

KLASIFIKASI TINGKAT PEMBUSUKAN APEL HIJAU BERDASARKAN PERUBAHAN WARNA MENGGUNAKAN METODE CNN

Sofya Amanda Abbygail Tampubolon^{1*)}, Adlian Jefiza¹

¹Jurusan Teknik Elektro, Politeknik Negeri Batam, Jl. Ahmad Yani, Batam Kota, Kota Batam, Kepulauan Riau 29461, Indonesia

**Penulis korespondensi, E-mail: sofya.abbygail@gmail.com*

Abstrak

Pembusukan pada apel hijau merupakan masalah kritis yang menurunkan kualitas fisik serta nilai ekonomi produk dalam rantai distribusi pangan. Identifikasi dini terhadap perubahan visual pada kulit apel sangat penting untuk meminimalkan kerugian ekonomi. Penelitian ini berhasil mengembangkan sistem klasifikasi tingkat pembusukan apel hijau berbasis *Deep Learning* menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50. Sistem dirancang dan diimplementasikan dengan dua mode masukan yang fleksibel, yaitu klasifikasi secara *real-time* melalui webcam dan masukan berupa file citra statis, di mana hasil klasifikasi dari kedua mode tersebut secara otomatis disimpan untuk keperluan pencatatan data (*data logging*). Tahapan pengolahan data yang dilakukan meliputi *cropping*, *resizing*, *renaming*, dan augmentasi data untuk memperkaya variasi pembelajaran model. Fitur utama pada tahap *preprocessing* sistem adalah penerapan konversi ruang warna dari RGB ke HSV (*Hue, Saturation, Value*), yang terbukti efektif dalam menonjolkan karakteristik perubahan warna akibat pembusukan. Apel diklasifikasikan ke dalam tiga kategori, yaitu segar, mulai membusuk, dan busuk. Berdasarkan hasil pengujian menggunakan metrik evaluasi *accuracy*, *precision*, *recall*, dan *F1-score*, sistem memperoleh tingkat akurasi sebesar 98,88% pada dataset pengujian. Model yang menggunakan strategi *Fine-Tuning* menunjukkan kinerja terbaik dan mencapai konvergensi pada epoch ke-12. Hasil tersebut menunjukkan bahwa sistem yang dikembangkan merupakan solusi otomatisasi yang efektif dan terdokumentasi bagi industri dalam memantau kualitas buah, meskipun kinerjanya pada mode *real-time* masih dipengaruhi oleh kestabilan pencahayaan lingkungan.

Kata kunci: CNN, Apel Hijau, HSV, Real-time Klasifikasi, ResNet-50.

Abstract

Spoilage in green apples is a critical issue that compromises the physical quality and economic value of products within the food distribution chain. Early identification of visual changes on the apple skin is essential to minimize economic losses. This study has successfully developed a classification system for green apple decay levels based on Deep Learning using the Convolutional Neural Network (CNN) method with ResNet-50 architecture. The system has been designed and implemented with two flexible input modes: real-time classification via webcam and static image file input, wherein classification results from both modes are automatically saved for data logging. The data processing stages performed included cropping, resizing, renaming, and data augmentation to enrich the model's learning variation. A key feature in the system's preprocessing is the implementation of color space conversion from RGB to HSV (*Hue, Saturation, Value*), which proved effective in highlighting color change features caused by decay. Apples were classified into three categories: fresh, starting to rot, and rotten. Based on testing results using accuracy, precision, recall, and F1-score evaluation metrics, the system achieved an accuracy rate of 98.88% on the test dataset. The model utilizing the Fine-Tuning strategy demonstrated the best performance and achieved convergence at the 12th epoch. These results indicate that the developed system serves as an effective and documented automation solution for the industry in monitoring fruit quality, although its performance in real-time mode remains influenced by environmental lighting stability.

Keywords: CNN, Green Apple, HSV, Real-time Classification, ResNet-50.

1. Pendahuluan

Permintaan terhadap buah apel, khususnya apel hijau, terus meningkat di pasar Indonesia karena kandungan nutrisi dan manfaat kesehatannya, sehingga menjadi salah satu buah yang banyak dikonsumsi oleh masyarakat yang menerapkan pola hidup sehat. Namun, kualitas apel hijau yang beredar di pasaran sering kali tidak konsisten. Salah satu penyebab utamanya adalah proses pemilihan dan penyortiran buah yang masih dilakukan secara manual dengan mengandalkan pengamatan visual serta penilaian subjektif dari pekerja. Proses manual tersebut sering menimbulkan ketidakkonsistenan dalam penilaian kualitas serta rendahnya efisiensi penyortiran buah, baik di pasar maupun di sektor pertanian [1]. Selain itu, faktor lingkungan seperti suhu, kelembapan, dan curah hujan dapat mempercepat proses pembusukan apel hijau yang tidak selalu dapat dideteksi melalui pengamatan visual. Akibatnya, apel yang tampak segar saat dijual dapat mengalami pembusukan dalam waktu singkat sebelum sampai ke tangan konsumen. Kondisi ini tidak hanya menyebabkan kerugian bagi konsumen, tetapi juga memengaruhi efisiensi rantai pasok serta menurunkan keuntungan produsen dan distributor [2]. Untuk mengatasi permasalahan tersebut, diperlukan suatu sistem yang mampu mengidentifikasi dan mengklasifikasikan tingkat pembusukan apel hijau secara otomatis, cepat, dan akurat. Salah satu pendekatan yang menjanjikan adalah penerapan teknologi *computer vision* yang dipadukan dengan pengolahan citra digital. Pendekatan ini memungkinkan ekstraksi berbagai fitur visual, seperti warna, bentuk, dan tekstur, yang bermanfaat untuk mengevaluasi kualitas buah. Di antara berbagai model warna, Hue, Saturation, dan Value (HSV) banyak digunakan karena mampu merepresentasikan informasi warna secara lebih konsisten pada kondisi pencahayaan yang berbeda dibandingkan dengan model warna RGB. Mengingat apel hijau mengalami perubahan warna yang cukup jelas selama proses pembusukan, fitur warna HSV dapat dimanfaatkan sebagai parameter utama dalam proses klasifikasi tingkat kesegaran buah [3]. Berbagai penelitian sebelumnya telah menunjukkan bahwa metode *deep learning* efektif dalam mendeteksi tingkat kesegaran buah. Sebuah model *Convolutional Neural Network* (CNN) berbasis arsitektur LeNet-5 berhasil mencapai akurasi sebesar 93% dalam mengklasifikasikan apel segar dan apel busuk [4]. Namun, penelitian tersebut hanya berfokus pada klasifikasi dua kelas sehingga belum mampu mengenali fase transisi pembusukan yang memiliki karakteristik visual lebih kompleks. Penelitian lain menggunakan metode CNN untuk membedakan stroberi matang dan belum matang dengan tingkat akurasi sebesar 98% serta *validation loss* sebesar 0,0383 [5]. Meskipun menghasilkan akurasi tinggi, penelitian tersebut lebih menitikberatkan pada tingkat kematangan buah, bukan proses pembusukan yang melibatkan perubahan warna secara bertahap. Selain itu, penelitian mengenai klasifikasi buah dan sayuran segar

