



Perbandingan Algoritma Logistic Regression dan K-Nearest Neighbor Dalam Klasifikasi Kematangan Buah Pepaya

Wildan Amin Wiharja*, Tohirin Al Mudzakir, Hilda Yulia Novita, Jamaludin Indra

Fakultas Ilmu Komputer, Teknik Informatika, Universitas Buana Perjuangan Karawang, Karawang, Indonesia
Email: ¹if21.wildanwiharja@mhs.ubpkarawang.ac.id, ²tohirin@ubpkarawang.ac.id, ³hilda.yulia@ubpkarawang.ac.id,
⁴jamaludin.indra@ubpkarawang.ac.id

Email Penulis Korespondensi: if21.wildanwiharja@mhs.ubpkarawang.ac.id

Abstrak—Penentuan kematangan buah pepaya secara visual sering kali tidak konsisten dan kurang akurat. Untuk mengatasi hal ini, penelitian menggunakan algoritma Logistic Regression dan K-Nearest Neighbor (K-NN) dalam klasifikasi otomatis berbasis pemrosesan citra digital. Dataset awal berisi 300 gambar yang kemudian diperbesar melalui preprocessing dan augmentasi menjadi 1.200 gambar. Fitur diekstraksi menggunakan Gray Level Co-occurrence Matrix (GLCM), kemudian data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian. Tujuan penelitian adalah membandingkan performa kedua algoritma dan memahami cara klasifikasi masing-masing. Hasilnya, K-NN dengan $k=1$ mencapai akurasi 87%, sedangkan Logistic Regression dengan regulasi L2 memperoleh 73%, menunjukkan K-NN lebih unggul dalam mengklasifikasikan tingkat kematangan pepaya.

Kata Kunci: Klasifikasi Kematangan Buah, GLCM, Logistic Regression, K-Nearest Neighbor

Abstract—Visual assessment of papaya ripeness often leads to inconsistent and low accuracy results. To address this, the study applies Logistic Regression and K-Nearest Neighbor (K-NN) algorithms for automatic classification using digital image processing. The initial dataset consisted of 300 images, which were expanded to 1,200 through preprocessing and augmentation. Features were extracted using the Gray Level Co-occurrence Matrix (GLCM) method, and the data was split into 80% for training and 20% for testing. The study aims to compare the performance of both algorithms and understand their classification mechanisms. Results show that K-NN with $k=1$ achieved an accuracy of 87%, while Logistic Regression with L2 regularization reached 73%, indicating that K-NN outperforms Logistic Regression in classifying papaya ripeness levels.

Keywords: Fruit Ripeness Classification; GLCM; Logistic Regression; K-Nearest Neighbor

1. PENDAHULUAN

Sebagai buah tropis yang sangat disukai di Indonesia, Pepaya yang tersedia sepanjang tahun memiliki kandungan gizi yang beragam, seperti vitamin A, vitamin C, serat, kalsium, dan magnesium, yang berperan penting dalam menjaga kesehatan. Berkat kandungan gizinya yang melimpah, pepaya disukai oleh berbagai lapisan masyarakat. Namun, pepaya kerap dipanen dan dikirim sebelum mencapai tingkat kematangan sempurna, sehingga saat tiba di tangan konsumen, tingkat kematangannya bisa berbeda-beda[1].

Secara visual, tingkat kematangan pepaya memang dapat dilihat, namun perbedaan intensitas warnanya seringkali sulit dibedakan. Biasanya, pepaya mentah berwarna hijau pekat, lalu berubah menjadi hijau kekuningan saat setengah matang, dan akan berwarna oranye saat benar-benar matang[2]. Warna memiliki peran penting dalam membantu manusia mengenali dan mengidentifikasi objek secara jelas, karena mata manusia dapat membedakan berbagai macam warna. Dalam menilai tingkat kematangan buah seperti pepaya, warna kulit biasanya dijadikan indikator utama. Namun, ketepatan manusia dalam membedakan warna tidak selalu sempurna. Beberapa faktor dapat memengaruhi hal ini, seperti penurunan kemampuan penglihatan seiring bertambahnya usia, kebutuhan konsentrasi lebih saat membandingkan warna, serta perbedaan persepsi warna antar individu. Oleh sebab itu, diperlukan pendekatan yang lebih akurat dan objektif dalam mengklasifikasikan tingkat kematangan buah pepaya, karena perbedaan warnanya kerap sulit dikenali dengan pengamatan langsung.

