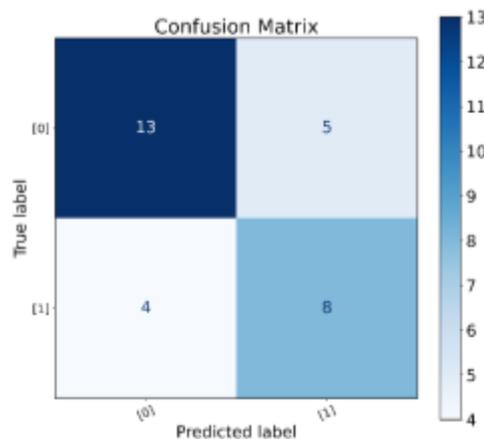
Pada table 1.12. dan table 1.13. merupakan visualisasi hasil proses pembelajaran selama 100 iterasi yang menunjukkan hasil akurasi dan nilai error (*loss*) pada model, dimana hasil proses pembelajaran model mendapatkan akurasi sebesar 89%



Gambar 14 Grafik Pembelajaran model

d. Hasil Prediksi

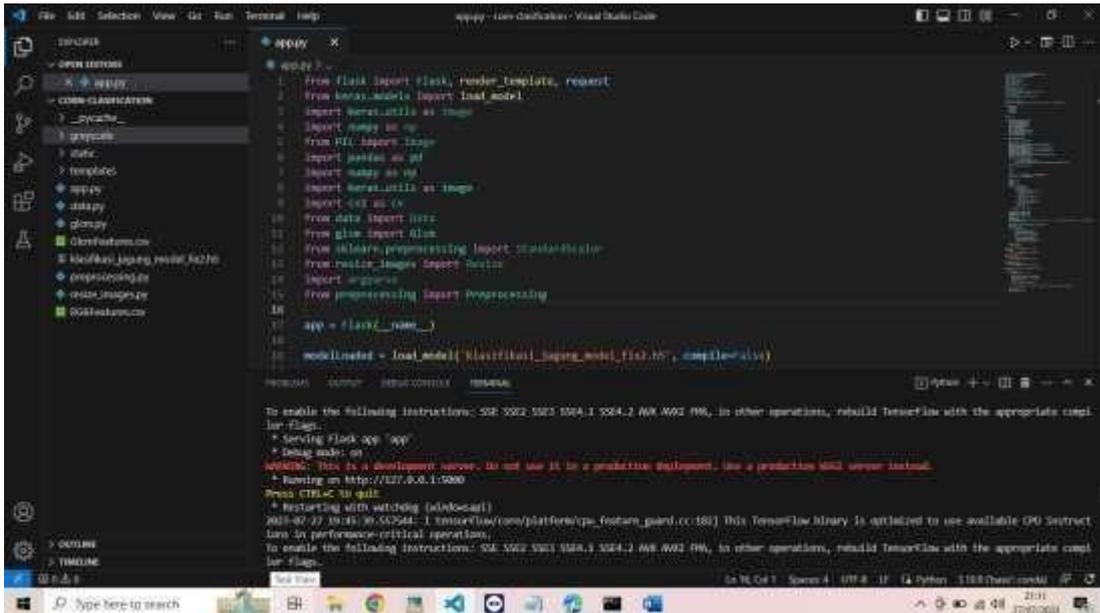
Tahapan hasil prediksi menjelaskan *confusion matrix* menampilkan hasil prediksi terhadap data validasi ada 13 data yang benar atau yang terdeteksi label 0, ada 5 data yang salah prediksi label 1, ada 4 data yang salah prediksi yang terdeteksi 0, ada 8 data yang benar atau terdeteksi label 1.



