

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

4.1.1 Lingkungan Implementasi

Sistem klasifikasi tingkat kematangan buah alpukat pada penelitian ini diimplementasikan menggunakan bahasa pemrograman Python yang dijalankan pada lingkungan Google Colaboratory. Lingkungan ini dipilih karena menyediakan sumber daya komputasi berbasis cloud yang dapat langsung terhubung dengan penyimpanan Google Drive, sehingga proses pembacaan dataset citra dapat dilakukan tanpa memerlukan instalasi maupun konfigurasi perangkat keras tambahan pada sisi pengguna.

Proses implementasi memanfaatkan beberapa pustaka pendukung yang saling melengkapi pada tiap tahapan pipeline penelitian. Pustaka OpenCV digunakan untuk seluruh operasi pengolahan citra, meliputi pembacaan berkas gambar, resizing, ekstraksi region of interest, hingga konversi ruang warna. Pustaka NumPy dan Pandas berperan dalam manipulasi data numerik serta penyusunan struktur tabular hasil ekstraksi fitur, sementara pustaka SciPy digunakan secara khusus untuk menghitung nilai skewness dan kurtosis pada distribusi warna tiap channel. Selanjutnya, tahap pemodelan dan evaluasi memanfaatkan scikit-learn, yang menyediakan modul KNeighborsClassifier, MinMaxScaler, train_test_split, serta sejumlah fungsi metrik evaluasi. Visualisasi hasil, baik berupa citra tahapan

preprocessing maupun grafik akurasi, dihasilkan menggunakan Matplotlib dan Seaborn.

Dataset citra disusun ke dalam tiga folder berdasarkan kelas kematangan, yaitu mentah, setengah_matang, dan matang, yang masing-masing memuat 70 berkas citra sehingga total keseluruhan mencapai 210 citra. Susunan ini konsisten dengan rancangan dataset yang telah dijelaskan pada Bab III. Pengecekan ketersediaan folder beserta jumlah berkas dilakukan pada tahap awal implementasi untuk memastikan tidak terdapat data yang hilang atau tidak terbaca sebelum memasuki tahap pra-pemrosesan.

4.1.2 Implementasi Alur Sistem

Implementasi alur sistem diawali dengan tahap verifikasi ketersediaan dataset pada Google Drive. Proses pengecekan folder menunjukkan bahwa ketiga kelas kematangan masing-masing memiliki 70 citra, sehingga total dataset yang tersedia berjumlah 210 citra dengan komposisi yang seimbang antar kelas. Kondisi dataset yang balanced ini penting untuk mencegah bias klasifikasi yang umumnya muncul akibat dominasi kelas mayoritas pada algoritma berbasis jarak seperti KNN.

Selanjutnya, seluruh 210 citra diproses melalui fungsi `extract_roi` yang menggabungkan tahap resizing menjadi ukuran 100×100 piksel dan ekstraksi ROI menggunakan GrabCut. Proses ini dijalankan secara iteratif terhadap setiap citra pada ketiga folder kelas, dan hasil verifikasi menunjukkan bahwa jumlah ROI yang terbentuk konsisten dengan jumlah label, yaitu masing-masing 210 data.

Konsistensi jumlah data pada tahap ini mengonfirmasi bahwa tidak terdapat citra yang gagal dibaca atau dilewati selama proses ekstraksi.

Tahap berikutnya adalah penerapan Gaussian Blur berkernel 5×5 terhadap seluruh 210 citra hasil ROI, dilanjutkan dengan normalisasi intensitas piksel ke rentang 0 hingga 1. Verifikasi numerik terhadap array hasil normalisasi menunjukkan nilai minimum 0,0 dan nilai maksimum 1,0, yang mengindikasikan bahwa proses normalisasi intensitas berjalan sebagaimana mestinya tanpa kesalahan komputasi pada seluruh data.

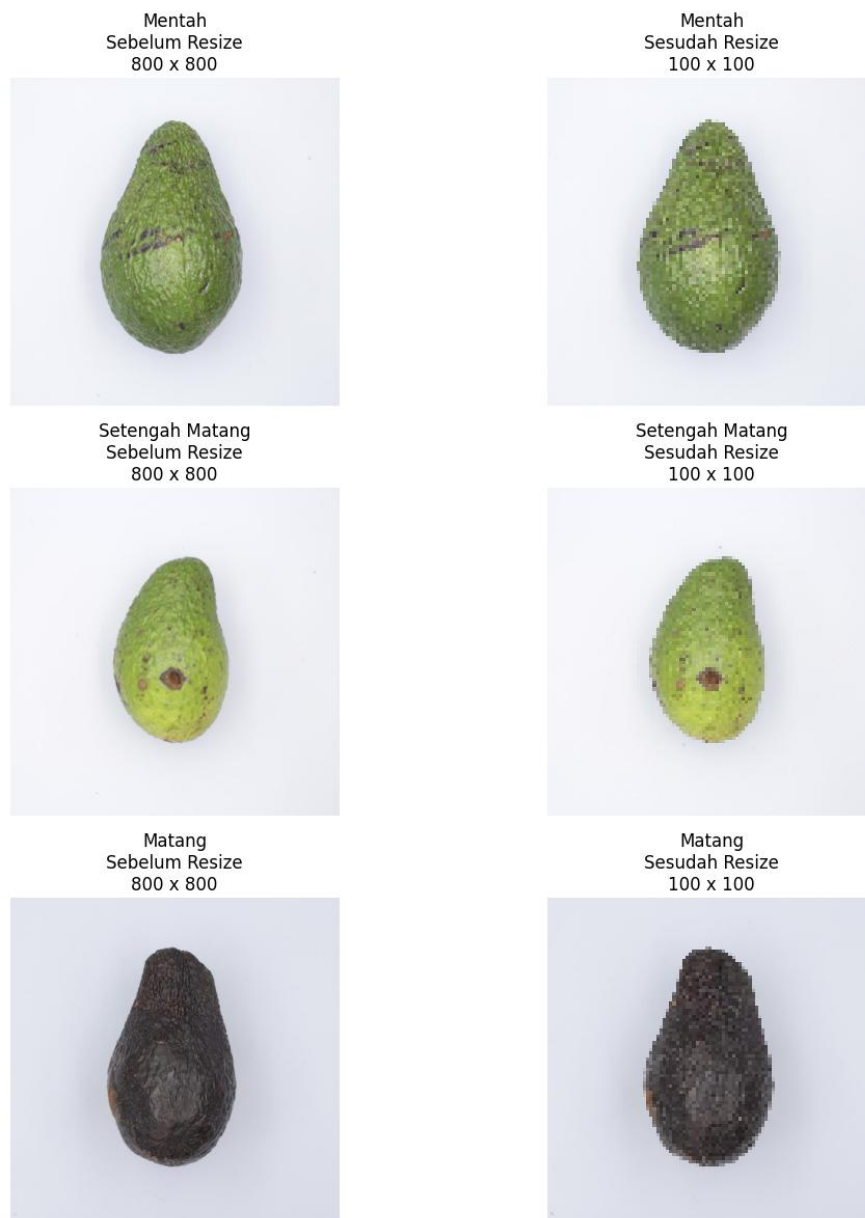
Citra yang telah dinormalisasi kemudian dikonversi dari ruang warna RGB ke HSV dan diekstraksi menjadi 27 fitur statistik. Hasil ekstraksi membentuk dataset berukuran 210 baris dan 28 kolom, yaitu 27 kolom fitur ditambah satu kolom label kelas. Dataset training dan dataset testing dibaca langsung dari folder yang telah dipisahkan sebelumnya. Selanjutnya dilakukan preprocessing, ekstraksi fitur HSV, normalisasi fitur menggunakan Min-Max Scaling, kemudian data training digunakan untuk membangun model KNN dan data testing digunakan untuk proses evaluasi. Alur implementasi tersebut berjalan secara berurutan tanpa penyimpangan dari desain sistem yang telah ditetapkan pada Bab III, dan setiap tahapan disertai proses verifikasi untuk memastikan validitas data sebelum dilanjutkan ke tahap berikutnya.

