

# KLASIFIKASI KEMATANGAN BUAH SAWIT BERBASIS WEBSITE MENGGUNAKAN DEEP LEARNING

Claudia Delvia Pangati<sup>1</sup>, Genrawan Hoendarto<sup>2</sup>, Hendro<sup>3</sup>

<sup>12</sup>Informatika, <sup>3</sup> Sistem Informasi, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak  
<sup>1</sup>21421422\_claudia\_d\_p@widyadharma.ac.id, <sup>2</sup>genrawan@widyadharma.ac.id, <sup>3</sup>hendro@widyadharma.ac.id

## Abstract

*Oil palm fruit is one of the main agricultural commodities in Indonesia. The maturity level of the fruit significantly influences both the quality and quantity of the oil produced. Manual determination of fruit maturity is still frequently carried out; however, this method is subjective and inefficient. This study aims to develop a web-based application for classifying the maturity level of oil palm fruit using artificial intelligence. The model employed is a Convolutional Neural Network (CNN) with the ResNet50V2 architecture, trained on a dataset of oil palm fruit images categorized into three maturity levels: unripe, ripe, and overripe. The model was then integrated into a web-based application using the Flask framework as the user interface. The test results show that the model is capable of performing classification with high accuracy and fast response time. The application enables users to upload images of oil palm fruit and automatically obtain classification results through a responsive web interface. The implementation of this technology can enhance accuracy, objectivity, and efficiency in the process of determining fruit maturity. The findings of this study indicate that artificial intelligence-based systems hold great potential for supporting the harvesting process in the palm oil industry.*

**Keywords:** Image Classification, ResNet50V2, Convolutional Neural Network, Web Application

## Abstrak

Buah kelapa sawit merupakan salah satu komoditas utama di Indonesia. Tingkat kematangan buah sangat memengaruhi kualitas dan kuantitas minyak yang dihasilkan. Penentuan tingkat kematangan secara manual masih sering dilakukan, namun metode tersebut bersifat subjektif dan kurang efisien. Penelitian ini bertujuan untuk mengembangkan aplikasi klasifikasi tingkat kematangan buah kelapa sawit berbasis *web* menggunakan kecerdasan buatan. Model yang digunakan adalah *Convolutional Neural Network* (CNN) dengan arsitektur ResNet50V2, yang dilatih menggunakan dataset citra buah kelapa sawit pada tiga kategori tingkat kematangan, yaitu belum masak, masak, dan terlalu masak. Model kemudian diintegrasikan ke dalam aplikasi berbasis *web* menggunakan *framework Flask* sebagai antarmuka pengguna. Hasil pengujian menunjukkan bahwa model mampu melakukan klasifikasi dengan tingkat akurasi yang tinggi serta waktu respons yang cepat. Aplikasi ini memberikan fasilitas bagi pengguna untuk mengunggah citra buah kelapa sawit dan memperoleh hasil klasifikasi secara otomatis melalui tampilan *web* yang responsif. Penerapan teknologi ini dapat meningkatkan akurasi, objektivitas, dan efisiensi dalam proses penentuan kematangan buah sawit. Hasil penelitian ini menunjukkan bahwa sistem berbasis kecerdasan buatan memiliki potensi besar untuk diterapkan dalam mendukung proses panen pada industri kelapa sawit.

**Kata kunci:** Klasifikasi Citra, ResNet50V2, *Convolutional Neural Network*, Aplikasi *Web*

## 1. PENDAHULUAN

Kelapa sawit (*Elaeis guineensis*) merupakan salah satu komoditas strategis dalam sektor agribisnis Indonesia<sup>[1]</sup>. Sebagai negara penghasil kelapa sawit terbesar di dunia, Indonesia memiliki peran penting dalam memenuhi permintaan minyak kelapa sawit atau *Crude Palm Oil* (CPO) di pasar global<sup>[2]</sup>. Kualitas dan kuantitas CPO sangat dipengaruhi oleh tingkat kematangan buah sawit saat panen. Oleh karena itu, proses identifikasi tingkat kematangan buah menjadi aspek krusial dalam rantai produksi kelapa sawit.

Secara konvensional, klasifikasi tingkat kematangan buah kelapa sawit dilakukan secara manual oleh tenaga kerja terlatih. Namun, pendekatan ini memiliki kelemahan utama, yaitu subjektivitas dalam penilaian, potensi kesalahan manusia, serta keterbatasan dalam efisiensi waktu dan tenaga. Permasalahan ini menimbulkan kebutuhan akan sistem klasifikasi otomatis yang akurat, konsisten, dan mudah digunakan di lapangan. Seiring berkembangnya teknologi, pendekatan berbasis kecerdasan buatan, khususnya *deep learning*, telah digunakan secara luas untuk memecahkan berbagai permasalahan pengenalan citra. Salah satu arsitektur yang terbukti efektif dalam klasifikasi gambar adalah *Convolutional Neural Network* (CNN). CNN mampu mengenali pola visual secara otomatis melalui proses ekstraksi fitur dari data citra, seperti warna dan tekstur. Berbagai penelitian telah menunjukkan bahwa CNN unggul dalam akurasi klasifikasi objek visual dibandingkan metode tradisional. Salah satu arsitektur CNN yang digunakan adalah ResNet50V2, yang memiliki kemampuan untuk menangani permasalahan degradasi performa pada jaringan yang sangat dalam.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang aplikasi klasifikasi tingkat kematangan buah kelapa sawit berbasis *web* menggunakan model CNN dengan arsitektur ResNet50V2. Aplikasi ini dirancang agar dapat menerima input berupa citra buah kelapa sawit, memprosesnya melalui model *deep learning*, dan menghasilkan klasifikasi tingkat kematangan secara otomatis ke dalam kategori belum masak, masak, dan terlalu masak. Implementasi sistem dilakukan dalam lingkungan pengujian terkendali, dengan pencahayaan normal, dan hasil klasifikasi ditampilkan melalui antarmuka *web* yang ramah pengguna.

## 2. METODE PENELITIAN

### 2.1 Metode Penelitian

#### 2.1.1 Teknik Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan dengan mengambil dataset citra buah kelapa sawit menggunakan data publik yang tersedia di *platform* kaggle. Dataset ini berisi kumpulan gambar buah kelapa sawit dengan tiga tingkat kematangan, yaitu belum masak, masak, dan terlalu masak. Untuk meningkatkan kemampuan model dalam mengenali gambar yang tidak termasuk dalam distribusi data pelatihan (*out of distribution/OOD*) dilakukan modifikasi pada dataset dengan menambahkan satu kelas tambahan, yaitu kelas *unknown*, kelas ini berisi kumpulan gambar acak yang tidak terkait dengan buah kelapa sawit, seperti gambar buah lain, objek alam, atau benda umum lainnya. Penambahan kelas ini bertujuan agar sistem dapat mengenali dan mengklasifikasikan gambar-gambar yang berada di luar tiga kategori utama, sehingga model menjadi lebih robust dan adaptif terhadap input yang tidak sesuai atau asing terhadap data pelatihan aslinya.



