



Implementasi Metode GLCM, HSV dan K-Nearest Neighbor untuk Klasifikasi Kualitas Jeruk Nipis

Tugas Akhir

Oleh:

Ezi Alfindeo Perangin Angin (4212201086)

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2025**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul: "Implementasi Metode GLCM, HSV dan K-Nearest Neighbor untuk Klasifikasi Kualitas Jeruk Nipis" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri**. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 06 Juli 2026



Ezi Alfindo Perangin angin
NIM: 4212201086

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

Oleh:
Ezi Alfindeo Perangin Angin (4212201086)

Tanggal Sidang: 06 Juli, 2026

Disetujui oleh:



1. Fitriyanti Nakul, S.Pd., M.Si
NIK:118197



1. Adlian Jefiza, S.Pd., M.T
NIK:199220



2. Muhammad Naufal Airlangga
Diputra, S.Pd., M.P.H
NIK:122281



2. Fadli Firdaus S.Pd
NIK:122271

[Implementasi Metode GLCM, HSV dan K-Nearest Neighbor untuk Klasifikasi Kualitas Jeruk Nipis]

Abstrak

Proses sortir jeruk nipis yang masih dilakukan secara manual dengan mengandalkan pengamatan visual manusia cenderung menghasilkan hasil yang subjektif dan tidak konsisten. Hal ini berpotensi menyebabkan tercampurnya buah berkualitas rendah dengan buah berkualitas baik. Penelitian ini bertujuan untuk merancang sistem klasifikasi otomatis kualitas jeruk nipis berbasis citra digital menggunakan kombinasi fitur tekstur Gray Level Co-occurrence Matrix (GLCM) dan fitur warna HSV, serta algoritma klasifikasi K-Nearest Neighbor (KNN). Fitur yang digunakan terdiri dari fitur tekstur (contrast, dissimilarity, homogeneity, energy), fitur warna HSV (Hue, Saturation, Value), serta fitur tambahan berupa shape dan statistical features yang digabungkan menjadi vektor fitur berdimensi 131. Untuk meningkatkan efisiensi, dilakukan reduksi dimensi menggunakan Principal Component Analysis (PCA) dengan mempertahankan 95% variansi data. Model dievaluasi menggunakan akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa kombinasi metode GLCM, HSV, dan KNN mampu mengklasifikasikan kualitas jeruk nipis ke dalam tiga kelas (segar, tidak segar, dan busuk) dengan akurasi mencapai 97.06% pada data uji. Sistem ini diharapkan dapat menjadi solusi praktis untuk meningkatkan konsistensi proses sortir buah.

Kata kunci: Jeruk nipis, GLCM, KNN, klasifikasi kualitas.

[Implementation Of GLCM, HSV and K-Nearest Neighbor Methods for Classifying the Quality of Lime]

Abstract

The lime sorting process, which is still performed manually and relies on human visual inspection, tends to produce subjective and inconsistent results. This can potentially lead to low-quality fruit being mixed with high-quality fruit. This study aims to design an automatic lime quality classification system based on digital images using a combination of Gray Level Co-occurrence Matrix (GLCM) texture features and HSV color features, as well as the K-Nearest Neighbor (KNN) classification algorithm. The features used consist of texture features (contrast, dissimilarity, homogeneity, energy), HSV color features (Hue, Saturation, Value), as well as additional features in the form of shape and statistical features combined into a 131-dimensional feature vector. To improve efficiency, dimensionality reduction was performed using Principal Component Analysis (PCA) while retaining 95% of the data variance. The model was evaluated using accuracy, precision, recall, and F1-score. The results showed that the combination of GLCM, HSV, and KNN methods was able to classify lime quality into three classes (fresh, not fresh, and rotten) with an accuracy of 97.06% on the test data. This system is expected to provide a practical solution for improving the consistency of the fruit sorting process.

Keywords: Lime, GLCM, KNN, quality classification.

Kata Pengantar

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik. Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan jenjang Diploma IV pada Program Studi Teknik Mekatronika, Politeknik Negeri Batam.

Tugas Akhir ini membahas perancangan dan implementasi sistem deteksi serta klasifikasi Kualitas Jeruk Nipis berbasis citra digital dengan memanfaatkan metode GLCM, HSV dan KNN. Penelitian ini bertujuan untuk menghasilkan sistem yang mampu mengidentifikasi dan mengelompokan kualitas jeruk nipis secara otomatis, sehingga dapat membantu proses sortir menjadi lebih efektif, cepat, dan akurat. Sistem yang dikembangkan diharapkan dapat menjadi solusi pendukung dalam menjaga standar kualitas produk pertanian jeruk nipis.

Dalam penyusunan Laporan Tugas Akhir ini, penulis menyadari bahwa keberhasilan penyelesaian laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan YME atas segala nikmat, kesehatan, dan kesempatan yang di berikan.
2. Kedua orang tua dan keluarga, atas doa dan dukungannya.
3. Bapak Diono, S.T., M.Eng, selaku Kepala Program Studi Teknik Mekatronika.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng, selaku Kepala Jurusan Teknik Elektro.
5. Bapak Adlian Jefiza, S.Pd, M.T, dan bapak Fadli Firdaus, S.Pd selaku dosen pembimbing tugas akhir yang telah memberikan arahan dan bimbingan selama proses pengerjaan laporan akhir ini.
6. Seluruh dosen Teknik Elektro, atas ilmu dan pengalaman yang telah diberikan selama proses perkuliahan.
7. Teman-teman Teknik Mekatronika, yang telah memberikan dukungan, bantuan, dan kebersamaan selama proses penyusunan buku laporan akhir ini.

Batam, 6 Juli 2026

Penulis

Daftar Isi

Pernyataan Keaslian Tugas Akhir	i
Lembar Pengesahan	ii
Abstrak	iv
<i>Abstract</i>	v
Kata Pengantar	vi
Daftar Gambar	xi
Daftar Tabel	xii
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Batasan	4
Bab 2 Tinjauan Pustaka	5
2.1 Jeruk Nipis	5
2.2 Pengolahan Citra	5
2.3 Principal Component Analysis (PCA)	5
2.4 HSV	5
2.5 Gray Level Co-occurrence Matrix (GLCM)	5
2.6 K-Nearest Neighbor (KNN)	6
2.7 Train-Test Split	7
2.8 Cross Validation	7
2.9 Evaluasi Performa Model	7
2.10 You Only Look Once (YOLO)	8
2.11 Penelitian Terkait	8
2.12 Analisis dan Gap Penelitian	11
Bab 3 Metodologi	12

3.1 Diagram Alir Penelitian	12
3.1.1 Studi Literatur	12
3.1.2 Persiapan Alat dan Bahan	13
3.2 Pengumpulan Dataset	14
3.3 Preprocessing	15
3.3.1 Load Dataset	15
3.3.2 Split Data	15
3.3.3 Augmentasi	15
3.3.4 Segmentasi	16
3.4 Ekstraksi Fitur	16
3.4.1 Ekstraksi Fitur Tekstur Menggunakan GLCM	16
3.4.2 Ekstraksi Fitur Warna Menggunakan HSV	17
3.4.3 Ekstraksi Fitur Bentuk (<i>Shape Features</i>)	17
3.4.4 Ekstraksi Fitur Statistik (<i>Statistical Features</i>)	18
3.4.5 Penggabungan Seluruh Fitur	18
3.5 Pelatihan Model	18
3.5.1 Normalisasi Fitur	18
3.5.2 Reduksi Dimensi Menggunakan (PCA)	19
3.5.3 Klasifikasi Menggunakan <i>K-Nearest Neighbor</i> (KNN)	19
3.5.4 Evaluasi Model Menggunakan <i>K-Fold Cross Validation</i>	20
3.6 Implementasi Sistem	20
3.6.1 Penyimpanan Model	20
3.6.2 Perancangan <i>Graphical User Interface</i> (GUI)	21
3.6.3 Visualisasi Hasil Klasifikasi	21
3.7 Analisis	21
3.7.1 Analisis Performa Klasifikasi (Confusion Matrix)	21
3.7.2 Analisis Metrik Performa per kelas	21
3.7.3 Pengujian Parameter K dan Fungsi Jarak	21
3.7.4 Analisis K-Fold Cross Validation	22

3.7.5 Analisis Rotasi Citra pada GLCM	22
3.7.6 Analisis Kesalahan Klasifikasi	22
Bab 4 Hasil dan Pembahasan	23
4.1 Pengambilan Data	23
4.2 Augmentasi.....	24
4.2.1 Analisis Pengaruh Augmentasi terhadap akurasi model	24
4.3 Segmentasi	26
4.3.1 Dampak Segmentasi terhadap Akurasi Model	27
4.4 Preprocessing dan Ekstraksi Fitur.....	28
4.4.1 Preprocessing	30
4.4.2 Ekstraksi Fitur Warna (HSV).....	30
4.4.3 Ekstraksi Fitur GLCM.....	33
4.5 Tabel hasil Analisis Metrik Performa per kelas.....	35
4.5.1 Analisis Kesalahan Klasifikasi	36
4.6 Tabel hasil Pengujian Parameter K dan fungsi jarak.....	38
4.6.1 Jarak Euclidean	40
4.6.2 Jarak Manhattan.....	40
4.6.3 Jarak Minkowski ($p=3$)	40
4.7 Tabel hasil Pengujian K-Fold Cross Validation	40
4.8 Analisis Performa Klasifikasi (Confusion Matrix) pada Data Test	41
4.9 Analisis Performa Klasifikasi (Confusion Matrix) pada Data Training ..	43
4.10 Analisis Kesalahan Klasifikasi	44
4.11 Evaluasi.....	45
4.12 Perancangan Graphical User Interface.....	46
4.12.1 Visualisasi Pengujian GUI (<i>Load Image</i>).....	48
4.12.2 Visualisasi Pengujian GUI (<i>Real Time</i>).....	50
4.12.3 Performa Sistem pada Kondisi Conveyor Bergerak	52
Bab 5 Kesimpulan dan Saran	54
5.1 Kesimpulan	54

5.2 Saran	55
Daftar Pustaka	56
Biodata	57
Lampiran	58

Daftar Gambar

Gambar 1 Diagram Alir	12
Gambar 2. Desain Conveyor	14
Gambar 3 Tahapan visualisasi hasil	20
Gambar 4. Distribusi Kelas	23
Gambar 5 hasil pengambilan citra 3 kualitas buah jeruk nipis	24
Gambar 6. Visualisasi augmentasi data	25
Gambar 7 Visualisasi Hasil GrabCut Segmentasi	26
Gambar 8. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk Segar	28
Gambar 9. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk Busuk	29
Gambar 10. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk kurang segar	29
Gambar 11. Confusion Matrix data Test	42
Gambar 12. Confusion Matrix data Training	43
Gambar 13. Confusion Matrix tanpa HSV	45
Gambar 14. Tampilan Gui	46
Gambar 15. Load Image Jeruk Segar	49
Gambar 16. Load Image Jeruk Tidak Segar	49
Gambar 17. Load Image Jeruk Busuk	50
Gambar 18. Real time (Jeruk Segar dan Tidak Segar)	51
Gambar 19. Real Time (Jeruk Tidak Segar dan Busuk)	51
Gambar 20. Real Time (Jeruk Segar dan Busuk)	52

Daftar Tabel

Table 1. Penelitian Terkait.....	8
Table 2. Alat dan Bahan.....	13
Table 3. Teknik augmentasi data.....	24
Table 4. Jumlah data hasil augmentasi.....	24
Table 5. Channel Hue	30
Table 6. Channel saturation	31
Table 7. Channel value	32
Table 8. Parameter kelas segar	34
Table 9. Parameter kelas tidak segar.....	34
Table 10. Parameter kelas busuk	35
Table 11. Metrik performa per kelas.....	36
Table 12. Analisis kesalahan klasifikasi.....	36
Table 13. Hasil pengujian parameter nilai K dan fungsi jarak.....	39
Table 14. Hasil pengujian K-Fold Cross Validation.....	41
Table 16. Fitur Tombol dan Fungsinya	48

Bab 1 Pendahuluan

1.1 Latar Belakang

Kualitas hasil panen merupakan salah satu faktor penting yang menentukan nilai jual dan daya saing produk pertanian. Salah satu komoditas hortikultura yang banyak dimanfaatkan dalam industri makanan, minuman, farmasi, maupun rumah tangga adalah jeruk nipis (*Citrus aurantiifolia*). Agar kualitas produk tetap terjaga hingga sampai ke konsumen, diperlukan proses penyortiran berdasarkan tingkat kesegaran dan kondisi fisik buah. Namun, pada praktiknya proses sortir jeruk nipis masih banyak dilakukan secara manual dengan mengandalkan pengamatan visual manusia. Cara tersebut memiliki beberapa kelemahan, di antaranya bersifat subjektif, bergantung pada pengalaman pekerja, mudah dipengaruhi oleh kelelahan, serta menghasilkan penilaian yang tidak konsisten. Akibatnya, buah dengan kualitas yang berbeda dapat tercampur sehingga menurunkan mutu produk dan berpotensi mengurangi kepercayaan konsumen.

Perkembangan teknologi pengolahan citra digital (*digital image processing*) memberikan alternatif solusi untuk mengatasi permasalahan tersebut. Melalui pengolahan citra, karakteristik visual suatu objek dapat dianalisis secara otomatis sehingga proses klasifikasi kualitas buah menjadi lebih objektif, cepat, dan konsisten. Berbagai penelitian telah memanfaatkan metode ekstraksi fitur dan algoritma pembelajaran mesin untuk melakukan klasifikasi kualitas buah berdasarkan karakteristik visual yang dimiliki.

Metode ekstraksi tekstur yang umum digunakan adalah *Gray Level Co-occurrence Matrix* (GLCM) [1]. Metode ini mampu merepresentasikan karakteristik tekstur permukaan objek melalui hubungan spasial antar piksel sehingga efektif untuk membedakan permukaan buah yang halus, kasar, keriput, maupun memiliki bercak akibat pembusukan. Pada jeruk nipis, perubahan kualitas menyebabkan perubahan tekstur kulit, seperti munculnya keriput, permukaan yang semakin kasar, serta kerusakan jaringan akibat proses pembusukan. Oleh karena itu, fitur tekstur yang dihasilkan GLCM dapat menjadi salah satu indikator penting dalam proses klasifikasi kualitas buah.

Meskipun demikian, penggunaan GLCM saja belum sepenuhnya mampu menggambarkan karakteristik kualitas jeruk nipis. Hal ini disebabkan karena perubahan kualitas buah tidak hanya ditandai oleh perubahan tekstur, tetapi juga perubahan warna kulit buah. Jeruk nipis segar umumnya memiliki warna hijau yang merata, sedangkan buah yang mulai mengalami penurunan kualitas akan berubah menjadi hijau kekuningan hingga kecoklatan ketika mengalami pembusukan. Apabila sistem hanya memanfaatkan informasi tekstur, maka objek yang memiliki tekstur relatif mirip tetapi warna yang berbeda masih berpotensi menghasilkan kesalahan klasifikasi. Kondisi tersebut menunjukkan bahwa

informasi warna memiliki peranan penting sebagai fitur pembeda antar kelas kualitas jeruk nipis.

Berdasarkan kondisi tersebut, penelitian ini menambahkan fitur warna *Hue*, *Saturation*, dan *Value* (HSV) sebagai pelengkap fitur tekstur GLCM. Ruang warna HSV dipilih karena mampu memisahkan informasi warna menjadi tiga komponen utama, yaitu *Hue* yang merepresentasikan jenis warna, *Saturation* yang menunjukkan tingkat kejenuhan warna, dan *Value* yang menggambarkan tingkat kecerahan. Dengan mengombinasikan fitur tekstur GLCM dan fitur warna HSV, sistem diharapkan mampu memperoleh representasi karakteristik buah yang lebih lengkap sehingga proses klasifikasi menjadi lebih akurat dibandingkan apabila hanya menggunakan salah satu jenis fitur saja.

Penelitian klasifikasi kualitas buah yang dilakukan hanya menggunakan citra yang telah dipotong (cropping) atau dipilih secara manual sehingga proses klasifikasi dilakukan pada objek yang sudah diketahui lokasinya. Pendekatan tersebut masih kurang sesuai apabila sistem akan diterapkan pada proses sortir otomatis menggunakan kamera, karena citra yang diperoleh tidak hanya berisi buah, tetapi juga latar belakang dan kemungkinan terdapat lebih dari satu objek. Akibatnya, sistem tidak dapat secara otomatis menentukan lokasi buah yang akan diklasifikasikan.

Oleh karena itu, diperlukan tahap deteksi objek sebelum proses klasifikasi dilakukan. Tahap ini bertujuan untuk menemukan posisi buah jeruk nipis pada citra sehingga proses ekstraksi fitur GLCM dan HSV hanya dilakukan pada area buah. Tanpa proses deteksi objek, fitur yang diekstraksi dapat tercampur dengan informasi latar belakang sehingga berpotensi menurunkan akurasi klasifikasi.

Pada penelitian ini digunakan algoritma *You Only Look Once* (YOLO) sebagai metode deteksi objek karena mampu mendeteksi lokasi objek secara cepat dalam satu kali proses (*single-stage object detection*). Kemampuan tersebut menjadikan YOLO sesuai untuk implementasi pada sistem sortir berbasis conveyor yang memerlukan pemrosesan secara real-time. Hasil deteksi YOLO kemudian digunakan sebagai masukan pada proses ekstraksi fitur GLCM dan HSV sehingga hanya area buah jeruk nipis yang dianalisis.