4.2 Hasil Preprocessing

4.2.1 Hasil Resizing

Tahap resizing dilakukan terhadap seluruh 210 citra dataset untuk menyeragamkan dimensi citra menjadi 100×100 piksel, terlepas dari resolusi asli

tiap berkas yang bervariasi tergantung sumber perolehannya. Perbandingan citra sebelum dan sesudah resizing untuk masing-masing kelas kematangan ditampilkan pada berikut.



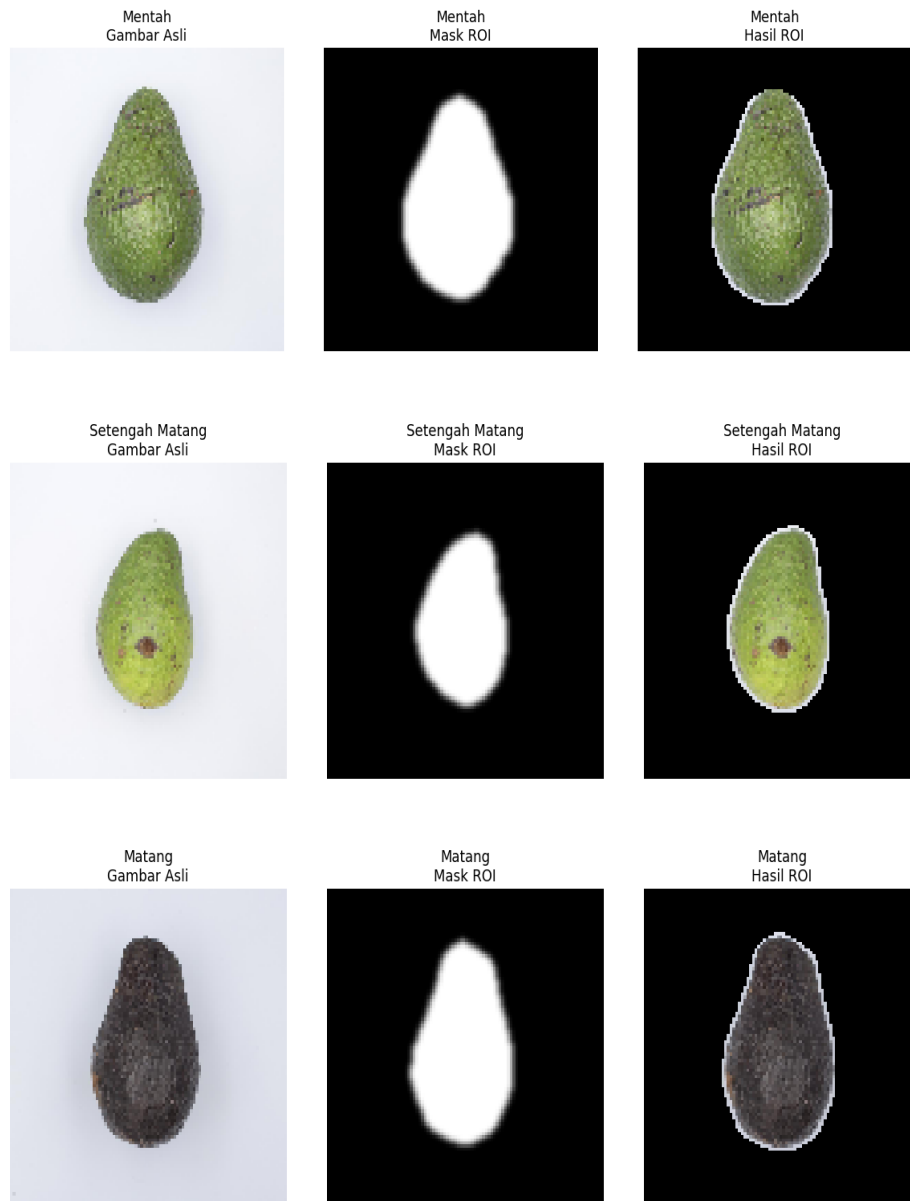
Gambar 4.1 Hasil Resizing Citra pada Setiap Kelas

Berdasarkan Gambar 4.1, seluruh citra pada ketiga kelas berhasil diseragamkan menjadi ukuran 100×100 piksel tanpa mengalami distorsi bentuk objek yang signifikan. Proporsi warna dan pola visual utama pada permukaan kulit alpukat tetap dapat dikenali meskipun resolusi citra diturunkan. Hal tersebut menunjukkan bahwa ukuran 100×100 piksel yang dipilih sudah memadai untuk mempertahankan informasi warna yang relevan bagi proses klasifikasi, sekaligus menekan beban komputasi pada tahap-tahap selanjutnya.