Pengolahan citra merupakan proses manipulasi data dua dimensi yang digunakan dalam berbagai kebutuhan, seperti peningkatan kualitas gambar, pengenalan pola, serta kompresi atau penyimpanan citra untuk keperluan transmisi. Salah satu tahapan krusial dalam proses ini adalah ekstraksi fitur, yaitu proses mengambil karakteristik tertentu dari bentuk atau tekstur citra[3]. Melalui *Gray Level Co-occurrence Matrix (GLCM)*, fitur tekstur gambar dapat diekstraksi, sehingga memungkinkan analisis lebih detail terhadap karakteristik visual citra.[4]

Penelitian terdahulu telah membahas klasifikasi tingkat kematangan buah menggunakan berbagai pendekatan. Salah satunya dilakukan oleh Suryanti dan Rohman yang meneliti proses klasifikasi kualitas apel dengan memanfaatkan warna dan bentuk sebagai parameter, menggunakan metode *K-Nearest Neighbor (KNN)* dan teknik ekstraksi fitur *GLCM*. Penelitian tersebut menggunakan 117 citra sebagai dataset dan berhasil mencapai tingkat akurasi sebesar 83,37%. [5]. Raysyah, Arinal, dan Mulyana melakukan penelitian mengenai klasifikasi tingkat kematangan buah kopi dengan memanfaatkan deteksi warna menggunakan metode *K-Nearest Neighbor (KNN)* dan *Principal Component Analysis (PCA)*. Dengan nilai $K=3$ dan jumlah dataset sebanyak 135 citra, penelitian ini berhasil mencapai akurasi sebesar 97,77%. [6]. Zulkifli dan Fajri dalam penelitiannya membahas tentang klasifikasi kematangan buah stroberi menggunakan metode *Logistic Regression*. Penelitian ini menggunakan 521 data citra stroberi yang diklasifikasikan ke dalam dua kategori kematangan, yaitu matang dan mentah, dan berhasil memperoleh akurasi sebesar 97,7%. [7]. Heru dan Eko dalam penelitiannya mengkaji penggunaan ekstraksi fitur warna *GLCM* yang diterapkan pada algoritma *K-Nearest Neighbor (KNN)* untuk klasifikasi tingkat kematangan buah rambutan. Data latih yang digunakan sebanyak 180 gambar, dengan 45 di antaranya merupakan gambar buah busuk, 45 citra setengah matang, 45 citra mentah, dan hanya 5 citra matang.

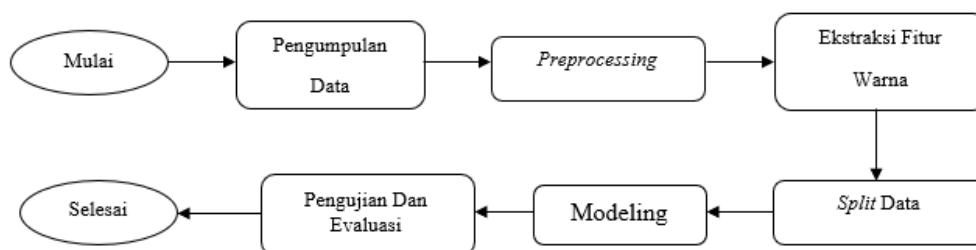


Sementara itu, data uji berjumlah 80 citra, masing-masing terdiri dari 20 citra untuk setiap kategori: busuk, matang, setengah matang, dan mentah. Dari hasil penelitian, didapatkan bahwa akurasi terbaik sebesar 98,75% tercapai saat $K=1$, sedangkan akurasi terendah sebesar 92,5% ditemukan pada $K=7$ dan $K=9$ [8]. Dalam penelitian yang dilakukan oleh Krisna, Wahyu, dan Veronica, klasifikasi kematangan buah kersen dilakukan menggunakan citra berbasis model warna *HSI* dan algoritma *K-Nearest Neighbor (KNN)*, dengan dataset sebanyak 27 gambar yang menghasilkan akurasi 89%[9].

Penelitian ini bertujuan untuk mengklasifikasikan tingkat kematangan buah pepaya menggunakan pengolahan citra digital, serta membandingkan kinerja algoritma *Logistic Regression* dan *K-Nearest Neighbor* dalam mengelompokkan citra ke dalam kategori mentah, mengkal, dan matang.

2. METODOLOGI PENELITIAN

Penelitian ini, objek yang digunakan merupakan citra buah pepaya dengan tiga klasifikasi tingkat kematangan buah dibagi menjadi tiga, yaitu mentah, setengah matang, serta matang. Pepaya mentah memiliki warna hijau pekat, sedangkan setengah matang memperlihatkan perubahan warna hijau menjadi lebih pucat dengan nuansa kuning. Sementara itu, pepaya matang berwarna oranye. Langkah-langkah prosedur penelitian ditampilkan pada Gambar 1.



Gambar 1. Prosedur penelitian

Pada penelitian ini, proses pengumpulan data dilakukan dengan cara mencari dan memilih dataset yang relevan melalui *Kaggle*, sebuah platform yang banyak digunakan dalam bidang *Data Science* dan *Machine Learning* karena menyediakan lebih dari 6000 dataset dalam format *CSV* dan gambar[10]. Dataset yang digunakan terdiri dari 300 gambar buah pepaya, dengan rincian 100 gambar untuk masing-masing tingkat kematangan mentah, setengah matang, dan matang. Selanjutnya, dilakukan kajian literatur melalui jurnal dan buku yang relevan mengenai klasifikasi tingkat kematangan buah menggunakan algoritma *Logistic Regression* dan *K-Nearest Neighbor*.