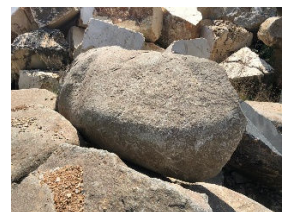
Belum Masak



Masak



Terlalu Masak



Unknown

Gambar 1. Sampel Dataset Setiap Kelas

#### 2.1.2 Teknik Analisis Sistem

Analisis dilakukan dengan pendekatan berorientasi objek menggunakan *Unified Modeling Language* (UML) untuk menggambarkan proses kerja sistem klasifikasi tingkat kematangan buah kelapa sawit.

#### 2.1.3 Aplikasi Perancangan Sistem

Perancangan sistem menggunakan bahasa pemrograman Python dengan Visual Studio Code untuk pengembangan aplikasi, serta Google Colab untuk pembuatan model.

### 2.2 Landasan Teori

#### 2.2.1 Tingkat Kematangan

Tingkat kematangan adalah Kumpulan persyaratan dari semua tingkat kematangan yang lebih rendah, di semua dimensi yang ditentukan<sup>[3]</sup>. Tingkat kematangan merupakan dimensi yang menunjukkan tahapan kematangan mulai dari yang terendah sampai tertinggi yang didasarkan atas *level* dan nama dari CMM<sup>[4]</sup>.

#### 2.2.2 Klasifikasi

Klasifikasi adalah proses pengelompokan data ke dalam kelas atau kategori yang telah ditentukan berdasarkan fitur atau atribut data<sup>[5]</sup>. Klasifikasi adalah suatu proses untuk menemukan satu model atau fungsi yang menjelaskan dan memisahkan *class* dari suatu data dengan tujuan agar dapat menggunakan model atau fungsi tersebut untuk memprediksi suatu *class* dari satu objek baru yang belum dikenali<sup>[6]</sup>.

#### 2.2.3 Deep Learning

*Deep learning* merupakan salah satu cabang dari *machine learning* yang bertujuan meniru metode kerja otak manusia dengan memakai *artificial neural network* atau jaringan saraf tiruan<sup>[7]</sup>. *Deep learning* merupakan teknik *machine learning* yang mengajarkan komputer untuk belajar dari data<sup>[8]</sup>.

#### 2.2.4 CNN

*Convolutional Neural Network* (CNN) adalah salah satu metode dari *Machine Learning* (ML) yang dikembangkan dari *Multi-Layer Perceptron* (MLP) diciptakan untuk mengerjakan data dua dimensi<sup>[9]</sup>. *Convolutional Neural Network* merupakan salah satu jenis algoritma *deep learning* yang dapat menerima *input* berupa gambar, menentukan aspek atau obyek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk “belajar” mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya<sup>[10]</sup>.

#### 2.2.5 ResNet50V2

ResNet50V2 merupakan *pre-trained* model CNN yang memiliki 48 lapis konvolusi dan 2 lapis *pooling*<sup>[11]</sup>. ResNet50v2 adalah arsitektur jaringan saraf konvolusional yang dibangun di atas model ResNet dan merupakan varian ResNet50 yang telah disempurnakan dengan penyertaan lapisan normalisasi *batch*, yang membantu menstabilkan proses pelatihan dengan mengurangi efek pergeseran kovariat<sup>[12]</sup>.

### 2.2.6 Pemrosesan Gambar

Pemrosesan gambar atau *image processing* adalah suatu bentuk pemrosesan sinyal dengan *input* berupa gambar sebagai *output* dengan teknik tertentu<sup>[13]</sup>. Pemrosesan gambar merupakan proses dimana data gambar dianalisis dan dimodifikasi menggunakan berbagai teknik atau algoritma<sup>[14]</sup>.

### 2.2.7 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) adalah bahasa pemodelan standar yang umum di bidang rekayasa perangkat lunak yang meliputi teknik notasi grafik untuk membuat model abstrak dari sistem tertentu<sup>[15]</sup>. *Unified Modelling Language* (UML) merupakan alat bantu untuk pemodelan sistem dengan menggunakan OOAD<sup>[16]</sup>.

### 2.2.8 Confusion Matrix

*Confusion matrix* adalah metrik evaluasi pengukuran yang digunakan dalam algoritma klasifikasi pembelajaran mesin untuk menemukan keakuratan model dengan menunjukkan kombinasi dari kelas *actual* dengan kelas prediksi<sup>[17]</sup>. *Confusion matrix* adalah label persegi yang digunakan untuk mengevaluasi kinerja model klasifikasi *machine learning* dengan membandingkan hasil prediksi model dengan nilai yang sebenarnya dalam *dataset* yang diketahui<sup>[18]</sup>. *Confusion matrix* digunakan untuk menghitung Accuracy, Precision, Recall, dan F1-Score berdasarkan data True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), False Positive Rate (FPR), True Negative Rate (TNR). Persamaan 1 digunakan untuk menghitung persentase accuracy, persamaan 2 digunakan untuk menghitung persentase precision, persamaan 3 digunakan untuk menghitung persentase recall, persamaan 4 digunakan untuk menghitung persentase F1-Score dengan jumlah keluaran kelas yang lebih dari dua.

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^N c_{i,i}}{\sum_{i=1}^N \sum_{j=1}^N c_{i,j}} \quad (1)$$

$$\text{Recall}_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FN_{\text{class}}} \quad (2)$$

$$\text{Precision}_{\text{class}} = \frac{TP_{\text{class}}}{TP_{\text{class}} + FP_{\text{class}}} \quad (3)$$

$$\text{F1-Score}_{\text{class}} = \frac{2 \times TP_{\text{class}}}{2 \times TP_{\text{class}} + FN_{\text{class}} + FP_{\text{class}}} \quad (4)$$

$$\text{FPR}_{\text{class}} = \frac{FP_{\text{class}}}{FP_{\text{class}} + TN_{\text{class}}} \quad (5)$$

$$\text{TNR}_{\text{class}} = \frac{TN_{\text{class}}}{FP_{\text{class}} + TN_{\text{class}}} \quad (6)$$

Keterangan:

True Positives (TP) adalah jumlah kasus positif yang secara benar diprediksi sebagai positif.

True Negatives (TN) adalah jumlah kasus negative yang secara benar diprediksi sebagai negative.

False Positives (FP) adalah jumlah kasus negative yang salah diprediksi sebagai positif.

False Negatives (FN) adalah jumlah kasus positif yang salah diprediksi sebagai negative.