Selain aspek komputasi, penyeragaman ukuran turut berperan penting dalam menjaga konsistensi jumlah piksel yang dianalisis pada tahap ekstraksi fitur statistik. Apabila ukuran citra dibiarkan bervariasi, jumlah piksel yang menyusun setiap channel HSV akan berbeda antar sampel, yang berpotensi memengaruhi validitas perbandingan nilai statistik seperti mean dan standard deviation antar kelas. Dengan demikian, proses resizing menjadi fondasi penting yang menjamin setiap citra diperlakukan secara setara pada tahap pengolahan berikutnya.

4.2.2 Hasil ROI Extraction

Setelah proses resizing, tahap berikutnya adalah pemisahan objek buah alpukat dari latar belakang menggunakan metode GrabCut yang dikombinasikan dengan operasi morfologi opening dan closing serta pengambilan largest contour. Hasil ekstraksi ROI pada masing-masing kelas kematangan ditampilkan pada gambar berikut.



Gambar 4.2 Hasil Ekstraksi ROI pada Setiap Kelas

Gambar 4.2 memperlihatkan bahwa metode GrabCut berhasil memisahkan objek utama alpukat dari latar belakang pada ketiga kelas kematangan, ditandai dengan area di luar kontur objek yang telah dihitamkan (bernilai nol). Keberhasilan segmentasi ini bergantung pada inisialisasi rectangle yang mencakup sebagian besar area objek, sehingga algoritma GrabCut dapat melakukan estimasi model

foreground dan background secara iteratif dengan baik. Pada beberapa sampel, masih terdapat sisa area latar belakang berukuran kecil di sekitar tepi objek, namun proporsinya relatif tidak signifikan terhadap keseluruhan area citra.

Keberhasilan tahap ROI extraction memiliki peran penting terhadap validitas fitur warna yang diekstraksi pada tahap selanjutnya. Tanpa pemisahan objek dari latar belakang, statistik warna seperti mean dan variance pada channel H, S, dan V akan turut dipengaruhi oleh warna latar belakang yang tidak relevan dengan karakteristik kematangan buah, sehingga berpotensi menurunkan diskriminasi antar kelas. Dengan latar belakang yang telah dihilangkan, fitur statistik yang diekstraksi selanjutnya dapat lebih merepresentasikan karakteristik warna kulit alpukat secara murni.

4.2.3 Hasil Gaussian Blur

Tahap reduksi noise dilakukan menggunakan Gaussian Blur dengan ukuran kernel 5×5 terhadap seluruh citra hasil ROI extraction. Perbandingan visual antara citra sebelum dan sesudah proses Gaussian Blur pada setiap kelas ditampilkan pada gambar berikut.



Gambar 4.3 Hasil Penerapan Gaussian Blur pada Setiap Kelas

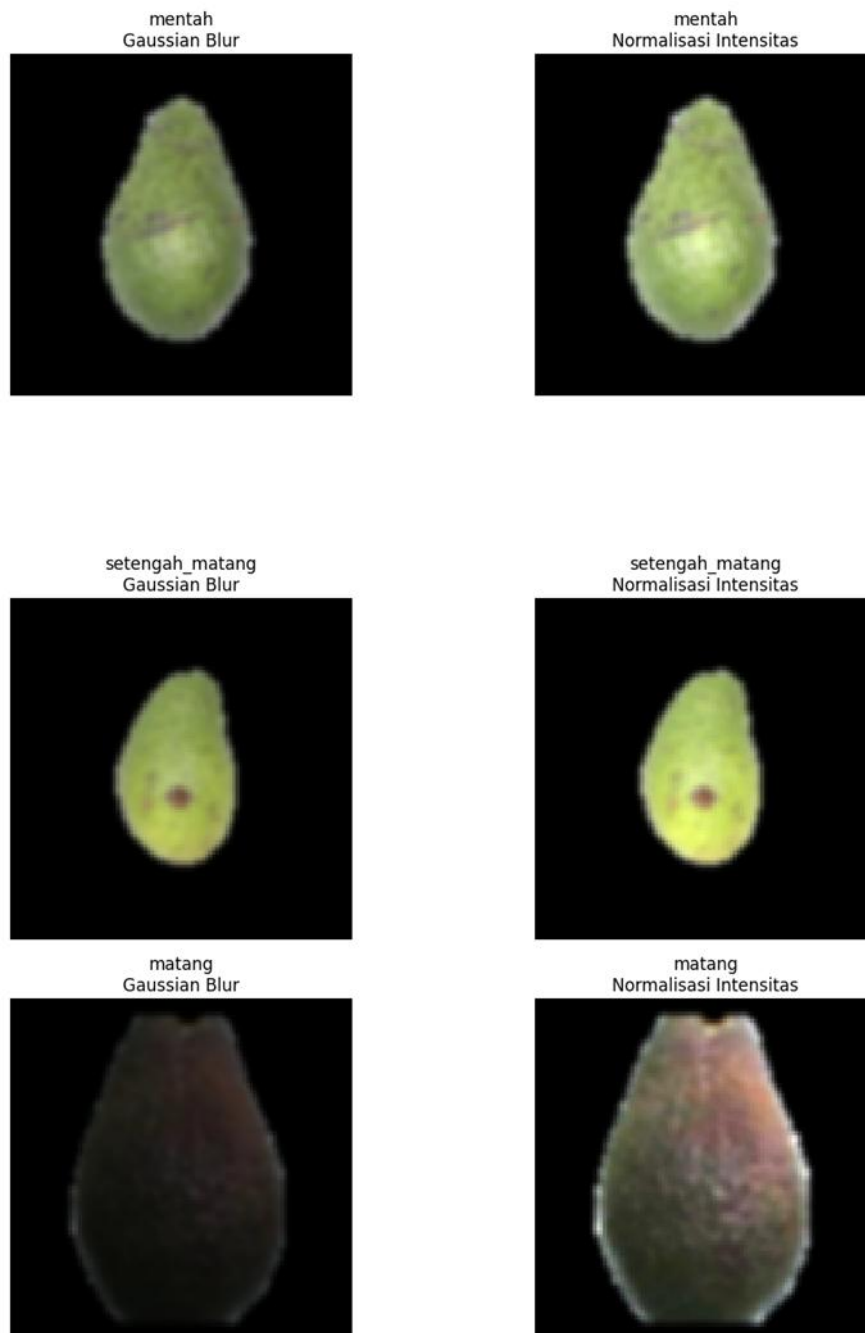
Berdasarkan Gambar 4.3, penerapan Gaussian Blur menghasilkan citra dengan tekstur yang lebih halus dibandingkan citra ROI sebelumnya, ditandai dengan berkurangnya variasi piksel bertetangga yang tajam akibat noise sensor

kamera maupun artefak kompresi citra. Meskipun demikian, distribusi warna global pada tiap kelas tetap dapat dipertahankan, mengingat ukuran kernel 5×5 tergolong kecil sehingga hanya menghaluskan variasi lokal tanpa menghilangkan pola warna dominan pada permukaan kulit buah.