Dataset awal terdiri dari 300 citra buah pepaya. Setelah melalui tahapan *preprocessing* dan augmentasi untuk meningkatkan kualitas dan keragaman data, dilanjutkan dengan ekstraksi fitur tekstur menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*. Hasil akhir dari proses ini menghasilkan 1.200 citra yang siap digunakan untuk analisis dan pemodelan. Tahapan *preprocessing* ini mencakup beberapa langkah, salah satunya adalah menghapus *background* pada gambar jika ada elemen latar belakang yang mengganggu, sehingga analisis dapat fokus pada objek utama.[11]. Proses *preprocessing* mencakup beberapa langkah, yaitu :

a. *Resize*

Resize gambar adalah tahapan mengubah dimensi gambar dengan menyesuaikan jumlah piksel yang menyusunnya, baik dengan memperbesar maupun memperkecil ukuran gambar[12].

Seluruh citra dalam dataset diubah ukurannya menjadi 240×240 piksel pada tahap *resize*.

b. Rotasi

Rotasi gambar merupakan proses transformasi yang mengubah posisi gambar dengan memturnya ke arah searah atau berlawanan jarum jam[13].

Pada tahap rotasi, seluruh citra diputar sebesar 30 derajat sebagai teknik augmentasi guna memperkaya variasi data latih dan mendukung generalisasi model.

c. *Flipping*

Flipping adalah proses memutar dan membalik gambar dalam dataset. Proses ini dapat dilakukan dengan pembalikan secara *horizontal (horizontal flip)* maupun *vertikal (vertical flip)*[14].

Pada tahap *flipping*, citra dibalik secara *horizontal* dan *vertikal* sesuai sudut 185° dan 95° , sebagai teknik augmentasi untuk menambah variasi data.

d. *Zooming*

Zooming merupakan teknik augmentasi yang dilakukan dengan memperbesar gambar sesuai dengan nilai yang telah ditentukan[15].

Tahap selanjutnya adalah *zooming*, yaitu teknik augmentasi citra dengan memperbesar gambar sebesar 1.5 kali dari ukuran aslinya untuk menambah variasi data.

e. *Cropping*

Cropping image merupakan proses mengambil bagian tertentu dari gambar untuk mempermudah analisis serta mengurangi ukuran file gambar[16].



Pada tahap cropping, citra dipotong berdasarkan koordinat (50, 50) dengan ukuran 150×150 piksel sebagai bagian dari teknik augmentasi untuk menambah variasi data.

f. *Brightness/Contrast adjustment*

Penyesuaian kecerahan dan kontras dilakukan untuk membuat model lebih tahan terhadap variasi pencahayaan [17]. Pada tahap *brightness adjustment*, citra diperjelas dengan menambah kecerahan sebesar 2.0 dan mengalikan kontras dengan faktor 2.0 sebagai teknik augmentasi.

g. Segmentasi *Otsu*

Metode *thresholding* otomatis yang digunakan untuk memisahkan objek dari latar belakang dalam citra *grayscale* [18]. Pada segmentasi *Otsu*, citra *grayscale* disesuaikan dengan *brightness* dan *contrast* sebesar 2.0, lalu diterapkan *Gaussian Blur* 5×5 untuk mengurangi noise sebelum segmentasi.

Tahapan *preprocessing* dan augmentasi pada penelitian ini dilakukan menggunakan *library Pillow (PIL)*, yang mencakup proses *resize*, *rotasi*, *flipping*, *zooming*, *cropping*, serta penyesuaian *brightness* dan *contrast*. Setelah itu, dilakukan segmentasi citra menggunakan metode *Otsu* untuk memisahkan objek buah dari latar belakang sebelum tahap analisis lebih lanjut.

Langkah selanjutnya setelah *preprocessing* adalah melakukan ekstraksi fitur menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*, yang merupakan sebuah matriks untuk memetakan frekuensi kemunculan pasangan piksel dengan intensitas tertentu dalam citra [19]. Fungsi *GLCM* adalah menghitung jumlah kemunculan pasangan piksel dengan jarak dan arah tertentu yang kemudian menjadi acuan dalam ekstraksi fitur *GLCM*. Setiap nilai dalam matriks ini menunjukkan frekuensi pasangan piksel yang memiliki intensitas spesifik dan berjarak 1 piksel, dengan arah 0° , 45° , 90° , serta 135° [20].

Dataset yang sudah diproses melalui tahapan *preprocessing* dan ekstraksi fitur dengan *GLCM* dibagi menjadi dua bagian, yakni 80% untuk proses *training* model dan 20% untuk proses *testing*. Model digunakan untuk mempelajari pola kematangan buah pepaya, sedangkan data pengujian dipakai untuk mengukur kemampuan klasifikasi terhadap data baru. Pembagian ini memastikan model tidak hanya akurat, tetapi juga mampu melakukan generalisasi. Selanjutnya, dilakukan pembangunan dan perbandingan dua model klasifikasi.

Sebelum melanjutkan ke tahap pengujian menggunakan kedua algoritma, terlebih dahulu dilakukan pemodelan dengan algoritma *Logistic Regression* yang menggunakan regulasi L2 untuk mencegah *overfitting*, serta *K-Nearest Neighbor (KNN)* dengan nilai k terbaik yaitu $k=1$, yang diperoleh melalui pencarian nilai k optimal berdasarkan akurasi pada data validasi.

a. *Logistic Regression*

Algoritma *Logistic Regression* dipakai untuk menganalisis keterkaitan antara variabel bebas dan variabel terikat yang bersifat biner. Variabel dependen dalam model ini umumnya hanya memiliki dua kemungkinan nilai, seperti 0 dan 1, yang merepresentasikan dua kondisi berbeda, misalnya keberhasilan dan kegagalan [21].

b. *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor (K-NN)* mengklasifikasikan data baru dengan melihat kemiripan atau jarak terdekat terhadap K data dalam dataset pelatihan, di mana jarak terpendek digunakan sebagai dasar penentuan kelas [22].