## 3. HASIL DAN PEMBAHASAN

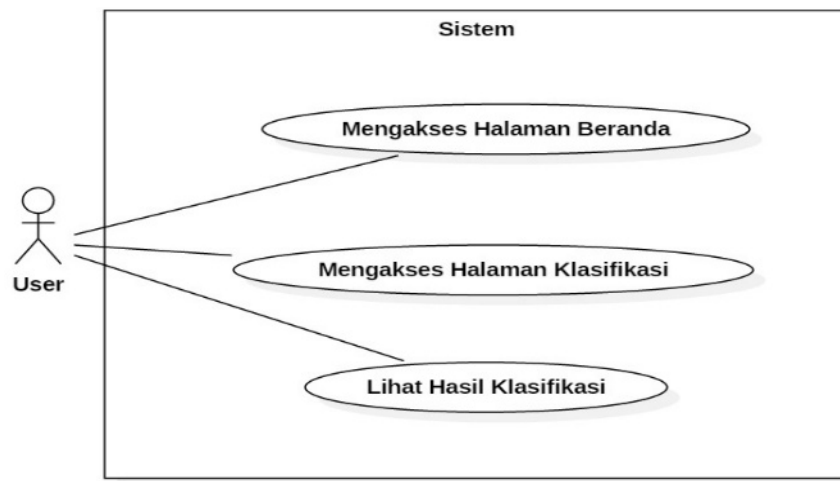
### 3.1 Unified Modeling Language (UML) Pada Sistem Klasifikasi Tingkat Kematangan Sawit

*Unified Modeling Language* (UML) pada sistem klasifikasi tingkat kematangan buah kelapa sawit digunakan untuk memvisualisasikan dan memahami desain sistem klasifikasi berbasis *web* yang dikembangkan. UML membantu menggambarkan bagaimana pengguna, seperti petani atau operator kebun, berinteraksi dengan sistem melalui diagram *use case*, diagram *sequence*, dan diagram *activity*. Diagram *use case* menjelaskan peran dan interaksi pengguna dengan sistem. Diagram *sequence* menggambarkan urutan interaksi antara objek-objek dalam sistem saat menjalankan proses tertentu, misalnya saat pengguna mengunggah gambar dan sistem memproses gambar tersebut menggunakan model deep learning untuk menentukan tingkat kematangan buah. Sementara itu, diagram *activity* menggambarkan alur proses logika kerja sistem secara detail, dimulai dari pengguna membuka aplikasi hingga memperoleh hasil klasifikasi.

#### 3.1.1 Diagram Use Case

*Use case* diagram pada Gambar 2 menunjukkan interaksi antara seorang aktor, yaitu *user* dengan sistem. Diagram ini menggambarkan tiga aktivitas utama yang dapat dilakukan oleh pengguna terhadap sistem. Pertama, pengguna dapat mengakses halaman beranda, yang berisi tampilan awal atau menu utama dari aplikasi. Pada menu utama ini ditampilkan alur kerja sistem, sehingga pengguna dapat memahami tahapan penggunaan aplikasi dengan lebih mudah. Kedua, pengguna dapat mengakses halaman klasifikasi, yaitu halaman yang memungkinkan pengguna untuk memasukkan data atau memilih parameter yang akan diklasifikasikan oleh sistem. Ketiga, pengguna dapat melihat hasil klasifikasi yang telah diproses oleh sistem melalui fitur lihat hasil klasifikasi. Setiap

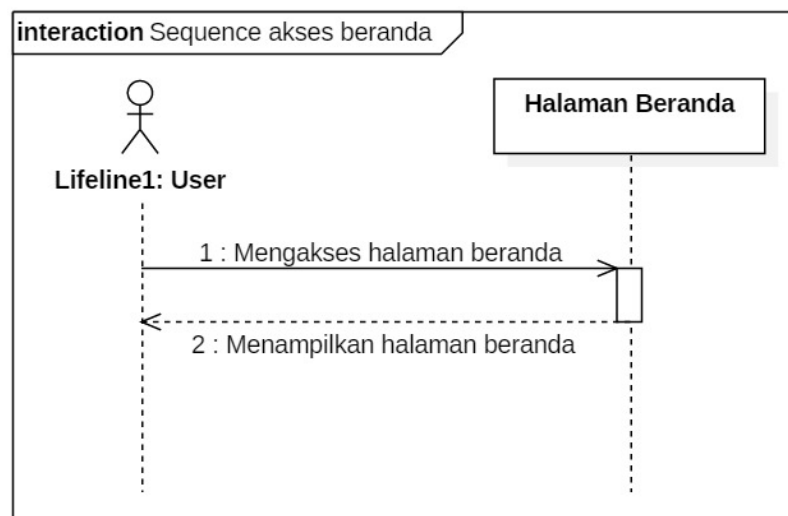
aktivitas atau *use case* ini digambarkan dalam bentuk *ellipse* dan dihubungkan langsung dengan aktor pengguna melalui garis asosiasi, yang menandakan adanya interaksi langsung. Diagram ini memberikan gambaran umum tentang fungsi utama yang tersedia dalam sistem serta peran pengguna dalam mengakses dan memanfaatkan fungsi-fungsi tersebut.



Gambar 2. Use Case Diagram Sistem Klasifikasi

### 3.1.2 Diagram Sequence Akses Halaman Beranda

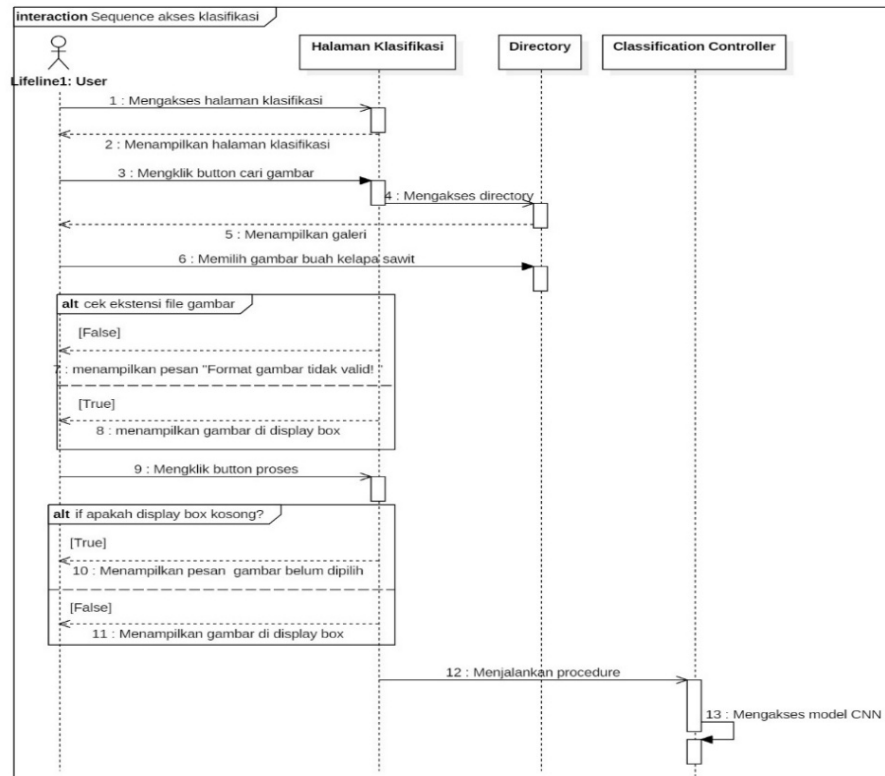
*Sequence* diagram pada Gambar 3 menggambarkan interaksi sederhana ketika pengguna mengakses halaman beranda. Proses dimulai ketika aktor user mengirim permintaan ke sistem dengan aksi “mengakses halaman beranda”. Permintaan ini diteruskan ke objek sistem yang terlibat, dalam hal ini ditandai sebagai “halaman klasifikasi”. Sistem merespon permintaan tersebut dengan mengembalikan tampilan berupa halaman beranda kepada pengguna. Respon ini digambarkan dengan panah kembali bertuliskan “menampilkan halaman beranda”. Diagram ini menekankan pada komunikasi dua arah yang cepat dan langsung antara pengguna dan sistem dalam konteks navigasi ke halaman utama aplikasi.