Setelah proses ekstraksi fitur dilakukan, diperlukan algoritma klasifikasi yang mampu mengelompokkan data berdasarkan kemiripan karakteristik fitur tersebut. Pada penelitian ini digunakan algoritma *K-Nearest Neighbor* (KNN) karena memiliki konsep yang sederhana, mudah diimplementasikan, serta terbukti memberikan performa yang baik pada dataset berukuran kecil hingga menengah. Selain itu, performa KNN sangat dipengaruhi oleh pemilihan nilai K dan fungsi jarak yang digunakan dalam mengukur tingkat kemiripan antar data. Oleh sebab itu, penelitian ini juga melakukan pengujian beberapa nilai K serta membandingkan fungsi jarak *Euclidean*, *Manhattan*, dan *Minkowski* untuk memperoleh konfigurasi klasifikasi yang memberikan performa terbaik.

Beberapa penelitian sebelumnya telah menunjukkan efektivitas kombinasi metode GLCM dan KNN dalam klasifikasi buah dan menghasilkan akurasi yang tinggi. (C. Suryanti 2023) [2] Berhasil mengklasifikasikan kualitas buah apel berdasarkan warna dan bentuk dengan akurasi tinggi menggunakan KNN. Sementara itu (Wiastopo 2024) [3] menerapkan kombinasi KNN dan GLCM untuk mendeteksi kesegaran ikan kembung dengan akurasi, presisi, dan recall sebesar 96%. Namun hasil kajian terdahulu masih ditemukan beberapa keterbatasan, di antaranya penggunaan fitur yang hanya berfokus pada tekstur atau warna saja, belum adanya pengujian variasi fungsi jarak pada algoritma KNN, serta implementasi yang masih terbatas pada klasifikasi citra statis tanpa dukungan sistem deteksi objek secara real-time. Oleh karena itu, penelitian ini mengusulkan sistem klasifikasi kualitas jeruk nipis yang mengombinasikan fitur tekstur GLCM dan fitur warna HSV, menggunakan algoritma K-Nearest Neighbor sebagai metode klasifikasi, melakukan evaluasi terhadap beberapa fungsi jarak dan nilai K, serta mengintegrasikan algoritma YOLO untuk mendukung proses deteksi objek secara real-time. Sistem kemudian diimplementasikan dalam bentuk *Graphical User Interface* (GUI) sehingga mampu melakukan deteksi dan klasifikasi kualitas jeruk nipis secara otomatis dan lebih siap diterapkan pada sistem sortir buah.

1.2 Rumusan Masalah

1. Bagaimana mengimplementasikan *Gray Level Co-occurrence Matrix* (GLCM) untuk mengekstraksi fitur tekstur citra buah jeruk nipis?
2. Bagaimana mengimplementasikan algoritma *K-Nearest Neighbor* (KNN) untuk mengklasifikasikan kualitas buah jeruk nipis ke dalam tiga kategori, yaitu segar, tidak segar, dan busuk?
3. Bagaimana menentukan kombinasi parameter nilai K dan fungsi jarak pada algoritma KNN yang menghasilkan performa klasifikasi terbaik?
4. Bagaimana merancang dan mengimplementasikan sistem klasifikasi berbasis GUI yang mampu menampilkan hasil klasifikasi secara visual dan *real-time*

1.3 Tujuan

1. Mengimplementasikan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk mengekstraksi fitur tekstur dari citra jeruk nipis.
2. Mengimplementasikan algoritma *K-Nearest Neighbor* untuk mengklasifikasikan kualitas buah ke dalam tiga kelas yaitu, Segar, tidak segar, dan busuk.
3. Menentukan kombinasi parameter K dan fungsi jarak terbaik pada algoritma KNN melalui pengujian dan evaluasi performa.
4. Mengembangkan sistem klasifikasi berbasis GUI yang mampu menampilkan hasil klasifikasi secara visual dan real-time.

1.4 Manfaat

1. Memberikan solusi praktis berupa sistem kualifikasi buah jeruk nipis.
2. Meningkatkan pemahaman mahasiswa terhadap Agroteknologi.
3. Menjadi dasar pengembangan sistem klasifikasi kualitas buah lainnya.

1.5 Batasan

1. Klasifikasi kualitas dibatasi pada tiga kelas: Segar, tidak segar, dan busuk, berdasarkan pengamatan visual.
2. Fitur citra yang digunakan hanya berasal dari hasil ekstraksi tekstur menggunakan metode GLCM dan fitur warna (HSV histogram) atau fitur bentuk.
3. Algoritma klasifikasi yang digunakan terbatas pada *K-Nearest Neighbor* (KNN).
4. Visualisasi dilakukan melalui antarmuka digital, dan implementasi ke alat sortir sederhana.

Bab 2 Tinjauan Pustaka

2.1 Jeruk Nipis

Jeruk nipis (*Citrus aurantiifolia* (Christm) Swingle) adalah salah satu kelompok jeruk yang memiliki ukuran kecil, kulit tipis, serta rasa yang sangat asam. Buah ini banyak digunakan dalam industri makanan, minuman, kesehatan, hingga produk pembersih karena kandungan vitamin C, antioksidan, dan asam sitrat yang tinggi. Sehingga menjadi salah satu buah yang sangat diminati untuk berbagai keperluan komersial maupun Kesehatan[4].

2.2 Pengolahan Citra

Pengolahan citra merupakan teknik untuk meningkatkan kualitas citra agar lebih mudah dikenali dan dianalisis oleh manusia maupun komputer. Proses ini bertujuan mengurangi gangguan seperti noise, blur, serta memperbaiki ketajaman warna. Baik input maupun output berupa citra, namun hasilnya memiliki kualitas yang lebih baik untuk keperluan identifikasi atau klasifikasi[5][6].

2.3 Principal Component Analysis (PCA)

Principal Component Analysis adalah suatu metode untuk mengidentifikasi suatu pola dan menonjolkan adanya suatu perbedaan ataupun kesamaan dalam suatu kumpulan data. Metode ini biasa digunakan untuk mereduksi dimensi data, menjadi bentuk yang berada pada bidang nilai yang berbeda. Metode ini bekerja dengan cara menghitung covariance matrix dari data, dan kemudian mencari eigenvectors dan eigenvalues[7].

2.4 HSV

Model HSV adalah model yang mendefinisikan warna dalam terminologi *Hue*, *Saturation*, dan *Value*. Hue menyatakan warna sebenarnya, saturation menyatakan tingkat kemurnian suatu warna, yakni dengan mengidentifikasi seberapa banyak warna putih diberikan warna, sedangkan value merupakan atribut yang menyatakan banyaknya cahaya yang diterima oleh warna. Sehingga perhitungan HSV cocok untuk diterapkan pada klasifikasi tingkat kualitas buah jeruk nipis melalui ekstraksi warna[8].

2.5 Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) adalah metode ekstraksi fitur dalam pengolahan citra yang digunakan untuk menganalisis tekstur suatu gambar. Metode ini bekerja dengan menghitung frekuensi kemunculan pasangan piksel

dengan nilai keabuan tertentu pada jarak dan arah dalam citra. Hasil perhitungan ini membentuk sebuah matriks yang disebut matriks *Co-occurrence*.

GLCM menggunakan pendekatan statistik untuk mengevaluasi hubungan spasial antar piksel, sehingga dapat menggambarkan pola tekstur yang tidak dapat dilihat hanya dari warna atau bentuk saja. Dari matriks GLCM, diperoleh beberapa parameter fitur tekstur penting, di antaranya *Contrast*, *Correlation*, *Homogeneity*, dan *Energi*. [9].

Fitur-fitur ini dapat digunakan untuk mengklasifikasikan objek berdasarkan tekstur permukaannya, seperti dalam penentuan kualitas buah jeruk nipis

2.6 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah algoritma *supervised learning* yang digunakan untuk klasifikasi berdasarkan kedekatan data baru dengan data pelatihan. KNN menghitung jarak antara data uji dan seluruh data latih, kemudian menetapkan kelas berdasarkan mayoritas dari K tetangga terdekat [10]. Metode ini dikenal sederhana, tidak memerlukan pelatihan model secara eksplisit, dan efektif dalam klasifikasi dengan data set berukuran kecil hingga menengah. Algoritma KNN mengukur kedekatan antara data uji dan data latih menggunakan fungsi jarak. Beberapa jenis pengukuran jarak yang umum digunakan dalam KNN antara lain:

1. *Euclidean Distance*

Pengukuran jarak ini dilakukan dengan menghitung jarak garis lurus antar data yang sudah di latih dengan data baru, semakin dekat jaraknya maka semakin mirip data jeruk baru tersebut dengan data yang sudah di latih sebelumnya.

$$d(p, q) = \sum_{i=1}^n (p_i - q_i)^2$$

2. *Manhattan Distance*

Pengukuran jarak ini dilakukan dengan menghitung jarak antara data yang sudah di latih terhadap data baru secara horizontal + vertikal. Sama seperti *euclidean* semakin dekat nilai jarak antara kedua data maka jeruk akan semakin mirip.

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

3. *Minkowski Distance*

Pengukuran jarak ini merupakan metrik umum yang dapat disesuaikan untuk berbagai jenis perhitungan jarak dalam klasifikasi. *Minkowski distance* menggunakan parameter p untuk menentukan cara

perhitungan jarak antara data jeruk nipis yang sudah di latih dengan data jeruk baru yang akan di klasifikasi

$$d(p, q) = \left(\sum_{i=1}^n |x_i - y_i|^r \right)^{1/r}$$

Pengujian beberapa jenis jarak ini penting untuk memilih fungsi jarak paling sesuai dengan karakteristik data [11].

2.7 Train-Test Split

Train-Test Split adalah metode pembagian data set menjadi dua bagian utama: data latih dan data uji. Biasanya sekitar 70-80% data yang digunakan untuk melatih model dan sisanya untuk menguji performa. Meskipun metode ini sederhana, hasil evaluasi model sangat bergantung pada cara pembagian data tersebut. Sehingga metode ini dapat menghasilkan bias jika data tidak dibagi secara representative [12].

2.8 Cross Validation

Cross-validation merupakan metode evaluasi performa model yang lebih menyeluruh dibandingkan *train-test split*. Salah satu teknik yang paling umum digunakan adalah *K-Fold Cross Validation*. Teknik ini membagi data set ke dalam K sub-bagian atau "fold". Model kemudian di latih menggunakan *K-1 Fold* dan diuji pada satu *fold* yang tersisa. Proses ini diulang sebanyak K kali, sehingga setiap *fold* berperan sebagai data uji satu kali. Hasil dari seluruh iterasi kemudian dirata-ratakan untuk memberikan estimasi performa model yang lebih akurat dan mengurangi risiko bias[13].

Cross-validation berguna untuk:

1. Meminimalkan bias evaluasi yang mungkin muncul dari pembagian data yang tidak merata.
2. Memastikan model tidak hanya cocok untuk data tertentu (*overfitting*), tetapi juga dapat digunakan secara umum (*generalisasi*).

2.9 Evaluasi Performa Model

Evaluasi performa model bertujuan untuk mengukur seberapa baik algoritma klasifikasi bekerja. Salah satu metode evaluasi yang umum digunakan adalah *Confusion Matrix*, yaitu tabel evaluasi yang memperlihatkan perbandingan antara label aktual dan hasil prediksi model klasifikasi. Metrik evaluasi yang digunakan meliputi:

1. *Accuracy* : Proporsi prediksi benar terhadap seluruh prediksi.

$$Accuracy = \frac{TP}{TP + TN + FP + FN} \times 100$$

2. **Precision:** Seberapa tepat model memprediksi suatu kelas.

$$Precision = \frac{TP}{TP + FP} \times 100$$

3. **Recall:** Seberapa banyak data relevan yang berhasil diprediksi.

$$Recall = \frac{TP}{TP + FN} \times 100$$

4. **F1-Score:** Rata-rata harmonis antara presisi dan recall.

$$F1 - Score = \frac{2 \times Presisi \times recall}{Presisi \times Recall} \times 100$$

2.10 You Only Look Once (YOLO)

YOLO (You Look Only Once) adalah sebuah sistem untuk melakukan pendeteksian objek, yang ditargetkan untuk pemrosesan secara *real-time*. YOLO membingkai pendeteksian objek sebagai masalah regresi tunggal dimana dari piksel gambar langsung ke kotak pembatas (*bounding box*) spasial yang terpisah dan probabilitas kelas yang terkait. YOLO melakukan pendeteksian dan pengenalan objek dengan sebuah jaringan syaraf tunggal (single memprediksi neural kotak-kotak network), pembatas yang dan probabilitas kelas secara langsung dalam satu evaluasi[14].

2.11 Penelitian Terkait

Berikut adalah beberapa penelitian terkait terdahulu yang relevan dengan topik ini:

Table 1. Penelitian Terkait

N o	Tahu n	Penulis	Judul	Metode	Kelemahan
1	2021	Siti Raysyah, Veri Arinal, Dadang Iskandar Mulyana	Klasifikasi Tingkat Kematangan Buah Kopi Berdasarkan Deteksi Warna Menggunakan KNN Dan PCA.	KNN & PCA	Tidak ada pengujian variasi kondisi

2	2023	Jauhari Maulani, Mayang Sari	Komparasi Metode K-Nearest Neighbor (KNN) Dengan Support Vector Machine (Svm) Terhadap Tingkat Akurasi Klasifikasi Kualitas Air.	SVM & KNN	Tidak menjelaskan penentuan parameter (K pada KNN dan Kernel pada SVM)
3	2025	Mutiara Mega Amelia, Hani Harafani, Muhammad Rafa Maarif, Bintang Maulana Fazrin.	Meningkatkan Akurasi KNN Menggunakan Metode Particle Swarm Optimization pada Klasifikasi Kualitas Buah Apel.	KNN & PSO	Tidak ada pengujian variasi kondisi
4	2024	Yuza Reswan, Rozali Toyib, Harry Witriyono, Ani Anggraini	Klasifikasi Tingkat Kematangan Buah Nanas Berdasarkan Fitur Warna Menggunakan Metode K-Nearest Neighbor.	KNN	Jumlah Data set yang terbatas
5	2021	I Gede Wirayudhana	Klasifikasi Mutu Buah Jambu Biji Getah Merah Berdasarkan Tekstur	GLCM & KNN	akurasi rendah

			Menggunakan Grey Level Co-Occurrence Matrix (GLCM) dengan Klasifikasi KNN.		
6	2022	Rifqi Syahrul Ilhamy, Ucta Pradema Sanjaya	Algoritma K-Nearest Neighbors (KNN) untuk Klasifikasi Citra Buah Pisang dengan Ekstraksi Ciri Gray Level Co-Occurrence.	KNN & GLCM	Akurasi tidak stabil
7	2022	Frencis Matheos Sarimole, Achmad Syaeful	Classification of Durian Types Using Features Extraction Gray Level Co-Occurrence Matrix (GLCM)	GLCM & KNN	Tidak ada perbandingan metode
8	2020	Listya & Rokhman	Classification of Tangerine (Citrus Reticulata Blanco) Quality Using Combination of GLCM, HSV, and K-NN.	GLCM + HSV + KNN	Data set kecil akurasi tidak stabil
9	2023	Wenjia Geng	Attention-Aware CNN for Fruit Image Classification	BasicCNN + CBAM	Akurasi pada citra nyata rendah

10	2024	Nisa Fatimah Indrianti, Ade Kania Ningsih, Ridwan Ilyas	IMPLEMENTASI DATA MINING UNTUK KLASIFIKASI PENYAKIT GAGAL GINJAL KRONIS MENGUNAKAN METODE K- NEAREST NEIGHBOR	KNN	Dataset Terbatas
----	------	--	--	-----	---------------------

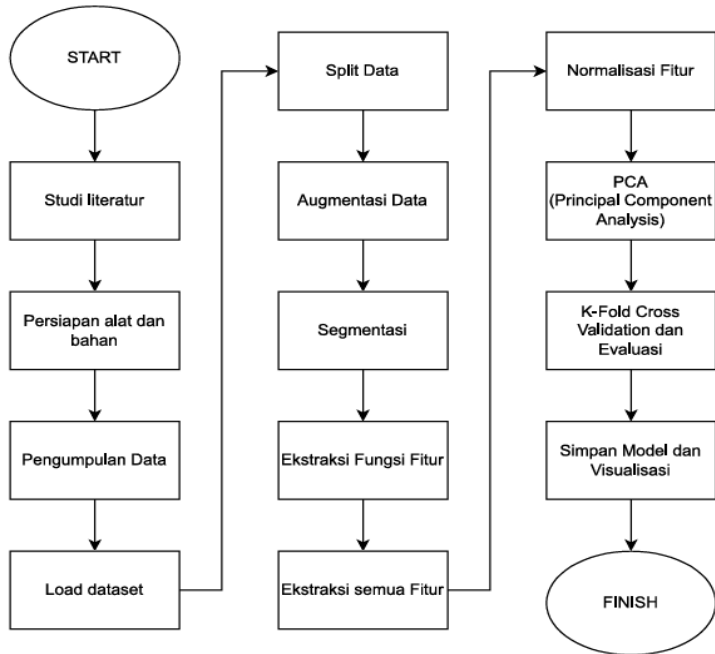
2.12 Analisis dan Gap Penelitian

1. Data set kecil: beberapa peneliti, sehingga hasil model kurang general.
 2. Tidak adanya pengujian fungsi jarak: penelitian sebelumnya jarang membandingkan fungsi jarak.
 3. Evaluasi terbatas: banyak peneliti hanya mengukur akurasi tanpa menyertakan *precision*, *recall*, atau *F1-score*.
 4. Kurangnya visualisasi sistem: tidak semua peneliti menampilkan hasil klasifikasi dalam bentuk antarmuka atau visualisasi grafis.
- Tugas akhir ini berusaha mengatasi beberapa gap tersebut dengan:
1. Memperbanyak dataset dengan melakukan augmentasi.
 2. Menguji performa KNN dengan berbagai fungsi jarak.
 3. Melakukan analisis menyeluruh menggunakan *accuracy*, *precision*, *recall*, dan *f1-score*.
 4. Menyediakan visualisasi hasil klasifikasi untuk penggunaan akhir.
- Dengan demikian, tugas akhir ini tidak hanya mengulang metode yang sudah ada, tetapi juga memberikan kontribusi baru dalam konteks penerapan teknologi pengolahan citra digital di lingkungan petani local.