Model yang telah dibuat akan dievaluasi menggunakan *confusion matrix*. *Confusion matrix* adalah tabel berukuran $m \times m$ yang digunakan untuk menilai kinerja model klasifikasi dengan membandingkan antara label prediksi dan label aktual. Sebuah model yang baik akan menunjukkan nilai tinggi pada diagonal utama (*True Positives* dan *True Negatives*), sementara nilai di luar diagonal sebaiknya rendah atau mendekati nol. Berikut adalah *representasi confusion matrix* untuk kasus 2×2 .

Dalam evaluasi model klasifikasi, N adalah jumlah data dengan label negatif, P' adalah total prediksi berlabel positif, dan N' adalah total prediksi berlabel negatif. *True Positives (TP)* menunjukkan data yang benar diprediksi sebagai positif, sedangkan *True Negatives (TN)* adalah data yang benar diprediksi sebagai negatif. *False Positives (FP)* terjadi saat data negatif diprediksi sebagai positif, dan *False Negatives (FN)* terjadi saat data positif diprediksi sebagai negatif. Istilah-istilah ini digunakan untuk menghitung metrik evaluasi model.

Dari *confusion matrix*, terdapat tiga metrik evaluasi utama yang sering digunakan, yaitu:

a. *Accuracy*: Mengukur seberapa baik model secara keseluruhan dalam membuat prediksi yang benar.

$$\text{Akurasi} = \frac{TP + TN}{TP + FN + FP + T} \times 100\% \quad (1)$$

b. *Precision*: Mengukur tingkat keakuratan model dalam membuat prediksi positif.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

c. *Recall*: Mengukur tingkat keberhasilan model dalam mendeteksi semua data yang seharusnya positif

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

d. *F1-Score*: Menilai keseimbangan antara *precision* dan *recall*, sangat berguna dalam menangani masalah ketidakseimbangan kelas pada dataset.



$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Evaluasi kinerja klasifikasi dilakukan melalui confusion matrix yang menghasilkan nilai *accuracy*, *precision*, *recall* dan *F1-Score* dalam bentuk persentase (0–100%) [23].

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan data

Dalam penelitian ini, dataset yang digunakan memuat gambar buah pepaya yang terbagi menjadi tiga kategori kematangan, masing-masing kategori berisi 100 gambar, jumlah total citra yang digunakan dalam penelitian ini adalah 300. Gambar 2. memperlihatkan contoh buah pepaya dari masing-masing tingkat kematangan.




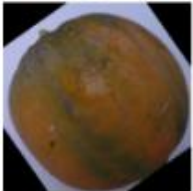
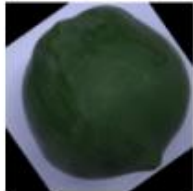
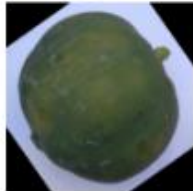








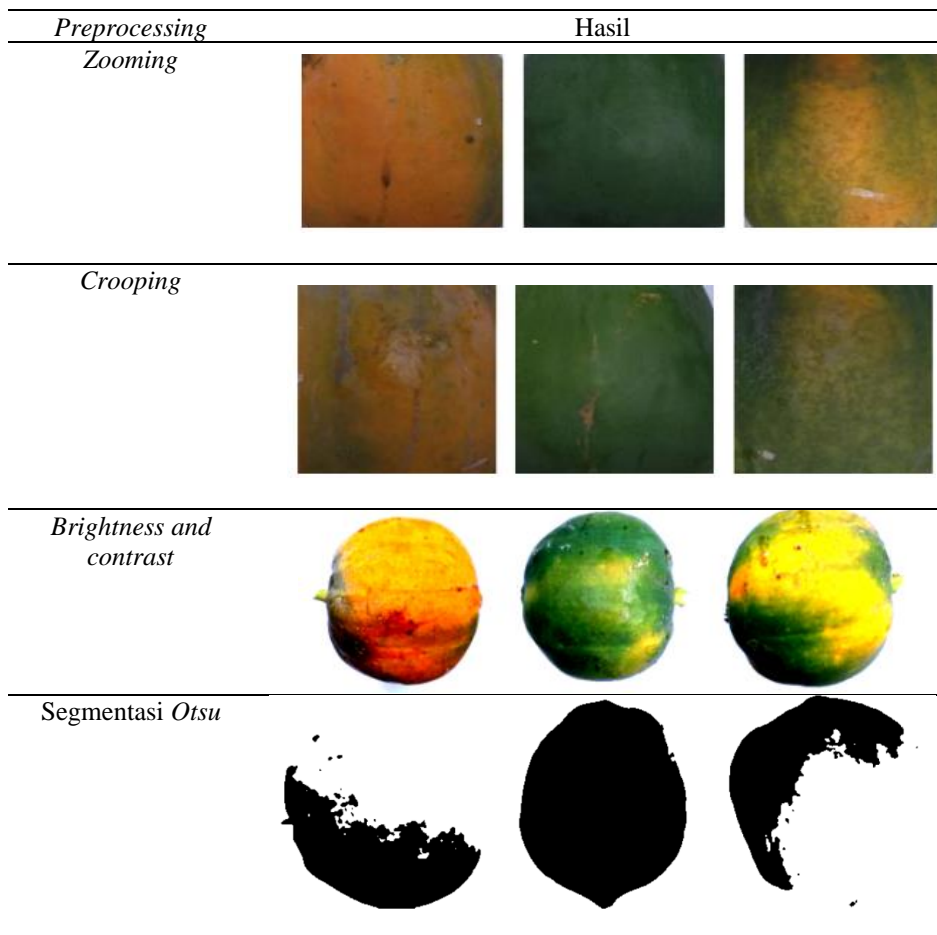
Gambar 2. Buah pepaya dengan 3 tingkat kematangan, mentah, mengkal dan matang