Gambar 3. Diagram Sequence Akses Halaman Beranda

### 3.1.3 Diagram Sequence Akses Halaman Klasifikasi

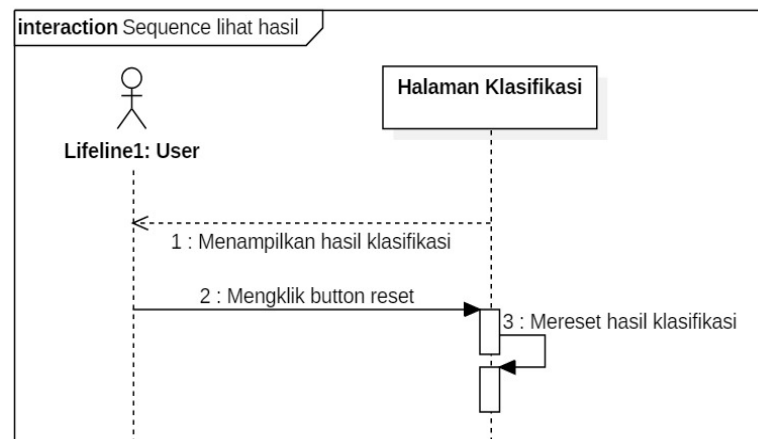
*Sequence* diagram pada Gambar 4 menggambarkan alur interaksi antara pengguna (user) dengan sistem saat melakukan proses klasifikasi gambar buah kelapa sawit. Proses Alur dimulai saat pengguna mengakses halaman klasifikasi, dan sistem merespons dengan menampilkan halaman tersebut. Setelah itu, pengguna mengklik tombol “cari gambar”, yang kemudian sistem akan mengakses direktori (galeri) perangkat untuk memungkinkan pengguna memilih gambar buah kelapa sawit. Setelah gambar dipilih, sistem melakukan pengecekan terhadap ekstensi file gambar. Jika ekstensi file tidak valid (bukan jpg, jpeg, atau png), sistem akan menampilkan pesan kesalahan “Format gambar tidak valid!”. Jika ekstensi valid, sistem akan menampilkan gambar tersebut pada display box. Selanjutnya, pengguna menekan tombol “proses” untuk memulai klasifikasi. Sistem kemudian memeriksa apakah display box kosong. Jika kosong, sistem akan menampilkan pesan “Gambar belum dipilih!”. Jika display box berisi gambar, sistem akan melanjutkan untuk menjalankan prosedur klasifikasi. Prosedur ini diteruskan ke Classification Controller, yang kemudian mengakses model CNN (Convolutional Neural Network) untuk melakukan klasifikasi gambar. Setelah klasifikasi dilakukan oleh model, sistem akan menampilkan hasilnya kepada pengguna.



Gambar 4. Diagram Sequence Akses Halaman Klasifikasi

### 3.1.4 Diagram Sequence Lihat Hasil Klasifikasi

*Sequence* diagram pada Gambar 5 menggambarkan alur interaksi antara pengguna (*user*) dengan sistem saat menampilkan dan mereset hasil dari proses klasifikasi. Proses dimulai ketika sistem menampilkan hasil klasifikasi kepada pengguna setelah proses klasifikasi selesai dilakukan. Kemudian, pengguna memiliki opsi untuk menghapus atau mengatur ulang hasil tersebut dengan mengklik tombol *reset*. Setelah tombol diklik, sistem akan menjalankan perintah untuk *mereset* hasil klasifikasi yang telah ditampilkan sebelumnya.



Gambar 5. Diagram Sequence Lihat Hasil Klasifikasi

Diagram ini menunjukkan interaksi yang sederhana namun penting dalam proses klasifikasi, yaitu bagaimana sistem memberikan fleksibilitas bagi pengguna untuk mengatur ulang hasil analisis jika diperlukan. Hal ini penting, misalnya ketika pengguna ingin melakukan klasifikasi ulang terhadap gambar lain atau jika terjadi kesalahan dalam pemilihan gambar sebelumnya.

## 3.2 Arsitektur dan Penyetelan Model

### 3.2.1 Base Model

Pada tahap awal, model ResNet50V2 digunakan sebagai base model. ResNet50V2 merupakan model Convolutional Neural Network (CNN) yang telah dilatih sebelumnya pada dataset besar ImageNet, sehingga memiliki kemampuan ekstraksi fitur yang kuat dan bersifat umum untuk berbagai tugas pengenalan gambar. Parameter “weights=imagenet” menunjukkan bahwa bobot awal model diperoleh dari pelatihan pada dataset ImageNet dan “include\_top=False” berarti bagian klasifikasi terakhir fully connected layer model dihilangkan,

sehingga hanya bagian convolutional base yang digunakan sebagai ekstraktor fitur. Selanjutnya, input shape = 256, 256, 3 menetapkan ukuran gambar input yang diterima oleh model, yakni  $256 \times 256$  piksel dengan tiga saluran warna (RGB).

### 3.2.2 Penambahan Layer Costum

Setelah mendapatkan fitur dari base model, dilakukan proses pooling dan beberapa layer tambahan untuk memproses fitur menjadi prediksi kelas:

#### a) GlobalAveragePooling2D

Layer ini digunakan untuk untuk mereduksi dimensi data fitur yang dihasilkan oleh base model ResNet50V2. Sebagai pengganti Flatten, GlobalAveragePooling2D menghitung rata-rata dari setiap channel fitur map, sehingga hasilnya berupa satu nilai untuk setiap channel.

#### b) BatchNormalization

Layer ini berfungsi untuk menormalkan *output* dari layer sebelumnya. Dengan mengatur ulang distribusi aktivasi agar memiliki mean mendekati 0 dan standar deviasi mendekati 1, proses pelatihan menjadi lebih stabil dan cepat konvergen. *Batch normalization* juga dapat mengurangi sensitivitas terhadap inisialisasi bobot dan bertindak sebagai regularisasi tambahan karena menambahkan sedikit *noise* selama pelatihan.

#### c) Dropout

Dropout adalah teknik regularisasi yang digunakan untuk mencegah model menjadi terlalu bergantung pada neuron tertentu (*overfitting*). Neuron dinonaktifkan secara acak setiap iterasi pelatihan, sehingga jaringan belajar untuk menyebarkan informasi lebih merata ke seluruh jaringan. Teknik ini mendorong model untuk menjadi lebih robust dan general terhadap data baru.

#### d) Dense

Lapisan Dense, atau yang dikenal juga sebagai fully connected layer, merupakan bagian dari jaringan saraf tiruan di mana setiap neuron terhubung dengan semua neuron pada lapisan sebelumnya. Fungsi utama dari lapisan ini adalah untuk menggabungkan semua fitur yang telah diekstraksi oleh lapisan konvolusional dan pooling menjadi representasi akhir yang digunakan untuk klasifikasi.

