

Penerapan *Support Vector Machine* pada Klasifikasi Kondisi Kematangan Cabai Rawit

1st Anggie Wiyani Putri
 Universitas Buana Perjuangan Karawang
 Karawang, Indonesia
 if21.anggieputri@mhs.ubpkarawang.ac.id

2nd Ahmad Fauzi
 Universitas Buana Perjuangan Karawang
 Karawang, Indonesia
 afauzi@ubpkarawang.ac.id

3rd Dwi Sulisty Kusumaningrum
 Universitas Buana Perjuangan Karawang
 Karawang, Indonesia
 dwi.sulisty@ubpkarawang.ac.id

4th Sutan Faisal
 Universitas Buana Perjuangan Karawang
 Karawang, Indonesia
 sutan.faisal@ubpkarawang.ac.id

Abstract— Cabai rawit (*Capsicum frutescens*) merupakan komoditas penting di Indonesia, namun ketergantungan impor masih tinggi akibat kendala sortasi dan distribusi yang tidak merata. Proses sortasi manual seringkali tidak akurat, menyebabkan cabai mentah, matang, atau rusak tercampur, sehingga menurunkan kualitas dan nilai jual. Penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis kondisi kematangan cabai rawit menggunakan algoritma *Support Vector Machine* (SVM) dengan ekstraksi fitur kombinasi HSV, GLCM, dan *Local Binary Pattern* (LBP). Metode yang digunakan meliputi augmentasi data, segmentasi berbasis *HSV thresholding*, serta ekstraksi fitur warna dan tekstur. Hasil evaluasi menunjukkan model SVM dengan kernel RBF mencapai rata-rata F1-Score sebesar 84% dengan performa terbaik pada kelas matang dengan F1-Score sebesar 92%. Namun, kelas rusak memiliki F1-Score lebih rendah (71%) akibat ketidakseimbangan data. Simpulan penelitian ini menunjukkan bahwa kombinasi fitur HSV, GLCM, dan LBP efektif untuk klasifikasi, namun perlu pengembangan dataset dan pendekatan *deep learning* seperti CNN-SVM untuk meningkatkan akurasi berdasarkan F1-Score pada kelas minoritas.

Kata kunci—Cabai Rawit, Kondisi Kematangan, Klasifikasi, Machine Learning, Support Vector Machine (SVM), HSV, GLCM, LBP

I. PENDAHULUAN

Di Indonesia, Cabai Rawit (*Capsicum frutescens*) adalah salah satu jenis tanaman pertanian yang paling populer. Menurut Muhammad Taufiq Ratule, Plt. Dirjen Hortikultura Kementerian, menyatakan bahwa populasi di Indonesia mengonsumsi 1,17 juta ton cabai setiap tahunnya [1]. Cabai adalah salah satu komoditas penting dalam perdagangan domestik dan internasional karena nilainya yang tinggi, selain itu cabai rawit menjadi bumbu utama dalam banyak masakan.

Laporan Pusat Data dan Sistem Informasi Kementerian Pertanian menunjukkan bahwa impor cabai segar pada tahun 2023 mencapai 50,3 ribu ton, dengan defisit 120 juta USD [2]. Dinyatakan oleh Plt. Dirjen Hortikultura Kementerian Muhammad Taufiq Ratule dalam Forum Cabai Nasional, kendala sortasi dasar dan distribusi yang tidak merata di beberapa wilayah Indonesia menyebabkan ketergantungan impor ini [1]. Seperti halnya jeruk nipis, Cabai Rawit sebagai tanaman hortikultura pun memiliki kondisi yang apabila cabai mentah (berwarna hijau pucat) disatukan dengan cabai matang (berwarna orange) akan mempercepat proses kematangan pada cabai mentah, begitu pula dengan pencampuran cabai mentah, matang, dan rusak dalam distribusi—seperti yang terjadi pada jeruk [3] dapat mempercepat kerusakan dan menurunkan nilai jual. Proses memilih atau menyortir kondisi kematangan cabai rawit seringkali bergantung pada pengamatan visual yang kurang baik dan tidak memiliki standar objektif, yang berdampak pada kualitas cabai yang didistribusikan. Kondisi kematangan sayuran dan buah memengaruhi kandungan nutrisi dan daya tahan produk selama penyimpanan serta rasa dan teksturnya [4]. Akibatnya, cabai yang terlalu matang atau belum matang sering lolos hingga terjual, hal ini dapat menurunkan kepuasan pelanggan, mengurangi daya jual, dan mengganggu stabilitas pasar domestic. Oleh karena itu, dibutuhkan sistem klasifikasi otomatis berbasis komputer untuk meningkatkan kemampuan sortasi kondisi kematangan cabai rawit.

Berbagai penelitian sebelumnya telah berhasil mengembangkan pendekatan untuk menentukan tingkat kematangan cabai dengan tingkat akurasi yang bervariasi. Pada Data *train* dengan model *Backpropagation* berbasis GLCM mencapai akurasi 81,4% dan 74,2% Data *Testing* [5], *Gradient Boosting* 76% [6], dan *hybrid CNN-KNN* 99,33% [7]. Sementara penelitian dengan SVM menggunakan kombinasi HSV-GLCM mencapai akurasi sebesar 92,12% [8] dan *hybrid CNN-SVM* mencapai F1-Score sebesar 94,9% [9]. Namun, penelitian-penelitian tersebut masih memiliki keterbatasan dalam dataset dan pemanfaatan fitur gabungan. Oleh karena itu, penelitian ini memilih SVM dengan ekstraksi fitur HSV, GLCM, dan LBP untuk meningkatkan akurasi klasifikasi pada dataset yang lebih besar. Implementasi ini diharapkan dapat memberikan solusi praktis bagi industri pertanian dalam meningkatkan kualitas distribusi dan daya saing cabai di pasar domestik maupun internasional.