3.2 Preprocessing

Preprocessing dalam klasifikasi kematangan buah melibatkan serangkaian langkah seperti *resize*, rotasi, *flipping*, *zooming*, *cropping*, serta penyesuaian *brightness* dan *contrast*. Tujuan dari langkah-langkah ini adalah untuk menyamakan ukuran gambar, mengurangi noise, memperkaya variasi data, dan menonjolkan fitur penting seperti warna dan tekstur kulit pepaya. Proses ini membantu model dalam mengenali pola visual yang konsisten dan meningkatkan akurasi dalam membedakan tingkat kematangan buah. Hasil dari tahap *preprocessing* dapat dilihat pada Tabel 1.

Tabel 1. Hasil *Preprocessing*

<i>Preprocessing</i>	Hasil		
<i>Resize</i>			
<i>Rotasi</i>			
<i>Flipping Horizontal</i>			
<i>Flipping Vertikal</i>			



3.3 Ekstraksi Fitur Warna

Ekstraksi fitur warna menggunakan metode *GLCM* dilakukan untuk menangkap informasi tekstur dari citra grayscale buah pepaya yang mencerminkan tingkat kematangan. *GLCM* menghitung fitur statistik seperti *contrast*, *dissimilarity*, *homogeneity*, *energy*, *correlation*, dan *asm*, yang menggambarkan perubahan tekstur kulit buah. Fitur-fitur ini membantu algoritma klasifikasi dalam membedakan kategori kematangan buah, seperti mentah, mengkal, dan matang, dengan lebih akurat.

Tabel 2. Hasil ekstraksi fitur warna

Tingkat Kematangan	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	ASM
Pepaya Matang	30,44%	2,26%	0,52%	0,04%	0,99%	0,00%
Pepaya Mentah	32,17%	1,96%	0,59%	0,05%	0,99%	0,00%
Pepaya Mengkal	38,44%	2,24%	0,53%	0,04%	0,99%	0,00%

3.4 Split Data

Dari total 300 citra buah pepaya yang terbagi dalam tiga kategori tingkat kematangan (mentah, mengkal, dan matang), Setelah melalui tahapan preprocessing, jumlah dataset mengalami peningkatan menjadi 1.200 data sebagai hasil dari penerapan teknik augmentasi pada data awal, dataset akan dibagi menjadi dua bagian. Sebanyak 80% atau 960 citra digunakan menjadi data training untuk melatih model klasifikasi, sementara 20% atau 240 citra sisanya digunakan menjadi data testing untuk mengukur performa dan akurasi model yang telah dilatih. Tujuan dari pembagian ini adalah agar model dapat belajar dari beragam variasi data serta mampu menggeneralisasi dengan efektif terhadap data baru yang belum dikenal.

Tabel 3. Split data

Persentase data		Jumlah	
Training	Testing	Training	Testing
80%	20%	960	240
Total		1.200	



3.5 Modeling

Tahap pemodelan dilakukan terlebih dahulu sebelum pengujian, dengan menerapkan dua algoritma klasifikasi, yakni *Logistic Regression* dan *K-Nearest Neighbor (K-NN)*, untuk membandingkan performa keduanya terhadap dataset yang telah dipraproses.

```

from sklearn.linear_model import LogisticRegression

# Inisialisasi model Logistic Regression
logreg_model = LogisticRegression(random_state=42, max_iter=1000, C=1.0, solver='lbfgs')

# Melatih model dengan data pelatihan
# Gunakan X_train_flat yang sudah di-flatten
logreg_model.fit(X_train_flat, y_train)
    
```

Gambar 3. Modeling *Logistic Regression*

Pada Gambar 3 menunjukkan proses pemodelan klasifikasi menggunakan algoritma *Logistic Regression*, menggunakan data yang telah melewati tahapan preprocessing, dengan data awal sebanyak 300 menjadi 1.200 data, di mana model diinisialisasi dan dilatih untuk mengenali pola dalam data guna menentukan tingkat kematangan buah pepaya.

```

[ ] from sklearn.neighbors import KNeighborsClassifier

# Implementasi model knn
# Menggunakan nilai K terbaik dari data uji
knn_model = KNeighborsClassifier(n_neighbors=optimal_k_test) # Mengganti optimal_k_accuracy dengan optimal_k_test
knn_model.fit(X_train, y_train)
    
```