Pengurangan noise pada tahap ini secara langsung berkontribusi terhadap stabilitas nilai fitur statistik yang akan dihitung pada tahap ekstraksi fitur, khususnya untuk parameter yang sensitif terhadap outlier seperti standard deviation dan kurtosis. Tanpa proses ini, variasi piksel yang bersifat acak berpotensi meningkatkan nilai variance secara semu, sehingga mengaburkan perbedaan variance yang sesungguhnya disebabkan oleh perubahan warna akibat proses pematangan.

4.2.4 Hasil Normalisasi Intensitas

Tahap akhir preprocessing adalah normalisasi intensitas piksel ke rentang 0 hingga 1, yang bertujuan menyeragamkan distribusi intensitas antar citra sehingga perbedaan kondisi pencahayaan saat pengambilan gambar dapat diminimalkan. Hasil perbandingan antara citra setelah Gaussian Blur dan setelah normalisasi intensitas ditampilkan pada gambar berikut.



Gambar 4.4 Hasil Normalisasi Intensitas pada Setiap Kelas

Secara visual, Gambar 4.4 tidak menunjukkan perbedaan yang mencolok dibandingkan citra hasil Gaussian Blur, karena proses normalisasi intensitas pada dasarnya hanya mengubah skala nilai piksel tanpa mengubah pola relatif antar

piksel dalam citra. Namun demikian, hasil verifikasi numerik terhadap keseluruhan array citra menunjukkan nilai minimum tepat 0,0 dan nilai maksimum tepat 1,0, yang mengonfirmasi bahwa proses normalisasi telah diterapkan secara konsisten pada seluruh 210 citra dalam dataset.

Normalisasi intensitas pada tahap ini berbeda dengan normalisasi fitur yang dilakukan setelah ekstraksi fitur statistik, karena keduanya bekerja pada level yang berbeda, yaitu level piksel citra dan level fitur numerik hasil ekstraksi. Normalisasi intensitas piksel berfungsi menyeragamkan kondisi pencahayaan sebelum konversi ruang warna dilakukan, sedangkan normalisasi fitur bertujuan menyeimbangkan kontribusi setiap fitur dalam perhitungan jarak Euclidean pada algoritma KNN. Kedua tahap normalisasi tersebut bersifat saling melengkapi dan sama-sama diperlukan untuk menghasilkan performa klasifikasi yang optimal.

4.3 Hasil Ekstraksi Fitur

4.3.1 Konversi RGB ke HSV

Seluruh 210 citra yang telah melalui tahap preprocessing dikonversi dari ruang warna RGB ke HSV menggunakan fungsi `cv2.cvtColor()` pada OpenCV. Sebelum konversi dilakukan, nilai piksel yang semula berada pada rentang 0 hingga 1 dikembalikan terlebih dahulu ke rentang 0 hingga 255 dalam tipe data `uint8`, mengingat fungsi konversi ruang warna pada OpenCV mensyaratkan format tersebut agar transformasi matematis dapat dihitung secara tepat.

Proses konversi menghasilkan tiga channel terpisah, yaitu Hue, Saturation, dan Value, masing-masing dalam bentuk matriks berukuran 100×100 piksel untuk setiap citra. Verifikasi terhadap proses ini menunjukkan bahwa konversi berhasil

dilakukan pada keseluruhan 210 citra tanpa kegagalan, sehingga seluruh data siap digunakan pada tahap ekstraksi fitur statistik berikutnya. Representasi HSV yang dihasilkan pada tahap ini menjadi dasar bagi perhitungan sembilan parameter statistik pada masing-masing channel yang dibahas pada subbab selanjutnya.

4.3.2 Hasil Ekstraksi 27 Fitur

Ekstraksi fitur menghasilkan 27 fitur statistik untuk setiap citra, yang terdiri atas mean, standard deviation, variance, skewness, kurtosis, minimum, maximum, median, dan interquartile range pada masing-masing channel H, S, dan V. Guna memberikan gambaran umum mengenai karakteristik warna pada setiap kelas kematangan, nilai rata-rata (mean) dari ketiga channel HSV untuk masing-masing kelas ditampilkan pada tabel berikut, sedangkan rincian lengkap 27 fitur untuk seluruh sampel dapat dilihat pada lampiran.

Tabel 4.1 Rata-Rata Nilai Fitur Mean Channel H, S, dan V per Kelas

Kelas	H_Mean	S_Mean	V_Mean
Mentah	21,81	0,0698	0,1487
Setengah Matang	18,98	0,1239	0,1675
Matang	47,17	0,0462	0,1097

Tabel 4.1 memperlihatkan pola yang menarik pada channel Hue, di mana nilai H_Mean kelas matang (47,17) jauh lebih tinggi dibandingkan kelas mentah (21,81) maupun setengah matang (18,98). Pola ini konsisten dengan karakteristik biokimia pematangan alpukat yang melibatkan degradasi klorofil, sehingga kulit buah mengalami pergeseran rona warna dari hijau menuju kecoklatan yang tercermin dalam peningkatan nilai Hue pada representasi HSV OpenCV. Perlu

dicatat bahwa karena konversi HSV pada penelitian ini dilakukan terhadap citra yang telah dinormalisasi ke rentang intensitas 0-1 (bertipe float32), nilai channel Hue yang dihasilkan OpenCV berada pada skala 0-360 derajat, sedangkan channel Saturation dan Value berada pada skala 0-1, berbeda dengan skala 0-255 yang umum digunakan pada citra 8-bit biasa. Menariknya, nilai H_Mean antara kelas mentah dan setengah matang relatif berdekatan, yang mengindikasikan bahwa perubahan rona warna pada awal proses pematangan berlangsung secara bertahap dan tidak semencolok perubahan menjelang kematangan penuh.

