OPTIMALISASI CONVOLUTION NEURAL NETWORK (CNN) DALAM MENGINDENTIFIKASI TINGKAT KEMATANGAN BUAH CPO

SKRIPSI

DISUSUN OLEH

JIHAN SHAKILA
2109010009

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025

OPTIMALISASI CONVOLUTION NEURAL NETWORK (CNN) DALAM MENGINDENTIFIKASI TINGKAT KEMATANGAN BUAH CPO

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Sistem Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

JIHAN SHAKILA

NPM. 2109010009

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

LEMBAR PENGESAHAN

Judul Skripsi	:	OPTIMALISASI CONVOLUTION NEURAL NETWORK (CNN) DALAM MENGINDETIFIKASI TINGKAT KEMATANGAN BUAH CPO
Nama Mahasiswa	:	Jihan Shakila
NPM	:	2109010009
Program Studi	:	Sistem Informasi

Menyetujui
Komisi Pembimbing

(Assoc. Prof. Dr. Al-Khowarizmi, M.Kom)
NIDN. 0127099201

Ketua Program Studi

(Dr. Firahmi Rizky, S.Kom.,M.Kom)
NIDN. 0116079201

Dekan

(Assoc. Prof.Dr. Al-Khowarizmi, M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

OPTIMALISASI CONVOLUTION NEURAL NETWORK (CNN) DALAM MENGINDENTIFIKASI TINGKAT KEMATANGAN BUAH CPO

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 01 Juli 2025

Yang membuat pernyataan

Jihan Shakila

NPM. 2109010009

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Jihan Shakila
NPM	: 2109010009
Program Studi	: Sistem Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

OPTIMALISASI CONVOLUTIONA NEURAL NETWORK (CNN) DALAM MENGINDENTIFIKASI KEMATANGAN BUAH CPO

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 01 Juli 2025

Yang membuat pernyataan

Jihan Shakila

NPM. 2109010009

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Jihan Shakila
Tempat dan Tanggal Lahir	: Medan, 22 Mei 2003
Alamat Rumah	: Jl Marelan psr 4 Barat
Telepon/Faks/HP	081536742923
E-mail	: jihanshakila03@gmail.com
Instansi Tempat Kerja	:
Alamat Kantor	:

DATA PENDIDIKAN

SD	: SD Negeri 091566 Bah Jambi	TAMAT: 2015
SMP	: SMP Harapan Mekar	TAMAT: 2018
SMA	: SMA YAPIM Marelan	TAMAT: 2021

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Prof. Dr. Akrim, M.Pd selaku Wakil Rektor II Universitas Muhammadiyah Sumatera Utara.
3. Bapak Assoc. Prof. Dr. Rudianto, S.Sos., M.Si selaku Wakil Rektor III Universitas Muhammadiyah Sumatera Utara.
4. Bapak Assoc. Prof. Dr. Al-Khowarizmi, M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Bapak Halim Maulana., ST., M.Kom selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi.
6. Bapak lutfi basit, S.Sos., M.I.Kom selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
7. Bapak Martiano, S.Pd, S.Kom., M.Kom selaku Ketua Jurusan Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi.
8. Bapak Assoc. Prof. Dr. Al-Khowarizmi, M.Kom selaku dosen pembimbing saya telah mempermudah saya dalam pembimbingan sehingga skripsi saya dapat selesai tepat waktu.

9. Ayah dan Bunda saya selaku orang tua yang telah memberi support dan doa dalam pelaksanaan skripsi saya
10. Nazwa selaku adik saya yang pendukung dan teman curhat saya dalam menyelesaikan skripsi ini.
11. Diri saya sendiri Jihan Shakila yang sudah berusahan keras dan berjuang sejauh ini. Mampu menguatkan dan meyakinkan bahwa semuanya akan selesai pada waktunya.
12. Dini selaku sahabat saya yang mendukung dan menemani saya dalam menyelesaikan skripsi saya.
13. Teman saya Alifa, Nazwa (jon), ica (mbak tik), riski perdana (jon 2) yang terlibat langsung dalam mendukung saya dari awal semester perkuliahan hingga akhir semester.