II. METODELOGI PENELITIAN

Agar mendapatkan *output* sesuai yang diharapkan tentu perlu tahap penggeraan yang jelas sehingga dapat menjawab tujuan dari masalah yang telah dipaparkan sebelumnya. Berikut tahapananya:



Gambar 1. Tahapan Prosedur Penelitian

A. Pengumpulan Data

Penelitian ini menggunakan dataset sebanyak 523 citra cabai rawit yang diambil dengan kamera Google Pixel 6 Pro pada jarak 15 cm untuk memastikan detail optimal. Citra diperoleh dari pasar tradisional, mencakup tiga kategori kematangan: matang (278 citra), mentah (161 citra), dan rusak (84 citra), sehingga merepresentasikan kondisi nyata di lapangan.



Gambar 2. Contoh citra dari tiap label

B. Pre Processing Data

Pada Tahapan ini, dataset citra akan terlebih dahulu diunggah ke dalam *google drive* sehingga mampu untuk melanjutkan ke tahap Selanjutnya. Setelah dataset berhasil di muat, dataset akan melewati tahap augmentasi dan Segmentasi. Selanjutnya data hasil *preprocessing* siap untuk melanjutkan tahap ekstraksi Fitur HSV dan GLCM lalu dilatih pada pemodelan *Support Vector Machine*.

a. Augmentasi

Tahap augmentasi ini diperlukan untuk menambah variasi pada data train tanpa mengubah infomasi mendasar yang bertujuan untuk mencegah terjadinya *overfitting* dan meningkatkan generalisasi model dan membantu model mengenali pola yang lebih *robust*.

Tabel 1. Augmentasi Data

Algoritma	: Augmentasi Data
Input	: Dataset Citra Cabai Rawit
Output	: Data hasil augmentasi
1.	Membaca Dataset berisi 3 label (Matang, Mentah, dan Rusak)
2.	Memutar Citra dari sudut 30° hingga 90° untuk variasi perspektif (<i>Rotate</i>)
3.	Membalik citra secara horizontal/vertikal (<i>Flip</i>)
4.	Menambah pencahayaan dan kontras
5.	Menambahkan <i>Motion Blur</i>
6.	Menambahkan gangguan digital (<i>Noise Gaussian</i>)
7.	Menghilangkan sebagian kecil citra
8.	Hasil augmentasi

b. Segmentasi

Metode segmentasi di tambahkan guna memisahkan objek cabai dengan *backgroundnya*, ini dapat memberikan bantuan kepada model agar dapat lebih terfokus dalam mengekstraksi objek dan dapat memaksimalkan proses pemodelan.

Tabel 2. Segmentasi Data

Algoritma	: Segmentasi Data
Input	: Data Hasil Augmentasi
Output	: Data hasil Segmentasi dan Masking
1.	Input Hasil Augmentasi
2.	Menentukan warna dengan nilai HSV yang ditentukan oleh fungsi <i>Lower</i> dan <i>Upper</i>
3.	Mengganti <i>background</i> dengan warna hitam yang selanjutnya akan dijadikan <i>masking</i>
4.	Hasil Segmentasi dan <i>Masking</i>

C. Ekstraksi Fitur

Tahapan ini cukup krusial karena dapat mempengaruhi hasil prediksi berdasarkan warna dan tekstur dari objek citra cabai itu sendiri. Hasil nilai yang di peroleh dari ekstraksi ini akan menjadi input bagi pelatihan model SVM.

a. Fitur Ruang Warna HSV

Umumnya RGB adalah komponen warna primer yang terdiri dari *Red* (Merah), *Green* (Hijau), dan *Blue* (Biru) menjadikan RGB memiliki kelebihan sebagai ruang warna yang luas, namun hal ini juga menjadi kelemahan dalam segmentasi. Oleh karena itu diperlukan konversi ke HSV sehingga mampu menghasilkan nilai rata-rata *hue*, *saturation*, dan *value* sebagai ekstraksi fitur warna [5]. Adapun rumus konversi RGB terhadap HSV:

Tabel 3. Ekstraksi Ruang Warna HSV

Algoritma	: HSV
Input	: Citra RGB Hasil Segmentasi
Output	: 3 fitur warna <i>mean</i> H, S, dan V
1.	Membaca Dataset Menentukan <i>Min Max</i> . $R = \frac{R}{255}$, $G = \frac{G}{255}$, $B = \frac{B}{255}$
2.	Menghitung V dengan mencari nilai <i>Max</i> dari RGB
3.	Menghitung $V_m = V - \min(r,g,b)$
4.	Menghitung $S = \frac{v - (r,g,b)}{v}$
5.	Menghitung nilai H berdasarkan nilai R, G, dan B
6.	Memisahkan <i>channel</i> H, S, dan V
7.	Hitung rata – rata setiap <i>channel</i> .

b. Fitur Tekstur GLCM

Selanjutnya, dilakukan ekstraksi fitur tekstur dengan matriks GLCM. Matriks pada GLCM dapat merepresentasikan hubungan suatu derajat keabuan antara dua piksel dengan intensitas tertentu dalam jarak di simbolkan d dan arah sudut disimbolkan Θ , sudut yang biasa digunakan dalam GLCM untuk menghitung *co-occurrence* terdapat pada sudut 0° , 45° , 90° , dan 135° .