Gambar 4. Modeling *K-Nearest Neighbor*

Gambar 4 menunjukkan proses pemodelan klasifikasi menggunakan algoritma *K-Nearest Neighbor (KNN)*. Model diinisialisasi dengan jumlah tetangga terdekat terbaik (nilai k optimal) yaitu K=1, menggunakan data yang telah melewati tahapan *preprocessing*, dengan data awal sebanyak 300 menjadi 1.200 data, kemudian dilatih untuk mengenali pola dalam data guna mengklasifikasikan tingkat kematangan buah pepaya.

3.6 Pengujian Dan Evaluasi

Penelitian ini menggunakan dua algoritma untuk pengujian, yaitu *Logistic Regression* dan *K-Nearest Neighbor*. Sebelum dilakukan pengujian menggunakan algoritma *K-Nearest Neighbor* terlebih dahulu mencari nilai K terbesar. Pada gambar 4.3 hasil dari mencari nilai K terbesar.

K	Akurasi
0 1	0.870833
1 3	0.866667
2 5	0.845833
3 7	0.829167
4 9	0.816667

Nilai K terbaik berdasarkan akurasi di data uji: 1

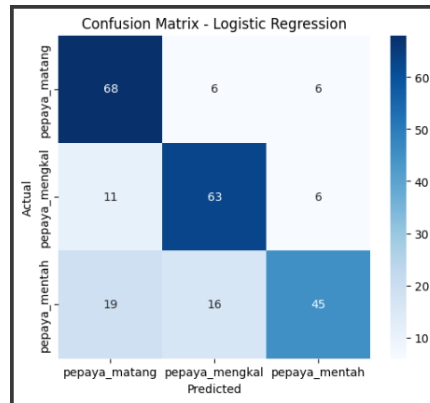
Gambar 5. Hasil Mencari nilai K terbesar

Setelah diuji dengan kedua algoritma tersebut, hasilnya dievaluasi menggunakan *confusion matrix* yang mencakup *metrik accuracy, precision, recall, dan f-measure (f1-score)*. Berikut adalah tabel yang menunjukkan hasil pengujian dan evaluasi dari kedua algoritma dengan pembagian dataset 80:20.

Tabel 4. Hasil dan evaluasi

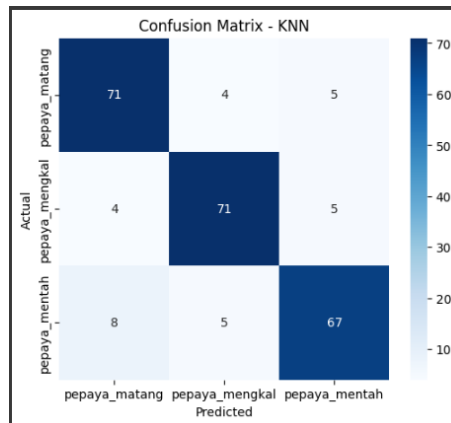
Model	Tingkat Kematangan	Accuracy	Precision	Recall	F1-Score
<i>Logistic Regression</i>	Pepaya matang	73%	69%	85%	76%
<i>Logistic Regression</i>	Pepaya mengkal	73%	74%	79%	76%
<i>Logistic Regression</i>	Pepaya mentah	73%	79%	56%	66%
<i>K-Nearest Neighbor</i>	Pepaya matang	87%	86%	89%	87%
<i>K-Nearest Neighbor</i>	Pepaya mengkal	87%	89%	89%	89%
<i>K-Nearest Neighbor</i>	Pepaya mentah	87%	87%	84%	85%

Dari Tabel 4, kedua algoritma dengan dataset yang telah melewati *preprocessing*, augmentasi dan ekstraksi fitur *GLCM* dengan total data awal 300 menjadi 1.200 data menunjukkan hasil yang berbeda akurasi, untuk algoritma *Logistic Regression* dengan regulasi L2 menunjukkan hasil akurasi sebesar 73%, dan algoritma *K-Nearest Neighbor* dengan nilai terbaik $K=1$ hasil akurasi 87%.



Gambar 6. *Confusion matrix Logistic Regression*

Pada Gambar 6, *confusion matrix* algoritma *Logistic Regression*, terlihat bahwa model memiliki performa klasifikasi yang cukup baik dalam mengklasifikasikan tingkat kematangan pepaya menjadi tiga kategori: matang, mengkal, dan mentah. Untuk kelas pepaya matang, sebanyak 68 data berhasil diklasifikasikan dengan benar, namun terdapat 6 data yang salah diklasifikasikan sebagai pepaya mengkal dan 6 data lainnya sebagai pepaya mentah. Pada kelas pepaya mengkal, model mengklasifikasikan 63 data secara benar, sementara 11 data salah diklasifikasikan sebagai pepaya matang dan 6 data lainnya sebagai pepaya mentah. Adapun untuk kelas pepaya mentah, hanya 45 data yang diklasifikasikan dengan benar, sedangkan 19 data salah diklasifikasikan sebagai pepaya matang dan 16 data sebagai pepaya mengkal. Hasil ini menunjukkan bahwa model memiliki kecenderungan untuk mengalami kesalahan klasifikasi, terutama pada kelas pepaya mentah, yang seringkali tertukar dengan dua kelas lainnya.