maupun busuk menggunakan metode *Deep Learning* memperoleh akurasi sebesar 80,83% menggunakan optimizer *Stochastic Gradient Descent* (SGD) [6]. Akan tetapi, performa tersebut masih menunjukkan adanya peluang peningkatan melalui penggunaan arsitektur yang lebih dalam (*deep architecture*) dan strategi optimasi yang lebih efektif. Berdasarkan penelitian-penelitian terdahulu, masih terdapat beberapa kesenjangan penelitian (*research gap*). Pertama, sebagian besar penelitian hanya melakukan klasifikasi biner antara buah segar dan busuk sehingga belum mampu mengidentifikasi fase transisi "mulai membusuk" yang memiliki karakteristik visual sangat mirip dengan kedua kelas lainnya. Kedua, sebagian besar penelitian masih menggunakan arsitektur CNN konvensional yang memiliki keterbatasan dalam mengekstraksi fitur visual yang kompleks. Ketiga, sebagian besar penelitian masih memanfaatkan ruang warna RGB yang relatif sensitif terhadap perubahan pencahayaan, sehingga performa sistem berpotensi menurun ketika diterapkan pada kondisi nyata dengan intensitas cahaya yang bervariasi. Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan sistem klasifikasi tingkat pembusukan apel hijau dengan mengombinasikan ekstraksi fitur ruang warna HSV dan arsitektur ResNet-50 berbasis *Transfer Learning*. Arsitektur ResNet-50 dipilih karena memiliki mekanisme *residual learning* yang mampu mengatasi permasalahan *vanishing gradient* pada jaringan yang dalam, sehingga dapat mengekstraksi fitur visual secara lebih mendalam dibandingkan arsitektur CNN konvensional. Sementara itu, penggunaan ruang warna HSV diharapkan mampu meningkatkan representasi perubahan warna yang terjadi selama proses pembusukan, sehingga model lebih robust terhadap variasi pencahayaan. Kontribusi utama penelitian ini terletak pada pengembangan sistem klasifikasi tiga tingkat pembusukan apel hijau, yaitu segar, mulai membusuk, dan busuk, yang diimplementasikan dalam aplikasi berbasis *Graphical User Interface* (GUI) dengan kemampuan klasifikasi citra statis maupun *real-time*. Berbeda dengan penelitian sebelumnya yang umumnya hanya berfokus pada klasifikasi biner, sistem yang dikembangkan mampu mengidentifikasi fase transisi "mulai membusuk" yang merupakan kondisi paling sulit dibedakan karena memiliki karakteristik visual yang saling tumpang tindih (*overlapping*) dengan kelas lainnya. Selain itu, penelitian ini menghasilkan akurasi klasifikasi sebesar 98,88% pada data pengujian, yang menunjukkan bahwa kombinasi ruang warna HSV dan arsitektur ResNet-50 mampu meningkatkan kemampuan model dalam membedakan perubahan warna apel hijau secara lebih akurat. Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi tingkat pembusukan apel hijau berbasis *Deep Learning* menggunakan kombinasi ruang warna HSV dan arsitektur ResNet-50. Sistem yang dikembangkan diharapkan mampu melakukan identifikasi tiga tingkat pembusukan secara otomatis dengan tingkat akurasi yang tinggi serta

dapat diterapkan sebagai solusi pendukung proses penyortiran buah pada sektor pertanian dan industri pangan.

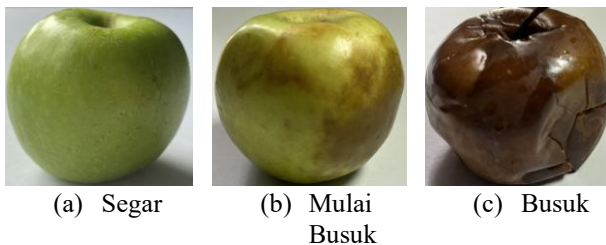
2. Metode

2.1. Pengumpulan Dataset

Tahapan ini menguraikan proses awal dalam penelitian, yang dimulai dari akuisisi citra digital buah apel hijau hingga penerapan serangkaian teknik prapemrosesan. Tujuan utama dari tahapan ini adalah untuk mempersiapkan data mentah agar memiliki format yang seragam dan kualitas yang optimal sebelum digunakan dalam proses pelatihan model *Convolutional Neural Network* (CNN) [7].

2.1.1 Pengumpulan dan Pembagian Dataset

Dataset dikumpulkan secara mandiri menggunakan kamera *smartphone* dengan metode *multi-angle photography* untuk mendapatkan representasi visual yang menyeluruh. Total dataset berjumlah 869 citra, yang didistribusikan ke dalam kelas Segar (330 citra), Mulai Membusuk (297 citra), dan Busuk (242 citra). Dataset kemudian dibagi menjadi data latih (80%), data validasi (10%), dan data uji (10%) menggunakan *seed* acak tetap untuk menjaga konsistensi distribusi.



Gambar 1. Contoh visual dari dataset apel hijau yang dikumpulkan

2.2 Pra-pemrosesan Dataset

Sebelum citra diproses oleh model klasifikasi, tahapan pra-pemrosesan (preprocessing) dilakukan untuk menstandarisasi input dan meningkatkan kualitas data latih. Tahapan ini diimplementasikan menggunakan modul *transforms*. Compose pada PyTorch dengan langkah-langkah sebagai berikut:

2.2.1 Pembagian Dataset (*Data Splitting*)

Agar evaluasi model dapat dilakukan secara objektif, dataset dibagi menjadi tiga subset independen menggunakan pustaka *splitfolders*. Rasio pembagian yang diterapkan adalah 80:10:10, yang dialokasikan untuk 80% data latih (*training set*), 10% data validasi (*validation set*), dan 10% data uji (*testing set*). Proses ini dilakukan secara acak dengan menyertakan nilai *seed* (nilai acak tetap) agar

distribusi data tetap konsisten dan dapat direproduksi pada pengujian selanjutnya.

2.2.2 Penyeragaman Dimensi Citra (*Resizing*)

Seluruh citra masukan, baik pada himpunan data latih, validasi, maupun uji, diseragamkan dimensinya menjadi 224×224 piksel. Penyesuaian ukuran ini merupakan standar yang diterapkan untuk menyesuaikan dengan arsitektur lapisan masukan (*input layer*) pada model CNN yang digunakan.