Pola yang berbeda ditunjukkan oleh channel Saturation, di mana kelas setengah matang justru memiliki nilai S_Mean tertinggi (0,1239) dibandingkan kelas mentah (0,0698) dan matang (0,0462). Kondisi tersebut mengindikasikan bahwa kejenuhan warna kulit alpukat mencapai puncaknya pada fase transisi menuju matang, kemudian menurun kembali seiring munculnya warna kecoklatan yang cenderung lebih pudar pada fase matang penuh. Sementara itu, channel Value menunjukkan tren peningkatan dari kelas matang (0,1097) menuju kelas mentah (0,1487) dan setengah matang (0,1675), yang mengindikasikan variasi tingkat kecerahan permukaan kulit turut berubah seiring proses pematangan, meskipun tidak seteratur pola pada channel Hue.

Secara keseluruhan, hasil ekstraksi fitur menunjukkan bahwa ketiga channel HSV, terutama Hue dan Saturation, memberikan informasi yang saling melengkapi dalam membedakan ketiga kelas kematangan. Kombinasi ke-27 fitur statistik dari ketiga channel tersebut selanjutnya digunakan sebagai representasi numerik untuk proses klasifikasi menggunakan algoritma KNN, dengan harapan variasi antar kelas

yang telah teridentifikasi pada tahap ini dapat dimanfaatkan secara optimal oleh algoritma klasifikasi berbasis jarak.

4.3.3 Hasil Normalisasi Fitur

Nilai 27 fitur hasil ekstraksi memiliki rentang dan satuan yang berbeda-beda, misalnya nilai mean dan median yang berkisar pada rentang puluhan hingga ratusan, sementara nilai skewness cenderung berada pada rentang yang jauh lebih kecil. Perbedaan skala tersebut berpotensi menyebabkan fitur dengan rentang nilai besar mendominasi perhitungan jarak Euclidean pada algoritma KNN. Oleh karena itu, seluruh fitur dinormalisasi menggunakan Min-Max Scaling ke rentang 0 hingga 1. Contoh hasil normalisasi untuk beberapa fitur representatif ditampilkan pada tabel berikut.

Tabel 4.2 Contoh Hasil Normalisasi Fitur Menggunakan Min-Max Scaling

Kelas	H_Mean	S_Mean	V_Mean
Mentah	0,1728	0,1982	0,3170
Setengah Matang	0,0549	0,1805	0,2179
Matang	0,2530	0,2766	0,4284

Berdasarkan Tabel 4.2, nilai-nilai fitur yang semula memiliki rentang beragam berhasil ditransformasikan ke rentang 0 hingga 1 tanpa menghilangkan pola relatif antar kelas yang telah teridentifikasi sebelumnya pada Tabel 4.1. Verifikasi terhadap keseluruhan dataset fitur yang telah dinormalisasi menunjukkan nilai minimum 0,0 dan nilai maksimum 1,0, yang mengonfirmasi bahwa proses Min-Max Scaling telah diterapkan secara benar pada seluruh 27 fitur dan 210 sampel data.

Proses normalisasi fitur ini menjadi tahap krusial sebelum data digunakan pada algoritma KNN, mengingat KNN menghitung jarak antar sampel menggunakan seluruh dimensi fitur secara bersamaan. Tanpa normalisasi, fitur dengan skala nilai besar seperti mean atau median akan memberikan kontribusi jarak yang jauh lebih dominan dibandingkan fitur dengan skala kecil seperti skewness, meskipun keduanya berpotensi sama-sama informatif untuk membedakan kelas kematangan. Dengan normalisasi yang telah dilakukan, seluruh fitur memberikan kontribusi yang lebih seimbang dalam proses perhitungan jarak pada tahap pemodelan.

4.4 Hasil Pemodelan

4.4.1 Pembagian Dataset

Dataset yang telah dinormalisasi berjumlah 210 sampel dibagi menjadi data latih dan data uji dengan proporsi 80% dan 20% dipisahkan secara manual menjadi folder training dan testing sebelum proses klasifikasi dilakukan. Hasil pembagian dataset ditampilkan pada tabel berikut.

Tabel 4.3 Distribusi Data Latih dan Data Uji

Kelas	Data Latih	Data Uji	Total
Mentah	56	14	70
Setengah Matang	56	14	70
Matang	56	14	70
Total	168	42	210

Tabel 4.3 menunjukkan bahwa hasil pembagian dataset menghasilkan 168 data latih dan 42 data uji, dengan komposisi yang tepat seimbang, yaitu 56 data latih dan 14 data uji untuk masing-masing kelas. Keseimbangan proporsi ini penting

untuk menghindari bias evaluasi yang umumnya terjadi apabila salah satu kelas terlalu mendominasi data uji, sehingga metrik evaluasi yang diperoleh pada tahap selanjutnya dapat merepresentasikan performa model secara adil terhadap ketiga kelas kematangan.

4.4.2 Pengujian Variasi Nilai K

Pengujian dilakukan terhadap sembilan variasi nilai k, yaitu 1, 2, 3, 5, 7, 9, 11, 13, dan 15, dengan metrik jarak Euclidean dan pembobotan seragam antar tetangga. Hasil akurasi yang diperoleh pada data uji untuk masing-masing nilai k ditampilkan pada tabel berikut.

Tabel 4.4 Hasil Pengujian Akurasi pada Variasi Nilai K

Nilai K	Accuracy
1	0,8333
2	0,7143
3	0,7381
5	0,6190
7	0,5952
9	0,6190
11	0,5952
13	0,6190
15	0,6190

Berdasarkan Tabel 4.4, nilai akurasi tertinggi diperoleh pada $k = 1$, yaitu sebesar 0,8333 atau 83,33%. Ketika nilai k dinaikkan menjadi 2, akurasi menurun menjadi 71,43%, kemudian sedikit meningkat pada $k = 3$ menjadi 73,81%. Selanjutnya, pada $k = 5$ hingga $k = 15$, akurasi cenderung berfluktuasi pada kisaran 59,52% hingga 61,90%, tanpa menunjukkan tren yang stabil. Hasil tersebut menunjukkan bahwa peningkatan nilai k pada dataset penelitian ini secara konsisten

menurunkan performa klasifikasi, dan penggunaan satu tetangga terdekat ($k = 1$) tetap memberikan hasil terbaik dibandingkan seluruh variasi k yang diuji.