### 3.2.3 Freezing Base Model

Freezing base model merupakan langkah penting dalam transfer learning yang dilakukan dengan membekukan semua layer dari model dasar ResNet50V2, agar bobotnya tidak diperbarui selama proses pelatihan. Tujuan utama dari proses ini adalah untuk mempertahankan pengetahuan yang telah dipelajari model tersebut dari dataset besar seperti ImageNet, sehingga model tidak perlu belajar dari nol. Dengan membekukan base model, proses pelatihan menjadi lebih efisien karena hanya bagian kepala klasifikasi yang dilatih, dan ini sangat membantu terutama ketika dataset yang digunakan berukuran kecil atau terbatas. Selain itu, freezing juga mencegah terjadinya *overfitting*, karena fitur visual seperti tepi, bentuk, dan tekstur yang telah dikuasai model pretrained tetap dipertahankan. Teknik ini digunakan pada tahap awal pelatihan, dan setelah model menunjukkan performa yang stabil, beberapa layer dari base model dapat dibuka kembali (*unfreeze*) untuk melakukan fine tuning agar model lebih menyesuaikan diri dengan karakteristik spesifik data yang digunakan.

### 3.2.4 Kompilasi Model

Setelah arsitektur model selesai dibangun, tahap selanjutnya adalah melakukan kompilasi model. Proses ini melibatkan penetapan beberapa parameter penting yang menentukan cara model dilatih dan dievaluasi. Model kemudian dikompilasi menggunakan optimizer Adam dengan learning rate sebesar  $1e-5$ , serta loss function `categorical_crossentropy` yang sesuai untuk klasifikasi multi-kelas dengan label dalam format one-hot encoding.

### 3.2.5 Pelatihan Model

Pada tahap awal pelatihan, dilakukan proses freezing terhadap seluruh layer pada base model ResNet50V2, sehingga bobot-bobot hasil pre-training dari ImageNet tidak diperbarui. Strategi ini diterapkan agar pelatihan lebih efisien dan menghindari *overfitting*, terutama karena jumlah data pelatihan yang terbatas. Hanya bagian custom head yang dilatih untuk menyesuaikan diri dengan karakteristik data sawit. Setelah model menunjukkan kestabilan, dilakukan fine-tuning dengan membuka 35 layer terakhir dari base model, sehingga layer-layer tersebut dapat dilatih ulang dan lebih spesifik terhadap pola visual pada data yang digunakan.

### 3.2.6 Evaluasi Model

Evaluasi model dilakukan dengan menggunakan confusion matrix dan classification report. Confusion matrix digunakan untuk melihat distribusi prediksi model terhadap label yang sebenarnya, sedangkan classification report memberikan metrik evaluasi berupa precision, recall, F1-score, dan support untuk masing-masing kelas.

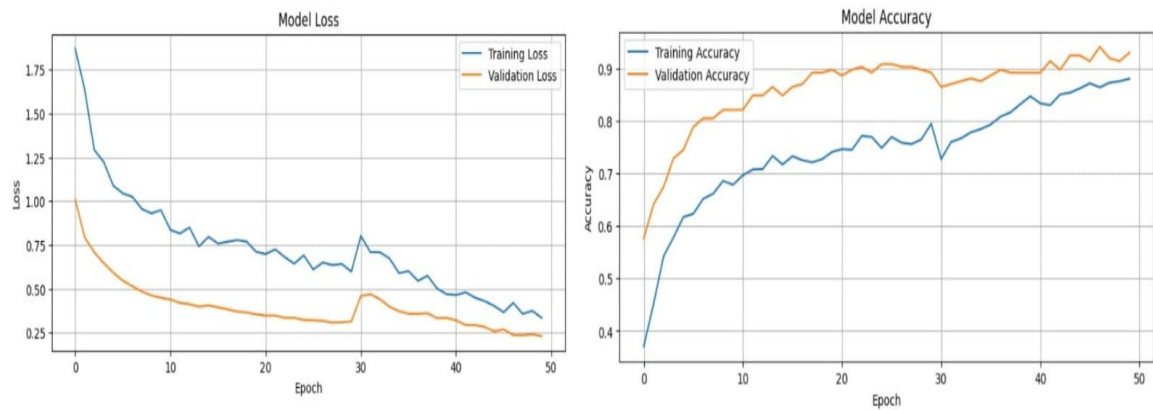
### 3.2.7 Penyimpanan Model

Model yang dilatih disimpan menggunakan `model.save()` dengan format .h5 untuk digunakan dalam sistem.

## 3.3. Hasil dan Pengujian

Evaluasi terhadap model dilakukan secara bertahap dengan menguji kinerja model pada dataset khusus, yaitu *dataset* validasi yang digunakan selama proses pelatihan, serta *dataset* pengujian yang sepenuhnya terpisah dan tidak pernah dilihat oleh model sebelumnya. Proses evaluasi ini mencakup berbagai metrik, seperti akurasi, nilai *loss*, analisis melalui *confusion matrix*.

### 3.3.1 Grafik Evaluasi Accuracy dan Loss

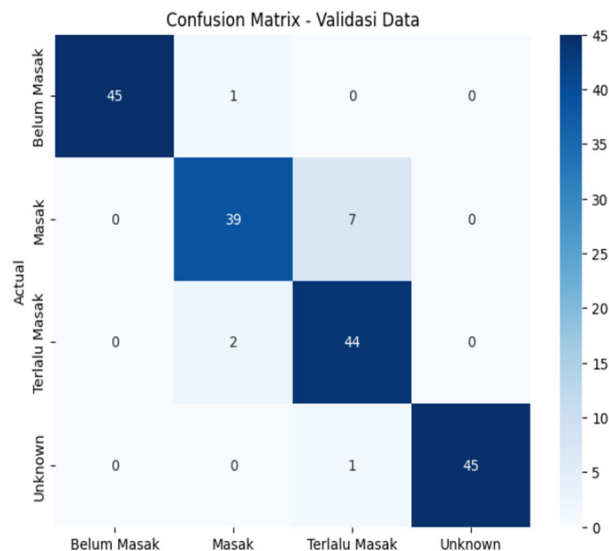


Gambar 6. Grafik Evaluasi Loss dan Accuracy

Grafik pada Gambar 6 menunjukkan performa model selama proses pelatihan selama 50 epoch, yang terdiri dari dua bagian: akurasi dan *loss*. Pada grafik akurasi, terlihat bahwa akurasi model pada data pelatihan (*training accuracy*) dan data validasi (*validation accuracy*) terus meningkat seiring bertambahnya *epoch*. Akurasi validasi mencapai angka di atas 90 persen, menunjukkan bahwa model mampu melakukan generalisasi dengan baik terhadap data yang tidak dilatih. Sementara itu, grafik *loss* menunjukkan penurunan yang konsisten baik pada data pelatihan maupun validasi, yang menandakan bahwa kesalahan prediksi model semakin kecil. Tidak terlihat indikasi *overfitting* yang signifikan karena *loss* validasi tidak mengalami kenaikan tajam dan tetap berada di bawah *loss* pelatihan.