Gambar 7. *Confusion matrix K-Nearest Neighbor*

Gambar 7 menunjukkan *Confusion matrix* pada algoritma *K-Nearest Neighbor (KNN)*, model menunjukkan performa klasifikasi yang sangat baik terhadap ketiga kelas tingkat kematangan buah pepaya, yaitu matang, mengkal, dan mentah. Untuk kelas pepaya matang, sebanyak 71 data diklasifikasikan dengan benar, sementara 4 data salah diklasifikasikan sebagai pepaya mengkal dan 5 data sebagai pepaya mentah. Pada kelas pepaya mengkal, model juga mampu mengklasifikasikan 71 data secara akurat, dan hanya melakukan kesalahan pada 4 data yang diklasifikasikan sebagai pepaya matang serta 5 data sebagai pepaya mentah. Sedangkan pada kelas pepaya mentah, 67 data berhasil diklasifikasikan dengan benar, dengan kesalahan klasifikasi sebanyak 8 data ke kelas pepaya matang dan 5 data ke kelas pepaya mengkal. Secara keseluruhan, hasil ini menunjukkan bahwa algoritma *KNN* mampu menghasilkan klasifikasi yang akurat dan seimbang, dengan kesalahan yang relatif kecil pada ketiga kelas.

4. KESIMPULAN

Algoritma *Logistic Regression* dan *K-Nearest Neighbor (KNN)* dapat digunakan untuk mengklasifikasikan tingkat kematangan buah pepaya matang, mengkal, dan mentah dengan memanfaatkan citra sebagai input yang telah melalui tahapan *preprocessing*, augmentasi, dan ekstraksi fitur. Berdasarkan hasil pengujian, *KNN* dengan nilai $k=1$ menunjukkan akurasi yang lebih tinggi, yaitu 87%, dibandingkan dengan *Logistic Regression* yang menggunakan regulasi L2 dan hanya



mencapai 73%. Hal ini menunjukkan bahwa KNN lebih efektif dan optimal dalam mengklasifikasikan tingkat kematangan pada dataset yang digunakan. Untuk meningkatkan performa model, penambahan jumlah data dan variasi citra sangat dianjurkan agar model mampu melakukan generalisasi dengan lebih baik dan menghasilkan prediksi yang lebih akurat. Selain itu, eksplorasi penggunaan algoritma lain seperti *Random Forest*, *Support Vector Machine (SVM)*, maupun metode *Deep Learning* perlu dipertimbangkan. Pengembangan teknik ekstraksi fitur juga dapat dilakukan dengan menggunakan model warna yang lebih beragam seperti *HSV*, *LAB*, atau *histogram* warna, sehingga dapat memperoleh metode klasifikasi yang paling efektif dan representatif dalam mendeteksi tingkat kematangan buah pepaya.