Bab 3 Metodologi

3.1 Diagram Alir Penelitian

Tugas Akhir terdiri dari beberapa tahapan utama yang digambarkan dalam bentuk diagram alir. Diagram ini menunjukkan alur proses mulai dari studi literatur hingga evaluasi performa dan visualisasi hasil klasifikasi.



Gambar 1 Diagram Alir

3.1.1 Studi Literatur

Tahap ini dilakukan untuk mengumpulkan referensi, teori, dan metode yang relevan dengan pengolahan citra digital, fitur tekstur GLCM, fitur warna HSV, PCA (Principal Component Analysis) serta klasifikasi menggunakan (KNN) *K-Nearest Neighbor*. Informasi ini akan menjadi dasar utama untuk merancang dan mengimplementasikan sistem klasifikasi kualitas buah jeruk nipis.

3.1.2 Persiapan Alat dan Bahan

Table 2. Alat dan Bahan

No	Alat/Bahan	Fungsi
1	Webcam C-270 HD	Mengambil gambar buah jeruk nipis
2	Laptop	Membuat program
3	Software Visual Studio Code	Proses ekstraksi fitur, dan klasifikasi
4	Dataset jeruk nipis	Data citra digunakan sebagai bahan pelatihan dan pengujian
5	Conveyor	Implementasi model pada alat sortir

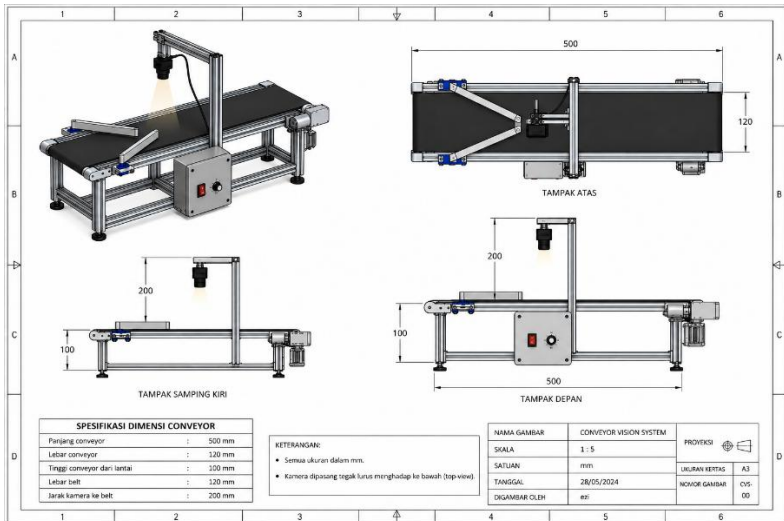
Pada tahap ini dilakukan persiapan alat dan bahan yang digunakan dalam proses penelitian. Persiapan ini bertujuan untuk memastikan seluruh kebutuhan dalam pengambilan data, pengolahan citra, serta implementasi sistem klasifikasi dapat berjalan dengan baik dan terstruktur.

Alat dan bahan yang digunakan dalam penelitian ini disajikan pada Tabel 3.1.2. Setiap komponen memiliki fungsi masing-masing yang mendukung proses penelitian, mulai dari pengambilan citra, pengolahan data, hingga implementasi sistem pada alat sortir.

Webcam digunakan sebagai perangkat utama untuk mengambil citra buah jeruk nipis dengan resolusi yang cukup untuk kebutuhan pengolahan citra. Laptop berfungsi sebagai media utama dalam pengolahan data dan pengembangan program. Perangkat lunak Visual Studio Code digunakan untuk melakukan proses pemrograman, termasuk ekstraksi fitur dan klasifikasi menggunakan algoritma KNN.

Dataset jeruk nipis merupakan bahan utama dalam penelitian ini yang digunakan sebagai data latih dan data uji. Dataset ini terdiri dari citra buah jeruk nipis dengan tiga kategori kualitas, yaitu segar, tidak segar, dan busuk.

Selain itu, digunakan juga conveyor sebagai media implementasi sistem klasifikasi secara sederhana. Conveyor berfungsi sebagai alat bantu untuk mensimulasikan proses sortir buah secara otomatis, di mana buah yang telah diklasifikasikan dapat diarahkan sesuai dengan kategori kualitasnya.



Gambar 2. Desain Conveyor

Gambar 2 menunjukkan desain conveyor yang digunakan sebagai media implementasi sistem klasifikasi kualitas jeruk nipis. Conveyor memiliki panjang 50 cm, lebar 20 cm, dan tinggi 10 cm dari permukaan lantai. Belt conveyor yang digunakan memiliki lebar 12 cm, sehingga mampu menopang buah jeruk nipis selama proses perpindahan. Sistem akuisisi citra menggunakan webcam yang dipasang pada rangka di atas conveyor dengan jarak 20 cm dari permukaan belt. Kamera diposisikan tegak lurus menghadap ke bawah (*top-view*) agar seluruh permukaan buah dapat tertangkap secara optimal. Dimensi dan konfigurasi tersebut dipilih untuk menghasilkan proses pengambilan citra yang stabil, meminimalkan perubahan perspektif, serta mendukung proses deteksi dan klasifikasi kualitas jeruk nipis secara real-time. Penempatan kamera pada ketinggian 20 cm juga memberikan bidang pandang yang cukup untuk menangkap seluruh objek tanpa mengurangi detail citra yang diperlukan pada proses ekstraksi fitur dan klasifikasi.

3.2 Pengumpulan Dataset

Pada tahap pengumpulan citra, data gambar buah jeruk nipis diambil menggunakan kamera beresolusi tinggi. Pengambilan gambar dilakukan dengan latar belakang putih. Kemudian di klasifikasikan secara manual kedalam tiga kelas, total dataset yang dikumpulkan sebanyak 450 data. Dengan masing masing 150 data untuk setiap kelas

1. Buah Jeruk Nipis Segar
Buah Jeruk Nipis Segar memiliki permukaan kulit halus, berwarna hijau, tidak terdapat bercak atau kerusakan fisik, dan tekstur buah yang padat saat ditekan.
2. Buah Jeruk nipis Tidak Segar
Buah Jeruk Nipis Busuk memiliki permukaan kulit yang tampak kering, warna tidak merata terdapat warna kuning pucat dengan bercak kehijauan.
3. Buah Jeruk Nipis Busuk
Buah Jeruk Nipis Busuk memiliki permukaan kulit yang keropos, terdapat kerusakan fisik, permukaan buah menjadi lembek, terdapat keriput pada permukaan buah, dan berwarna kecoklatan.

3.3 Preprocessing

3.3.1 Load Dataset

Load dataset adalah tahapan pertama yang dilakukan yang bertujuan untuk mengubah data gambar yang tersimpan dalam struktur folder menjadi data array agar bisa masuk ke dalam memori computer. Pada tahapan ini setiap kelas diberi label angka (0-2) sesuai kelasnya, sehingga model bisa belajar dan membedakan mana kelas segar, tidak segar dan busuk.

3.3.2 Split Data

Pembagian dataset menjadi data latih (training set) dan data uji (testing set) dengan rasio 70% data latih dan 30% data uji. Pemilihan rasio 70:30 didasarkan pada pertimbangan berikut:

1. Rasio 70:30 adalah rasio yang paling umum yang digunakan dalam machine learning, Ketika dataset sudah seimbang.
2. Dataset setiap kelas memiliki jumlah sample yang sama, yaitu 150 gambar per kelas (total 450 gambar).

Dengan demikian, rasio 70:30 menghasilkan 315 gambar latih dan 135 gambar uji, yang memberikan jumlah data uji yang cukup besar untuk evaluasi model yang stabil dan representatif.

3.3.3 Augmentasi

Teknik augmentasi adalah manipulasi gambar untuk memperbesar data set, mengatasi potensi overfitting pada model KNN dan meningkatkan robustitas terhadap variasi, orientasi, dan sudut pengambilan citra jeruk nipis di dunia nyata. Teknik augmentasi yang digunakan meliputi rotasi (90°, 180°, dan 270°), flipping horizontal dan vertikal, serta perubahan intensitas cahaya (*brightness*

adjustment). Augmentasi ini dilakukan pada data train saja, bertujuan untuk membuat model lebih konsisten terhadap variasi nyata di lapangan.

3.3.4 Segmentasi

Tahapan segmentasi objek jeruk nipis dilakukan untuk memisahkan foreground (jeruk) dan background secara akurat meskipun terdapat variasi pencahayaan dan keberadaan objek pengganggu. Proses ini diinisialisasi dengan persegi panjang otomatis dan dilanjutkan dengan operasi morfologi closing untuk menghilangkan noise dan lubang kecil pada hasil segmentasi. Tahapan ini sangat krusial karena memastikan bahwa fitur warna, tekstur, dan bentuk yang diekstraksi benar-benar berasal dari objek jeruk nipis, sehingga meningkatkan akurasi klasifikasi secara signifikan.

3.4 Ekstraksi Fitur

3.4.1 Ekstraksi Fitur Tekstur Menggunakan GLCM

Setelah citra melalui tahap preprocessing dan segmentasi, langkah selanjutnya adalah melakukan ekstraksi fitur tekstur menggunakan metode Gray Level Co-occurrence Matrix (GLCM). Metode ini digunakan untuk merepresentasikan karakteristik tekstur permukaan kulit jeruk nipis berdasarkan hubungan spasial antar piksel pada citra grayscale. Fitur tekstur dipilih karena perubahan kualitas jeruk nipis umumnya ditandai oleh perubahan permukaan kulit, seperti munculnya keriput, bercak, serta tekstur yang semakin kasar akibat proses pembusukan.

Sebelum proses ekstraksi dilakukan, citra hasil segmentasi dikonversi terlebih dahulu dari format RGB menjadi grayscale. Konversi ini diperlukan karena perhitungan GLCM menggunakan nilai intensitas keabuan setiap piksel sebagai dasar pembentukan matriks ko-okurensi. Selanjutnya, matriks GLCM dibentuk dengan menghitung frekuensi kemunculan pasangan piksel yang memiliki nilai keabuan tertentu pada jarak (*distance*) dan arah (*angle*) tertentu.

Pada penelitian ini digunakan tiga nilai jarak, yaitu 1, 2, dan 3 piksel, serta empat orientasi sudut, yaitu 0°, 45°, 90°, dan 135°. Penggunaan beberapa jarak dan orientasi bertujuan agar informasi tekstur dapat direpresentasikan secara lebih menyeluruh, karena pola tekstur permukaan jeruk nipis dapat muncul pada berbagai arah dan skala.

Nilai setiap parameter dihitung pada seluruh kombinasi jarak dan orientasi, kemudian dirata-ratakan sehingga diperoleh satu nilai representatif untuk masing-masing fitur tekstur. Proses ini dilakukan pada setiap citra jeruk nipis sehingga dihasilkan enam fitur tekstur yang selanjutnya digabungkan dengan fitur warna, bentuk, dan statistik sebelum memasuki tahap normalisasi dan reduksi dimensi menggunakan Principal Component Analysis (PCA).

3.4.2 Ekstraksi Fitur Warna Menggunakan HSV

Selain fitur tekstur menggunakan Gray Level Co-occurrence Matrix (GLCM), penelitian ini juga memanfaatkan fitur warna menggunakan ruang warna Hue, Saturation, Value (HSV). Penggunaan fitur warna bertujuan untuk melengkapi informasi tekstur karena perubahan kualitas jeruk nipis tidak hanya ditandai oleh perubahan permukaan kulit, tetapi juga oleh perubahan warna. Jeruk nipis segar umumnya memiliki warna hijau yang merata, sedangkan buah yang mulai mengalami penurunan kualitas akan mengalami perubahan warna menjadi hijau kekuningan hingga kecokelatan akibat proses pematangan dan pembusukan. Oleh karena itu, fitur warna berperan penting dalam membedakan setiap kelas kualitas jeruk nipis.

Pada tahap ini, citra hasil segmentasi terlebih dahulu dikonversi dari ruang warna RGB ke ruang warna HSV.

Setelah proses konversi selesai, masing-masing kanal HSV diproses secara terpisah untuk memperoleh distribusi warna menggunakan histogram. Pada penelitian ini digunakan 16 bin untuk setiap kanal *Hue*, *Saturation*, dan *Value* sehingga total diperoleh 48 fitur histogram. Histogram kemudian dinormalisasi agar distribusi nilai tidak dipengaruhi oleh ukuran objek maupun jumlah piksel pada citra.

Selain histogram warna, penelitian ini juga menghitung fitur statistik pada setiap kanal HSV yang terdiri atas nilai rata-rata (*mean*) dan simpangan baku (*standard deviation*). Fitur statistik digunakan untuk menggambarkan kecenderungan distribusi warna dan tingkat variasinya pada setiap kanal. Dengan tiga kanal warna, diperoleh enam fitur statistik tambahan yang melengkapi representasi warna citra.

Seluruh fitur histogram dan fitur statistik kemudian digabungkan menjadi satu vektor fitur warna berdimensi 54 fitur, yang terdiri atas 48 fitur histogram dan 6 fitur statistik. Vektor fitur ini selanjutnya dikombinasikan dengan fitur tekstur GLCM, fitur bentuk (*shape features*), dan fitur statistik lainnya untuk membentuk representasi karakteristik jeruk nipis secara lebih lengkap sebelum memasuki tahap normalisasi, reduksi dimensi menggunakan Principal Component Analysis (PCA), dan proses klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN).

3.4.3 Ekstraksi Fitur Bentuk (*Shape Features*)

Selain memanfaatkan fitur tekstur dan warna, penelitian ini juga menggunakan fitur bentuk (*shape features*) untuk menggambarkan karakteristik geometris objek jeruk nipis. Penambahan fitur bentuk bertujuan untuk melengkapi informasi yang diperoleh dari GLCM dan HSV, karena perubahan kualitas jeruk nipis tidak hanya memengaruhi tekstur dan warna, tetapi juga dapat menyebabkan perubahan bentuk fisik, seperti penyusutan, keriput, maupun perubahan kontur akibat proses pembusukan.

Ekstraksi fitur bentuk dilakukan pada citra hasil segmentasi sehingga hanya area objek jeruk nipis yang dianalisis. Tahap pertama adalah memperoleh kontur objek (contour) menggunakan metode deteksi kontur pada citra biner hasil segmentasi. Kontur yang memiliki luas terbesar dipilih sebagai representasi objek jeruk nipis. Selanjutnya, berbagai parameter geometris dihitung berdasarkan kontur tersebut untuk menghasilkan representasi numerik bentuk objek.

3.4.4 Ekstraksi Fitur Statistik (*Statistical Features*)

Selain fitur tekstur, warna, dan bentuk, penelitian ini juga memanfaatkan fitur statistik sebagai fitur pelengkap untuk menggambarkan karakteristik distribusi intensitas piksel pada citra jeruk nipis. Fitur statistik digunakan karena mampu memberikan informasi global mengenai penyebaran nilai piksel yang tidak secara langsung direpresentasikan oleh fitur GLCM, HSV, maupun fitur bentuk. Dengan demikian, informasi yang diperoleh dari citra menjadi lebih lengkap sehingga diharapkan dapat meningkatkan kemampuan model dalam membedakan kualitas jeruk nipis.

Ekstraksi fitur statistik dilakukan pada citra hasil segmentasi sehingga seluruh perhitungan hanya berasal dari area objek jeruk nipis. Setiap citra dianalisis berdasarkan distribusi nilai intensitas piksel untuk memperoleh beberapa parameter statistik yang merepresentasikan karakteristik citra secara keseluruhan.

3.4.5 Penggabungan Seluruh Fitur

Setelah proses ekstraksi fitur selesai dilakukan, seluruh fitur yang diperoleh dari metode GLCM, HSV, shape features, dan statistical features digabungkan menjadi satu vektor fitur yang merepresentasikan karakteristik setiap citra jeruk nipis. Proses ini bertujuan agar informasi tekstur, warna, bentuk, dan distribusi intensitas piksel dapat dimanfaatkan secara bersamaan dalam proses klasifikasi.

Pada penelitian ini, diperoleh 6 fitur GLCM, 54 fitur HSV, 13 fitur bentuk, dan 10 fitur statistik, sehingga total terdapat 83 fitur untuk setiap citra. Vektor fitur hasil penggabungan kemudian digunakan sebagai masukan pada tahap normalisasi menggunakan StandardScaler, sebelum dilakukan reduksi dimensi menggunakan Principal Component Analysis (PCA) dan proses klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN).

3.5 Pelatihan Model

3.5.1 Normalisasi Fitur

Setelah semua fitur digabungkan menjadi vector 131 dimensi, dilakukan normalisasi fitur. Normalisasi ini sangat penting karena ketiga domain fitur memiliki skala dan rentang nilai yang sangat berbeda. Tanpa normalisasi, fitur

dengan skala besar (seperti area) akan mendominasi perhitungan jarak pada algoritma KNN, sehingga fitur penting lainnya (seperti tekstur atau warna) menjadi tidak berpengaruh. Dengan normalisasi fitur, semua 131 fitur memiliki bobot yang seimbang sehingga klasifikasi menjadi lebih akurat dan stabil. Hasil normalisasi ini akan menjadi input bagi tahapan reduksi dimensi menggunakan Principal Component Analysis (PCA) dan pelatihan model KNN.