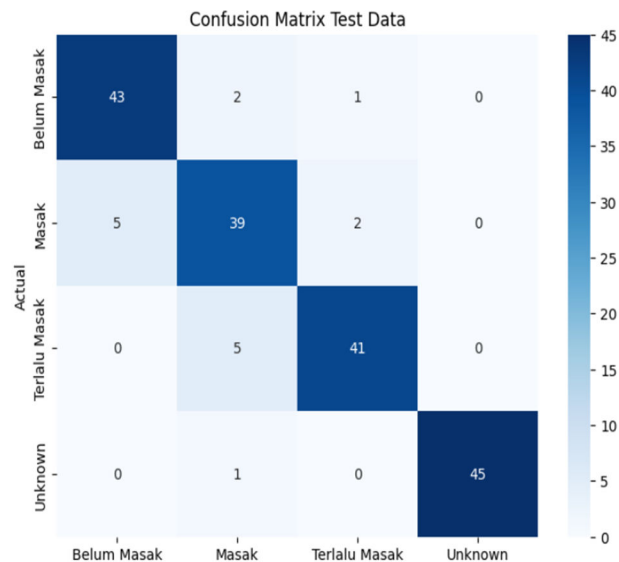
#### 3.3.2 Confusion Matrix

*Confusion matrix* pada Gambar 7 menunjukkan performa model klasifikasi terhadap data validasi dengan tiga kelas kematangan kelapa sawit yaitu belum masak, masak, terlalu masak, dan satu kelas unknown. Gambar ini menunjukkan perbandingan antara label aktual dan prediksi model untuk setiap kelas. Pada baris belum masak, sebanyak 45 dari 46 data berhasil diprediksi dengan benar. Untuk kelas masak, model memprediksi 39 data dengan benar. Pada kelas terlalu masak, 44 data diklasifikasikan dengan benar. Sementara itu, kelas unknown dengan 45 data diprediksi benar. Dari hasil ini, dapat disimpulkan bahwa model memiliki performa yang sangat baik, terutama dalam membedakan kelas-kelas dengan ciri visual yang cukup kontras.



Gambar 7. Confusion Matrix Pada Data Validasi

Confusion matrix pada Gambar 8 menunjukkan performa model klasifikasi terhadap data uji dengan tiga kelas kematangan kelapa sawit yaitu belum masak, masak, terlalu masak, dan satu kelas unknown. Dari gambar terlihat bahwa pada kelas belum masak, sebanyak 43 dari 46 data diklasifikasikan dengan benar. Pada kelas masak, 39 data diprediksi benar. Untuk kelas terlalu masak, 41 data diprediksi benar. Sementara itu, kelas unknown menunjukkan performa terbaik dengan 45 data diklasifikasikan dengan benar.



Gambar 8. Confusion Matrix Pada Data Uji

### 3.4 Hasil Evaluasi Model

Tabel 1. Hasil Evaluasi Model pada Data Validasi

Loss	Accuracy	Precision	Recall	F1-Score
0.2307	93.48%	0.9349	0.9348	0.9342

Pada Tabel 1 menunjukkan performa model pada data validasi, yaitu data yang digunakan selama proses pelatihan untuk memantau generalisasi model terhadap data yang tidak dilatih secara langsung. Nilai loss sebesar 0.2307 menunjukkan tingkat kesalahan prediksi model yang relatif rendah. Kemudian, nilai accuracy sebesar 93.48 persen data validasi berhasil diklasifikasikan dengan benar oleh model. Nilai precision 0.9349, recall 0.9348, dan F1-score 0.9342 yang menunjukkan bahwa model memiliki kinerja yang seimbang dan konsisten dalam mengidentifikasi kelas dengan benar, menghindari kesalahan positif maupun negatif secara proporsional.

Tabel 2. Hasil Evaluasi Model pada Data Uji

Loss	Accuracy	Precision	Recall	F1-Score
0.1090	97.28%	0.9740	0.9728	0.9729

Pada Tabel 2 menyajikan hasil performa model pada data uji, yaitu data yang benar-benar belum pernah dilihat oleh model sebelumnya dan digunakan untuk mengukur kemampuan generalisasi secara final. Nilai loss meningkat menjadi 0.1090 dengan nilai accuracy yang masih cukup tinggi yaitu 97.28 persen dari data uji berhasil diklasifikasikan dengan benar. Nilai precision 0.9740, recall 0.9728, dan F1-score 0.9729, menandakan bahwa performa model masih sangat baik dan tidak terlalu mengalami penurunan drastis meskipun diterapkan pada data baru.

Tabel 3. Hasil Evaluasi Model Per Kelas

Kelas	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Belum Masak	0.0629	97.82%	0.2088	97.82%	0.2454	93.47%
Masak	0.2834	89.67%	0.4191	82.60%	0.1316	97.82%
Terlalu Masak	0.1345	95.10%	0.2819	93.47%	0.0271	100.00%
Unknown	0.0424	98.64%	0.0129	100.00%	0.0320	97.82%

Pada Tabel 3 ditampilkan performa model klasifikasi terhadap masing-masing kelas, yaitu belum masak, masak, terlalu masak, dan unknown. Evaluasi dilakukan menggunakan metrik *train loss*, *train accuracy*, *validation loss*, *validation accuracy*, *test loss*, dan *test accuracy*. Kelas belum masak menunjukkan performa yang sangat baik dengan *train accuracy* dan *validation accuracy* sebesar 97,82 persen, serta *test accuracy* 93,47 persen, yang mengindikasikan konsistensi model dalam mengenali kelas ini di seluruh tahap pelatihan, validasi, hingga pengujian. Pada kelas masak, performa model tampak sedikit lebih rendah, dengan *train accuracy* sebesar 89,67 persen dan *validation accuracy* hanya 82,60 persen, yang menunjukkan bahwa model mengalami kesulitan dalam mengenali pola pada kelas ini, dengan *test accuracy* yang kembali tinggi di angka 97,82 persen. Kelas terlalu masak memberikan hasil yang sangat baik dengan *train accuracy* sebesar 95,10 persen, *validation accuracy* 93,47 persen, dan *test accuracy* mencapai 100 persen, serta nilai *test loss* yang sangat rendah, menandakan generalisasi model yang kuat terhadap kelas ini. Kelas *unknown* menunjukkan performa akurasi 98,64 persen *train accuracy*, dan 100 persen pada *validation accuracy*, serta *test accuracy* sebesar 97,82 persen disertai loss yang sangat kecil.

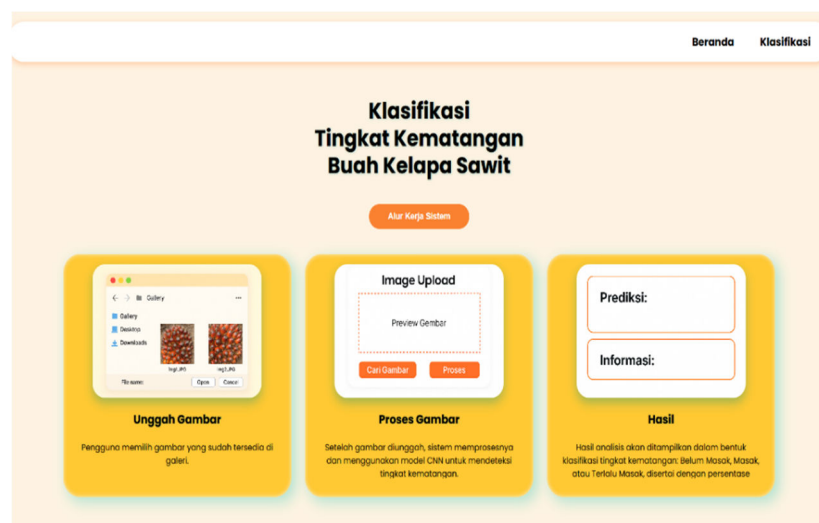


### 3.2 Perancangan Antarmuka

Antarmuka pengguna pada sistem ini telah dirancang secara baik dan terperinci, dengan menggabungkan berbagai komponen yang memberikan tampilan informatif dan mudah dipahami oleh pengguna. Desain antarmuka yang terstruktur ini mempermudah pengguna dalam mengoperasikan sistem, berikut ini merupakan beberapa tampilan dari sistem klasifikasi tingkat kematangan buah kelapa sawit.