Setelah Melakukan penentuan awal matriks berdasarkan dua pasang piksel, dilakukan *transpose* dengan membalikan nilai dari x (baris) menjadi y (kolom) begitu pula sebaliknya. Hasil *transpose* dinormalisasikan dengan membagi nilai dengan jumlah keseluruhan nilai.

Tabel 4. Eskstraksi Fitur Tekstur GLCM

Algoritma	: GLCM
Input	: Citra <i>Greyscale</i> Hasil Segmentasi
Output	: 6 fitur tekstur
1.	Membaca citra yang sudah di konversi ke <i>Grayscale</i>
2.	Menghitung jarak 1, 3, 5 piksel
3.	Menghitung Sudut arah 0° , 45° , 90° , dan 135°
4.	Mengekstraksi 5 properti statistik GLCM
5.	Menghitung nilai <i>entropy</i> GLCM
6.	Hasil 6 Fitur (<i>contrast</i> , <i>energy</i> , <i>homogeneity</i> , <i>correlation</i> , <i>dissimilarity</i> , <i>entropy</i>).

c. Fitur Tekstur LBP

Local Binary Pattern (LBP) merupakan metode ekstraksi fitur tekstur yang efektif untuk gambar *grayscale*. Metode ini bekerja dengan membandingkan intensitas piksel pusat dengan 8 tetangganya (radius=1), memberikan nilai 1 jika intensitas tetangga \geq piksel pusat, atau 0 jika lebih rendah. Pola biner 8-bit yang dihasilkan kemudian dikonversi ke nilai desimal sebagai representasi fitur lokal [10]. Nilai fitur dinormalisasi untuk memastikan komparabilitas antar gambar, menghasilkan vektor fitur akhir.

Tabel 5. Ekstraksi Fitur LBP

Algoritma	: LBP
Input	: Citra <i>Greyscale</i> Hasil Segmentasi
Output	: 10 fitur pola lokal
1.	Membaca citra yang sudah di konversi ke <i>Grayscale</i>

-
2. Menerapkan operator LBP dengan radius 1 dengan 8 titik tetangga
 3. Menghitung histogram pola biner
 4. Normalisasi Histogram
-

Setelah berhasil mendefinisikan dan mendapatkan nilai fiturnya, langkah selanjutnya adalah mengintegrasikan seluruh fitur seperti berikut:

Tabel 6. Kombinasi Fitur

Algoritma	: Kombinasi Fitur
Input	: <i>Output HSV + GLCM + LBP</i>
Output	: Vektor fitur kombinasi (20 Fitur)
1.	Menggabungkan semua fitur seperti, 3 fitur HSV, 6 fitur GLCM, 10 fitur LBP, dan 1 fitur <i>entropy greyscale</i>
2.	Normalisasikan dengan <i>StandardScaler</i>

D. Implementasi Klasifikasi Algoritma *Support Vector Machine*

Dataset dibagi 80:20 (*train:test*) untuk *tuning* parameter. Algoritma SVM *One-vs-Rest* diaplikasikan setelah ekstraksi fitur, dengan *class_weight* untuk menangani ketidakseimbangan kelas minoritas (misal: cabai rusak). Parameter ini meningkatkan sensitivitas model terhadap kelas minoritas, mengoptimalkan metrik *recall* dan *F1-Score* [10].

Tabel 7. Klasifikasi Algoritma SVM

Algoritma	: Klasifikasi SVM
Input	: Data <i>Train</i>
Output	: Model Prediksi dan Hasil Prediksi Klasifikasi
1.	Membaca data <i>train</i>
2.	Melakukan <i>tuning</i> pada kernel <i>linear</i> , RBF, <i>Polynomial</i> , Sigmoid
3.	Mencari <i>hyperplane</i> terbaik untuk memisahkan data menggunakan kernel hasil <i>Tuning</i>
4.	Penerapan parameter <i>Class Weight</i> untuk menyeimbangkan bobot label
5.	Menggunakan metode SVM <i>One-vs-Rest</i>
6.	Hasil prediksi klasifikasi

E. Evaluasi Model

Pada tahap terakhir ini, pengujian dilakukan dengan menggunakan model yang telah dilatih sebelumnya. Setelah itu, proses evaluasi dilakukan menggunakan metode *Confusion Matrix* dan ROC AUC untuk mengevaluasi seberapa baik kinerja model.

a. *Confusion Matrix*

Pada tahap ini, metode *Confusion Matrix* digunakan untuk mengetahui nilai dari akurasi, presisi, *recall*, dan *F1-Score*. Evaluasi ini sangat penting karena hasil prediksi akan menunjukkan kinerja model yang telah dilatih sehingga mereka dapat menjawab rumusan masalah sebelumnya.

- Akurasi : proporsi prediksi yang benar dari semua prediksi

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3.1)$$

- Presisi : Proporsi prediksi positif yang benar dari semua prediksi positif.

$$\text{Presisi} = \frac{TP}{(TP+FP)} \quad (3.2)$$

- *Recall* : Proporsi sampel positif yang benar-benar diklasifikasikan sebagai positif.