3.5.2 Reduksi Dimensi Menggunakan (PCA)

Setelah normalisasi, dilakukan reduksi dimensi menggunakan Principal Component Analysis (PCA), Metode ini digunakan untuk menyederhanakan jumlah fitur tanpa mengurangi informasi penting yang terdapat pada data dengan target mempertahankan 95% variansi data asli. PCA dipilih karena 131 fitur yang diekstraksi mengandung banyak redundansi dan korelasi antar fitur (misalnya: kontras, dan dissimilarity GLCM, mean, variance statistik, dll). PCA tidak menghapus informasi, melainkan mentransformasi fitur menjadi ruang baru yang lebih ringkas namun tetap mewakili karakteristik citra secara optimal. Dalam sistem ini PCA digunakan setelah seluruh fitur di ekstrak. Hal ini bertujuan untuk meningkatkan performa model dengan memberikan input yang lebih bersih dan terstruktur ke algoritma klasifikasi.

Nilai 95% dipilih karena merupakan nilai yang umum digunakan pada reduksi dimensi menggunakan PCA. Persentase ini mampu mempertahankan sebagian besar informasi penting, sekaligus mengurangi redundansi antar fitur sehingga proses klasifikasi menjadi lebih efisien.

3.5.3 Klasifikasi Menggunakan *K-Nearest Neighbor* (KNN)

tahap selanjutnya adalah melakukan klasifikasi untuk menentukan kualitas buah jeruk nipis ke dalam tiga kategori, yaitu segar, tidak segar, dan busuk. Proses klasifikasi dilakukan menggunakan algoritma *K-Nearest Neighbor* (KNN). model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1 score untuk membandingkan performa klasifikasi berdasarkan fungsi jarak yang digunakan. Nilai K merupakan parameter penting dalam algoritma KNN. Pemilihan nilai K yang terlalu kecil dapat menyebabkan model terlalu sensitif terhadap noise (overfitting), sedangkan nilai K yang terlalu besar dapat membuat model kehilangan detail penting (underfitting). Oleh karena itu, Untuk mendapatkan nilai K yang optimal dan konsisten pada data uji, dilakukan eksperimen dengan beberapa nilai K (K = 3, 5, 7, 9 dan 11), K terbaik akan 15 ditentukan dengan evaluasi akurasi tertinggi, dan gunakan confusion matrix sebagai dasar evaluasi.

untuk mengevaluasi kinerja sistem klasifikasi kualitas buah jeruk nipis yang dikembangkan menggunakan metode Gray Level Co-occurrence Matrix (GLCM) untuk ekstraksi fitur dan *K-Nearest Neighbor* (KNN) untuk sistem klasifikasi.

1. *Accuracy*: Persentase prediksi benar terhadap total prediksi

2. *Precision*: Tingkat ketepatan sistem dalam mengidentifikasi satu kelas
3. *Recall*: Kemampuan sistem menangkap seluruh data dari satu kelas
4. *F1-score*: Rata-rata harmonis antara presisi dan recall.

3.5.4 Evaluasi Model Menggunakan *K-Fold Cross Validation*

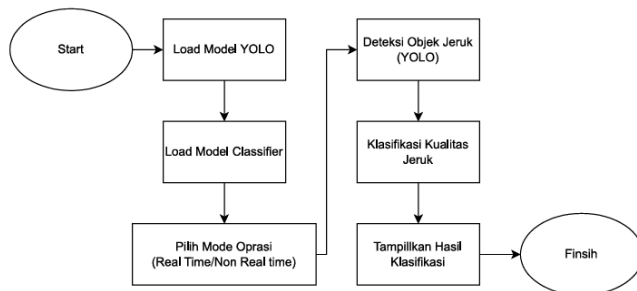
Setelah model K-Nearest Neighbor (KNN) diperoleh dari proses pelatihan, dilakukan evaluasi menggunakan metode K-Fold Cross Validation untuk mengukur konsistensi dan kemampuan generalisasi model. Metode ini dilakukan dengan membagi data menjadi 10 fold, di mana pada setiap iterasi satu fold digunakan sebagai data pengujian, sedangkan sembilan fold lainnya digunakan sebagai data pelatihan. Proses ini diulang hingga setiap fold pernah menjadi data pengujian sebanyak satu kali.

Pada setiap iterasi dihitung nilai accuracy, kemudian seluruh hasil evaluasi dirata-ratakan untuk memperoleh performa model secara keseluruhan. Penggunaan 10-Fold Cross Validation bertujuan untuk mengurangi bias akibat pembagian data serta memastikan bahwa model memiliki performa yang stabil pada berbagai variasi data sebelum diterapkan pada proses klasifikasi.

3.6 Implementasi Sistem

3.6.1 Penyimpanan Model

Tahap penyimpanan model dan visualisasi hasil merupakan bagian akhir dari sistem yang dikembangkan. Pada tahap ini, model yang telah dilatih dan diuji akan disimpan agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang. Selain itu, hasil deteksi dan klasifikasi divisualisasikan untuk memudahkan pengguna dalam memahami keluaran sistem.



Gambar 3 Tahapan visualisasi hasil

3.6.2 Perancangan *Graphical User Interface* (GUI)

Graphical User Interface (GUI) dirancang sebagai antarmuka yang memudahkan pengguna dalam melakukan proses klasifikasi kualitas jeruk nipis secara *real-time*. GUI dibangun menggunakan bahasa pemrograman Python dengan menyediakan beberapa fitur utama, seperti pemilihan sumber citra, proses klasifikasi, serta tampilan hasil prediksi. Selain itu, GUI juga terhubung dengan model K-Nearest Neighbor (KNN) yang telah dilatih sehingga pengguna dapat melakukan proses klasifikasi secara mudah tanpa harus menjalankan program melalui kode.

3.6.3 Visualisasi Hasil Klasifikasi

Setelah proses klasifikasi selesai dilakukan, GUI menampilkan hasil prediksi kualitas jeruk nipis berdasarkan kelas yang telah ditentukan, yaitu Segar, Busuk, Terinfeksi, dan Busuk Terinfeksi. Selain hasil klasifikasi, GUI juga menampilkan citra hasil segmentasi dan informasi pendukung, seperti nilai probabilitas atau tingkat keyakinan model (apabila tersedia). Visualisasi hasil ini bertujuan untuk memudahkan pengguna dalam memahami hasil klasifikasi secara cepat dan intuitif.

3.7 Analisis

Subbab ini menjelaskan rencana analisis terhadap sistem klasifikasi kualitas buah jeruk nipis menggunakan metode ekstraksi fitur GLCM dan algoritma *K-Nearest Neighbor* (KNN). Analisis dilakukan terhadap performa system berdasarkan data uji. Hasil analisis akan disajikan dalam bentuk table pada bab hasil dan pembahasan.

3.7.1 Analisis Performa Klasifikasi (Confusion Matrix)

Analisis pertama dilakukan menggunakan confusion matrix untuk mengetahui jumlah prediksi benar dan salah dari model klasifikasi.

3.7.2 Analisis Metrik Performa per kelas

Setelah diperoleh confusion matrix, akan dilakukan analisis terhadap masing-masing kelas untuk mengetahui seberapa akurat sistem mengenali buah jeruk nipis segar, tidak segar, dan busuk.

3.7.3 Pengujian Parameter K dan Fungsi Jarak

Untuk memperoleh kombinasi parameter terbaik pada algoritma KNN, dilakukan analisis terhadap beberapa nilai K dan fungsi jarak. Nilai K yang diuji antara lain: 3, 5, dan 7, dengan fungsi jarak *Euclidean*, *Manhattan*, dan *Minkowski*.

3.7.4 Analisis K-Fold Cross Validation

Analisis dilakukan dengan menggunakan metode K-Fold Cross Validation untuk menilai kestabilan model.

3.7.5 Analisis Rotasi Citra pada GLCM

Karena metode GLCM sensitif terhadap arah, maka dilakukan analisis terhadap performa sistem pada citra yang diputar dengan sudut 0° , 45° , 90° dan 135° . Tujuannya untuk mengetahui apakah nilai yang dihasilkan setiap kelas pada saat melakukan ekstraksi GLCM dapat mendukung algoritma KNN dalam proses klasifikasi.

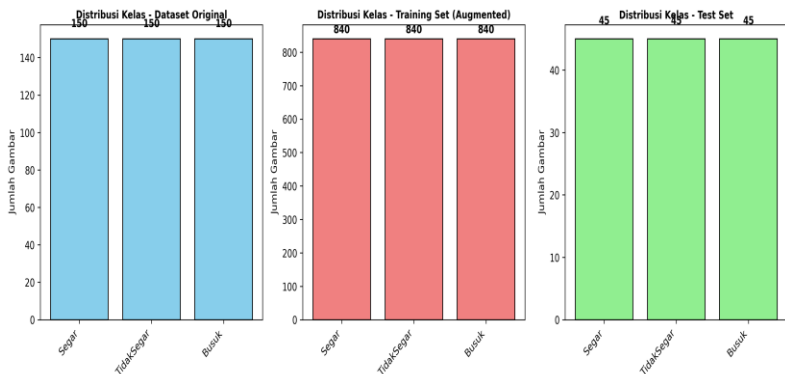
3.7.6 Analisis Kesalahan Klasifikasi

Untuk lebih memahami keterbatasan sistem, akan dicatat dan dianalisis beberapa contoh salah klasifikasi.

Bab 4 Hasil dan Pembahasan

4.1 Pengambilan Data

Pengambilan data citra buah jeruk nipis dilakukan dirumah penulis , total data citra yang terkumpul 450 citra asli yang terbagi menjadi tiga kelas kualitas, yaitu segar (150 citra), tidak segar (150 citra) dan busuk (150 citra). Semua citra diambil menggunakan kamera smartphone menggunakan latar belakang polos agar memudahkan proses segmentasi.



Gambar 4. Distribusi Kelas

Pada gambar 2 adalah contoh gambar yang dijadikan dataset, gambar ini merupakan hasil pengambilan foto dengan kondisi pencahayaan yang baik, mencakup variasi kualitas buah jeruk nipis (Segar, Tidak Segar dan Busuk).



Gambar 5 hasil pengambilan citra 3 kualitas buah jeruk nipis

4.2 Augmentasi

Augmentasi data dilakukan setelah proses split data, proses ini dilakukan hanya pada data training, pada sub bab 3.1.6 sudah dijelaskan bahwa tujuan melakukan augmentasi data adalah meningkatkan jumlah variasi data latih, mengatasi overfitting, dan meningkatkan robustitas terhadap variasi, orientas.

Table 3. Teknik augmentasi data

No	Teknik Augmentasi	Jumlah Variasi
1	Rotasi (90°, 180°, dan 270°)	3
2	Flip Horizontal	1
3	Flip Vertical	1
4	Penyesuaian Kecerahan (+30% dan -30% dari nilai asli)	2
Total Variasi per citra asli		8 variasi

Table 4. Jumlah data hasil augmentasi

Kelas	Jumlah Citra Asli	Jumlah augmentasi per citra	Total citra setelah augmentasi
Segar	105	8	840
Tidak segar	105	8	840
Busuk	105	8	840
Total	315	-	2.520

Setelah Augmentasi jumlah data latih meningkat 8 kali lipat dari 315 menjadi 2.520 citra dengan proporsi kelas yang tetap seimbang (1:1:1). Data uji sebanyak 135 citra (45 per kelas) tidak di augmentasi agar tetap merepresentasikan kondisi citra asli di dunia nyata dan menghindari kebocoran informasi (data leakage) ke proses evaluasi.

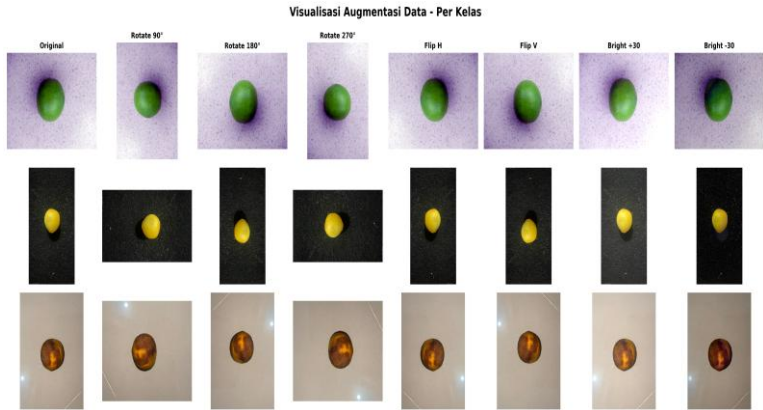
Dengan strategi augmentasi ini, model KNN memperoleh kemampuan generalisasi yang jauh lebih baik sehingga siap digunakan pada kondisi pengambilan citra yang bervariasi di lapangan

4.2.1 Analisis Pengaruh Augmentasi terhadap akurasi model

Penerapan augmentasi data memberikan kontribusi langsung terhadap peningkatan performa klasifikasi, yang dapat dianalisis dari beberapa aspek berikut.

1. Peningkatan Variasi Fitur yang Dipelajari Model

Augmentasi seperti rotasi, flipping, perubahan intensitas cahaya, dan transformasi geometris menghasilkan citra jeruk dengan perbedaan sudut pandang, variasi Pencahayaan, dan perubahan orientasi objek. Hal ini menyebabkan fitur tekstur (GLCM) dan warna (HSV) yang diekstraksi menjadi lebih beragam. Akibatnya, distribusi fitur pada data latih menjadi lebih luas dan tidak terkonsentrasi pada pola tertentu saja. Kondisi ini membantu model mengenali karakteristik kelas secara lebih umum, bukan hanya menghafal pola spesifik dari data asli.



Gambar 6. Visualisasi augmentasi data

2. Reduksi Overfitting pada Model

Tanpa augmentasi, model cenderung mengalami *overfitting*, yaitu kondisi di mana model sangat baik mengenali data latih tetapi kurang mampu mengklasifikasikan data uji. Sehingga dengan bertambahnya data latih hasil augmentasi model tidak hanya bergantung pada satu pola tekstur atau warna tertentu dan sensitivitas model terhadap noise dan variasi kecil pada citra menjadi berkurang.

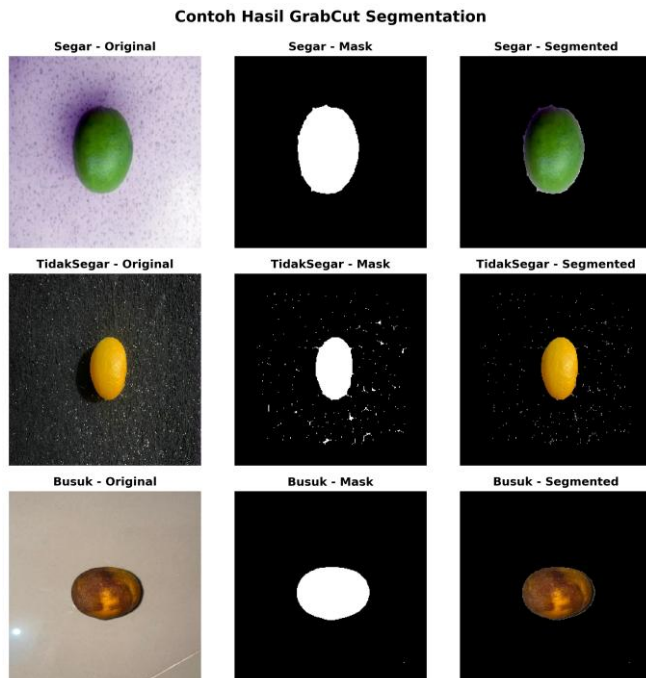
3. Dampak Terhadap Algoritma KNN

Algoritma K-Nearest Neighbor (KNN) sangat bergantung pada kepadatan dan distribusi data di ruang fitur. Augmentasi data menyebabkan titik-titik data latih tersebar lebih merata dan jarak antar sampel dalam kelas yang sama menjadi lebih representatif, akibatnya, proses pencarian tetangga terdekat menjadi lebih akurat sehingga Keputusan klasifikasi pada data uji menjadi lebih stabil dan konsisten.

Meskipun augmentasi terbukti meningkatkan performa model, perlu dicatat bahwa augmentasi tidak sepenuhnya menghilangkan kesalahan klasifikasi. Hal ini menunjukkan bahwa augmentasi bersifat memperkaya data, bukan menciptakan informasi baru, kemiripan karakteristik antar kelas masih menjadi faktor utama kesalahan. Dengan demikian augmentasi berfungsi sebagai pendukung performa, sementara peningkatan lebih lanjut dapat dicapai dengan penambahan fitur yang lebih diskriminatif atau penggunaan metode klasifikasi yang lebih kompleks.

4.3 Segmentasi

Segmentasi citra merupakan tahapan krusial dalam sistem ini, tujuan nya Adalah memisahkan foreground (buah jeruk nipis) dan background, tanpa segmentasi yang baik, fitur tekstur GLCM dan warna akan tercampur dengan noise latar belakang sehingga menurunkan akurasi klasifikasi secara signifikan, hasil visualisasi segmentasi pada 3 kelas buah jeruk nipis dapat dilihat pada gambar 7.