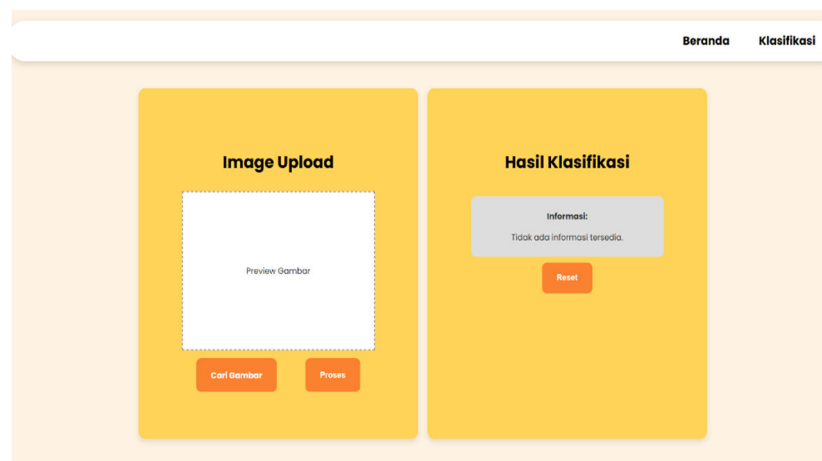
#### 3.2.1 Halaman Beranda

Tampilan halaman beranda pada sistem ini dirancang untuk memberikan gambaran menyeluruh mengenai fungsi dan tujuan dari sistem klasifikasi tingkat kematangan buah kelapa sawit. Di bagian tengah halaman, pengguna akan menemukan penjelasan mengenai alur kerja sistem yang disusun secara terstruktur dan mudah dipahami. Alur kerja ini terdiri dari tiga tahap utama yang harus dilalui untuk mendapatkan hasil klasifikasi yang akurat. Tahap pertama adalah unggah gambar, di mana pengguna diminta untuk mengunggah gambar buah kelapa sawit yang akan dianalisis. Gambar tersebut harus memenuhi ketentuan tertentu agar sistem dapat memprosesnya secara optimal. Setelah gambar berhasil diunggah, sistem akan memasuki tahap kedua, yaitu proses gambar. Pada tahap ini, sistem secara otomatis akan menganalisis gambar menggunakan model kecerdasan buatan yang telah dilatih untuk mengenali tingkat kematangan buah sawit berdasarkan ciri-ciri visual tertentu. Tahap terakhir adalah hasil, di mana sistem akan menampilkan output berupa tingkat kematangan buah kelapa sawit disertai dengan informasi pendukung seperti tingkat akurasi prediksi dan penjelasan tambahan yang relevan.



Gambar 9. Tampilan Antarmuka Halaman Beranda

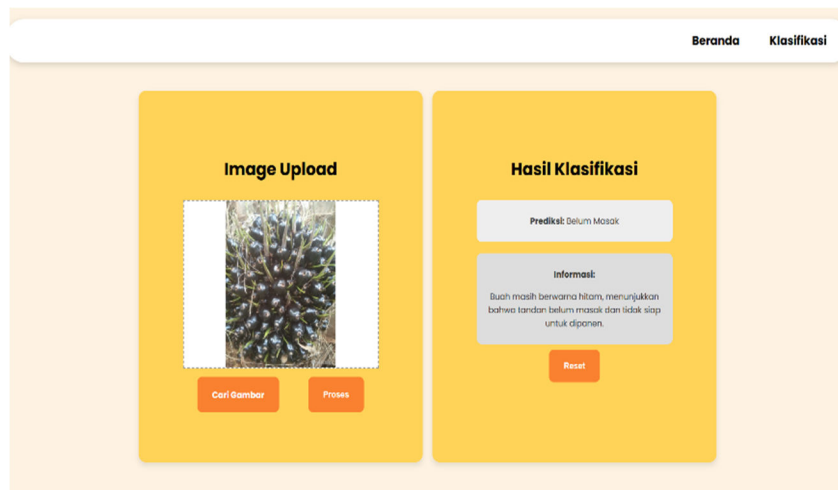
#### 3.2.2 Halaman Klasifikasi



Gambar 10. Tampilan Antarmuka Halaman Klasifikasi

Tampilan pada gambar ini merupakan halaman Klasifikasi. Halaman ini terbagi menjadi dua bagian utama. Di sisi kiri terdapat fitur Image Upload, tempat pengguna mengunggah gambar buah kelapa sawit dari perangkat melalui tombol “Cari Gambar”, dan gambar yang dipilih akan ditampilkan dalam kotak preview. Setelah gambar dimuat, pengguna dapat menekan tombol “Proses” untuk memulai analisis gambar. Di sisi kanan terdapat bagian Hasil Klasifikasi, yang menampilkan hasil analisis berupa tingkat kematangan dan informasi akurasi dari model. Saat belum ada gambar yang diproses, sistem menampilkan pesan “Tidak ada informasi tersedia”. Selain itu, tersedia tombol “Reset” untuk menghapus hasil klasifikasi dan memulai proses dari awal.

### 3.2.3 Halaman hasil klasifikasi



Gambar 11. Tampilan Antarmuka Halaman Output Klasifikasi

Tampilan halaman hasil proses klasifikasi yang dirancang untuk menyajikan informasi prediksi tingkat kematangan buah kelapa sawit secara otomatis. Setelah pengguna mengunggah gambar buah sawit melalui sistem, gambar tersebut diproses. Hasil prediksi yang ditampilkan pada halaman ini mencakup label klasifikasi yang sesuai dengan kondisi buah pada gambar, serta persentase akurasi dari model terhadap prediksi tersebut.