4.4.3 Pemilihan Model Terbaik

Berdasarkan hasil pengujian variasi nilai k pada Tabel 4.4, nilai $k = 1$ dipilih sebagai konfigurasi model terbaik karena menghasilkan nilai akurasi tertinggi, yaitu 83,33% pada data uji. Model KNN dengan $k = 1$ kemudian digunakan sebagai model akhir untuk melakukan klasifikasi terhadap seluruh data uji sehingga hasil prediksinya menjadi dasar dalam evaluasi performa model pada subbab berikutnya.

Pemilihan nilai $k = 1$ menunjukkan bahwa keputusan berdasarkan satu tetangga terdekat memberikan hasil klasifikasi terbaik pada dataset ini. Meskipun demikian, penggunaan nilai k yang kecil memiliki sensitivitas yang lebih tinggi terhadap data yang menyimpang (outlier), sehingga pada penelitian selanjutnya dapat dilakukan pengujian menggunakan dataset yang lebih besar maupun teknik validasi lain seperti cross-validation untuk memperoleh model yang lebih robust.

4.5 Hasil Evaluasi Model

4.5.1 Akurasi

Model KNN dengan $k = 1$ menghasilkan nilai accuracy sebesar 0,8333 atau 83,33% pada data uji. Nilai tersebut menunjukkan bahwa sebanyak 35 dari 42 citra berhasil diklasifikasikan sesuai dengan kelas kematangannya, sedangkan 7 citra mengalami kesalahan klasifikasi.

Nilai akurasi sebesar 83,33% menunjukkan bahwa kombinasi ekstraksi fitur warna HSV dan algoritma K-Nearest Neighbor mampu mengidentifikasi tingkat kematangan buah alpukat dengan tingkat ketepatan yang cukup baik, meskipun masih terdapat ruang untuk perbaikan dibandingkan performa ideal.

4.5.2 Presisi

Nilai precision yang diperoleh menggunakan metode weighted average adalah sebesar 0,8567 atau 85,67%. Nilai tersebut menunjukkan bahwa sebagian besar prediksi yang dihasilkan model telah sesuai dengan kelas sebenarnya, meskipun masih terdapat sejumlah kesalahan prediksi (false positive) pada kelas-kelas tertentu, khususnya kelas setengah matang yang beberapa kali menerima prediksi keliru dari kelas lain.

4.5.3 Recall

Nilai recall weighted-average yang diperoleh sebesar 0,8333 atau 83,33%. Nilai tersebut menunjukkan bahwa secara rata-rata model mampu mengenali sekitar 83% sampel pada masing-masing kelas kematangan, namun dengan tingkat keberhasilan yang tidak merata antar kelas, sebagaimana akan dibahas lebih rinci melalui confusion matrix pada subbab berikutnya.

4.5.4 F1-Score

Nilai F1-score weighted-average yang diperoleh sebesar 0,8326 atau 83,26%. Nilai tersebut merupakan rata-rata harmonik (harmonic mean) antara nilai precision dan recall. F1-score digunakan untuk memberikan gambaran mengenai

keseimbangan kemampuan model dalam menghasilkan prediksi yang tepat sekaligus mengenali sampel pada setiap kelas.

Nilai F1-score sebesar 83,26% menunjukkan bahwa model K-Nearest Neighbor dengan $k = 1$ mampu memberikan keseimbangan antara precision dan recall dalam melakukan klasifikasi tingkat kematangan buah alpukat berdasarkan fitur warna HSV.

Ringkasan hasil evaluasi model KNN terbaik dengan $k = 1$ berdasarkan empat metrik utama disajikan pada Tabel 4.5.

Tabel 4.5 Ringkasan Metrik Evaluasi Model KNN Terbaik ($k=1$)

Metrik	Nilai
Accuracy	83,33%
Precision	85,67%
Recall	83,33%
F1-Score	83,26%

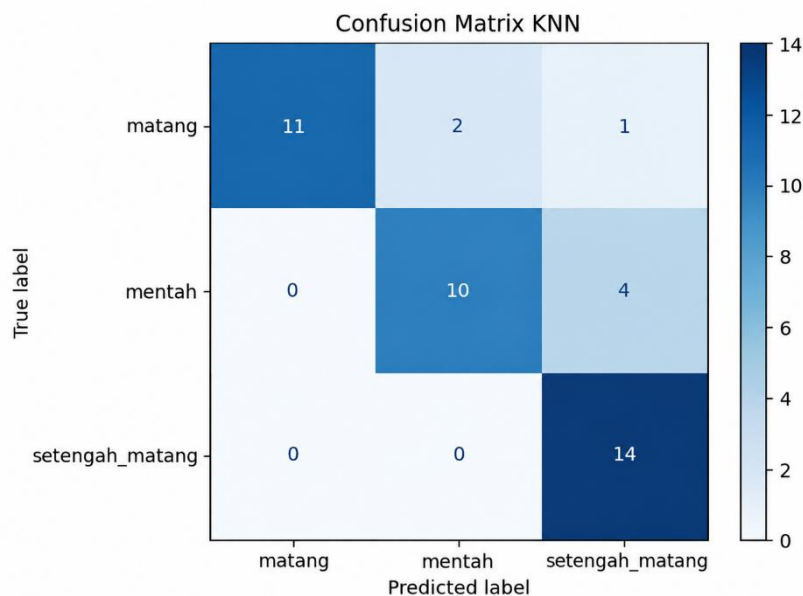
Berdasarkan Tabel 4.5, nilai accuracy sebesar 83,33%, precision sebesar 85,67%, dan F1-score sebesar 83,26% menunjukkan bahwa model K-Nearest Neighbor dengan $k = 1$ mampu melakukan klasifikasi terhadap tingkat kematangan buah alpukat dengan hasil yang cukup konsisten pada data pengujian. Nilai precision yang lebih tinggi dibandingkan accuracy dan F1-score menunjukkan adanya perbedaan antara kemampuan model dalam menghasilkan prediksi positif yang tepat dengan kemampuan keseluruhannya dalam mengklasifikasikan data.