Gambar 15. Gambar Hasil Prediksi

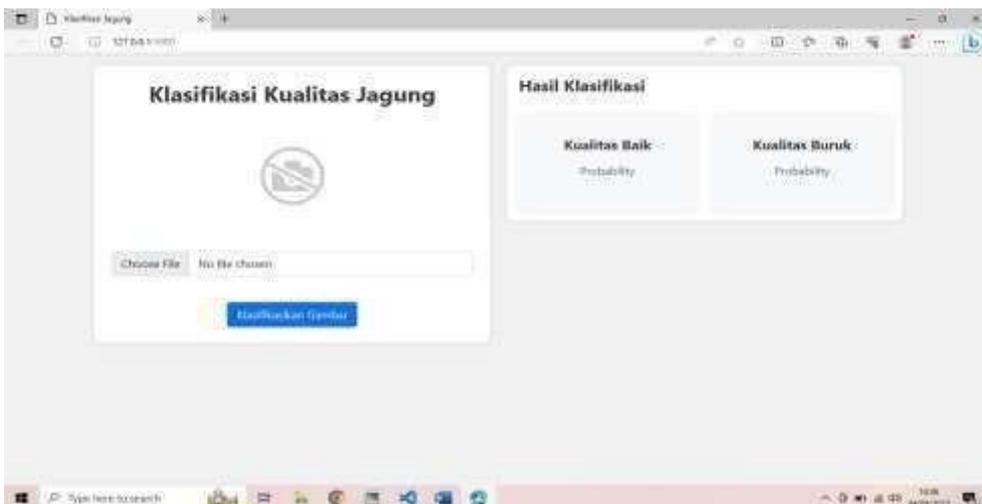
3.5. Implementasi sistem

Implementasi yang digunakan dalam penelitian ini menggunakan *website*. Aplikasi yang dibuat berfungsi untuk mengklasifikasikan gambar dengan menggunakan model yang telah dilatih dan disimpan sebelumnya. *Framework* yang digunakan adalah *Flask*. Pada implementasi ini menggunakan *VsCode (Visual Studio Code)* pada gambar 19.287 melakukan *proses run* dan cara menjalankan aplikasi *flask* dengan cara klik Terminal kemudian klik *Run python* file.



Gambar 16. Segmen Proses Run

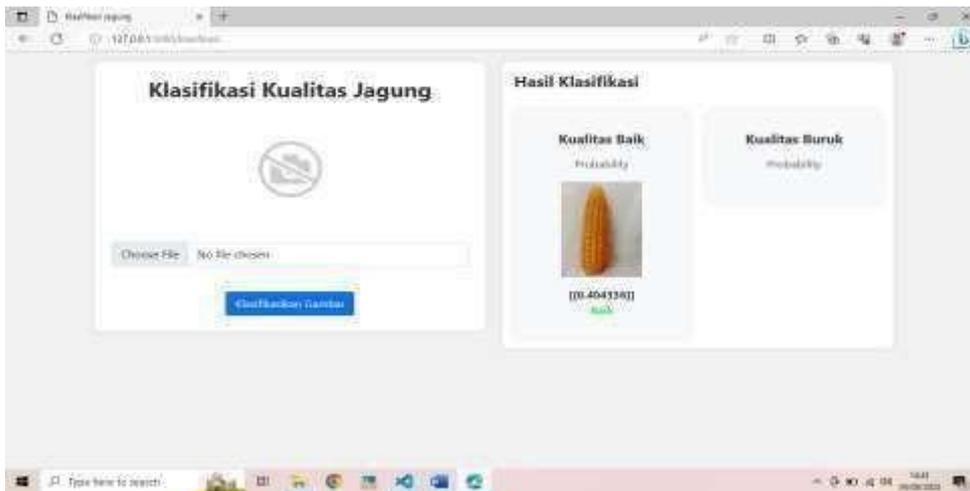
Setelah dijalankan akan mendapatkan server internal <http://127.0.0.1:5000/> klasifikasi dengan mengklik server kemudian akan muncul tampilan desain website Klasifikasi kualitas biji jagung seperti pada gambar 28 berikut :



Gambar 17. Desain Tampilan Website

Pada tampilan website, terdapat dua tombol, yaitu "pilih file" yang digunakan untuk mengambil gambar dan "klasifikasi gambar" yang digunakan untuk mengklasifikasikan gambar yang akan diklasifikasikan.