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (3.3)$$

- *F1-Score* : Nilai harmonik rata-rata antara *precision* dan *recall*.

$$F1 - Score = \frac{2(Precision + Recall)}{(Precision + Recall)} \quad (3.4)$$

b. ROC AUC

Evaluasi kinerja model juga mencakup pengukuran ROC AUC (*Receiver Operating Characteristic – Area Under Curve*) untuk menilai kemampuan klasifikasi *multiclass*, khususnya pada distribusi kelas yang tidak seimbang. ROC menggambarkan hubungan antara TPR (*True Positive Rate*) dan FPR (*False Positive Rate*), sedangkan AUC merepresentasikan luas area di bawah kurva ROC. Pada kasus *multiclass*, digunakan pendekatan *One-vs-Rest* (OvR) dan hasil AUC dirata-ratakan dengan metode *macro-average* untuk menggambarkan performa model secara keseluruhan. Adapun permasalahan nya sebagai berikut :

- *True Positive Rate (Recall)* : $TPR = \frac{TP}{TP+FN}$ (3.5)

- *False Positive Rate* : $FPR = \frac{FP}{FP+TN}$ (3.6)

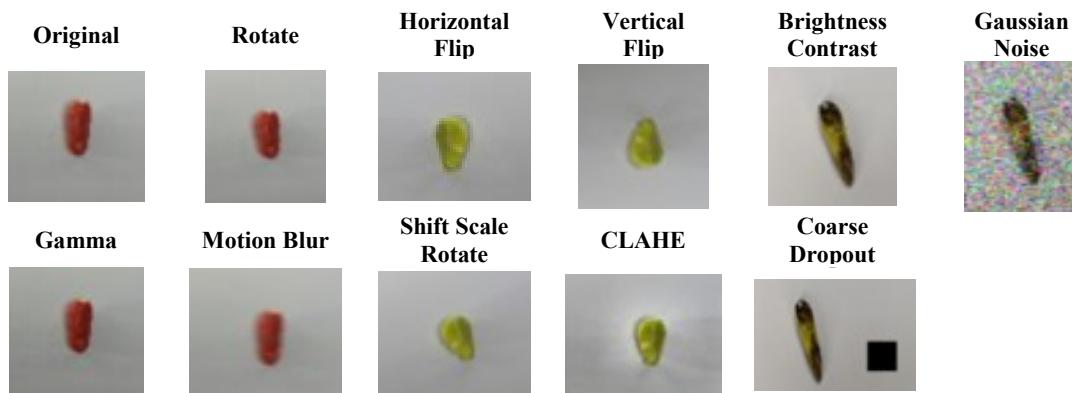
III. HASIL DAN PEMBAHASAN

A. Hasil Pre Processing Data

Dataset citra cabai rawit diproses melalui dua teknik: augmentasi dan segmentasi. Terlebih dahulu, tahap augmentasi dilakukan untuk meningkatkan variasi dataset latih. Kemudian, tahap segmentasi dilakukan untuk memisahkan objek dari latar belakang atau *background*, sehingga ekstraksi fitur berikutnya hanya berkonsentrasi pada menghitung nilai warna serat tekstur dari objek itu sendiri. Ini tampaknya dapat meningkatkan kinerja model klasifikasi. Ini adalah hasil dari proses augmentasi dan segmentasi:

a. Hasil Augmentasi

Gambar di bawah menunjukkan sampel cabai dari tiap kelas yang berhasil memuat variasi data pada data *train* di setiap kelas. Dapat dilihat bahwa gambar asli kini telah berputar, terbalik, memiliki *noise*, dicerahkan dan diperbesar.



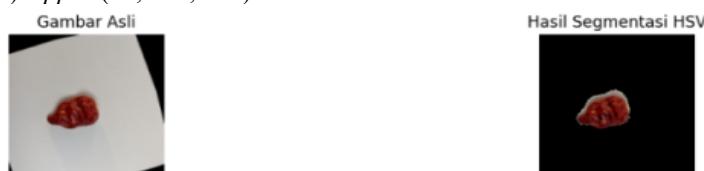
Gambar 6. Hasil Augmentasi

b. Hasil Segmentasi

HSV Tresholding, atau penggambaran berbasis ruang warna, berhasil membedakan objek cabai rawit dari *background* atau latar belakang berdasarkan ciri warna masing-masing kelas. Dalam hal ini, citra awal yang masih dalam format BGR diubah ke dalam format HSV, dan nilai atas dan bawahnya ditetapkan sebagai batasan warna untuk membedakan warna objek dari *background*.

a). Kelas Matang :

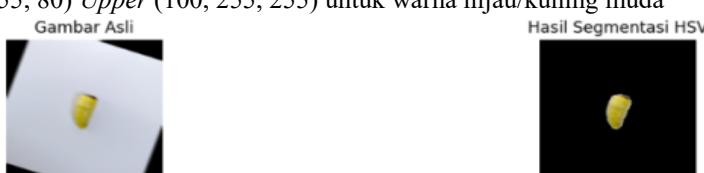
- *Lower* (20, 90, 100) *Upper* (35, 255, 255) untuk warna Orange
- *Lower* (0, 100, 50) *Upper* (10, 255, 255) untuk warna Merah



Gambar 7. Hasil Segmentasi Kelas Matang

b). Kelas Mentah :

- *Lower* (85, 55, 80) *Upper* (100, 255, 255) untuk warna hijau/kuning muda



Gambar 8. Hasil Segmentasi Kelas Mentah

c). Kelas Rusak:

- *Lower* (15, 30, 25) *Upper* (25, 25, 15) untuk coklat gelap



Gambar 9. Hasil Segmentasi Kelas Rusak

B. Hasil Ekstraksi Fitur

Ekstraksi Fitur dilakukan menggunakan data dari segmentasi yang telah di simpan sementara. Fitur yang diadopsi berupa 3 fitur rata-rata dari ruang warna HSV dan 6 fitur Tekstur GLCM seperti (*contrast, energy, homogeneity, correlation, dissimilarity, entropy*) serta 10 fitur pola lokal dari LBP dalam menentukan keriput pada permukaan fisik cabai rawit. Berikut hasil dari ekstraksi fitur pada kelas matang :



Gambar 10. Sampel Asli kelas Matang di ekstraksi

Tabel 8. Contoh Statistik Hasil Ekstraksi Fitur (Normalisasi)

	Mean	Std	Min	Max
Mean_H	-0.642	0.0	-0.642	-0.642
Mean_S	-0.484	0.0	-0.484	-0.484
Mean_V	-0.672	0.0	-0.672	-0.672
Contrast	-0.624	0.0	-0.624	-0.624
Energy	0.643	0.0	0.643	0.643
Homogeneity	0.635	0.0	0.635	0.635
Correlation	0.379	0.0	0.379	0.379
Dissimilarity	-0.636	0.0	-0.636	-0.636
GLCM_Entrropy	-0.641	0.0	-0.641	-0.641
LBP_0	-0.585	0.0	-0.585	-0.585
LBP_1	-0.619	0.0	-0.619	-0.619
LBP_2	-0.414	0.0	-0.414	-0.414
LBP_3	-0.49	0.0	-0.49	-0.49
LBP_4	-0.439	0.0	-0.439	-0.439
LBP_5	-0.283	0.0	-0.283	-0.283
LBP_6	0.056	0.0	0.056	0.056
LBP_7	0.284	0.0	0.284	0.284
LBP_8	0.649	0.0	0.649	0.649
LBP_9	-0.386	0.0	-0.386	-0.386
Image_Entrropy	-0.635	0.0	-0.635	-0.635

Setelah ekstraksi fitur di terapkan ke seluruh kelas maka hasil dari ekstraksi fitur berupa vektor 2D yang kemudian di simpan sebagai *array* ke dalam fungsi *x_features.shape* lalu di normalisasikan dan disimpan lagi kedalam fungsi *x_normalized* yang jelas akan menjadi input pada tahap klasifikasi SVM.

C. Hasil Model Klasifikasi SVM

Sebelum pelatihan, 5.753 citra dibagi menjadi data train (4.602) dan test (1.151) dengan rasio 80:20. Tahap *tuning* menggunakan empat kernel SVM (*Linear*, *RBF*, *Polynomial*, *Sigmoid*) untuk mengoptimasi klasifikasi fitur kompleks. Peneliti mengadopsi pendekatan dari studi Irma [11] yang menggunakan kernel *linear* karena efisien untuk data *linier* dan minim penyesuaian parameter. Replikasi tersebut mencapai akurasi 99% namun mengalami *overfitting*, membuktikan ketidakmampuannya menangani kompleksitas visual citra. Hasil tuning mengungkap kernel *linear*, meski efisien, kurang efektif untuk data *non-linier*. Pengujian menggunakan SVC (*Scikit-Learn*) dengan parameter *class_weight* untuk menyeimbangkan kelas minoritas ('Rusak').

== PERBANDINGAN KINERJA KERNEL SVM ==					
	Kernel	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
1	rbf	73.935708	68.534169	69.024688	68.559717
0	linear	70.981755	66.868684	67.880797	66.362990
2	poly	66.463944	61.477886	62.739444	61.694014
3	sigmoid	42.050391	51.736817	41.936478	41.735311

Gambar 11. Hasil Proses Tuning

Hasil menunjukkan kernel RBF memberikan performa terbaik (akurasi 73,94%, F1-Score 68,56%), diikuti kernel *Linear*, sementara *Polynomial* dan *Sigmoid* menunjukkan hasil yang rendah. Oleh karena itu, pelatihan akhir model klasifikasi SVM dilakukan menggunakan kernel RBF dengan parameter $C=100$ dan $gamma=0.1$.

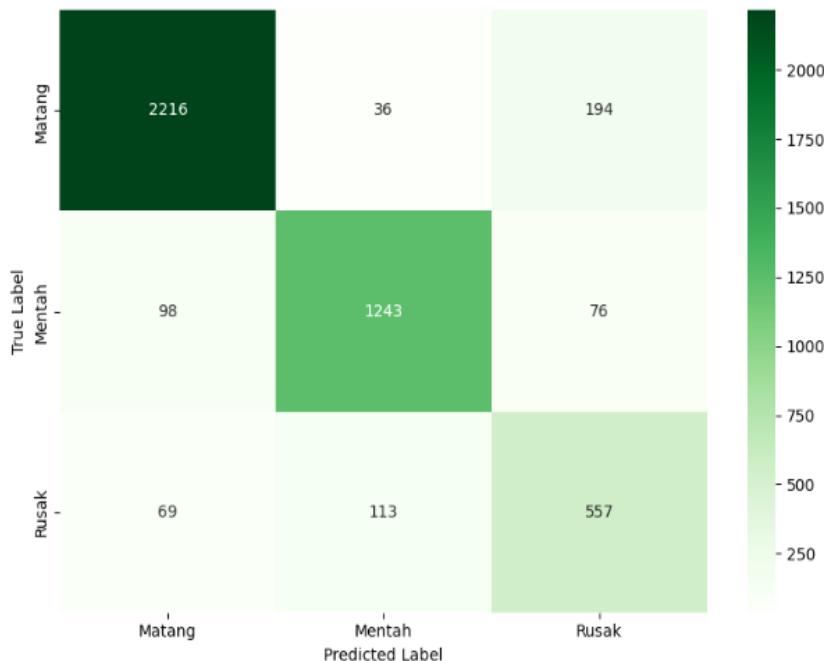
```
svm_grid_model = SVC(kernel=best_kernel_initial, class_weight='balanced',
                      decision_function_shape='ovr', random_state=42)
```