2.2.3 Augmentasi Data

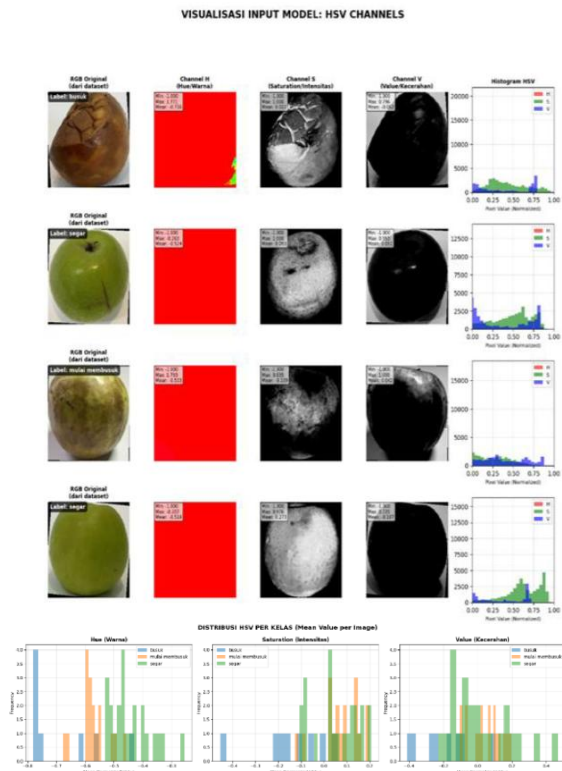
Teknik augmentasi diterapkan secara dinamis (*on-the-fly*) khusus pada data latih. Proses transformasi citra ini dilakukan secara langsung di dalam memori saat pelatihan tanpa menyimpan *file* baru. Transformasi yang diterapkan meliputi *Random Horizontal Flip* dengan probabilitas 0,3 (30%) dan *Random Rotation* sebesar 10 derajat. Pendekatan ini berfungsi untuk memperkaya variasi visual, membantu model mengenali fitur secara lebih umum (*generalisasi*), dan mencegah terjadinya *overfitting*.



Gambar 2. Visualisasi Augmentasi

2.2.4 Konversi Ruang Warna dan Normalisasi.

Mengingat indikator utama kesegaran apel hijau sangat bergantung pada perubahan warna kulit, ekstraksi fitur dimaksimalkan melalui konversi ruang warna citra dari RGB menjadi HSV (*Hue, Saturation, Value*). Setelah dikonversi, dilakukan normalisasi bertingkat. Tahap pertama adalah penskalaan manual pada setiap kanal (H dibagi 179; S dan V dibagi 255) untuk mencapai rentang nilai $[0,1]$. Tahap kedua adalah standarisasi menggunakan nilai *mean* dan standar deviasi sebesar 0,5, sehingga data terdistribusi secara terpusat pada rentang $[-1,1]$. Langkah normalisasi ini krusial untuk mempercepat proses konvergensi model selama pelatihan.

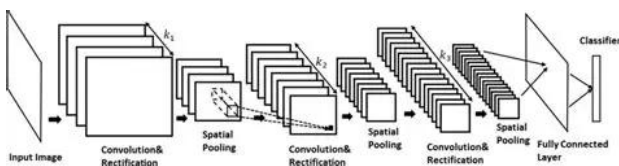


Gambar 3. Visualisasi HSV

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan varian dari Multilayer Perceptron (MLP) yang mengeksplorasi korelasi spasial lokal pada data berstruktur grid, seperti citra digital. Metode ini bekerja dengan menerapkan pola konektivitas lokal antar neuron pada lapisan yang berdekatan, sehingga sangat efektif digunakan untuk tugas ekstraksi fitur, deteksi objek, dan klasifikasi citra secara otomatis tanpa memerlukan rekayasa fitur (*feature engineering*) manual yang rumit [8].

2.3.1 Arsitektur CNN



Gambar 4. Arsitektur CNN

Arsitektur CNN dibangun melalui struktur berlapis yang secara hierarkis mempelajari pola dan fitur dari data citra masukan untuk melakukan tugas klasifikasi yang tangguh (*robust*) terhadap variasi distorsi kecil. Secara umum, arsitektur ini tersusun dari beberapa lapisan inti yang bekerja secara sekuensial. Proses dimulai dari Lapisan Konvolusi (*Convolutional Layer*) yang menjadi fondasi

utama dengan menggunakan sejumlah kernel atau filter untuk memindai (*scan*) citra input guna mendeteksi fitur lokal seperti tepi, tekstur, dan gradasi warna. Setiap filter tersebut menghasilkan *feature map* melalui operasi konvolusi dengan perumusan $S_t = x \times w_t$ di mana x merepresentasikan citra input, w merupakan kernel, dan t adalah piksel. Selanjutnya, Lapisan Fungsi Aktivasi (*Activation Layer*) ditempatkan untuk memperkenalkan sifat non-linearitas pada jaringan agar model mampu memetakan hubungan data yang kompleks. Fungsi aktivasi yang umum digunakan adalah ReLU (*Rectified Linear Unit*) dengan persamaan $f(x) = \max(0, x)$, yang membatasi hasil *output* pada batas minimum nol. Setelah tahap ekstraksi dan aktivasi, Lapisan *Pooling* (*Pooling Layer*), seperti operasi *Max Pooling*, diaplikasikan untuk mereduksi dimensi spasial dari *feature map* sambil mempertahankan informasi visual yang paling penting. Proses ini krusial untuk memangkas jumlah beban komputasi dan mengontrol laju *overfitting*. Hasil dari lapisan *pooling* yang masih berformat multidimensional (volume 3D) kemudian ditransformasi menjadi vektor satu dimensi melalui Lapisan Perataan (*Flattening Layer*) agar siap diproses oleh lapisan jaringan saraf tradisional. Vektor tersebut lalu diteruskan ke Lapisan Terhubung Penuh (*Fully Connected Layer*) yang bertindak sebagai pengambil keputusan akhir dengan mempelajari kombinasi hierarkis dari fitur-fitur yang telah diekstraksi. Sebagai tahap penyelesaian, Lapisan *Output* menghasilkan prediksi probabilitas akhir menggunakan fungsi aktivasi *Softmax* untuk mengolah keluaran dari lapisan sebelumnya menjadi distribusi probabilitas masing-masing kategori target pada kasus klasifikasi multi-kelas [9].

2.4 Skenario Pelatihan dan Evaluasi Model

Tahap pelatihan dan evaluasi merupakan fase krusial dalam pengembangan sistem klasifikasi untuk memastikan model mampu mengenali karakteristik visual dari setiap tingkat pembusukan apel hijau secara akurat. Pada subbab ini, diuraikan serangkaian proses komputasi yang diterapkan pada sistem, mulai dari penyesuaian arsitektur ResNet-50 melalui strategi *Transfer Learning*, penetapan hiperparameter dan mekanisme optimasi, kontrol otomatis pelatihan, hingga pengujian performa akhir. Seluruh skenario ini dirancang secara terstruktur agar model tidak hanya mampu mengidentifikasi pola kromatik HSV pada data latih, tetapi juga memiliki kemampuan generalisasi yang tinggi dan *error* yang minim saat dihadapkan pada data baru [10].