REFERENCES

- [1] E. Tanadi, S. Palimbong, and K. B. Lewerissa, "Potensi Pemanfaatan Buah Pepaya dalam Produk Es Krim," *Applicable Innovation of Engineering and Science Research (AVoER)*, pp. 1–8, 2020.
- [2] R. Kurniawan, "Klasifikasi Tingkat Kematangan Buah Pepaya Berdasarkan Warna Kulit Menggunakan Sensor Warna TCS3200," *Journal ICTEE*, vol. 4, no. 1, pp. 1–13, 2023.
- [3] F. M. Fathoni, C. A. Putra, and A. L. Nurlaili, "Klasifikasi Penyakit Daun Anggur menggunakan metode k-nearest neighbor Berdasarkan Gray level co-occurrence matrix," *Biner: Jurnal Ilmiah Informatika dan Komputer*, vol. 3, no. 1, pp. 8–15, 2024.
- [4] R. A. Saputra, D. Puspitasari, A. Supriyatna, D. F. Saefudin, R. A. Purnama, and K. Ramanda, "Hyperparameter Optimization in CNN Algorithm for Chili Leaf Disease Classification" ICAISD, 2023
- [5] C. Suryanti and M. G. Rohman, "Klasifikasi Kualitas Buah Apel Berdasarkan Warna dan Bentuk Menggunakan Metode KNN," *Generation Journal*, vol. 8, no. 1, pp. 34–41, 2024.
- [6] S. Raysyah, V. Arinal, and D. I. Mulyana, "Klasifikasi tingkat kematangan buah kopi berdasarkan deteksi warna menggunakan metode knn dan pca," *JSII (Jurnal Sistem Informasi)*, vol. 8, no. 2, pp. 88–95, 2021.
- [7] Z. Zulkifli and R. Fajri, "Klasifikasi Tingkat Kematangan Buah Strawberry Menggunakan Algoritma Logistic Regression," *Data Sciences Indonesia (DSI)*, vol. 4, no. 2, pp. 50–59, Dec. 2024, doi: 10.47709/dsi.v4i2.4850.
- [8] H. P. Hadi and E. H. Rachmawanto, "Ekstraksi Fitur Warna Dan GLCM Pada Algoritma KNN Untuk Klasifikasi Kematangan Rambutan," *JIP (Jurnal Informatika Polinema)*, vol. 8, no. 3, 2022
- [9] K. A. Pratama, W. Priyo Atmaja, and V. Lusiana, "Klasifikasi Tingkat Kematangan Buah Kersen Menggunakan Citra HSI Dengan Metode K-Nearest Neighbor (KNN)," *Jurnal Orang Pintar Komputer*, vol. 11, no. 1, 2022, doi 10.30591/smartcomp.v11i1.3171
- [10] A. Rahmat, M. Syafiih, and M. Faid, "Implementasi Klasifikasi Potensi Penyakit Jantung Dengan Menggunakan Metode C4.5 Berbasis Website (Studi Kasus Kaggle.Com)," *INFOTECH journal*, vol. 9, no. 2, pp. 393–400, Jul. 2023, doi: 10.31949/infotech.v9i2.6295.
- [11] Adi Rizky, Ayu Ratna, Tohirin, "Klasifikasi Daging Sapi Berdasarkan Ciri Warna Dengan Metode Otsu dan K-Nearest Neighbor", *Techno Explore Jurnal Ilmu Komputer dan Informasi*, Vol.6, No 1, 2021
- [12] N. Mega Saraswati, R. Cipta Sigitta Hariyono, and D. Chandra, "Face Recognition Menggunakan Metode Haar Cascade Classifier Dan Local Binary Pattern Histogram.", *Jurnal Media Elektrik*, Vol.20, No.3, 2023
- [13] J. Khatib Sulaiman, N. Pramesti Aprilia, T. Herlina Rochadiani, and U. Pradita, "Image Captioning untuk Gambar Rambu Lalu Lintas Indonesia Menggunakan Pretrained CNN dan Transformer," *Indonesian Journal of Computer Science*, Vol 13, No 3, 2024.
- [14] N. Mukaromah, S. Mulyono, and U. Islam Sultan Agung, "Implementasi Stable Diffusion Dan Fine-Tuning Low Rank Adaptation Untuk Pembuatan Logo," *Jurnal Rekaayasa Sistem Informasi dan Teknologi*, No 3-Februari, 2025
- [15] Ni luh, Nyoman, Made, "Klasifikasi Jajanan Khas Bali Untuk Preservasi Pengetahuan Kuliner Lokal Menggunakan Arsitektur VGG-16", *Journal Sintech*, vol 7, No.1-April 2024. Available: <https://doi.org/10.31598>
- [16] Fadli, Lucky, Lisawita, "Klasifikasi Buah Pinang Berdasarkan Tingkat Kematangan Menggunakan Metode Naïve Bayes", *Journal Semester*, Vol 3, No 1, 2024.
- [17] I. Putu, C. Jumariana, and P. Sugiartawan, "Identifikasi Pengenalan Pola Daun Kelor Kering Dengan Yolo V8," *IJEIS (Indonesian Journal of Electronics and Instrumentations Systems)*, vol. 14, no. 1, pp. 91–100, 2024, doi: 10.22146/ijeis.94871.
- [18] J. C. Lapendy, A. A. C. Resky, H. Makmur, A. B. Kaswar, D. D. Andayani, and F. Adiba, "Klasifikasi rasa jeruk siam berdasarkan warna dan tekstur berbasis pengolahan citra digital," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 2, pp. 756–767, 2024.
- [19] W. Shinta Sari and C. Atika Sari, "Klasifikasi Bunga Mawar Menggunakan KNN dan Ekstraksi Fitur GLCM dan HSV," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 145–156, 2022.
- [20] T. Pusdita and V. Lusiana, "JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) Journal homepage: <https://jurnal.stkipppgritulungagung.ac.id/index.php/jipi> Deteksi Motif Sarung Tenun Goyor Botolan Kabupaten Pemalang Menggunakan Metode KNN," vol. 10, no. 1, pp. 473–481, 2025, doi: 10.29100/jipi.v10i1.5778.
- [21] N. D. Azzahra, A. Ambarwati, A. Desiani, S. I. Maiyanti, and I. Ramayanti, "Perbandingan Algoritma K-Nearest Neighbor Dan Logistic Regression Dalam Klasifikasi Penyakit Kanker Serviks," *Energy : Jurnal Ilmiah Ilmu-Ilmu Teknik*, vol. 14, no. 1, pp. 1–8, May 2024, doi: 10.51747/energy.v14i1.1843.
- [22] T. Nurmayanti, D. Hartini, T. Rohana, S. A. P. Lestari, and D. Wahiddin, "Comparison of K-Nearest Neighbors and Convolutional Neural Network Algorithms in Potato Leaf Disease Classification," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 8, no. 1, pp. 360–372, 2024.
- [23] I. M. A. A. Pramana, I. W. Sudiarsa, and P. G. S. C. Nugraha, "Penerapan Algoritma Naïve Bayes Untuk Prediksi Penjualan Produk Terlaris Pada CV Akusara Jaya Abadi," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 10, no. 4, 2023, doi: <https://doi.org/10.35957/jatisi.v10i4.6498>