Gambar 7 Visualisasi Hasil GrabCut Segmentasi

Penggunaan algoritma GrabCut dengan inisialisasi rectangle otomatis terbukti sangat efektif dan efisien untuk memisahkan objek jeruk nipis dari latar belakang yang kompleks. Pada gambar 5 dapat dilihat proses segmentasi menggunakan grabcut menghilangkan 89,1% background sehingga fitur GLCM, HSV dan bentuk benar-benar merepresentasikan kondisi jeruk nipis saja, bukan noise latar belakang. Tanpa segmentasi (simple threshold) masih menyisakan 25% area background sehingga fitur contrast dan dissimilarity meningkat secara artifisial, histogram warna HSV juga akan tercampur dengan warna latar belakang.

4.3.1 Dampak Segmentasi terhadap Akurasi Model

Segmentasi citra menggunakan metode GrabCut bertujuan untuk memisahkan objek jeruk nipis (foreground) dari latar belakang (background), sehingga fitur yang diekstraksi benar-benar merepresentasikan karakteristik objek. Secara teori, proses segmentasi dapat meningkatkan akurasi klasifikasi karena mengurangi pengaruh noise dari background.

Namun, berdasarkan hasil pengujian pada penelitian ini, diperoleh bahwa akurasi model tanpa segmentasi sebesar 97,78%, sedangkan dengan segmentasi GrabCut sebesar 97,04%. Perbedaan ini menunjukkan adanya penurunan akurasi sebesar 0,74% setelah dilakukan segmentasi.

Penurunan ini tergolong sangat kecil dan tidak signifikan, yang secara praktis setara dengan perubahan klasifikasi pada sekitar satu sampel dari total 135 data uji. Hal ini menunjukkan bahwa segmentasi tidak memberikan peningkatan performa yang signifikan pada kondisi dataset yang digunakan.

Beberapa faktor yang menyebabkan fenomena ini antara lain:

1. Kondisi background yang sederhana

Seluruh citra diambil dengan latar belakang yang relatif polos dan seragam, sehingga tanpa segmentasi pun pengaruh noise dari background sudah sangat kecil. Hal ini membuat kontribusi segmentasi terhadap peningkatan akurasi menjadi tidak terlalu signifikan.

2. Perubahan distribusi fitur akibat segmentasi

Proses segmentasi GrabCut dapat mengubah komposisi piksel pada citra, terutama pada bagian tepi objek. Perubahan ini mempengaruhi nilai fitur tekstur (GLCM) dan warna (HSV), sehingga posisi data dalam ruang fitur ikut berubah. Karena algoritma KNN sangat sensitif terhadap jarak antar data, perubahan kecil ini dapat menyebabkan perbedaan hasil klasifikasi.

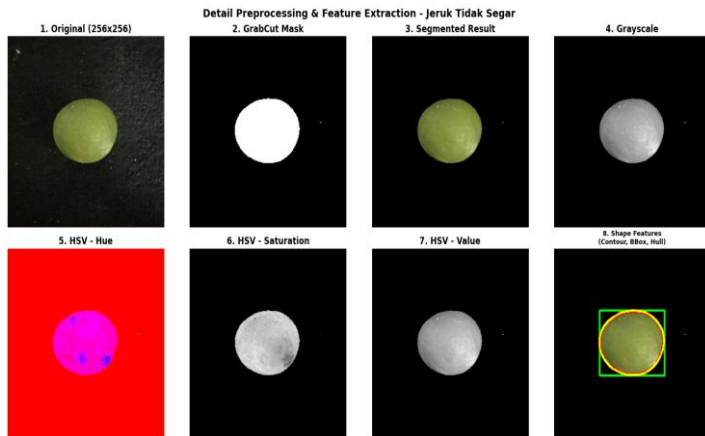
3. Kehilangan informasi penting (*over-segmentation*).

Dalam beberapa kasus, segmentasi dapat menghilangkan sebagian informasi visual seperti bayangan atau gradasi warna yang sebenarnya membantu proses klasifikasi. Hal ini menyebabkan model kehilangan sebagian informasi pembeda antar kelas.

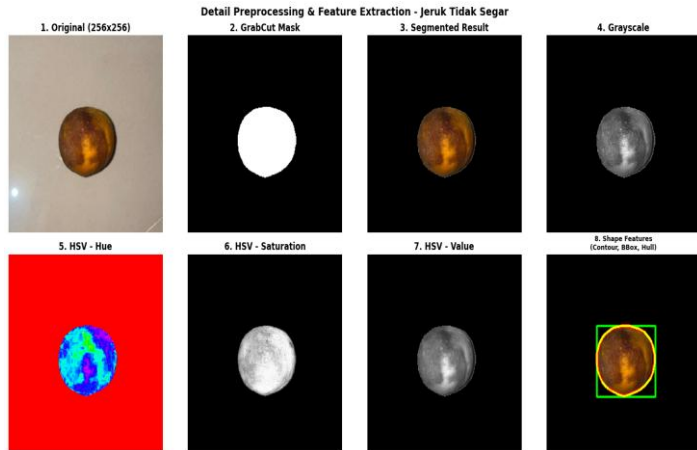
Meskipun tidak meningkatkan akurasi pada penelitian ini, penggunaan segmentasi tetap memiliki peran penting, terutama untuk kondisi citra dengan latar belakang yang kompleks atau tidak terkontrol. Oleh karena itu, segmentasi tetap direkomendasikan untuk meningkatkan robustitas sistem dalam penerapan di dunia nyata.

4.4 Preprocessing dan Ekstraksi Fitur

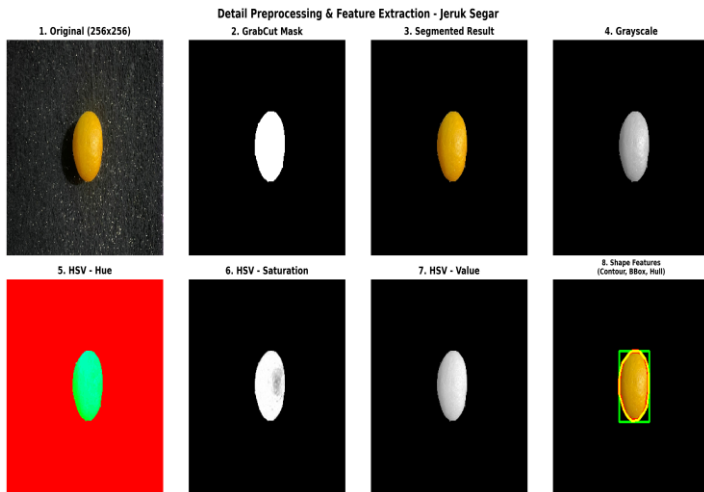
Pada tahapan ini sudah dilakukan serangkaian proses peningkatan kualitas citra serta pengambilan fitur penting yang di perlukan untuk membedakan kelas jeruk (segar, tidak segar, busuk). Proses ini terdiri dari beberapa langkah, mulai dari segmentasi objek hingga perhitungan fitur warna, tekstur dan bentuk. Hasil visualisasi tiap tahapan di tunjukan pada gambar dibawah.



Gambar 8. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk Segar



Gambar 9. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk Busuk



Gambar 10. Visualisasi Preprocessing dan Ekstraksi Fitur Jeruk kurang segar

4.4.1 Preprocessing

1. *Resize Citra*

Citra asli diubah ke ukuran 256 x 256 piksel agar konsisten dan sesuai untuk proses komputasi. Ukuran ini dipilih karena cukup besar untuk menjaga detail permukaan jeruk namun tetap efisien secara komputasi.

2. Segmentasi menggunakan *Grabcut*

Metode GrabCut digunakan untuk memisahkan objek jeruk dari latar belakang. Proses ini menghasilkan *binary mask* yang menandai piksel objek dengan nilai putih, sedangkan latar belakang ditandai hitam. Tahap ini penting untuk memastikan bahwa fitur yang dihitung hanya berasal dari jeruk dan tidak dipengaruhi oleh background.

3. Hasil Segmentasi (*Segmentasi Result*)

Mask GrabCut diterapkan ke citra asli sehingga hanya area jeruk yang dipertahankan. *Background* diset menjadi hitam. Citra hasil segmentasi ini digunakan untuk seluruh proses selanjutnya

4. Konversi ke *GrayScale*

Citra yang telah tersegmentasi lalu dikonversi ke grayscale untuk keperluan ekstraksi tekstur, terutama perhitungan fitur GLCM. Proses ini menyederhanakan intensitas warna sehingga pola tekstur lebih mudah dianalisis.

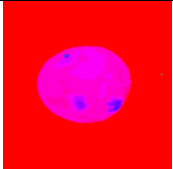
4.4.2 Ekstraksi Fitur Warna (HSV)

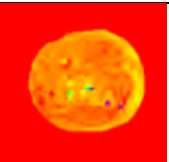
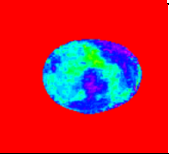
Ekstraksi warna dilakukan dengan mengonversi citra ke ruang warna HSV (Hue, Saturation, Value).

1. *Channel Hue*

Channel Hue merepresentasikan jenis warna atau Panjang gelombang warna yang dominan pada setiap piksel.

Table 5. Channel Hue

Kelas	Visual	
Buah Jeruk Segar		

Buah jeruk tidak segar			
Buah jeruk busuk			

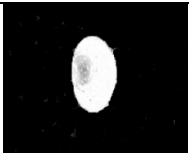
Channel *Hue* memiliki peran yang sangat penting dalam proses klasifikasi kualitas jeruk karena mampu merepresentasikan informasi warna secara efektif. Channel ini dapat membedakan warna kulit jeruk pada setiap kelas, di mana jeruk segar cenderung memiliki warna yang stabil, jeruk tidak segar mengalami pergeseran ke kombinasi warna kuning-oranye, sedangkan jeruk busuk menunjukkan perubahan spektrum warna akibat kerusakan pada permukaan kulit. Selain itu, *Hue* juga mampu mendeteksi area pembusukan yang umumnya muncul sebagai bercak dengan warna berbeda dibandingkan area kulit yang masih sehat, sehingga memudahkan sistem dalam mengenali bagian yang mengalami kerusakan. Keunggulan lainnya adalah *Hue* relatif lebih stabil terhadap perubahan intensitas pencahayaan dibandingkan komponen warna lainnya, sehingga informasi warna yang diperoleh menjadi lebih konsisten dan andal untuk digunakan dalam proses klasifikasi.

2. *Channel Saturation*

Channel saturation menggambarkan Tingkat kejenuhan warna atau seberapa kuat warna tersebut terlihat.

Table 6. Channel saturation

Kelas	Visual	
Buah Jeruk Segar		

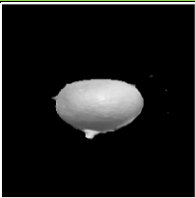
Buah Jeruk Tidak Segar			
Buah Jeruk Busuk			

Channel *Saturation* berperan penting dalam mengidentifikasi infeksi atau bercak jamur pada permukaan jeruk karena merepresentasikan tingkat kejenuhan warna. Jeruk segar umumnya memiliki permukaan yang cerah dengan warna yang lebih hidup sehingga menghasilkan nilai *Saturation* yang tinggi dan relatif merata. Sebaliknya, jeruk tidak segar maupun busuk mengalami perubahan struktur kulit yang menyebabkan warna tampak lebih kusam, kurang jenuh, dan pada beberapa bagian cenderung mengarah ke abu-abu. Selain itu, channel *Saturation* juga relatif lebih stabil terhadap pengaruh bayangan (*shading*) dibandingkan nilai intensitas secara langsung, sehingga informasi yang dihasilkan lebih konsisten dan dapat diandalkan sebagai fitur warna dalam proses klasifikasi kualitas jeruk.

3. Channel Value

Channel Value merepresentasikan tingkat kecerahan (*brightness*) suatu piksel, bukan warna.

Table 7. Channel value

Kelas	Visual	
Buah Jeruk Segar		

Buah jeruk tidak segar	
Buah Jeruk Busuk	

Channel Value berperan penting dalam analisis kualitas jeruk nipis karena merepresentasikan tingkat kecerahan atau intensitas cahaya pada permukaan buah. Channel ini membantu menganalisis tekstur permukaan, di mana jeruk segar umumnya memiliki nilai *Value* yang lebih cerah dan homogen, sedangkan jeruk busuk cenderung memiliki area yang lebih gelap dan permukaan yang tidak rata sehingga menghasilkan variasi nilai *Value* yang lebih ekstrem. Selain itu, *Value* juga dapat mengidentifikasi pola permukaan yang terbentuk dari pantulan cahaya pada kulit jeruk. Pola intensitas ini akan berubah ketika permukaan mengalami kerusakan atau pembusukan, sehingga perbedaan kondisi buah dapat diamati dengan lebih jelas. Karena channel *Value* hanya merepresentasikan intensitas cahaya tanpa dipengaruhi oleh informasi warna, fitur ini mampu memberikan informasi tekstur yang lebih murni dan sangat mendukung proses ekstraksi fitur tekstur menggunakan metode Gray Level Co-occurrence Matrix (GLCM).

4.4.3 Ekstraksi Fitur GLCM

GLCM tidak divisualisasikan dalam bentuk gambar karena GLCM tidak menghasilkan citra melainkan matriks berisi frekuensi pasangan nilai piksel. Fitur tekstur dihitung menggunakan metode *Grey Level Co-occurrence Matrix (GLCM)* pada beberapa sudut orientasi (0°, 45°, 90°, dan 135°). Parameter yang dianalisis meliputi contrast, dissimilarity, homogeneity, dan energy pada sudut 0°, 45°, 90°, dan 135°, serta nilai rata-rata (mean). Analisis ini bertujuan untuk melihat perbedaan karakteristik tekstur antar kelas.

1. Kelas Jeruk Segar

Pada kelas jeruk segar, diperoleh nilai *contrast_mean* sebesar 0,1481 dan *dissimilarity_mean* sebesar 0,1432. Nilai ini tergolong

rendah, yang menunjukkan bahwa variasi intensitas piksel pada permukaan jeruk relatif kecil. Hal ini mencerminkan kondisi kulit jeruk yang masih halus dan belum mengalami kerusakan. Nilai *homogeneity_mean* sebesar 0,9289 menunjukkan tingkat keseragaman tekstur yang tinggi, sedangkan *energy_mean* sebesar 0,5638 menandakan pola tekstur yang cukup teratur. Kombinasi nilai ini menunjukkan bahwa jeruk segar memiliki tekstur yang homogen dan stabil pada berbagai arah sudut.

Table 8. Parameter kelas segar

Kelas Segar	<i>Contrast</i>	<i>Dissimilarity</i>	<i>Homogeneity</i>	<i>Energy</i>
Segar 0°	0.1225	0.1206	0.9399	0.5722
Segar 45°	0.1715	0.1629	0.9194	0.5565
Segar 90°	0.1324	0.1292	0.9357	0.5701
Segar 135°	0.1661	0.1601	0.9205	0.5563
Mean	0.1481	0.1432	0.9289	0.5638

2. Kelas Jeruk Tidak Segar

Kelas jeruk tidak segar menunjukkan peningkatan signifikan pada nilai *contrast mean* (0.2093) dan *dissimilarity mean* (0.2077) dibandingkan jeruk segar. Peningkatan ini mengindikasikan adanya variasi intensitas dan ketidaksamaan tekstur yang lebih tinggi akibat proses pembusukan. Nilai *homogeneity mean* menurun menjadi 0.8963, yang menandakan berkurangnya keseragaman tekstur. Selain itu, *energy mean* sebesar 0,4148 menunjukkan bahwa pola tekstur menjadi kurang teratur. Kondisi ini sesuai dengan karakteristik fisik jeruk tidak segar yang mengalami perubahan warna dan tekstur.

Table 9. Parameter kelas tidak segar

Kelas Tidak Segar	<i>Contrast</i>	<i>Dissimilarity</i>	<i>Homogeneity</i>	<i>Energy</i>
Tidak segar 0°	0.1865	0.1857	0.9072	0.4234
Tidak segar 45°	0.2500	0.2472	0.8767	0.3989
Tidak segar 90°	0.1658	0.1654	0.9174	0.4325
Tidak segar 135°	0.2349	0.2326	0.8839	0.4047
Mean	0.2093	0.2077	0.8963	0.4148

3. Kelas Jeruk Busuk

Pada kelas jeruk busuk, nilai *contrast_mean* sebesar 0,0875 dan *dissimilarity_mean* sebesar 0,0862 tergolong sangat rendah. Hal ini menunjukkan bahwa jeruk busuk mengalami perubahan tekstur permukaan. Nilai *homogeneity_mean* sangat tinggi, yaitu 0,9570, serta *energy_mean* sebesar 0,3936, yang menunjukkan tekstur sangat seragam dan terstruktur.

Table 10. Parameter kelas busuk

Kelas Busuk	<i>Contrast</i>	<i>Dissimilarity</i>	<i>Homogeneity</i>	<i>Energy</i>
Busuk 0°	0.0691	0.0688	0.9656	0.4007
Busuk 45°	0.1056	0.1030	0.9487	0.3870
Busuk 90°	0.0796	0.0788	0.9607	0.3964
Busuk 135°	0.0957	0.0943	0.9530	0.3902
Mean	0.0875	0.0862	0.9570	0.3936

Berdasarkan hasil analisis, dapat disimpulkan bahwa setiap kelas jeruk memiliki pola fitur GLCM yang berbeda. Jeruk segar cenderung memiliki nilai *homogeneity* dan *energy* yang tinggi, sedangkan jeruk busuk menunjukkan peningkatan *contrast* dan *dissimilarity* yang signifikan. Perbedaan karakteristik ini menyebabkan fitur GLCM memiliki daya diskriminatif yang baik, sehingga mampu mendukung kinerja algoritma KNN dalam mengklasifikasikan kondisi jeruk secara akurat.