## 4. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah:

- Penelitian ini berhasil membangun sistem klasifikasi tingkat kematangan buah kelapa sawit berbasis arsitektur ResNet50V2, dengan empat kelas: belum masak, masak, terlalu masak, dan unknown. Model mencapai akurasi validasi 93,48 persen dan akurasi uji 97,28 persen, dengan precision, recall, dan F1-score di atas 97 persen.
- Jumlah data yang digunakan untuk setiap kelas disusun secara merata, yaitu masing-masing terdiri dari 460 citra untuk kategori belum masak, masak, terlalu masak, serta satu kelas tambahan unknown. Penggunaan jumlah data yang setara ini dilakukan secara terencana untuk mencegah ketimpangan distribusi data yang berpotensi menyebabkan model mengalami bias terhadap kelas tertentu.
- Penambahan kelas unknown dengan pendekatan out of distribution (OOD) terbukti membantu model dalam mengenali objek yang berada di luar kategori kematangan buah kelapa sawit, seperti gambar benda atau objek lain yang jelas bukan buah sawit. Namun demikian, model masih mengalami kesulitan ketika menerima masukan berupa gambar buah lain yang memiliki bentuk atau warna yang mirip dengan kelapa sawit. Hal ini menunjukkan bahwa meskipun pendekatan OOD efektif dalam meningkatkan ketahanan model terhadap data yang tidak sesuai distribusi, kemampuan model untuk membedakan antara buah kelapa sawit dan objek yang secara visual serupa masih perlu ditingkatkan..
- Aplikasi klasifikasi citra berhasil dibangun dan telah mampu menjalankan seluruh fitur fungsional dengan baik, sebagaimana dibuktikan melalui pengujian black box yang dijelaskan pada Bab IV. Pengujian ini difokuskan pada aspek fungsionalitas sistem, yang menunjukkan bahwa setiap fitur utama, seperti proses unggah gambar dan klasifikasi citra, telah berjalan sesuai dengan yang diharapkan. Namun demikian, sistem masih memiliki keterbatasan pada aspek input, yaitu saat ini hanya dapat mengunggah dan memproses satu gambar dalam satu kali proses.
- Pada sistem yang dibangun, hasil klasifikasi tidak disimpan ke dalam database. Hal ini menunjukkan bahwa sistem hanya berfungsi sebagai alat bantu klasifikasi tingkat kematangan buah kelapa sawit secara langsung tanpa menyimpan riwayat klasifikasi yang telah dilakukan

## 5. SARAN

Sebagai saran untuk pengembangan lebih lanjut, beberapa hal yang dapat dipertimbangkan:

- Menyimpan Hasil Klasifikasi dalam Database

Pada sistem yang dibangun hasil klasifikasi tidak disimpan ke dalam database, sehingga data hilang setelah halaman ditutup atau proses diulang. Hal ini menyulitkan pengguna untuk meninjau kembali hasil sebelumnya. Oleh karena itu, disarankan agar sistem dikembangkan dengan fitur penyimpanan ke database, sehingga riwayat klasifikasi dapat direkam dan dimanfaatkan untuk keperluan analisis, pelaporan, atau pengambilan keputusan di masa mendatang.

b. Pengembangan Sistem untuk Mendukung Multi-Batch Processing Gambar

Untuk pengembangan selanjutnya, sistem sebaiknya dapat menerapkan pemrosesan secara multi-batch guna meningkatkan kecepatan dan kemudahan dalam mengklasifikasikan tingkat kematangan buah kelapa sawit. Hal ini dikarenakan pada sistem saat ini, pengguna hanya dapat memilih satu gambar dalam satu kali proses, sedangkan dengan multi-batch, pengguna dapat mengunggah banyak gambar sekaligus dalam satu kali proses.

c. Penambahan Kelas pada Tingkat Kematangan Buah Sawit

Variasi kelas klasifikasi saat ini hanya mencakup belum masak, masak, dan terlalu masak. Pada pengembangan selanjutnya, dapat ditambahkan variasi kelas seperti mengkal dan busuk untuk klasifikasi yang lebih detail dan membantu pengambilan keputusan panen yang lebih akurat.

## UCAPAN TERIMA KASIH

Puji syukur peneliti panjatkan kepada Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya, sehingga jurnal ini dapat diselesaikan dengan baik. Peneliti juga mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, arahan, serta bantuan baik secara langsung maupun tidak langsung selama proses penyusunan jurnal ini.

## DAFTAR PUSTAKA

- [1] Wandana, F. A., Dhora, A., & Zukfika, Z. (2024). *Budidaya Kelapa Sawit: Panduan Lengkap dari Pembibitan Hingga Panen*. PT. Star Digital Publishing, Yogyakarta.
- [2] Putro, L. H. S. (2023). *Biometana dari Air Limbah Pabrik Pengolahan Kelapa Sawit*. PT. RajaGrafindo Persada – Rajawali Pers, Jakarta.
- [3] Aisyah, T., et al. (2024). *Transformasi Digital (Pengantar Literasi Digital untuk Mahasiswa)*. Seval Literindo Kreasi, Bandung.
- [4] Mahendrawathi. (2024). *Bussines Process Management, Konsep dan Implementasi*. CV Andi Offset, Yogyakarta.
- [5] Saptadi, T. S., et al. (2024). *DATA MINING*. Cendikia Mulia Mandiri, Yogyakarta.
- [6] Yendrizal. (2022). *Monograf Algoritma C4.5 Pada Teknik Klasifikasi Penyusutan Volume Pupuk*. CV. Azka Pustaka, Padang.
- [7] Widhiada, W., Kusuma, I. G. B. W., & Diputra, A. A. G. P. (2023). *Kaki Robot Bionic Berbasis Deep Learning Neural Network Control*. CV Budi Utama, Yogyakarta.
- [8] Setiawan, W. (2021). *Deep Learning menggunakan Convolutional Neural Network: Teori dan Aplikasi*. Media Nusa Creative, Surabaya.
- [9] Harani, N. H., & Hasanah, M. (2020). *Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Berbasis Python*. Kreatif Publis Nusantara, Jakarta.
- [10] Putro, E. C., Awangga, R. M., & Andarsyah, R. (2020). *Tutorial Object Detection People with Faster Region-Based Convolutional Neural Network (Faster R-CNN)*. Kreatif Industry Nusantara, Jakarta.
- [11] Lina. (2024). *Teknologi Kecerdasan Buatan untuk Pendeteksian Sel Darah*. CV Budi Utama, Yogyakarta.
- [12] Nguyen, Q.-V., Huynh, C.-P., Nguyen, N. T., & Le-Minh, H. (2023). *The 12th Conference on Information Technology and Its Applications*. Springer Nature Switzerland, Cham.
- [13] Putra, R. R., Isa, I. G. T., & Malyan, A. B. J. (2023). *Buku Ajar Pengantar Deep Learning dalam Pemrosesan Citra*. PT Nasya Expanding Management, Bandung.
- [14] Fatimah, et al. (2024). *Pencitraan Diagnostik Wanita: Prinsip, Patologi Umum dan Modalitas Imejing*. Cendikia Mulia Mandiri, Yogyakarta.
- [15] Sari, R. F., & Utami, A. (2021). *Rekayasa Perangkat Lunak Berorientasi Objek Menggunakan PHP*. Andi, Yogyakarta.
- [16] Putri, R. E., Tasril, V., & Lestari, A. N. (2024). *Pengembangan Sistem Bimbingan dan Konseling*. Serasi Media Teknologi, Surabaya.
- [17] Kahlil, et al. (2023). *Computer Vision Berbasis Deep Learning untuk Aplikasi Pertanian: Teori dan Praktik*. Syiah Kuala University Press, Banda Aceh.
- [18] Khozaimi. (2023). *Kecerdasan Komputasional*. PT. Literasi Nusantara Abadi Grup, Jakarta.