2.4.1. Konfigurasi Arsitektur dan Transfer Learning

Model klasifikasi citra diimplementasikan menggunakan pustaka *torchvision.models* dengan menerapkan strategi *Transfer Learning*. Arsitektur dasar yang digunakan adalah struktur *backbone* ResNet-50 yang dimuat beserta bobot pra-latih (*pre-trained weights*) dari dataset ImageNet untuk mengekstraksi fitur visual dasar seperti tepi dan tekstur.

Lapisan *input* disesuaikan untuk menerima tensor yang merepresentasikan ruang warna HSV, menggantikan format citra RGB yang konvensional. Selanjutnya, seluruh lapisan konvolusi pada *backbone* dibekukan dengan menonaktifkan kalkulasi gradien (*requires_grad = False*) agar bobot fitur dasar tidak rusak saat proses pelatihan awal. Sebagai tahap akhir dari konfigurasi, lapisan klasifikasi asli digantikan dengan blok *Sequential* baru yang mencakup *Dropout Layer* (0,3) untuk mencegah *overfitting*, serta *Linear Layer* yang memetakan 2048 fitur masukan menjadi 3 *node* keluaran untuk merepresentasikan probabilitas kelas [11].

2.4.2. Mekanisme Pelatihan dan Optimasi

Setelah arsitektur model dikonfigurasi, tahap selanjutnya adalah eksekusi proses pelatihan model. Berdasarkan log implementasi sistem, pelatihan dijalankan menggunakan konfigurasi parameter yang telah disesuaikan dengan kapasitas komputasi lingkungan *cloud* (Google Colab). Konfigurasi parameter pelatihan menetapkan ukuran *batch* sebesar 16 citra per iterasi, yang dipilih secara spesifik untuk menjaga stabilitas memori GPU pada lingkungan Colab sekaligus memastikan pembaruan gradien terjadi cukup sering. Algoritma optimasi yang digunakan adalah Adam dengan laju pembelajaran (*learning rate*) awal yang diset rendah untuk menjaga bobot fitur pralatih (*pre-trained*) agar tidak rusak. Optimasi ini dipadukan dengan penggunaan fungsi kerugian *Cross-Entropy Loss* yang sangat sesuai untuk menangani kasus klasifikasi multi-kelas. Mekanisme eksekusi berjalan dengan skema iteratif (*Train & Validation Loop*) yang melibatkan dua fase utama secara bergantian pada setiap *epoch*. Pada fase pelatihan, model melakukan *forward pass* untuk memprediksi kelas, menghitung *loss*, dan memperbarui bobot melalui proses *backpropagation*. Khusus pada penerapan strategi *Fine-Tuning*, sistem menerapkan teknik *Gradient Clipping* sebelum pembaruan bobot dilakukan untuk mencegah lonjakan gradien yang dapat memicu ketidakstabilan model. Setelah setiap *epoch* pelatihan selesai, sistem secara otomatis beralih ke fase validasi (tanpa pembaruan bobot) untuk mengukur performa model pada data yang belum pernah dipelajari [12].

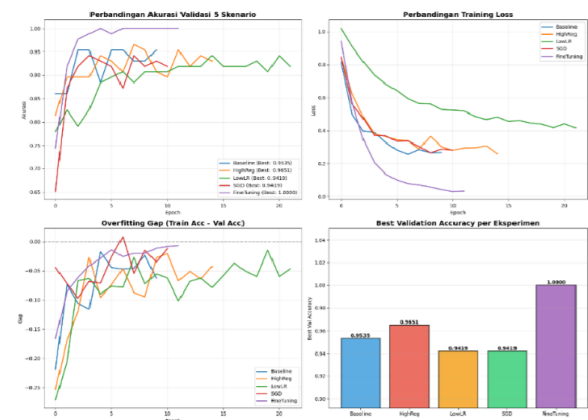
2.4.3. Kontrol Otomatis dan Penyimpanan Model

Untuk menjamin efisiensi komputasi dan performa yang optimal, sistem dilengkapi dengan mekanisme kontrol otomatis selama proses pelatihan berlangsung. Sistem menggunakan *Scheduler ReduceLROnPlateau* yang berfungsi menurunkan *learning rate* secara otomatis jika akurasi validasi mengalami stagnasi, sehingga memungkinkan model untuk mencapai titik konvergensi yang lebih presisi. Pelatihan ini dijadwalkan berjalan hingga batas maksimum 50 *epoch*. Namun, sistem menerapkan mekanisme *Early Stopping* dengan parameter *patience* bernilai 7. Artinya, jika akurasi validasi tidak mengalami peningkatan selama 7 *epoch* berturut-turut, pelatihan akan dihentikan secara otomatis untuk mencegah

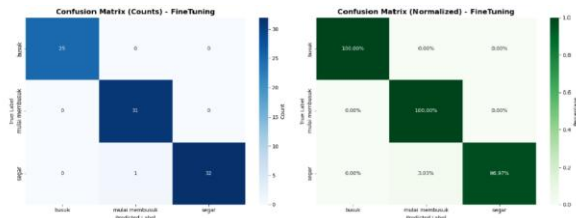
terjadinya *overfitting* serta menghemat waktu komputasi. Selain itu, sistem menerapkan mekanisme penyimpanan model terbaik (*checkpointing*) dengan memantau metrik akurasi validasi secara otomatis pada setiap akhir *epoch*. Model hanya akan disimpan ke dalam file *best_model.pth* jika akurasi validasi saat itu tercatat lebih tinggi dibandingkan *epoch* sebelumnya. Langkah ini memastikan bahwa model akhir yang akan digunakan dan ditanamkan pada sistem antarmuka (GUI) adalah versi model dengan performa generalisasi terbaik [13].

2.4.4. Evaluasi Pengujian Akhir (Testing Phase)

Kinerja sistem diukur secara objektif pada data yang belum pernah dilihat sebelumnya menggunakan 89 citra data uji (*unseen data*). Evaluasi ini memuat kembali file *best_model.pth* dari proses validasi untuk menjamin pengujian pada tingkat generalisasi tertinggi, dengan mengatur model ke mode evaluasi guna menonaktifkan lapisan *Dropout* dan menstabilkan hasil prediksi. Perbandingan performa dari berbagai skenario yang diuji dapat dilihat pada Gambar 6, yang mengilustrasikan dinamika akurasi validasi dan *training loss* selama proses pelatihan. Berdasarkan hasil pengujian tersebut, skenario *Fine-Tuning* berhasil mencapai tingkat akurasi tertinggi sebesar 98,88% dengan *Macro F1-Score* 0,9896. Analisis kinerja yang lebih mendalam divisualisasikan melalui Gambar 7 dalam bentuk *Confusion Matrix*. Matriks tersebut menunjukkan sensitivitas mutlak (*Recall* 100%) dalam mendeteksi kelas 'Busuk' dan 'Mulai Membusuk', dengan tingkat kesalahan klasifikasi yang sangat minim, yaitu hanya 1 sampel (1,12%) apel 'Segar' yang terprediksi sebagai 'Mulai Membusuk'. Kesalahan berupa *false alarm* ini dinilai masih berada dalam batas toleransi yang aman untuk standar industri keamanan pangan. Meskipun demikian, performa model saat inferensi nyata sangat dipengaruhi oleh intensitas pencahayaan. Model bekerja optimal pada kondisi terang dan merata, tetapi sensitivitasnya dapat berkurang pada kondisi *backlight*. Secara keseluruhan, arsitektur ResNet-50 dengan strategi *Fine-Tuning* terbukti sangat efektif diterapkan dengan syarat operasional dilakukan pada lingkungan pencahayaan yang terkondisi [14].