Nilai F1-score sebesar 83,26% menunjukkan adanya keseimbangan antara precision dan recall dalam model. Dengan demikian, berdasarkan hasil pengujian

pada data testing, model K-Nearest Neighbor dengan $k = 1$ dapat digunakan untuk melakukan klasifikasi tingkat kematangan buah alpukat berdasarkan fitur warna HSV.

4.5.5 Confusion Matrix

Guna memperoleh gambaran yang lebih rinci mengenai distribusi kesalahan klasifikasi antar kelas, hasil prediksi model KNN terbaik divisualisasikan dalam bentuk confusion matrix pada gambar berikut.



Gambar 4.5 Confusion Matrix Model KNN

Berdasarkan Gambar 4.5, model berhasil mengklasifikasikan 11 dari 14 data uji pada kelas matang dengan benar, sedangkan 2 sampel salah diprediksi sebagai kelas mentah dan 1 sampel salah diprediksi sebagai kelas setengah matang. Pada kelas mentah, sebanyak 10 dari 14 sampel berhasil diklasifikasikan dengan benar, sementara 4 sampel lainnya keliru diprediksi sebagai kelas setengah matang.

Adapun pada kelas setengah matang, model berhasil mengklasifikasikan seluruh 14 sampel dengan benar tanpa satupun kesalahan.

Hasil ini menunjukkan pola kesalahan yang berbeda dari yang umum diperkirakan berdasarkan analisis fitur warna: alih-alih kelas setengah matang yang paling sulit dibedakan, justru kelas setengah matang berhasil dikenali secara sempurna, sedangkan kesalahan klasifikasi terkonsentrasi pada kelas mentah yang beberapa sampelnya memiliki representasi fitur warna yang cukup dekat dengan kelas setengah matang. Kelas matang juga menyumbang sebagian kesalahan, dengan sampel yang keliru diprediksi sebagai mentah maupun setengah matang.

Secara keseluruhan, dari 42 data uji, sebanyak 35 sampel berhasil diklasifikasikan dengan benar, sehingga menghasilkan nilai accuracy sebesar 83,33%. Hasil ini menunjukkan bahwa kombinasi ekstraksi fitur warna HSV dan algoritma K-Nearest Neighbor mampu memberikan performa klasifikasi yang cukup baik, namun sistem masih mengalami kesulitan yang cukup berarti dalam membedakan kelas mentah dari kelas setengah matang.

4.6 Pembahasan

4.6.1 Analisis Hasil Klasifikasi

Hasil pengujian menunjukkan bahwa sistem klasifikasi tingkat kematangan buah alpukat menggunakan fitur warna HSV dan algoritma K-Nearest Neighbor mampu mencapai accuracy sebesar 83,33%. Dari 42 data uji, sebanyak 35 sampel berhasil diklasifikasikan sesuai dengan kelas sebenarnya, sedangkan 7 sampel mengalami kesalahan klasifikasi.

Berdasarkan hasil confusion matrix, kesalahan klasifikasi tidak tersebar merata pada ketiga kelas. Kelas setengah matang tidak mengalami kesalahan klasifikasi sama sekali (14 dari 14 benar), sedangkan kelas mentah menyumbang kesalahan paling banyak, yaitu 4 dari 14 sampel yang keliru diprediksi sebagai kelas setengah matang. Kelas matang turut mengalami 3 kesalahan, tersebar ke kelas mentah (2 sampel) dan setengah matang (1 sampel).

Kondisi tersebut didukung oleh hasil ekstraksi fitur HSV pada Tabel 4.1, khususnya nilai H_Mean yang menunjukkan bahwa kelas mentah memiliki nilai rata-rata 21,81, sedangkan kelas setengah matang memiliki nilai 18,98. Kedekatan nilai tersebut menyebabkan representasi fitur kedua kelas saling beririsan, sehingga algoritma KNN yang mengandalkan kedekatan jarak masih menghasilkan sejumlah kesalahan klasifikasi, khususnya sampel kelas mentah yang terdorong ke wilayah fitur kelas setengah matang. Sebaliknya, kelas matang memiliki nilai H_Mean sebesar 47,17, yang berbeda cukup jauh dibandingkan dua kelas lainnya, meskipun demikian tetap terdapat beberapa sampel matang yang salah diklasifikasikan, mengindikasikan adanya variasi warna pada sebagian sampel kelas matang yang tumpang tindih dengan kelas lain.

Secara keseluruhan, hasil penelitian menunjukkan bahwa fitur statistik warna HSV cukup mampu merepresentasikan karakteristik tingkat kematangan buah alpukat, dengan tingkat akurasi 83,33% yang menunjukkan performa klasifikasi yang cukup baik meskipun belum optimal, khususnya dalam membedakan kelas mentah dari kelas setengah matang.

4.6.2 Analisis Pengaruh Nilai K

Hasil pengujian variasi nilai k pada Tabel 4.4 menunjukkan bahwa nilai $k = 1$ menghasilkan performa terbaik dengan accuracy sebesar 83,33%. Ketika nilai k ditingkatkan, akurasi cenderung menurun dan berfluktuasi, dari 71,43% pada $k = 2$, sedikit naik menjadi 73,81% pada $k = 3$, kemudian turun dan berfluktuasi pada kisaran 59,52%–61,90% untuk $k = 5$ hingga $k = 15$. Hasil tersebut menunjukkan bahwa peningkatan nilai k pada dataset penelitian ini secara umum tidak diikuti oleh peningkatan performa klasifikasi, bahkan cenderung menurunkan akurasi secara signifikan.

Pada penelitian ini, penggunaan $k = 1$ memberikan hasil terbaik karena proses klasifikasi hanya mempertimbangkan satu tetangga terdekat yang memiliki karakteristik fitur paling mirip dengan data uji. Ketika nilai k diperbesar, proses majority voting melibatkan lebih banyak tetangga yang berasal dari kelas lain, terutama antara kelas mentah dan setengah matang yang memiliki kedekatan karakteristik warna, sehingga meningkatkan kemungkinan terjadinya kesalahan prediksi secara signifikan.