4.5 Tabel hasil Analisis Metrik Performa per kelas

Berdasarkan hasil analisis, setiap kelas jeruk memiliki karakteristik tekstur yang berbeda. Kelas tidak segar memiliki nilai *contrast* dan *dissimilarity* tertinggi, yang menunjukkan permukaan kulit mulai mengalami perubahan tekstur sehingga variasi intensitas piksel meningkat. Sebaliknya, kelas busuk memiliki nilai *contrast* dan *dissimilarity* yang lebih rendah disertai nilai *homogeneity* tertinggi. Hal ini menunjukkan bahwa proses pembusukan menyebabkan permukaan buah menjadi lebih homogen akibat dominasi area gelap dan kerusakan yang merata. Perbedaan karakteristik tersebut menunjukkan bahwa fitur GLCM mampu membedakan ketiga kelas sehingga mendukung proses klasifikasi menggunakan K-Nearest Neighbor:

1. Kelas Segar (F1: 97%):

Kelas ini memiliki karakteristik visual yang sangat jelas dan konsisten. Fitur warna HSV menunjukkan nilai Hue dan Saturation yang tinggi dengan warna hijau cerah yang sangat berbeda dari kelas lain. Tekstur kulit jeruk nipis segar sangat halus, yang tercermin dalam nilai GLCM

contrast yang rendah dan homogeneity yang tinggi. Bentuk buah masih bulat sempurna dengan circularity mendekati 1.0, dan tidak ada ambiguitas atau overlap fitur dengan kelas lainnya. Kombinasi fitur yang distinct ini membuat model dapat mengidentifikasi kelas Segar dengan confidence yang sangat tinggi

2. Kelas Tidak Segar (F1 : 96%):

Perubahan warna pada kelas Tidak Segar mulai terlihat dengan degradasi warna menuju coklat kekuningan yang menghasilkan nilai HSV Value lebih rendah dibandingkan kelas Segar, namun belum separah kelas Busuk. Bentuk buah masih relatif terjaga meskipun nilai circularity mengalami sedikit penurunan dibandingkan kondisi segar. Dengan F1-score sebesar 96%, model menunjukkan performa yang sangat baik dalam mengenali kelas ini, yang mengindikasikan bahwa fitur warna, tekstur, dan bentuk mampu membedakan kelas Tidak Segar dari kelas Segar maupun Busuk secara konsisten

3. Kelas Busuk (F1: 99%):

Perubahan warna pada kelas Busuk sangat mencolok, dengan degradasi ke arah coklat atau kuning tua yang menghasilkan nilai HSV Value yang rendah. Tekstur permukaan menjadi tidak rata akibat proses pembusukan, yang terdeteksi melalui peningkatan GLCM contrast. Bentuk buah mulai berubah dengan penurunan nilai circularity. secara umum model masih dapat membedakan kedua kelas ini dengan baik.

Table 11. Metrik performa per kelas

KELAS	Presisi (%)	Recall (%)	F1-Score (%)
SEGAR	100%	93.00%	97.00%
TIDAK SEGAR	94.00%	98.00%	96.00%
BUSUK	98.00%	100%	99.00%

4.5.1 Analisis Kesalahan Klasifikasi

Dari total 135 sampel test set, model membuat 4 kesalahan prediksi (error rate 2.96%). Analisis detail kesalahan klasifikasi menunjukkan pola-pola menarik:

Table 12. Analisis kesalahan klasifikasi

No	Prediksi	Realita	Komentar
----	----------	---------	----------

1	Index: 27 Asli : Segar Prediksi : TidakSegar 	Segar	Warna kulit jeruk yang sudah menguning menyebabkan nilai fitur HSV bergeser mendekati profil kelas TidakSegar, sehingga KNN salah mengklasifikasikan sampel ini.
2	Index: 78 Asli : TidakSegar Prediksi : Busuk 	Tidak Segar	Warna kuning tua dan permukaan mengkerut membuat fitur HSV dan GLCM menyerupai kelas Busuk, sehingga model salah memprediksi sampel ini sebagai Busuk.
3	Index: 109 Asli : Segar Prediksi : TidakSegar 	Segar	Meskipun termasuk kelas segar, distribusi warna pada permukaan jeruk terlihat tidak merata dengan dominasi warna kekuningan pada beberapa area. Hal ini menyebabkan nilai Hue dan Saturation mendekati karakteristik kelas tidak segar. Selain itu, pencahayaan yang relatif tinggi juga meningkatkan nilai Value, sehingga fitur HSV cenderung bias dan memengaruhi hasil klasifikasi KNN.

4	Index: 130 Asli : Segar Prediksi : TidakSegar 	Segar	Kesalahan klasifikasi pada sampel ini dipengaruhi oleh kemiripan warna dengan kelas tidak segar, khususnya pada bagian tengah dan pinggir buah yang menunjukkan perubahan warna. Selain itu, tekstur permukaan yang lebih halus menyebabkan fitur GLCM kurang memberikan pembeda yang signifikan, sehingga model lebih dipengaruhi oleh fitur warna dan menghasilkan prediksi yang tidak sesuai.
---	--	-------	---

kesalahan antar kelas yang bersebelahan secara kualitas, yaitu (Segar-Tidak segar) dan (Tidak segar-Busuk). Tidak ada kesalahan yang bersifat ekstrem seperti Segar diklasifikasikan sebagai Busuk atau sebaliknya. Hal ini menunjukkan bahwa model KNN telah mampu mempelajari batas antar kelas secara umum dengan baik, dan kesalahan yang terjadi hanya pada sampel-sampel yang memiliki karakteristik visual di ambang batas antar kelas (*borderline samples*). Kesalahan ini wajar terjadi mengingat proses pematangan jeruk bersifat kontinu, sehingga perbedaan fitur antar kelas yang berdekatan tidak selalu tegas.

4.6 Tabel hasil Pengujian Parameter K dan fungsi jarak

Pada tahap ini, telah dilakukan pengujian untuk menentukan kombinasi hyperparameter algoritma nilai KNN, yaitu nilai K (jumlah tetangga terdekat) dan fungsi jarak (*distance metric*) yang digunakan. Pengujian ini dilakukan untuk memastikan mode mana yang digunakan memiliki performa optimal dan tidak dipilih secara sembarangan.

Nilai K yang diuji Adalah 3, 5, 7, 9, dan 11, sedangkan untuk fungsi jarak yang diuji meliputi:

1. *Euclidean* (jarak lurus, standart pada KNN)
2. *Manhattan* (jarak kota atau L1 distance)
3. *Minkowski* dengan parameter $p=3$ (generalisasi dari *Euclidean* dan *Manhattan*).

Total terapat 15 kombinasi *hyperparameter* yang dievaluasi. Proses pengujian dilakukan dengan melatih model pada data latih yang telah di augmentasi dan

telah melalui reduksi dimensi menggunakan PCA, kemudian dievaluasi pada data uji independent (original, tanpa augmentasi) untuk menghindari data leakage. Hasil pengujian di tampilkan pada table 19.

Table 13. Hasil pengujian parameter nilai K dan fungsi jarak

Fungsi Jarak	Nilai K	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
<i>Euclidean</i>	3	97.04%	97.15%	97.04%	97.03%
<i>Manhattan</i>	3	98.52%	98.53%	98.52%	98.52%
<i>Minkowski (p=3)</i>	3	94.81%	94.92%	94.81%	94.81%
<i>Euclidean</i>	5	97.04%	97.15%	97.04%	97.03%
<i>Manhattan</i>	5	96.30%	96.48%	96.30%	96.32%
<i>Minkowski (p=3)</i>	5	94.07%	94.25%	94.07%	94.09%
<i>Euclidean</i>	7	95.56%	95.84%	95.56%	95.60%
<i>Manhattan</i>	7	94.81%	95.22%	94.81%	94.88%
<i>Minkowski (p=3)</i>	7	94.07%	94.25%	94.07%	94.09%
<i>Euclidean</i>	9	94.81%	95.12%	94.81%	94.84%
<i>Manhattan</i>	9	94.81%	95.22%	94.81%	94.88%
<i>Minkowski (p=3)</i>	9	94.07%	94.25%	94.07%	94.09%
<i>Euclidean</i>	11	94.07%	94.25%	94.07%	94.09%
<i>Manhattan</i>	11	95.56%	95.84%	95.56%	95.60%
<i>Minkowski (p=3)</i>	11	94.07%	94.25%	94.07%	94.09%

Hasil pengujian yang ditampilkan pada table 21 yang memperlihatkan nilai akurasi, presisi, recall, dan F1-Score untuk setiap kombinasi fungsi jarak dan nilai K terhadap kinerja klasifikasi citra buah jeruk nipis.

Secara umum, hasil pengujian menunjukkan bahwa nilai akurasi tertinggi diperoleh sebesar 98.52%, nilai presisi tertinggi sebesar 98.53%, nilai recall tertinggi sebesar 98.52%, dan nilai F1-Score tertinggi 98.52%, nilai tersebut dicapai oleh fungsi jarak Manhattan pada nilai K= 3. Tingginya nilai akurasi dan F1-score menunjukkan bahwa kombinasi preprocessing citra, ekstraksi fitur warna dan tekstur, serta klasifikasi KNN mampu membedakan kondisi jeruk nipis dengan baik.

4.6.1 Jarak Euclidean

Fungsi jarak Euclidean menunjukkan performa terbaik pada nilai $K=3$ dan $K=5$ dengan akurasi yang sama sebesar 97,04%. Hal ini mengindikasikan bahwa pada ruang fitur hasil PCA, jarak geometris lurus sudah cukup efektif untuk memisahkan ketiga kelas jeruk. Namun seiring bertambahnya nilai K , akurasi cenderung menurun secara konsisten hingga 94,07% pada $K=11$. Penurunan ini terjadi karena semakin banyak tetangga yang dilibatkan, semakin besar kemungkinan tetangga dari kelas yang berbeda ikut mempengaruhi keputusan klasifikasi, terutama pada sampel yang berada di batas antar kelas..

4.6.2 Jarak Manhattan

Fungsi jarak Manhattan mencapai performa tertinggi dari seluruh kombinasi yang diuji, yaitu akurasi 98,52% pada $K=3$. Nilai ini menjadi akurasi maksimum dalam keseluruhan pengujian Table 19. Manhattan distance bekerja lebih baik pada K kecil karena pola distribusi fitur GLCM dan HSV yang cenderung bervariasi secara independen per dimensi — kondisi di mana penjumlahan selisih absolut lebih representatif dibanding jarak Euclidean. Namun performa Manhattan turun signifikan pada $K=5$ (96,30%) dan terus menurun hingga $K=9$, sebelum sedikit naik kembali di $K=11$ (95,56%), menunjukkan ketidakstabilan relatif dibanding Euclidean pada K besar.

4.6.3 Jarak Minkowski ($p=3$)

Fungsi jarak Minkowski dengan parameter $p=3$ secara konsisten menghasilkan akurasi terendah di antara ketiga fungsi jarak yang diuji, dengan rentang akurasi 94,07% – 94,81%. Yang menarik, nilai akurasi, presisi, recall, dan F1-Score hampir identik pada $K=5$ hingga $K=11$ (semuanya 94,07%). Ini mengindikasikan bahwa pada $p=3$, model mengalami *Saturasi* penambahan nilai K tidak lagi memberikan perubahan keputusan klasifikasi yang berarti.

4.7 Tabel hasil Pengujian K-Fold Cross Validation

Pengujian performa sistem klasifikasi dilakukan menggunakan metode 10-Fold Stratified Cross-Validation pada data original. Metode ini dipilih untuk memastikan bahwa evaluasi model dilakukan secara objektif dan setiap kelas memiliki proporsi yang seimbang pada data latih dan data uji di setiap fold.

Table 14. Hasil pengujian K-Fold Cross Validation

K-Fold	Accuracy	Precision	Recall	F1-Score
1	93.33%	93.75%	93.33%	93.32%
2	100.00%	100.00%	100.00%	100.00%
3	97.78%	97.92%	97.78%	97.78%
4	100.00%	100.00%	100.00%	100.00%
5	100.00%	100.00%	100.00%	100.00%
6	100.00%	100.00%	100.00%	100.00%
7	95.56%	96.08%	95.56%	95.54%
8	95.56%	95.69%	95.56%	95.55%
9	100.00%	100.00%	100.00%	100.00%
10	100.00%	100.00%	100.00%	100.00%

Hasil pengujian 10-Fold Stratified Cross Validation menunjukkan bahwa model klasifikasi jeruk yang dibangun memiliki performa yang stabil dan konsisten. Rentang nilai akurasi yang diperoleh berada antara 93% hingga 100%, yang mengindikasikan bahwa sistem mampu mengklasifikasikan citra jeruk dengan cukup baik pada berbagai pembagian data.

Akurasi tertinggi sebesar 100% menunjukkan bahwa data uji pada fold tersebut memiliki karakteristik fitur yang dapat dipisahkan secara sempurna oleh model. Kondisi ini bersifat wajar dalam skema cross validation, karena setiap fold memiliki distribusi data yang berbeda meskipun proporsi kelas tetap seimbang.

Sementara itu, fold dengan akurasi terendah sebesar 93.33% tetap menunjukkan performa yang cukup tinggi. Hal ini mengindikasikan bahwa kesalahan klasifikasi yang terjadi bersifat minimal dan kemungkinan disebabkan oleh kemiripan fitur pada beberapa kelas.

Nilai standar deviasi sebesar $\pm 2.78\%$ menunjukkan bahwa perbedaan performa antar fold relatif kecil. Dengan demikian, model dapat dikatakan memiliki kemampuan generalisasi yang baik dan tidak bergantung pada satu pembagian data tertentu.

4.8 Analisis Performa Klasifikasi (Confusion Matrix) pada Data Test

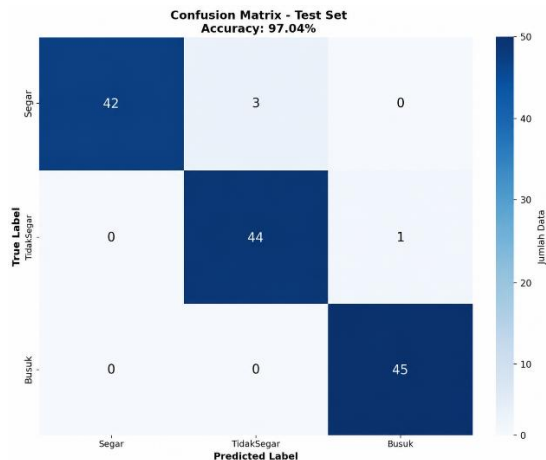
Confusion Matrix digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan label aktual dengan hasil prediksi model. Pada penelitian ini, confusion matrix digunakan untuk menganalisis performa model KNN dalam mengklasifikasikan kualitas buah jeruk nipis ke dalam tiga kelas kelas, yaitu Segar, tidak segar dan busuk.

Berdasarkan *Confusion Matrix* data tes, performa model menunjukkan kemampuan klasifikasi yang sangat baik pada ketiga kelas yang diuji.

Pada kelas Segar, dari total 45 data uji, sebanyak 42 data (93,33%) berhasil diklasifikasikan dengan benar sebagai Segar, sedangkan 3 data (6,67%) salah diklasifikasikan sebagai kelas lain. Hasil ini menunjukkan bahwa karakteristik jeruk segar dapat dikenali dengan baik oleh model.

Pada kelas Tidak Segar, dari 45 data uji, sebanyak 44 data (97,78%) berhasil diklasifikasikan dengan benar, sementara 1 data (2,22%) salah diklasifikasikan sebagai Segar, yang mengindikasikan bahwa model juga memiliki kemampuan yang sangat baik dalam membedakan jeruk tidak segar.

Sementara itu, pada kelas Busuk, seluruh 45 data uji (100%) berhasil diklasifikasikan dengan benar tanpa adanya kesalahan klasifikasi. Secara keseluruhan, hasil *Confusion Matrix* menunjukkan bahwa model memiliki performa yang sangat baik dalam mengidentifikasi kualitas jeruk nipis pada ketiga kelas, dengan tingkat kesalahan yang sangat rendah.



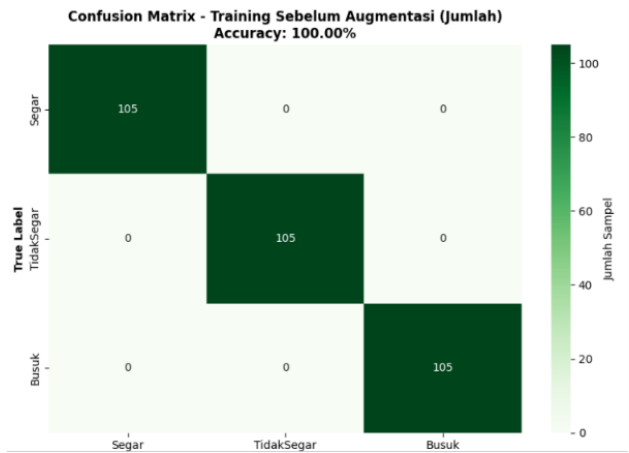
Gambar 11. Confusion Matrix data Test

Berdasarkan hasil pengujian, model klasifikasi mencapai akurasi sebesar 97,04% pada data uji. Nilai ini menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan kualitas jeruk nipis ke dalam tiga kelas yang ditentukan.