Gambar 12. Klasifikasi SVM

D. Hasil Evaluasi Model

Penjabaran berikut menyajikan hasil evaluasi dari kedua metode yang digunakan dalam menilai kinerja model, berdasarkan capaian performa yang diperoleh dari proses pemodelan.

a. Confusion Matrix



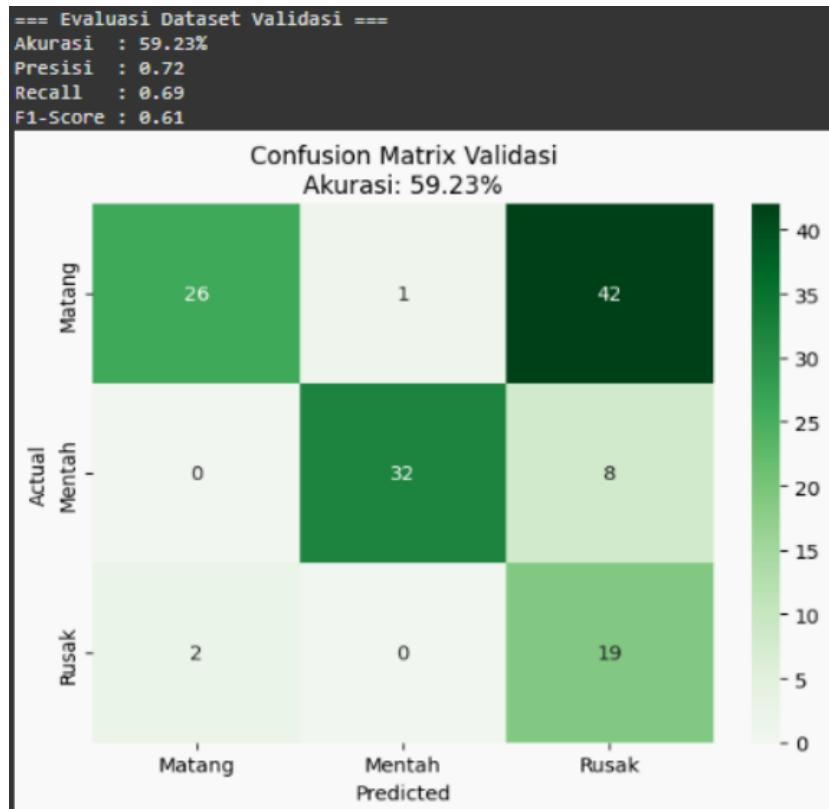
Gambar 13. Confusion Matrix Data Train

Proses pelatihan model di proses ulang menggunakan data *train*, sehingga menghasilkan performa akurasi mencapai 87%, dengan Presisi 83%, Recall 85%, dan F1-Score sebesar 84%. Berikut visualisasi dengan *confusion matrix* :

== Classification Report - Train Data ==				
	precision	recall	f1-score	support
Matang	0.93	0.91	0.92	2446
Mentah	0.89	0.88	0.89	1417
Rusak	0.67	0.75	0.71	739
accuracy			0.87	4602
macro avg	0.83	0.85	0.84	4602
weighted avg	0.88	0.87	0.87	4602

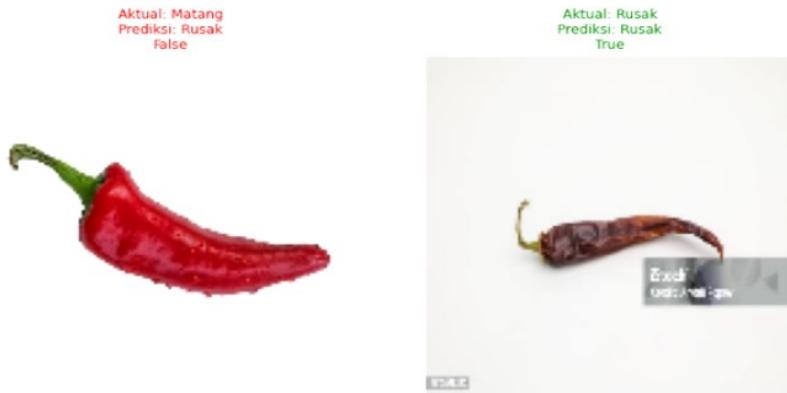
Gambar 14. Hasil Klasifikasi Data Train

Model juga diuji pada data validasi, yaitu sekumpulan citra cabai yang belum pernah digunakan selama proses pelatihan. Pengujian ini bertujuan untuk mengevaluasi kemampuan generalisasi model terhadap data baru serta melatih sistem model agar dapat mengenali objek berdasarkan tekstur dan warna, bukan hanya berdasarkan kemiripan dengan data *train*. Data validasi diperoleh dari berbagai sumber di internet yang dikumpulkan oleh penulis. Adapun hasil klasifikasinya disajikan sebagai berikut:



Gambar 15. Confusion Matrix Data Validasi

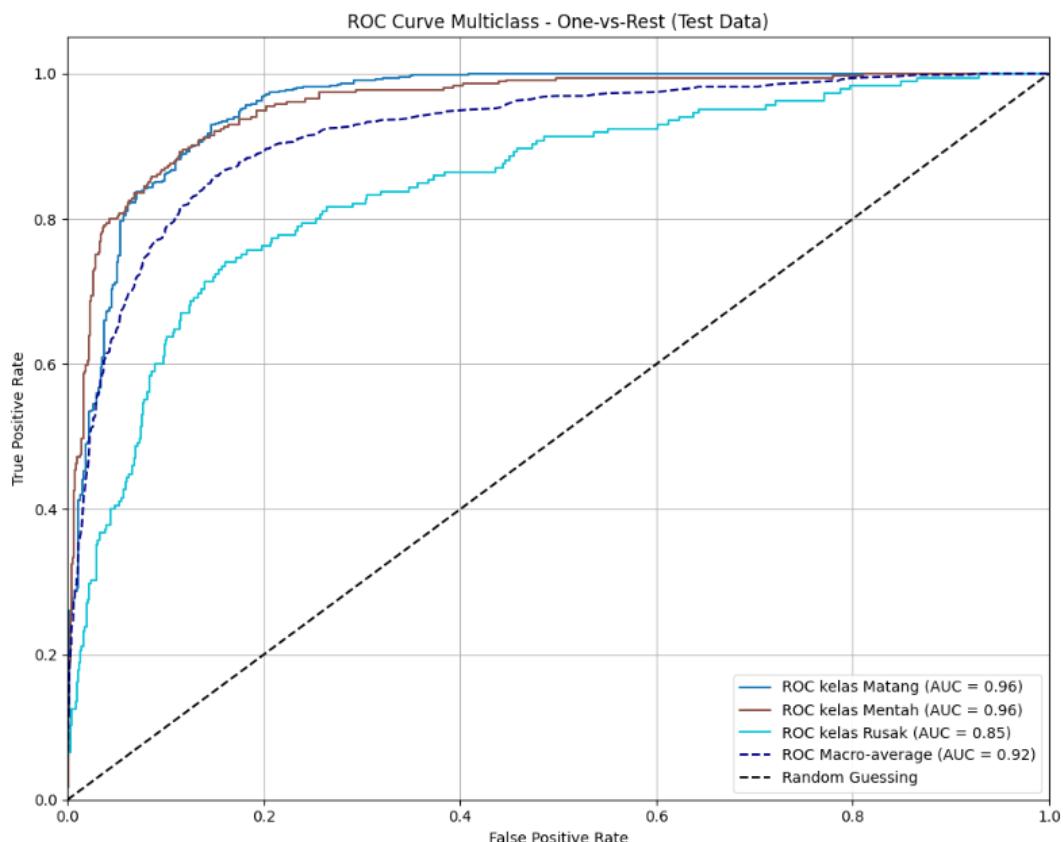




Gambar 16. Hasil Prediksi Data Validasi

Hasil prediksi data validasi menggunakan model SVM menunjukkan akurasi 59%, dengan beberapa kesalahan klasifikasi akibat kemiripan visual antara cabai matang dan rusak, khususnya pada warna dan tekstur. Fitur seperti bercak kehitaman, keriput, serta refleksi cahaya pada permukaan cabai memengaruhi hasil ekstraksi fitur (GLCM dan LBP), sehingga menurunkan akurasi model dalam membedakan kedua kelas tersebut. Temuan ini mengindikasikan bahwa *overlap* visual dan kondisi pencahayaan masih menjadi tantangan dalam klasifikasi citra berbasis fitur warna dan tekstur.

b. ROC AUC



Gambar 17. Grafik ROC AUC

```
== ROC AUC Score per Kelas ==
Matang: 0.956
Mentah: 0.956
Rusak: 0.845
Macro-average AUC: 0.920
```

Gambar 18. Score ROC AUC

Evaluasi menggunakan ROC AUC *One-vs-Rest* menunjukkan bahwa model SVM dengan kernel RBF mampu membedakan antar kelas dengan baik. Nilai AUC untuk kelas matang dan mentah sebesar 0,956, serta 0,845 untuk kelas rusak. Nilai *macro-average* AUC mencapai 0,920, mengindikasikan performa klasifikasi yang seimbang dan generalisasi model yang tinggi, sejalan dengan hasil F1-Score sebelumnya.

IV. KESIMPULAN

A. Kesimpulan

Penelitian ini menyimpulkan bahwa implementasi SVM efektif dalam mengklasifikasikan cabai rawit ke dalam tiga kelas, yaitu matang, mentah, dan rusak. Keberhasilan tersebut didukung oleh kombinasi fitur warna HSV serta tekstur GLCM dan LBP yang mampu merepresentasikan karakteristik masing-masing kelas secara akurat. Segmentasi HSV turut memberikan kontribusi signifikan dalam meningkatkan akurasi model, terutama terhadap nilai F1-Score. Di antara semua jenis kernel yang diuji, kernel RBF menunjukkan performa terbaik. Model SVM yang dilatih menggunakan data latih mencapai F1-Score rata-rata sebesar 84% setelah pelatihan ulang. Hasil ini diperkuat oleh evaluasi *confusion matrix* dan metrik AUC, di mana nilai AUC untuk kelas matang dan mentah masing-masing sebesar 0,956, serta 0,845 untuk kelas rusak, dengan rata-rata AUC sebesar 0,920. Hal ini menunjukkan bahwa model memiliki kemampuan klasifikasi yang baik, meskipun menghadapi ketidakseimbangan data antar kelas.