Gambar 5. Perbandingan 5 Skenario Eksperimen



Gambar 6. Confusion Matrix (Evaluasi Model Terbaik)

2.5. Implementasi Sistem dan Antarmuka Pengguna (GUI)

Implementasi GUI dibangun berbasis Python dan Tkinter untuk memudahkan pengguna dalam melakukan klasifikasi citra apel hijau secara interaktif. Sistem akan memproses citra menggunakan model dan menampilkan output berupa nilai HSV, parameter evaluasi, serta status kesegaran buah (segar, mulai membusuk, atau busuk). Selain tampilan visual, hasil diagnosis secara otomatis diekspor dan disimpan dalam format teks (.txt) ke folder 'hasil klasifikasi' untuk keperluan arsip data.



Gambar 7. Implementasi GUI

3. Hasil dan Analisa

Pada bagian ini, disajikan analisis terhadap hasil keluaran (output) yang diperoleh setelah sistem dijalankan melalui antarmuka grafis (GUI). Analisis ini bertujuan untuk memvalidasi apakah logika pemrosesan data pada sistem telah berfungsi sesuai dengan spesifikasi yang dirancang. Seluruh data hasil eksekusi program telah direkapitulasi dan disajikan dalam bentuk tabel-tabel berikut untuk memudahkan pembacaan dan evaluasi kinerja sistem secara keseluruhan.

3.1 Hasil Klasifikasi Pada Apel Hijau

Analisis hasil klasifikasi pada sistem ini didasarkan pada ekstraksi fitur visual menggunakan ruang warna HSV (*Hue*, *Saturation*, *Value*). Nilai-nilai yang tersaji pada hasil pengujian bukan merupakan angka acak, melainkan hasil komputasi matematis terhadap setiap piksel citra input, di mana nilai RGB dikonversi menjadi HSV untuk mendapatkan representasi warna yang perseptual. Parameter *Hue* (H) merepresentasikan panjang gelombang warna dominan, *Saturation* (S) merepresentasikan kepekatan warna, dan *Value* (V) merepresentasikan intensitas cahaya. Tabel 1 menampilkan data empiris hasil ekstraksi nilai rata-rata (*mean*) HSV dari 15 sampel citra uji yang digunakan untuk memvalidasi pola fitur setiap kelas pada kondisi pencahayaan standar.

Tabel 1. Sampel Nilai HSV Hasil Ekstraksi pada Citra Uji

No	Citra	Kondisi	H	S	V
1.	busuk_0004	BUSUK	16.93	104.64	117.39
2.	busuk_0013	BUSUK	23.88	98.76	109.82
3.	busuk_0497	BUSUK	19.16	119.46	82.79
4.	busuk_0484	BUSUK	19.82	126.36	79.79
5.	busuk_0389	BUSUK	16.13	106.74	117.76
6.	mulai membusuk_0004	MULAI	23.31	113.35	156.59
7.	mulai membusuk_0014	MULAI	25.67	134.73	127.04
8.	mulai membusuk_0389	MULAI	37.32	130.66	148.53
9.	mulai membusuk_0112	MULAI	30.72	117.36	134.34
10.	mulai membusuk_0347	MULAI	22.66	151.37	123.70
11.	segar_0640	SEGAR	35.74	142.10	131.58
12.	segar_0446	SEGAR	49.88	130.65	121.76
13.	segar_0082	SEGAR	46.16	155.78	184.40
14.	segar_0102	SEGAR	46.36	137.55	185.59
15.	segar_0378	SEGAR	37.10	136.03	119.38

Berdasarkan Tabel 11, dapat dirumuskan parameter batasan standar yang menjadi acuan logika klasifikasi model. Nilai *Hue* menjadi diskriminator utama pada kondisi ideal. Kelas "Busuk" ditandai dengan pergeseran nilai spektrum warna yang signifikan, berada pada rentang *Hue* terendah (16.13 – 23.88) dan rata-rata *Value* cenderung rendah (< 118). Hal ini membuktikan adanya perubahan pigmen dari hijau menjadi coklat gelap dan tekstur kulit *matte* yang menyerap iluminasi. Kelas "Segar" konsisten berada pada rentang *Hue* tinggi (35.74 – 49.88) dan nilai rata-rata *Value* tertinggi (> 120), merepresentasikan dominasi warna klorofil yang utuh dan

permukaan kulit apel yang licin mengilap. Sementara itu, kelas "Mulai Membusuk" berada pada rentang transisi (*Hue* 22.66 – 37.32) yang merepresentasikan campuran piksel hijau dan cokelat, menjadikannya fase yang paling kompleks karena memiliki irisan nilai (*overlapping*) dengan kelas lainnya. Setelah menganalisis karakteristik fitur input, evaluasi dilanjutkan pada hasil akhir prediksi sistem terhadap 89 data uji. Kinerja model direpresentasikan menggunakan *Confusion Matrix* pada Tabel 2 untuk memetakan perbandingan antara kelas prediksi sistem dengan kelas aktual.

Tabel 2. Tabel Confusion Matrix Hasil Klasifikasi Apel Hijau

Predicted Class	Fresh (Segar)	Starting to Rot (Mulai Membusuk)	Rotten (Busuk)
Fresh	32	0	0
Starting to Rot	1	31	0
Rotten	0	0	25

Berdasarkan Tabel 2, kinerja sistem mencatatkan performa 100% pada kelas "Busuk" dengan memprediksi seluruh 25 sampel secara tepat, mengonfirmasi bahwa fitur Hue dan Value rendah pada apel busuk sangat kuat dan mudah dibedakan oleh model. Kelas "Mulai Membusuk" juga mencapai performa 100%, di mana 31 sampel berhasil dikenali dengan sempurna. Pada kelas "Segar", sistem mencapai performa 97% dengan mengenali 32 dari total 33 sampel dengan benar. Analisis lebih mendalam terhadap satu sampel uji yang mengalami *False Negative* ini menunjukkan bahwa model mengalami ambiguitas akibat perpotongan nilai spektrum (*overlapping*) antara kelas "Segar" dan "Mulai Membusuk" pada rentang nilai Hue 35 hingga 37. Secara fisik, kulit apel hijau yang tergolong segar tidak selalu memiliki distribusi pigmen klorofil yang homogen dan seringkali memiliki bintik-bintik alami bawaan buah. Karena sistem prapemrosesan mengekstraksi nilai rata-rata (*mean*) HSV dari keseluruhan piksel citra masukan, anomali warna alami berskala kecil tersebut akan menarik rata-rata nilai Hue secara keseluruhan turun mendekati batas bawah kategori segar. Model CNN, yang telah dilatih secara sensitif untuk menangkap awal mula perubahan degradasi warna, menginterpretasikan pergeseran sempit pada ambang batas (*threshold*) ini sebagai inisiasi dari fase pembusukan. Meskipun demikian, bias prediksi ke arah "Mulai