Meskipun model klasifikasi menunjukkan performa yang sangat baik, masih terdapat beberapa kesalahan klasifikasi pada data uji. Kesalahan ini tidak terjadi secara acak, melainkan dipengaruhi oleh beberapa faktor utama yang berkaitan

dengan karakteristik visual data, metode ekstraksi fitur, serta batasan algoritma klasifikasi yang digunakan.

4.9 Analisis Perma Klasifikasi (Confusion Matrix) pada Data Training



Gambar 12. Confusion Matrix data Training

Berdasarkan Confusion Matrix data training sebelum augmentasi, performa model menunjukkan kemampuan klasifikasi yang sangat baik pada ketiga kelas yang digunakan dalam proses pelatihan. Pada kelas Segar, dari total 105 data latih, seluruh 105 data (100%) berhasil diklasifikasikan dengan benar sebagai kelas Segar tanpa adanya kesalahan klasifikasi. Hasil ini menunjukkan bahwa model mampu mempelajari karakteristik jeruk segar dengan sangat baik pada data pelatihan.

Pada kelas Tidak Segar, dari 105 data latih, seluruh 105 data (100%) juga berhasil diklasifikasikan dengan benar sebagai kelas Tidak Segar tanpa adanya kesalahan klasifikasi. Hal ini menunjukkan bahwa model mampu membedakan karakteristik jeruk tidak segar secara tepat pada data training.

Sementara itu, pada kelas Busuk, seluruh 105 data latih (100%) berhasil diklasifikasikan dengan benar sebagai kelas Busuk tanpa adanya kesalahan klasifikasi. Hasil tersebut menunjukkan bahwa model mampu mengenali ciri-ciri jeruk busuk secara sempurna pada data pelatihan.

Secara keseluruhan, hasil Confusion Matrix pada data training menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan

kualitas jeruk nipis pada ketiga kelas, dengan akurasi sebesar 100% dan tanpa adanya kesalahan klasifikasi. Kondisi ini mengindikasikan bahwa model mampu mempelajari pola pada data training dengan sangat baik. Namun, performa yang sempurna pada data training tetap perlu dibandingkan dengan hasil pengujian pada data testing untuk memastikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting.

4.10 Analisis Kesalahan Klasifikasi

Kesalahan klasifikasi paling banyak terjadi di (Segar-Tidak segar) dan (Tidak segar-Busuk). Hal ini disebabkan oleh kemiripan visual pada kedua kelas tersebut, terutama pada:

1. Tekstur permukaan jeruk

Tekstur permukaan jeruk pada kelas (segar-tidak segar) dan (tidak segar-busuk) memiliki kemiripan pola yang cukup tinggi, sehingga sulit dibedakan secara jelas oleh sistem. Pada jeruk tidak segar, permukaan mulai mengalami perubahan seperti sedikit keriput, penurunan kekenyalan, dan munculnya bercak halus. Sementara itu, pada jeruk busuk, kerusakan tekstur menjadi lebih lanjut seperti permukaan yang lebih lembek, keriput yang lebih jelas, serta adanya bagian yang rusak. perbedaan tingkat kerusakan tersebut sering kali tidak terlalu kontras dalam citra digital, terutama setelah dikonversi ke grayscale untuk ekstraksi fitur GLCM. Hal ini menyebabkan nilai fitur tekstur seperti contrast, dissimilarity, homogeneity, dan energy pada kedua kelas memiliki rentang yang saling tumpang tindih (overlap). Akibatnya, algoritma KNN mengalami kesulitan dalam menentukan batas yang tegas antara kelas tidak segar dan busuk, sehingga meningkatkan kemungkinan terjadinya kesalahan klasifikasi.

2. Perubahan warna yang bertahap

Warna jeruk tidak (segar-busuk) dan (tidak segar-busuk) sering berada pada rentang warna yang berdekatan (hijau-kuning) dan (kuning-coklat), sehingga sulit dibedakan secara tegas. Akibatnya, model cenderung mengalami kebingungan ketika batas visual antar kelas tidak memiliki perbedaan yang signifikan.

3. Keterbatasan Fitur Tekstur (GLCM)

Metode Gray Level Co-occurrence Matrix (GLCM) digunakan untuk mengekstraksi fitur tekstur seperti *contrast*, *homogeneity*, *energy*, dan *correlation*. Meskipun efektif untuk menggambarkan pola tekstur secara umum, GLCM memiliki keterbatasan, yaitu Sensitif terhadap pola tekstur global, tetapi kurang optimal untuk membedakan tingkat keparahan kerusakan dan tidak mampu menangkap informasi lokal

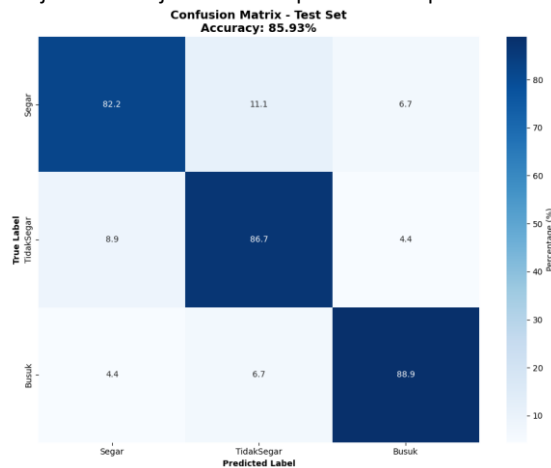
yang sangat halus, seperti penyebaran bercak kecil atau area busuk yang belum dominan.

4. Overlap Fitur Warna

Ekstraksi fitur warna menggunakan HSV Color Histogram juga berkontribusi terhadap kesalahan klasifikasi, karena Nilai Hue pada jeruk segar dan tidak segar sering berada pada rentang yang sama dan Perubahan Saturation dan Value tidak selalu konsisten, tergantung pada pencahayaan saat pengambilan citra.

4.11 Evaluasi

Berdasarkan hasil evaluasi pada Gambar 9, model yang tidak menggunakan fitur HSV masih menghasilkan beberapa kesalahan klasifikasi, terutama pada kelas yang memiliki karakteristik visual yang mirip. Hal ini menunjukkan bahwa penggunaan satu jenis fitur saja belum cukup untuk merepresentasikan seluruh



Gambar 13. Confusion Matrix tanpa HSV

Fitur Gray Level Co-occurrence Matrix (GLCM) mampu menangkap karakteristik tekstur, seperti kekasaran, kontras, dan pola permukaan objek. Namun, GLCM tidak mempertimbangkan informasi warna yang menjadi indikator penting dalam menentukan tingkat kesegaran buah. Di sisi lain, fitur HSV (Hue, Saturation, Value) secara efektif merepresentasikan informasi warna, seperti perubahan warna kulit jeruk dari segar hingga busuk. Dengan menggabungkan GLCM dan HSV, sistem dapat memanfaatkan keunggulan keduanya, yaitu informasi tekstur dan warna secara bersamaan. Kombinasi ini menghasilkan representasi fitur yang lebih komprehensif, sehingga meningkatkan kemampuan

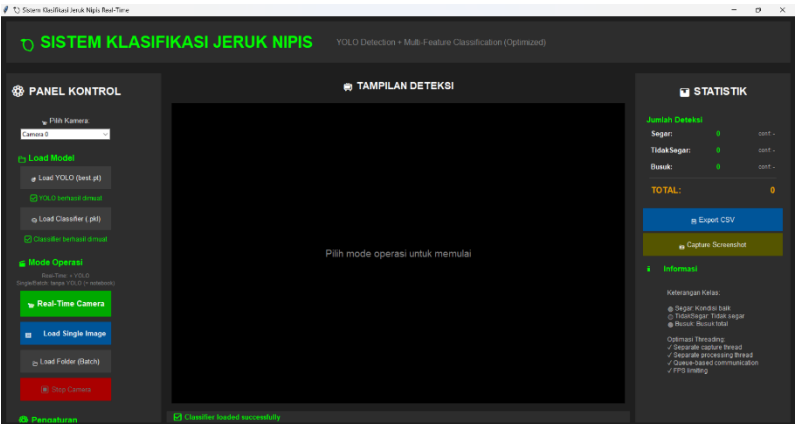
model dalam membedakan kelas yang memiliki kemiripan tekstur namun berbeda warna, maupun sebaliknya.

Oleh karena itu, penggabungan fitur GLCM dan HSV diharapkan dapat meningkatkan akurasi klasifikasi serta mengurangi kesalahan prediksi, terutama paa kelas yang sulit dibedakan jika hanya menggunakan satu jenis fitur.

4.12 Perancangan Graphical User Interface

Perancangan *Graphical User Interface* (GUI) pada sistem klasifikasi jeruk nipis ini bertujuan untuk menyediakan tampilan antarmuka yang intuitif, informatif, dan mudah dioperasikan oleh pengguna. GUI dikembangkan menggunakan library Tkinter berbasis Python dengan pendekatan desain multi-panel agar seluruh fungsi sistem dapat diakses dalam satu jendela aplikasi.

Secara keseluruhan, perancangan GUI ini dirancang agar operator dapat menjalankan seluruh proses mulai dari pemuatan model, pemilihan input, hingga pembacaan hasil klasifikasi secara efisien dalam satu tampilan yang terintegrasi.



Gambar 14. Tampilan Gui

Secara keseluruhan, perancangan GUI ini dirancang agar operator dapat menjalankan seluruh proses mulai dari pemuatan model, pemilihan input, hingga pembacaan hasil klasifikasi secara efisien dalam satu tampilan yang terintegrasi.

No	Fitur Interface	Lokasi Interface	Fungsi Utama
1	Pilih Kamera	Panel Kiri Atas	Dropdown untuk memilih perangkat kamera yang akan digunakan sebagai sumber input gambar secara <i>real-time</i> .

2	Load YOLO (.pt)	Panel Kiri	Tombol untuk memuat model deteksi objek berbasis YOLO (<i>best.pt</i>) yang digunakan untuk mendeteksi keberadaan jeruk nipis dalam frame gambar.
3	YOLO Terhasi Dimuat	Panel Kiri	Indikator status yang menampilkan konfirmasi bahwa model YOLO telah berhasil dimuat ke dalam.
4	Load Classifier (.pkl)	Panel Kiri	Indikator status yang menampilkan konfirmasi bahwa model KNN Classifier telah berhasil dimuat.
5	Classifier Berhasil Dimuat	Panel Kiri	secara langsung dari tangkapan kamera.
6	Real-Time Camera	Panel Kiri Tengah	Tombol untuk mengaktifkan mode kamera secara <i>real-time</i> , sehingga sistem dapat mendeteksi dan mengklasifikasikan jeruk nipis secara langsung dari tangkapan kamera.
7	Load Single Image	Panel Kiri Tengah	Tombol untuk memuat satu gambar statis dari penyimpanan komputer guna dilakukan proses deteksi dan klasifikasi secara individual.
8	Load Folder (Batch)	Panel Kiri Tengah	Tombol untuk memuat sekumpulan gambar dari satu folder sekaligus, memungkinkan proses deteksi dan klasifikasi secara massal (<i>batch processing</i>)
9	Stop Camera	Panel Kiri Bawah	Tombol untuk menghentikan proses pengambilan gambar dari kamera secara <i>real-time</i> dan mengakhiri sesi deteksi.
10	Tampilan Deteksi	Panel Tengah	Area utama yang menampilkan visual frame kamera atau gambar yang sedang diproses,

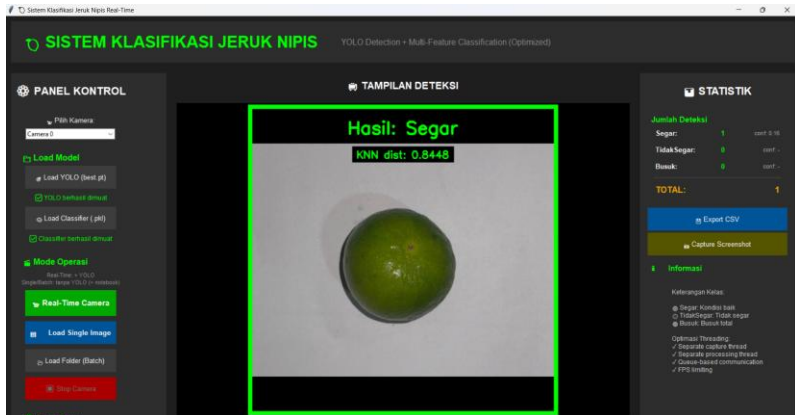
			lengkap dengan <i>bounding box</i> dan label hasil klasifikasi (Segar, Tidak Segar, atau Busuk).
11	Jumlah Deteksi (Statistik)	Panel Kanan Atas	Menampilkan rekap jumlah deteksi per kelas secara otomatis, meliputi total jeruk Segar, Tidak Segar, Busuk, serta total keseluruhan deteksi yang telah dilakukan.
12	Export CSV	Panel Kanan	Tombol untuk mengekspor seluruh data hasil deteksi dan klasifikasi ke dalam format file .csv sebagai dokumentasi dan keperluan analisis lebih lanjut.
13	Capture Screenshot	Panel Kanan	Tombol untuk mengambil tangkapan layar dari tampilan deteksi yang sedang aktif sebagai bukti visual hasil klasifikasi
14	Informasi Sistem	Panel Kanan Bawah	Menampilkan keterangan kelas (Segar, Tidak Segar, Busuk) beserta informasi teknis sistem seperti <i>optimasi threading, separate capture thread, dan queue-based communication</i> .

Table 15. Fitur Tombol dan Fungsinya

Pengujian visualisasi pada antarmuka grafis (GUI) bertujuan untuk memverifikasi bahwa sistem klasifikasi jeruk nipis yang telah dibangun dapat menampilkan hasil deteksi dan klasifikasi secara real-time dengan tampilan yang informatif dan mudah dipahami oleh pengguna.

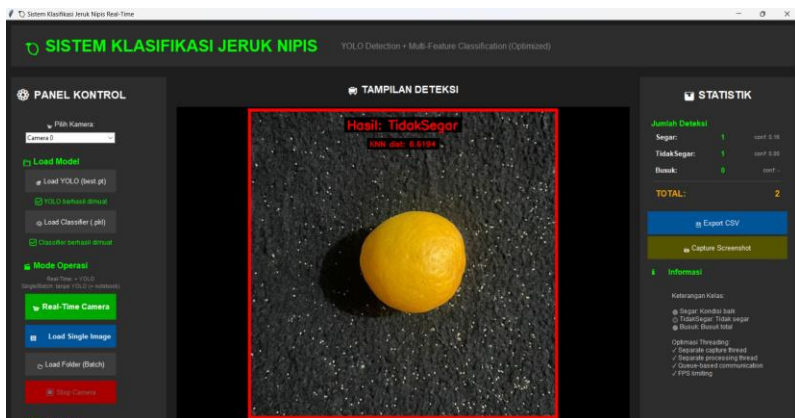
4.12.1 Visualisasi Pengujian GUI (*Load Image*)

Pengujian visualisasi pada mode *Load Image* bertujuan untuk memverifikasi bahwa sistem klasifikasi jeruk nipis yang telah dibangun dapat menampilkan hasil deteksi dan klasifikasi secara akurat terhadap gambar statis yang dimuat dari penyimpanan komputer. Pada mode ini, sistem memproses gambar secara langsung menggunakan pipeline yang identik dengan proses pelatihan.



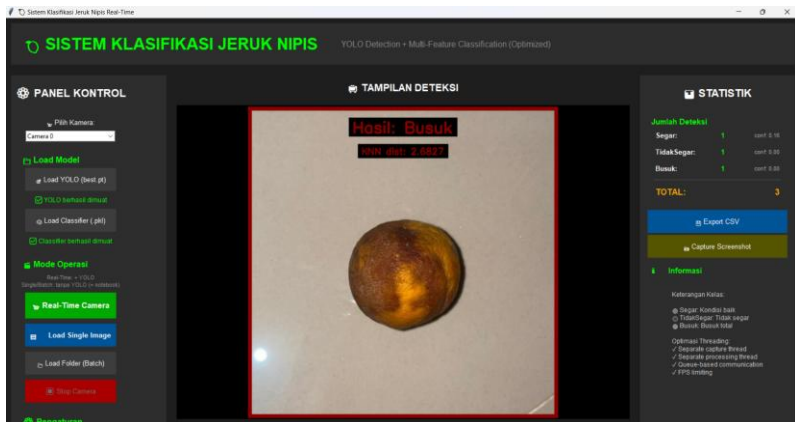
Gambar 15. Load Image Jeruk Segar

Pada Gambar 12, sistem berhasil mengklasifikasikan citra jeruk nipis kelas Segar dengan tepat. Hasil klasifikasi ditampilkan dalam bentuk *border* berwarna hijau mengelilingi gambar disertai label kelas di bagian atas. Keberhasilan ini didukung oleh karakteristik visual kelas Segar yang sangat khas, yaitu warna hijau cerah dengan nilai Hue dan Saturation yang tinggi, serta tekstur permukaan yang halus dan homogen yang tercermin dari nilai GLCM *contrast* rendah dan *homogeneity* tinggi.



Gambar 16. Load Image Jeruk Tidak Segar

Pada Gambar 13, sistem berhasil mengklasifikasikan citra jeruk nipis kelas TidakSegar dengan benar. Perubahan warna kulit jeruk yang mulai menguning dan tekstur permukaan yang sedikit berubah menghasilkan nilai fitur HSV dan GLCM yang berbeda dari kelas Segar, sehingga KNN dapat membedakan kedua kelas tersebut. Hasil ini sesuai dengan analisis metrik performa pada Tabel 19, di mana kelas TidakSegar memperoleh F1-Score sebesar 96%, menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali kelas ini.