B. Saran

Penelitian selanjutnya disarankan untuk menambah data kelas rusak dan menerapkan augmentasi berbasis GAN guna mengurangi bias. Penggunaan metode *hybrid* CNN-SVM direkomendasikan untuk menangani kompleksitas tekstur yang tidak terdeteksi oleh GLCM/LBP. Validasi sebaiknya menggunakan data real-time dari lingkungan produksi cabai.

PENGAKUAN

Makalah ini merupakan bagian dari penelitian Tugas Akhir penulis yang disusun sebagai kontribusi akademik dalam bidang visi komputer, dengan judul “Penerapan *Support Vector Machine* Pada Klasifikasi Kondisi Kematangan Cabai Rawit”. Penelitian ini dilakukan secara mandiri tanpa dukungan sponsor dari pihak mana pun, dan disusun sebagai bagian dari pemenuhan syarat akademik di Universitas Buana Perjuangan Karawang.

DAFTAR PUSTAKA

- [1] A. M. Fauzan, “Kementan Pastikan Ketersediaan Cabai Nasional Aman Hingga Akhir Tahun,” *Antara*, Sep. 03, 2024. Accessed: Jul. 27, 2025. [Online]. Available: <Https://Www.Antaranews.Com/Berita/4304459/Kementan-Pastikan-Ketersediaan-Cabai-Nasional-Aman-Hingga-Akhir-Tahun>
- [2] M. Manurung, Dalam *Analisis Kinerja Perdagangan Cabai Merah*, Pusat Data Dan Sistem Informasi Pertanian, Kementan, 2024, 2024, P. 25.
- [3] F. A. Giawa, Z. Siambaton, And T. Haramaini, “E-Monitoring Pada Alat Penyortiran Buah Jeruk Nipis Otomatis Berdasarkan Ukuran Dan Jenis Warna Berbasis Internet Of Things,” *Sudo Jurnal Teknik Informatika*, Vol. 3, No. 1, Pp. 41–59, May 2024, Doi: 10.56211/Sudo.V3i1.406.
- [4] F. O. Dayera, Musa Bundaris Palungan, “G-Tech : Jurnal Teknologi Terapan,” *G-Tech J. Teknol. Terap.*, Vol. 8, No. 1, Pp. 186–195, 2024, [Online]. Available: <Https://Ejournal.Uniramalang.Ac.Id/Index.Php/G-Tech/Article/View/1823/1229>
- [5] Z. E. Fitri, U. Nuhanatika, A. Madjid, And A. M. N. Imron, “Penentuan Tingkat Kematangan Cabe Rawit (Capsicum Frutescens L.) Berdasarkan Gray Level Co-Occurrence Matrix,” *Jurnal Teknologi Informasi Dan Terapan*, Vol. 7, No. 1, Pp. 1–5, Jun. 2020, Doi: 10.25047/Jtit.V7i1.121
- [6] U. Mahdiyah, “Klasifikasi Kualitas Citra Cabai Dengan Menggunakan Algoritma Gradien Boosting,” *Jami: Jurnal Ahli Muda Indonesia*, Vol. 4, No. 1, Pp. 61–69, Jun. 2023, Doi: 10.46510/Jami.V4i1.137.
- [7] Muhammad Rifki Bahrul Ulum, Basuki Rahmat, And Made Hanindia Prami Swari, “Implementasi Metode Cnn Dan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kematangan Tanaman Cabai Rawit,” *Modem : Jurnal Informatika Dan Sains Teknologi.*, Vol. 2, No. 3, Pp. 112–123, Jul. 2024, Doi: 10.62951/Modem.V2i3.131.
- [8] A. Sofyan, “Klasifikasi Tingkat Kematangan Cabai Rawit Dengan Ekstraksi Hsv-Glcm Dan Metode Klasifikasi Support Vector Machines (Svms),” Upn Veteran Jatim Repository. [Online]. Available: <Https://Repository.Upnjatim.Ac.Id/Id/Eprint/29630>
- [9] A. Agung Mujiono, K. Kartini, And E. Yulia Puspaningrum, “Implementasi Model Hybrid Cnn-Svm Pada Klasifikasi Kondisi Kesegaran Daging Ayam,” *Jati (Jurnal Mahasiswa Teknik Informatika)*, Vol. 8, No. 1, Pp. 756–763, Mar. 2024, Doi: 10.36040/Jati.V8i1.8855.
- [10] N. S. Fatimah And S. Agustin, “Klasifikasi Citra Batik Menggunakan Local Binary Pattern (Lbp) Dan Support Vector Machine (Svm),” *Jurnal Algoritma*, Vol. 22, No. 1, Pp. 185–196, May 2025, Doi: 10.33364/Algoritma/V.22-1.2208.
- [11] I. Irma, M. Muchtar, R. Adawiyah, And S. Sarimuddin, “Klasifikasi Tingkat Kematangan Cabai Merah Keriting Menggunakan Svm Multiclass Berdasarkan Ekstraksi Fitur Warna,” *Jurnal Informatika Dan Teknik Elektro Terapan*,