Langkah pertama klik choose file kemudian akan di arahkan ke directory untuk memilih gambar yang akan di klasifikasi. Langkah kedua yaitu klik "klasifikasi gambar" yang akan diklasifikasi kemudian akan muncul hasil dari klasifikasi jagung akan ditampilkan pada tabel disamping. Berikut adalah hasil dari klasifikasi kualitas biji jagung.



Gambar 18. Gambar Hasil Klasifikasi

3.6. Uji Coba

Pada tahapan uji coba dengan algoritma backpropagation yang sudah di implementasikan ke dalam website dengan menggunakan data baru sebanyak 20 data citra jagung terdiri dari 10 kualitas baik dan 10 kualitas buruk. Berikut adalah hasil uji coba website ditunjukkan pada tabel di bawah ini :

Tabel 3 Hasil Uji Coba

| Nama Citra | Kelas Awal | Prediksi | Hasil |
|------------|------------|----------|-------|
| br1 | Buruk | Buruk | Salah |
| br2 | Buruk | Buruk | Benar |
| br3 | Buruk | Buruk | Salah |
| br4 | Buruk | Buruk | Salah |
| br5 | Buruk | Buruk | Benar |
| br6 | Buruk | Buruk | Benar |
| br7 | Buruk | Buruk | Salah |

| | | | |
|------|-------|-------|-------|
| br8 | Buruk | Buruk | Benar |
| br9 | Buruk | Buruk | Benar |
| br10 | Buruk | Buruk | Salah |
| kb1 | Baik | Baik | Benar |
| kb2 | Baik | Baik | Benar |
| kb3 | Baik | Baik | Salah |
| kb4 | Baik | Baik | Benar |
| kb5 | Baik | Baik | Benar |
| kb6 | Baik | Baik | Benar |
| kb7 | Baik | Baik | Benar |
| kb8 | Baik | Baik | Salah |
| kb9 | Baik | Baik | Benar |
| kb10 | Baik | Baik | Benar |

Selanjutnya pada tabel diatas akan dilihat seberapa besar akurasinya dengan rumus berikut :

$$Akurasi = \frac{\sum Data Benar}{\sum Data Uji} \times 100$$

Pada tabel diatas terdapat 20 data dengan 5 yang data salah sehingga :

$$Akurasi = \frac{15}{20} \times 100 = 75\%$$

Jadi pada pengujian klasifikasi kualitas biji jagung dengan metode backpropagation neural network berbasis web didapatkan akurasi sebesar 75%. Karena model Machine Learning Overfitting terlalu mempelajari data terlalu detail, sehingga yang ditangkap bukan hanya datanya saja, namun noise yang ada juga direkam. Model overfitting akan menghasilkan nilai akurasi yang sangat tinggi saat proses training, namun memiliki akurasi rendah Ketika tesing.

4. Kesimpulan

Hasil penelitian menunjukkan bahwa aplikasi "Klasifikasi Benih Jagung Berdasarkan Tekstur dan Warna Menggunakan Metode Backpropagation Berbasis Web" berhasil mengklasifikasikan benih jagung dengan akurasi yang tinggi. Metode *Backpropagation* terbukti efektif dalam melakukan machine learning untuk mengenali pola tekstur dan warna, sehingga memberikan hasil klasifikasi yang handal. Interface web yang mudah digunakan memudahkan pengguna untuk mengunggah gambar biji jagung dan mendapatkan hasil klasifikasi dengan cepat. Keamanan data dan praktik privasi juga terjamin, sementara pemeliharaan rutin dan pembaruan model memastikan aplikasi tetap dioptimalkan dan responsif terhadap perubahan dataset atau kebutuhan pengguna..Hasil dari penelitian klasifikasi kualitas biji jagung berdasarkan fitur warna dan tekstur dengan metode *Backpropagation* dapat disimpulkan bahwa proses menentukan kualitas biji jagung berdasarkan fitur warna dan tekstur dengan total 150 data dengan ukuran format gambar minimal 250 x 250 pixel.. Maka hasil pengujian sistem klasifikasi diperoleh nilai akurasinya 75% dengan menggunakan metode backpropagation dapat memprediksi kualitas biji jagung sehingga dapat diimplementasikan ke sistem *Web Flask*.