Gambar 17. Load Image Jeruk Busuk

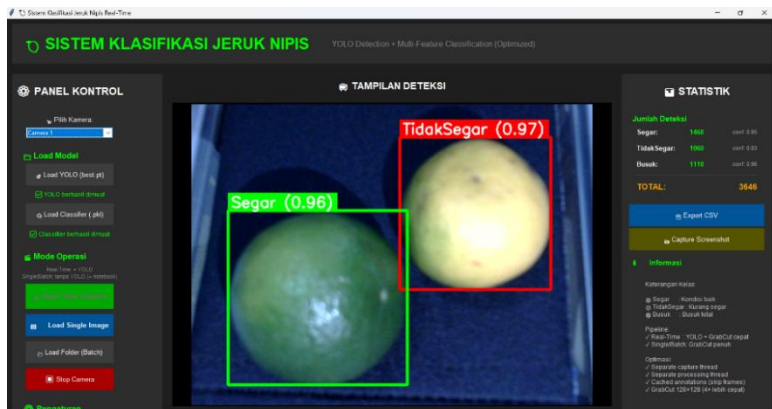
Pada Gambar 14, sistem berhasil mengklasifikasikan citra jeruk nipis kelas Busuk dengan tepat. Degradasi warna yang sangat mencolok ke arah coklat atau kuning tua serta perubahan tekstur permukaan akibat proses pembusukan menghasilkan pola fitur yang sangat berbeda dari kelas lainnya. Hal ini sesuai dengan hasil pengujian pada data uji di mana kelas Busuk memperoleh *recall* sebesar 100%, yang berarti seluruh sampel kelas Busuk berhasil dikenali dengan benar oleh model.

Secara keseluruhan, pengujian mode *Load Image* pada ketiga kelas menunjukkan bahwa sistem mampu mengklasifikasikan kualitas jeruk nipis secara akurat dari gambar statis. Hasil ini konsisten dengan akurasi model sebesar 97,04% yang diperoleh pada data uji, dan membuktikan bahwa integrasi antara model KNN yang telah dilatih dengan antarmuka GUI berjalan dengan baik.

4.12.2 Visualisasi Pengujian GUI (*Real Time*)

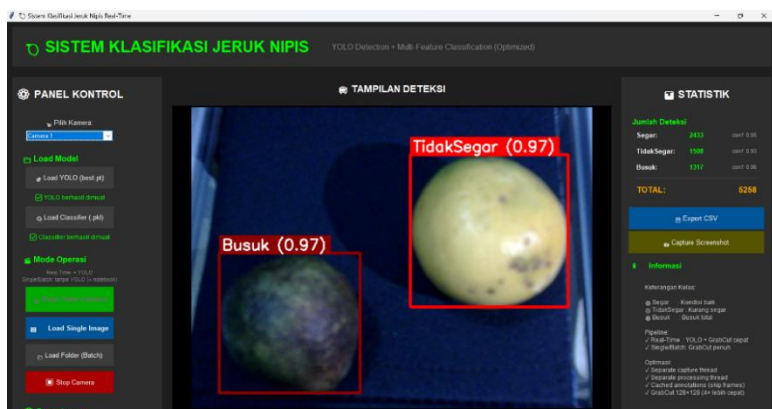
Gambar 15, 16 dan 17 menampilkan hasil pengujian antarmuka GUI sistem klasifikasi jeruk nipis pada mode *real-time*. Pengujian dilakukan dengan menempatkan dua buah jeruk nipis secara bersamaan di depan kamera, di mana

sistem berhasil mendeteksi dan mengklasifikasikan kedua objek secara simultan dalam satu *frame*.



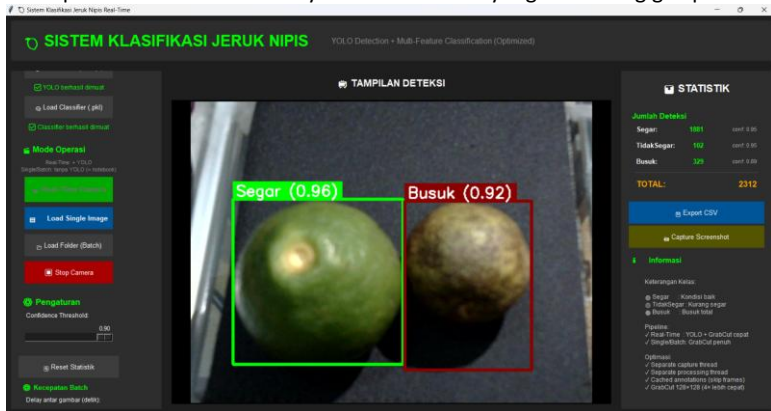
Gambar 18. Real time (Jeruk Segar dan Tidak Segar)

Pada Gambar 15, sistem mendeteksi dua objek jeruk nipis dengan hasil klasifikasi kelas Segar (*confidence* 0.96) ditandai kotak hijau, dan kelas TidakSegar (*confidence* 0.97) ditandai kotak merah. Tingginya nilai *confidence* pada kedua deteksi menunjukkan bahwa model mampu membedakan kondisi jeruk nipis yang masih segar dengan yang mulai menurun kualitasnya secara bersamaan dalam satu pengambilan gambar.



Gambar 19. Real Time (Jeruk Tidak Segar dan Busuk)

Pada Gambar 16, sistem menguji kombinasi jeruk kelas Busuk (*confidence* 0.97) dan TidakSegar (*confidence* 0.97). Kedua objek berhasil diklasifikasikan dengan nilai *confidence* yang tinggi. Hasil ini menunjukkan kemampuan sistem dalam membedakan dua kondisi penurunan kualitas yang berbeda tingkat, yaitu jeruk yang mulai tidak segar dan jeruk yang telah mengalami pembusukan penuh, meskipun secara visual keduanya memiliki warna yang cenderung gelap.



Gambar 20. Real Time (Jeruk Segar dan Busuk)

Pada Gambar 17, sistem mendeteksi kombinasi kelas Segar (*confidence* 0.96) dan Busuk (*confidence* 0.92). Perbedaan visual yang kontras antara kedua kelas ini jeruk segar berwarna hijau cerah dan jeruk busuk berwarna gelap kecoklatan membantu model memberikan hasil klasifikasi yang tepat. Panel statistik pada pengujian ini mencatat total akumulasi sebesar 2.312 deteksi, menunjukkan sistem berjalan secara stabil dan konsisten selama sesi pengujian berlangsung.

4.12.3 Performa Sistem pada Kondisi Conveyor Bergerak

Berdasarkan hasil pengujian, sistem mengalami penurunan performa deteksi pada kondisi *conveyor* bergerak, terutama pada kelas TidakSegar dan Busuk. Hal ini disebabkan oleh beberapa faktor.

1. Pertama, pergerakan *conveyor* menimbulkan *motion blur* pada *frame* kamera sehingga detail visual objek berkurang.
2. Kedua, kelas Segar memiliki karakteristik warna hijau cerah yang lebih mudah dikenali meskipun dalam kondisi blur, sedangkan kelas Tidak Segar dan Busuk memiliki warna yang lebih gelap dan kurang kontras terhadap latar belakang.

3. Ketiga, data pelatihan yang digunakan seluruhnya merupakan gambar statis sehingga model belum mampu menggeneralisasi kondisi objek yang bergerak.
4. Cahaya terlalu terang, redup, atau terdapat bayangan sehingga nilai HSV berubah dan memengaruhi hasil klasifikasi.

Secara keseluruhan, penurunan performa pada kondisi conveyor bergerak terutama disebabkan oleh ketidaksesuaian antara distribusi data pelatihan (gambar statis, kondisi terkontrol) dengan kondisi pengujian nyata (objek bergerak, pencahayaan bervariasi). Kelas Segar masih dapat dideteksi dengan baik karena warna hijau cerahnya memberikan kontras tinggi terhadap latar belakang conveyor, sehingga YOLO tetap mampu mengenali objek meskipun terjadi motion blur. Sementara itu, kelas TidakSegar dan Busuk memiliki warna yang lebih gelap dan kurang kontras, sehingga lebih rentan terhadap gangguan pergerakan.

Bab 5 Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan mengenai implementasi metode Gray Level Co-occurrence Matrix (GLCM) dan K-Nearest Neighbor (KNN) untuk klasifikasi kualitas buah jeruk nipis, dapat ditarik kesimpulan sebagai berikut.

1. Metode GLCM berhasil diimplementasikan untuk mengekstraksi fitur tekstur citra buah jeruk nipis pada empat arah sudut (0° , 45° , 90° , dan 135°). Fitur yang dihasilkan, meliputi contrast, dissimilarity, homogeneity, dan energy, terbukti mampu membedakan karakteristik tekstur antara kelas segar, tidak segar, dan busuk. Jeruk segar memiliki nilai contrast rendah (mean 0,1481) dan homogeneity tinggi (mean 0,9289), sedangkan jeruk busuk menunjukkan pola yang berbeda dengan contrast mean 0,0875 dan energy mean 0,3936. Perbedaan pola fitur GLCM antar kelas ini membuktikan bahwa metode tersebut memiliki daya diskriminatif yang baik untuk mendukung proses klasifikasi.
2. Algoritma KNN berhasil diterapkan untuk mengklasifikasikan kualitas buah jeruk nipis ke dalam tiga kelas, yaitu segar, tidak segar, dan busuk. Berdasarkan pengujian 15 kombinasi hyperparameter (nilai $K = 3, 5, 7, 9, 11$ dan fungsi jarak Euclidean, Manhattan, Minkowski $p=3$), kombinasi terbaik diperoleh menggunakan jarak Manhattan dengan $K = 3$ yang menghasilkan akurasi sebesar 98,52%, presisi 98,53%, recall 98,52%, dan F1-Score 98,52%. Hasil ini menunjukkan bahwa sistem klasifikasi yang dibangun mampu membedakan kondisi kualitas jeruk nipis secara konsisten dan akurat.
3. Evaluasi menggunakan 10-Fold Stratified Cross Validation menghasilkan rentang akurasi antara 93,33% hingga 100% dengan standar deviasi sebesar $\pm 2,78\%$, yang mengindikasikan model memiliki kemampuan generalisasi yang baik dan stabil. Pada pengujian data uji independen (135 citra), model mencapai akurasi sebesar 97,04% dengan hanya 4 kesalahan klasifikasi dari 135 sampel, di mana kelas busuk berhasil diklasifikasikan dengan sempurna (100% recall).
4. Antarmuka grafis (GUI) yang dikembangkan menggunakan library Tkinter berbasis Python berhasil mengintegrasikan seluruh proses klasifikasi dalam satu tampilan yang intuitif dan mudah dioperasikan. Sistem mendukung tiga mode operasi, yaitu real-time melalui kamera dengan deteksi objek berbasis YOLO, load single image, dan batch folder. Pengujian visualisasi menunjukkan bahwa sistem mampu mendeteksi

dan mengklasifikasikan lebih dari satu objek jeruk nipis secara simultan dalam satu frame dengan menampilkan bounding box, label kelas, dan nilai confidence secara langsung pada tampilan layar. Namun demikian, performa sistem pada kondisi conveyor bergerak masih memerlukan optimasi lebih lanjut, terutama untuk kelas TidakSegar dan Busuk yang lebih rentan terhadap gangguan motion blur.

5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut.

1. Penambahan jumlah dan variasi dataset disarankan untuk dilakukan pada penelitian selanjutnya, termasuk pengambilan citra pada berbagai kondisi pencahayaan, latar belakang, dan kamera yang berbeda, sehingga model lebih robust terhadap variasi kondisi nyata di lapangan.
2. Eksplorasi metode ekstraksi fitur tambahan, seperti Local Binary Pattern (LBP) atau fitur deep learning berbasis CNN, dapat dipertimbangkan untuk meningkatkan kemampuan sistem dalam membedakan kelas yang memiliki kemiripan visual tinggi, terutama antara kelas segar dan tidak segar.
3. Implementasi sistem pada perangkat keras nyata perlu dilakukan untuk memvalidasi performa sistem dalam kondisi sortir buah secara real-time di lingkungan produksi pertanian.
4. Perbandingan performa dengan algoritma klasifikasi lain, seperti Support Vector Machine (SVM) atau Random Forest, dapat dilakukan sebagai bahan evaluasi untuk memastikan bahwa KNN merupakan pilihan algoritma yang paling optimal untuk dataset dan karakteristik fitur yang digunakan dalam penelitian ini.
5. Pengembangan antarmuka pengguna (GUI) yang lebih lengkap dan mudah digunakan oleh operator non-teknis di lapangan dapat menjadi langkah lanjutan yang penting agar sistem ini dapat diadopsi secara praktis oleh petani atau pelaku usaha di industri pertanian jeruk nipis.

Daftar Pustaka

- [1] Sutisna and F. Chairulloh, "KLASIFIKASI TINGKAT KEMATANGAN BUAH NANAS BERDASARKAN TEKSTUR GRAY LEVEL CO-OCCURRENCE MATRIX DENGAN METODE SUPPORT VECTOR MACHINE," *J. Inform. Teknol. dan Sains*, vol. 4, no. 4, pp. 296–301, Nov. 2022, doi: 10.51401/jinteks.v4i4.2047.
- [2] C. Suryanti, Mg. Rohman, K. Kunci -Buah apel, and K. Knn, "Klasifikasi Kualitas Buah Apel Berdasarkan Warna dan Bentuk Menggunakan Metode KNN," *Gener. J.*, vol. 8, no. 1, pp. 2580–4952, 2023.
- [3] J. P. Wiastopo and I. Imelda, "Deteksi Kesegaran Ikan Kembung dengan Metode KNN Berdasarkan Fitur GLCM dan RGB-HSV," *J. Ticom Technol. Inf. Commun.*, vol. 13, no. 1, pp. 10–16, Sep. 2024, doi: 10.70309/TICOM.V13I1.137.
- [4] K. Yu *et al.*, "Advances in Computer Vision and Spectroscopy Techniques for Non-Destructive Quality Assessment of Citrus Fruits: A Comprehensive Review," *Foods 2025, Vol. 14, Page 386*, vol. 14, no. 3, p. 386, Jan. 2025, doi: 10.3390/FOODS14030386.
- [5] D. Ikhsan, E. Utami, and F. W. Wibowo, "METODE KLASIFIKASI MUTU GREENBEAN KOPI ARABIKA LANANG DAN BIASA MENGGUNAKAN K-NEAREST NEIGHBOR BERDASARKAN BENTUK," *J. Ilm. SINUS*, vol. 18, no. 2, pp. 1–8, Jul. 2020, doi: 10.30646/SINUS.V18I2.456.
- [6] A. Fadjeri,) Bayu, A. Saputra,) Dicki, K. A. Ariyanto, and L. Kurniatin, "Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital 1*)," *J. Ilm. Sinus*, vol. 20, no. 2, pp. 2548–4028, 2022, doi: 10.30646/sinus.v20i2.601.
- [7] "View of KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI BERDASARKAN DETEKSI WARNA MENGGUNAKAN METODE KNN DAN PCA", Accessed: Nov. 25, 2025. [Online]. Available: <https://ejournal.lppmunsera.org/index.php/jsii/article/view/3638/1782>
- [8] "View of Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV)." Accessed: Nov. 25, 2025. [Online]. Available: <https://join.if.uinsgd.ac.id/index.php/join/article/view/v1i15/15>
- [9] R. F. Naibaho and I. P. Sari, "Implementasi Metode Gray Level Co-Occurrence Matrix Menganalisis Tekstur Kulit Wajah," *sudo J. Tek. Inform.*, vol. 3, no. 4, pp. 172–182, Jun. 2024, doi: 10.56211/SUDO.V3I4.668.
- [10] "View of Klasifikasi Aroma Alkohol Menggunakan Metode KNN." Accessed: Jun. 12, 2025. [Online]. Available:

<https://jurnal.yoctobrain.org/index.php/ijodas/article/view/12/9>

- [11] F. Mahrus Fathoni, C. Aji Putra, A. Lina Nurlaili, F. Ilmu Komputer, U. Pembangunan Nasional, and J. Timur, "KLASIFIKASI PENYAKIT DAUN ANGGUR MENGGUNAKAN METODE K-NEAREST NEIGHBOR BERDASARKAN GRAY LEVEL CO-OCCURRENCE MATRIX," *Biner J. Ilm. Inform. dan Komput.*, vol. 3, no. 1, pp. 8–15, Jan. 2024, doi: 10.32699/BINER.V3I1.6332.
- [12] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A critical look at the current train/test split in machine learning".
- [13] N. Blesyova and F. N. Hasan, "ANALISIS SENTIMEN TERHADAP BEA CUKAI MENGGUNAKAN SUPPORT VECTOR MACHINE DAN K-FOLD CROSS VALIDATION," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 6, pp. 12051–12056, Nov. 2024, doi: 10.36040/JATI.V8I6.11774.
- [14] J. S. W. Hutaaruk, T. Matulatan, and N. Hayaty, "Deteksi Kendaraan secara Real Time menggunakan Metode YOLO Berbasis Android," *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 9, no. 1, pp. 8–14, May 2020, doi: 10.31629/SUSTAINABLE.V9I1.1401.

Biodata



Nama : Ezi Alfindo Perangin-Angin
TTL : 07 Nov 2003
Agama : Kristen Protestan
Alamat :

Email : ezialfindo09@gmail.com
Riwayat Pendidikan SMA/SMK : SMAN 01 Bintang
SMP : SMPN 12 Bintang Utara

Lampiran

https://drive.google.com/drive/folders/1_X2gUNWzV8D3o0scFXQL_MnVOyZwShfh?usp=sharing