Penurunan performa yang cukup tajam pada nilai k yang lebih besar (hingga di bawah 60% pada beberapa nilai k) mengindikasikan bahwa dataset penelitian ini memiliki sebaran fitur yang belum sepenuhnya terpisah secara baik antar kelas, sehingga model menjadi sensitif terhadap pemilihan jumlah tetangga. Berdasarkan hasil tersebut, dapat disimpulkan bahwa penggunaan $k = 1$ merupakan konfigurasi yang paling sesuai untuk dataset pada penelitian ini, namun hasil ini juga menunjukkan bahwa model belum sepenuhnya robust, sehingga penelitian

selanjutnya perlu mempertimbangkan penambahan jumlah data, teknik ekstraksi fitur tambahan, atau validasi seperti k-fold cross-validation untuk memperoleh model yang lebih stabil.

4.6.3 Pebandingan dengan Penelitian Terdahulu

Hasil penelitian ini menunjukkan bahwa algoritma K-Nearest Neighbor (KNN) yang dikombinasikan dengan ekstraksi fitur warna HSV mampu mengklasifikasikan tingkat kematangan buah alpukat dengan tingkat akurasi yang baik. Hasil tersebut sejalan dengan penelitian (Jaramillo-Acevedo et al., 2020) yang menunjukkan bahwa karakteristik warna pada citra digital dapat dimanfaatkan untuk mengidentifikasi tingkat kematangan alpukat. Perbedaannya, penelitian (Jaramillo-Acevedo et al., 2020) menggunakan algoritma Artificial Neural Network (ANN), sedangkan penelitian ini menggunakan algoritma K-Nearest Neighbor (KNN) yang lebih sederhana, mudah diimplementasikan, dan tidak memerlukan proses pelatihan model yang kompleks.

Selain itu, hasil penelitian ini juga memiliki kesamaan dengan penelitian (Pinto et al., 2019) yang memanfaatkan informasi spektral untuk mengklasifikasikan tingkat kematangan alpukat. Penelitian tersebut menggunakan citra hiperspektral yang memerlukan perangkat akuisisi khusus, sedangkan penelitian ini hanya menggunakan citra digital biasa yang dikonversi ke ruang warna HSV. Meskipun menggunakan metode akuisisi citra yang lebih sederhana, penelitian ini tetap mampu menghasilkan performa klasifikasi yang baik.

Hasil penelitian ini juga sejalan dengan penelitian (Aulia et al., 2023) yang menerapkan ruang warna HSV dan algoritma K-Nearest Neighbor (KNN) pada

klasifikasi tingkat kematangan buah. Kesamaan hasil tersebut menunjukkan bahwa ruang warna HSV mampu merepresentasikan perubahan warna selama proses pematangan dengan baik sehingga dapat digunakan sebagai fitur masukan yang efektif pada algoritma KNN. Hal ini memperkuat bahwa kombinasi fitur HSV dan algoritma KNN dapat diterapkan tidak hanya pada satu jenis buah, tetapi juga pada objek buah lain, termasuk alpukat.

Berdasarkan perbandingan dengan penelitian terdahulu, penelitian ini memberikan kontribusi dalam penerapan kombinasi fitur warna HSV dan algoritma K-Nearest Neighbor (KNN) untuk klasifikasi tingkat kematangan buah alpukat menggunakan citra digital. Pendekatan yang digunakan relatif sederhana, tidak memerlukan perangkat akuisisi citra khusus maupun algoritma yang kompleks, namun tetap mampu menghasilkan performa klasifikasi yang baik.

4.6.4 Kelebihan dan Keterbatasan Sistem

Sistem klasifikasi yang dikembangkan pada penelitian ini memiliki sejumlah kelebihan. Pertama, pendekatan yang digunakan tergolong sederhana dan efisien secara komputasi, mengingat algoritma KNN tidak memerlukan proses training yang kompleks, serta fitur yang diekstraksi berupa statistik deskriptif yang mudah diinterpretasikan secara langsung. Kedua, penggunaan ruang warna HSV mampu merepresentasikan karakteristik perubahan warna kulit alpukat dengan baik sehingga dapat digunakan sebagai fitur masukan pada proses klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN). Hasil penelitian menunjukkan bahwa fitur warna HSV memberikan representasi warna yang efektif untuk membedakan tingkat kematangan buah alpukat.

Ketiga, tahap preprocessing yang melibatkan ROI extraction berbasis GrabCut memungkinkan sistem untuk mengurangi pengaruh latar belakang terhadap fitur warna yang diekstraksi, sehingga fitur yang dihasilkan lebih merepresentasikan karakteristik objek buah secara murni. Kombinasi tahapan preprocessing, ekstraksi fitur, dan pemodelan yang terintegrasi tersebut menghasilkan sistem klasifikasi yang cukup andal untuk skala dataset penelitian ini.

Meskipun demikian, sistem ini juga memiliki sejumlah keterbatasan yang perlu diperhatikan. Pertama, ukuran dataset yang digunakan tergolong terbatas, yaitu 210 citra dengan 42 data uji, sehingga generalisasi hasil terhadap variasi kondisi pengambilan citra yang lebih luas, seperti perbedaan pencahayaan ekstrem atau latar belakang yang lebih kompleks, masih perlu diuji lebih lanjut. Kedua, hasil confusion matrix menunjukkan bahwa sistem masih mengalami kesulitan dalam membedakan kelas mentah dan setengah matang secara sempurna, yang mengindikasikan bahwa fitur warna HSV saja belum sepenuhnya mampu menangkap seluruh variasi transisi warna pada fase awal pematangan.

Ketiga, sebagai algoritma instance-based learning, KNN memiliki keterbatasan pada waktu prediksi yang akan meningkat seiring bertambahnya jumlah data latih, sehingga skalabilitas sistem pada dataset berskala jauh lebih besar perlu dipertimbangkan pada pengembangan selanjutnya. Keterbatasan-keterbatasan tersebut dapat menjadi arah pengembangan penelitian lanjutan, misalnya melalui penambahan jumlah data, keterbatasan tersebut dapat menjadi arah pengembangan penelitian selanjutnya, misalnya dengan menambah jumlah

dan variasi dataset, mengombinasikan fitur warna HSV dengan fitur tekstur, atau membandingkan performa K-Nearest Neighbor dengan algoritma klasifikasi lain seperti Support Vector Machine (SVM), Random Forest, maupun Artificial Neural Network (ANN).