Membusuk" ini dipandang sebagai peringatan dini (*False Alarm*) yang jauh lebih menguntungkan secara toleransi standar keamanan pangan dibandingkan jika sistem gagal mendeteksi buah yang benar-benar busuk. Selain evaluasi statistik, verifikasi kualitatif dilakukan dengan mengambil 15 sampel acak untuk diuji secara langsung melalui antarmuka pengguna (GUI). Hasil pengujian ini disajikan pada Tabel 3 untuk memastikan sistem mampu mengekstraksi fitur dan menampilkan hasil yang konsisten.

Tabel 3. Hasil Klasifikasi Perubahan Warna Pada Apel Hijau

No	Filename	Kelas	Hasil Klasifikasi
1	segar_0014	SEGAR	SEGAR
2	segar_0217	SEGAR	SEGAR
3	segar_0102	SEGAR	SEGAR
4	segar_0378	SEGAR	SEGAR
5	segar_0120	SEGAR	SEGAR
6	Mulai	MULAI	MULAI
7	membusuk_004	MEMBUSUK	MEMBUSUK
	Mulai	MULAI	MULAI
	membusuk_0014	MEMBUSUK	MEMBUSUK
8	Mulai	MULAI	MULAI
9	membusuk_0012	MEMBUSUK	MEMBUSUK
	Mulai	MULAI	MULAI
	membusuk_0378	MEMBUSUK	MEMBUSUK
10	Mulai	MULAI	MULAI
11	membusuk_0230	MEMBUSUK	MEMBUSUK
	Busuk_0017	BUSUK	BUSUK
	Busuk_0045	BUSUK	BUSUK
13	Busuk_0072	BUSUK	BUSUK
14	Busuk_0112	BUSUK	BUSUK
15	Busuk_0175	BUSUK	BUSUK

Berdasarkan Tabel 3, terlihat bahwa sistem berhasil mengidentifikasi seluruh 15 sampel uji dengan tingkat keberhasilan 100%. Kesesuaian mutlak antara label asli dan hasil klasifikasi pada berbagai variasi citra ini membuktikan bahwa alur pemrosesan data mulai dari konversi HSV hingga inferensi model ResNet-50 telah terintegrasi dengan stabil dan akurat pada aplikasi GUI. Untuk mengukur efektivitas model secara kuantitatif, dilakukan perhitungan terhadap empat parameter utama berdasarkan keseluruhan data uji. Rangkuman hasil evaluasi tersebut disajikan pada Tabel 4.

Tabel 4. Evaluasi Hasil Klasifikasi pada Apel hijau

Akurasi	Precision	Recall	F1-Score	Keterangan
98.88%	98.96%	98.99%	98.96%	Sangat Baik

Secara keseluruhan, sistem mencatatkan akurasi global sebesar 98,88%, menunjukkan bahwa model mampu memprediksi kondisi apel dengan tepat pada 88 dari total 89 sampel uji. Analisis metrik secara mendalam menunjukkan performa yang sangat seimbang: *Precision* mencapai 98.96% yang mengindikasikan tingkat kesalahan positif palsu sangat rendah, dan *Recall* sebesar 98.99% yang menunjukkan sensitivitas tinggi dalam meminimalkan kemungkinan *False Negative*. Nilai F1-Score sebesar 98.96% mengonfirmasi bahwa model memiliki keseimbangan yang optimal dan tangguh dalam menangani klasifikasi tanpa mengalami bias terhadap kelas mayoritas.

3.2 Pengujian Variasi Kedalaman Layer Pada Custom CNN

Pengujian ini membandingkan kinerja arsitektur *Custom CNN* dengan kedalaman 3-layer dan 4 layer. Berdasarkan Tabel 5, kedua model pada akhirnya mencapai performa evaluasi sempurna sebesar 100%. Model 3-layer menunjukkan stabilitas absolut di seluruh variasi *epoch* (50–500), sedangkan model 4-layer sempat mengalami sedikit fluktuasi akurasi (98%–98,67%) sebelum mencapai titik jenuh 100% pada 500 *epoch*. Fenomena akurasi 100% ini mengindikasikan adanya *overfitting to Instance* akibat tingginya homogenitas *dataset* dan potensi *data leakage* pada proses augmentasi rotasi. Hasil pengujian ini mengonfirmasi dua temuan utama terkait rendahnya kompleksitas data dan efisiensi kapasitas model. Pertama, fitur visual antarkelas memiliki tingkat keterpisahan yang tinggi (*linearly separable*), sehingga arsitektur sederhana (3 layer) sudah mampu melakukan pemetaan kelas secara sempurna tanpa kendala ambiguitas fitur. Kedua, penambahan layer menjadi 4 layer maupun peningkatan durasi latih hingga 500 *epoch* tidak memberikan peningkatan generalisasi yang signifikan. Arsitektur 3 layer terbukti sudah sangat memadai dan efisien karena mampu mencapai konvergensi optimal pada tahap awal pelatihan (di bawah 30 *epoch*) dengan bantuan mekanisme *Early Stopping*.

Tabel 5. Perbandingan Jumlah Layer

Jumlah Layer CNN	Akurasi	Precision	Recall	F1-Score
3 Layer	100%	100%	100%	100%
4 Layer	100%	100%	100%	100%

3.3 Eksperimen Optimasi Hiperparameter

Berdasarkan evaluasi komprehensif terhadap berbagai konfigurasi optimasi pada arsitektur ResNet-50, strategi *Fine-Tuning* terbukti secara empiris sebagai konfigurasi paling optimal. Skenario ini mencapai akurasi tertinggi hingga 100% pada pelatihan dengan nilai *loss* terendah sebesar 0,02. Selain unggul dalam akurasi, *Fine-Tuning* menunjukkan efisiensi komputasi terbaik dengan mencapai titik konvergensi lebih cepat, yakni pada *epoch* ke-29, dibandingkan dengan skenario *Baseline* (*epoch* 51) maupun SGD (*epoch* 52). Sebaliknya, intervensi lain seperti penurunan *Learning Rate* (Low LR = 0.0001) justru menghambat proses belajar dan menurunkan akurasi menjadi 94.67%, sementara penggunaan *optimizer* klasik SGD menghasilkan kurva konvergensi yang kurang stabil. Rekapitulasi perbandingan kinerja ini disajikan pada Tabel 6.