Daftar Pustaka

- [1] A. Wanto, "Prediksi Produktivitas Jagung Di Indonesia Sebagai Upaya Antisipasi Impor Menggunakan Jaringan Saraf Tiruan Backpropagation," *SINTECH (Science and Information Technology) Journal*, vol. 2, no. 1, pp. 53-62., 2019.
- [2] R. Aldillah, "Strategi pengembangan agribisnis jagung di Indonesia.," *Analisis Kebijakan Pertanian*, vol. 15, no. 1, pp. 43-66., 2017.
- [3] A. Wanto, D. Hartama, G. W. Bhawika, Z. Chikmawati, D. S. Hutauruk, P. H. Siregar and A. P. Windarto, "Model of Artificial Neural Networks in Predictions of Corn Productivity in an Effort to Overcome Imports in Indonesia," *In Journal of Physics: Conference Series*, vol. 1339, no. 1, p. 012057, 2019.
- [4] J. M. Maligan, D. D. Pratiwi and T. D. Widyaningsih, "Studi Preferensi Konsumen terhadap Nasi Putih dan Nasi Jagung Putih pada Pekerja Wanita di Kantor Pemerintah Kota Malang.," *Indonesian Journal of Human Nutrition*, vol. 6, no. 1, pp. 41-52, 2019.
- [5] D. Febrina, N. Khairunnisa and R. Febriyanti, "Pengaruh lama pemeraman dan metode pengolahan terhadap kualitas fisik dan kandungan nutrisi jerami jagung," *Jurnal Agripet*, vol. 20, no. 2, pp. 15-23, 2020.
- [6] A. R. Lapui, U. Nopriani and H. Mongi, "Analisis Kandungan Nutrisi Tepung Jagung (*Zea mays* Lam) dari Desa Uedele Kecamatan Tojo Kabupaten Tojo Una-Una untuk Pakan Ternak.," *Agropet*, vol. 18, no. 2, pp. 42-46, 2021.
- [7] M. I. Al-Qarazi, S. Sukardi and A. Anwar, "Analisis Peramalan Produksi, Konsumsi dan Harga Jagung Di Provinsi Nusa Tenggara Barat.," *Jurnal Agrimansion* , vol. 22, no. 1, pp. 49-60, 2021.
- [8] F. Sigaha, S. E. J. and S. Zainudin, "Evaluasi persentase karkas ayam kampung super dengan pemberian jermai jagung fermentasi.," *Jambura Journal of Animal Science*, vol. 2, no. 1, pp. 1-7, 2019.
- [9] G. Guntoro, L. Costaner and L. Lisnawita, "Prediksi Jumlah Kendaraan di Provinsi Riau Menggunakan Metode Backpropagation," *nformatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, vol. 14, no. 1, pp. 51-57., 2019.
- [10] A. Wanto, "Prediksi Produktivitas Jagung Di Indonesia Sebagai Upaya Antisipasi Impor Menggunakan Jaringan Saraf Tiruan Backpropagation," *SINTECH (Science and Information Technology) Journal*, vol. 2, no. 1, pp. 53-62, 2019.
- [11] A. Ciputra, E. H. Rachmawanto and A. Susanto, "Klasifikasi Tingkat Kematangan Buah Apel Manalagi Dengan Algoritma Naive Bayes Dan Ekstraksi Fitur Citra Digital," *Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, vol. 9, no. 1, pp. 465-472., 2018.