Tabel 6. Rekapitulasi Hasil Eksperimen pada Setiap Skenario

No	Nama Skenario	Akurasi %	Loss Terendah	Epoch Berhenti
1	Baseline	97.33%	0.18	51
2	Fine-Tuning	100%	0.02	29
3	SGD	96.00%	0.32	52
4	Low LR	94.67%	0.42	49
5	High Regularization	97.33%	0.21	36

Untuk memvalidasi keandalan model secara lebih mendalam, dilakukan pengujian lanjutan dengan memvariasikan hiperparameter khusus pada strategi *Fine-Tuning*. Eksperimen ini membandingkan pengaruh penyesuaian *learning rate*, pemilihan *optimizer*, dan strategi pembekuan lapisan (*layer freezing*). Hasil perbandingan tersebut diuraikan pada Tabel 7.

Tabel 7. Rekapitulasi Hasil Eksperimen Lanjutan Pada Finetuning

No	Skenario	Opt. (LR)	Strategy	Akurasi
1	Baseline Fine-Tune	Adam (0.0005)	All Layers Unfrozen	0.96
2	Lower LR	Adam (0.0001)	All Layers Unfrozen	0.98
3	High Dropout + SGD	SGD (0.01)	Frozen (Feature Extraction)	0.96
4	Optimal Mix	Adam (0.0003)	All Layers Unfrozen	0.97
5	Finetuning Gradual	Adam (0.00001)	Gradual Unfreezing (Layer4 Only)	1.00

Berdasarkan hasil eksperimen lanjutan tersebut, skenario *Fine-Tuning Gradual* (Skenario 5) terpilih sebagai konfigurasi paling optimal. Model ini berhasil mencapai Akurasi Validasi (*Best Validation Accuracy*) sempurna sebesar 1.00 (100%) pada fase pelatihan. Ketika model terbaik ini dievaluasi lebih lanjut menggunakan data uji (*Test Set*), sistem tetap menunjukkan konsistensi yang sangat tinggi dengan capaian Akurasi Pengujian (*Test Accuracy*) sebesar 98.88%. Selisih yang sangat minim antara akurasi validasi dan pengujian ini membuktikan bahwa model yang dikembangkan memiliki kemampuan generalisasi yang sangat baik dan terhindar dari indikasi *overfitting* yang signifikan.

3.4 Pengujian Dataset Baru

Pengujian lanjutan dilakukan dengan menginput citra apel baru ke dalam sistem berbasis *Graphical User Interface* (GUI) untuk memvalidasi performa model terhadap data yang belum pernah dipelajari sebelumnya. Contoh keberhasilan antarmuka sistem saat memproses dan mengenali citra baru dengan kondisi 'Mulai Membusuk' dapat dilihat secara visual pada Gambar . Secara keseluruhan, pengujian terhadap citra-citra baru membuktikan bahwa sistem berhasil mengklasifikasikan objek ke dalam kategori yang tepat sesuai dengan kondisi aslinya (*Ground Truth*). Meskipun model menunjukkan tingkat keyakinan (*confidence score*) yang tinggi dan konsisten pada kelas "Segar", terdapat penurunan tingkat keyakinan pada kategori "Mulai Membusuk" dan "Busuk". Penurunan nilai probabilitas ini dipicu oleh kemiripan nilai fitur warna HSV (*overlapping*) antar-kelas serta karakteristik fase transisi yang nilainya seringkali berada

tepat di perbatasan keputusan (*decision boundary*). Kendati dihadapkan pada variasi nilai fitur yang kompleks dan ambigu, sistem secara keseluruhan tetap terbukti andal dan mampu mengambil keputusan klasifikasi dengan akurat.



Gambar 8. Tampilan Hasil Pengujian pada GUI untuk Citra Dataset Baru

3.5 Hasil Pengambilan Real Time

Setelah pengujian dataset statis, evaluasi dilanjutkan secara real-time untuk mengetahui performa sistem dalam kondisi lingkungan nyata dengan pencahayaan alami ruangan. Contoh dokumentasi visual tangkapan layar dari antarmuka sistem saat melakukan ekstraksi dan deteksi objek secara langsung dapat dilihat pada Gambar 10. Secara keseluruhan, pengujian langsung ini membuktikan bahwa sistem tetap andal dan mampu mengklasifikasikan objek dengan tepat sesuai kondisi aslinya. Meskipun prediksi label secara visual terbukti akurat, terdapat fluktuasi dan penurunan drastis pada tingkat probabilitas (*confidence score*) menjadi 48% hingga 82% pada mode *real-time* dibandingkan dengan mode citra statis. Penurunan margin keyakinan ini secara langsung diakibatkan oleh interaksi dinamis antara intensitas cahaya ruangan dan ketidakstabilan sudut kamera (*camera angle*). Permukaan apel hijau segar yang licin memicu pantulan silau (*specular reflection*) saat terpapar pencahayaan yang tidak merata. Pantulan ini mendistorsi nilai piksel asli, menyebabkan fitur parameter *Saturation* (S) anjlok dan *Value* (V) meroket secara tidak wajar, sehingga citra keluar dari distribusi data latih idealnya. Selain itu, sudut pengambilan gambar kamera *webcam* yang tidak terkunci secara presisi sering kali menimbulkan bayangan (*shadow*) pada salah satu sisi lekukan buah. Area gelap akibat bayangan parsial tersebut dikomputasi oleh model sebagai sekumpulan piksel berintensitas rendah yang secara matematis menyerupai karakteristik bercak cokelat gelap pada apel kelas "Busuk". Distorsi fitur inilah yang

memaksa arsitektur ResNet-50 membagi nilai probabilitas distribusinya (*split probability*) ke beberapa kelas terdekat, sehingga menurunkan skor persentase akhir.



Gambar 9. Hasil Pengambilan Real Time

3.6 Hasil Pengujian Dataset Eksternal

Tahap pengujian akhir memvalidasi performa model menggunakan data eksternal yang bersumber dari internet untuk mengukur kemampuan generalisasi sistem terhadap karakteristik pencahayaan, latar belakang, dan sudut pandang yang lebih bervariasi. Secara keseluruhan, model terbukti sangat andal dengan keberhasilan mengklasifikasikan seluruh sampel uji (9 citra) secara tepat sesuai dengan label aslinya terlihat pada Gambar 11, yang membuktikan bahwa model tidak mengalami *overfitting* pada *dataset* internal. Berdasarkan analisis tingkat kepercayaan (*confidence score*), kelas "Segar" mencatatkan nilai tertinggi dan paling stabil pada rentang 84,36% hingga 98,15%, mengindikasikan bahwa fitur visual dan nilai HSV pada apel segar sangat konsisten sehingga mudah dikenali model. Sebaliknya, kelas "Mulai Membusuk" mengalami penurunan tingkat keyakinan hingga 50,00% akibat nilai HSV yang mulai berhimpit (*overlapping*) dengan kelas lain, sehingga menuntut kepastian lebih tinggi dari algoritma untuk membedakan degradasi warna yang tipis. Pada kelas "Busuk", hasil probabilitas juga cukup bervariasi (50,64% hingga 87,72%) dikarenakan rendahnya tingkat kecerahan (*Value*) pada area citra yang gelap, yang kerap mempersulit ekstraksi fitur secara maksimal. Meskipun terdapat fluktuasi persentase keyakinan pada kondisi transisi maupun area minim cahaya, integrasi antara ekstraksi fitur *Mean HSV* dan arsitektur model ini secara meyakinkan terbukti efektif dan tetap mampu memberikan prediksi label yang akurat pada data baru dari luar sistem.



Gambar 10. Hasil Pengujian Menggunakan Dataset Eksternal

3.7 Implikasi Praktis Sistem pada Industri Penyortiran Buah

Sistem klasifikasi otomatis yang berhasil mencapai akurasi pengujian sebesar 98,88% ini menawarkan implikasi praktis berskala besar bagi industri pascapanen dan rantai distribusi pangan di Indonesia. Mengingat proses penyortiran di pasar lokal maupun industri saat ini masih sangat bergantung pada penilaian visual pekerja yang subjektif dan inkonsisten, sistem berbasis GUI ini menyediakan parameter validasi (*quality control*) yang terstandardisasi dan mampu melakukan pencatatan data secara otomatis. Untuk menutupi kelemahan klasifikasi *real-time* yang rentan terhadap gangguan sudut kamera dan pencahayaan, implementasi sistem ini di lapangan (seperti di fasilitas *quality assurance* industri) dapat direalisasikan dengan merancang stasiun inspeksi mekanis. Desain sistem dapat diintegrasikan menggunakan struktur *jig* khusus; pengguna cukup menekan sistem operasional tersebut menggunakan tangan, dan mekanisme akan secara langsung mengunci posisi sensor pendeteksi dengan mantap. Penguncian posisi sensor ini akan mengeliminasi variasi sudut pengambilan gambar serta memastikan distribusi pencahayaan yang konsisten menyorot buah, sehingga performa optimal model klasifikasi dapat direplikasi secara stabil di jalur konveyor berkecepatan tinggi.

4. Kesimpulan

Berdasarkan hasil implementasi dan analisis pada sistem klasifikasi tingkat pembusukan apel hijau menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50, dapat disimpulkan bahwa penerapan strategi *Transfer Learning* melalui skenario *Fine-Tuning* terbukti

sangat efektif. Model mampu mencapai titik konvergensi optimal secara efisien pada *epoch* ke-12 berkat mekanisme *Early Stopping*, serta menghasilkan performa yang sangat tinggi dengan tingkat akurasi mencapai 98,88% dan nilai *loss* validasi 0,02 pada *dataset* uji. Pada pengujian kondisi nyata (*real-time*) dan validasi *dataset* eksternal, sistem secara konsisten berhasil mengidentifikasi ketiga kategori (Segar, Mulai Membusuk, dan Busuk) dengan prediksi label yang tepat. Meskipun demikian, terdapat tantangan berupa fluktuasi dan penurunan nilai probabilitas (*confidence score*) pada rentang 48% hingga 82% saat pengujian *real-time*. Penurunan ini dipicu oleh sensitivitas model terhadap variasi pencahayaan alami dan *noise* latar belakang, serta adanya ambiguitas nilai fitur warna pada kelas transisi "Mulai Membusuk" yang berisikan dengan dua kelas lainnya. Secara keseluruhan, kendati dipengaruhi oleh kompleksitas lingkungan dan fitur visual, sistem ini tetap tangguh dan mampu memberikan keputusan klasifikasi akhir yang akurat.

Referensi

- [1] J. G. A. K. Mahendran R, Application of Computer Vision Technique on Sorting and Grading of Fruits and Vegetables, *J. Food Process. Technol.*, vol. s1, 2015, doi: 10.4172/2157-7110.s1-001.
- [2] R. A. Sanad Alsbu, P. Yarlagadda, and A. Karim, An Empirical Model for Predicting the Fresh Food Quality Changes during Storage, *Foods*, vol. 12, no. 11, Jun. 2023, doi: 10.3390/foods12112113.
- [3] I. Rojas Santelices, S. Cano, F. Moreira, and Á. Peña Fritz, "Artificial Vision Systems for Fruit Inspection and Classification: Systematic Literature Review," Mar. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s25051524.
- [4] S. Serra, B. Anthony, F. B. Sesillo, A. Masia, and S. Musacchi, "Determination of Post-Harvest Biochemical Composition, Enzymatic Activities, and Oxidative Browning in 14 Apple Cultivars," 2021, doi: 10.3390/foods.
- [5] S. Ranani and S. M. Soomar, "Laboratory Based Experimental Study on Microbial Spoilage of Commercially Available Fruits," *J Microb Biochem Technol*, vol. 11, p. 423, 2019, doi: 10.35248/1948-5948.19.11.423.
- [6] D. Qiu *et al.*, "Classification of Apple Color and Deformity Using Machine Vision Combined with CNN," *Agriculture (Switzerland)*, vol. 14, no. 7, Jul. 2024, doi: 10.3390/agriculture14070978.
- [7] M. Khoiruddin and S. Tena, "Fruit and Vegetable Classification using Convolutional Neural Network with MobileNetV2," *Journal of Applied Research In Computer Science and Information Systems*, vol. 2, no. 2, pp. 203–210, Dec. 2024, doi: 10.61098/jarcis.v2i2.197.
- [8] J. Zhang, C. Li, Y. Yin, J. Zhang, and M. Grzegorzec, "Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1013–1070, Feb. 2023, doi: 10.1007/s10462-022-10192-7.
- [9] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
- [10] M. R. Bhat *et al.*, "Apple diseases: detection and classification using transfer learning," *Quality Assurance and Safety of Crops and Foods*, vol. 15, no. SP1, pp. 27–37, 2023, doi: 10.15586/qas.v15iSP1.1167.
- [11] H. S. Mputu, A. Abdel-Mawgood, A. Shimada, and M. S. Sayed, "Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers," *IEEE Access*, vol. 12, pp. 8283–8295, 2024, doi: 10.1109/ACCESS.2024.3352745.
- [12] D. Masters and C. Luschi, "Revisiting Small Batch Training for Deep Neural Networks," Apr. 2018, [Online]. Available: <http://arxiv.org/abs/1804.07612>
- [13] S. Noppitak and O. Surinta, "dropCyclic: Snapshot Ensemble Convolutional Neural Network Based on a New Learning Rate Schedule for Land Use Classification," *IEEE Access*, vol. 10, pp. 60725–60737, 2022, doi: 10.1109/ACCESS.2022.3180844.
- [14] M. A. Ahsan, A. Qayyum, J. Qadir, and A. Razi, "An Active Learning Method for Diabetic Retinopathy Classification with Uncertainty Quantification," Dec. 2020, [Online]. Available: <http://arxiv.org/abs/2012.13325>