Medan, 01 Juli 2025

Penulis

Jihan Shakila

OPTIMALISASI CONVOLUTION NEURAL NETWORK (CNN) DALAM MEGINDENTIFIKASI KEMATANGAN BUAH CPO

ABSTRAK

Kelapa sawit merupakan salah satu komoditas utama di Indonesia, yang hasil utamanya berupa minyak kelapa sawit (Crude Palm Oil/CPO) sangat bergantung pada tingkat kematangan buah saat dipanen. Penentuan tingkat kematangan buah kelapa sawit yang dilakukan secara manual oleh petani sering kali menyebabkan ketidaktepatan, sehingga diperlukan sistem otomatis yang mampu melakukan identifikasi dengan akurat. Penelitian ini bertujuan untuk mengidentifikasi tingkat kematangan buah kelapa sawit menggunakan algoritma Convolutional Neural Network (CNN). Dataset yang digunakan terdiri dari 429 gambar buah sawit yang diklasifikasikan ke dalam tiga kelas, yaitu Belum Masak, Masak, dan Terlalu Masak. Proses pelatihan model dilakukan menggunakan arsitektur InceptionV3 melalui metode transfer learning. Seluruh implementasi dilakukan di platform Google Colaboratory, dengan penerapan teknik augmentasi data, normalisasi, dan image enhancement sebagai bagian dari preprocessing. Antarmuka pengguna berbasis Gradio juga dikembangkan agar sistem dapat diakses dan digunakan secara interaktif. Hasil pelatihan menunjukkan bahwa model CNN mampu mencapai akurasi validasi sebesar 87.65%, menunjukkan performa yang cukup baik dalam klasifikasi tingkat kematangan buah sawit. Sistem yang dibangun diharapkan dapat menjadi solusi pendukung pengambilan keputusan dalam proses panen buah sawit secara lebih efisien dan akurat.

Kata Kunci: *Buah Kelapa Sawit, CNN, InceptionV3, Google Colab, Kematangan Buah, Gradio.*

OPTIMIZATION OF CONVOLUTION NEURAL NETWORK (CNN) IN IDENTIFYING CPO FRUIT RITH

ABSTRACT

Palm oil is one of the leading agricultural commodities in Indonesia. The quality of crude palm oil (CPO) largely depends on the ripeness of the fruit when harvested. Improper harvesting caused by manual estimation often results in decreased product quality and economic losses. To address this issue, this research proposes the development of a classification system for oil palm fruit ripeness using the Convolutional Neural Network (CNN) method. This study uses a dataset consisting of 429 images categorized into three classes: Unripe, Ripe, and Overripe. The model is trained using the InceptionV3 architecture with a transfer learning approach, combined with preprocessing techniques such as image enhancement, normalization, and data augmentation to optimize learning performance. The model training and testing were conducted using Google Colaboratory, which allows seamless access to GPU resources. A user-friendly interface was also built using Gradio to facilitate predictions by simply uploading an image. The results show that the CNN model can achieve a validation accuracy of 87.65%, indicating that the model is capable of classifying oil palm fruit ripeness with a high level of accuracy. It is expected that this system can assist farmers or plantation managers in making more accurate and efficient harvesting decisions.

Keywords*: Oil Palm Fruit, CNN, InceptionV3, Ripeness Classification, Gradio, Image Processing.*

DAFTAR ISI

DAFTAR ISI ..... **xi**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 3
BAB II LANDASAN TEORI **4**
2.1 Buah Kelapa Sawit (CPO) 4
2.2 Pengolahan Citra Digital 5
2.3 Deep Learning 5
2.4 Convolutional Neural Network (CNN) 7
2.5 Google Colaboratory 10
2.6 TensorFlow 11
2.7 Penelitian Terdahulu 13
BAB III METODOLOGI PENELITIAN **14**
3.1 Jenis Penelitian 14
3.2 Dataset 14
3.3 Alur penelitian 16
BAB IV HASIL DAN PEMBAHASAN **18**
4.1 Pengumpulan Dataset 18
4.2 Implementasi CNN 19
4.3 Prediksi Kematangan Buah Sawit Menggunakan UI Gradio 31
BAB V PENUTUP **38**
5.1 Kesimpulan 38
5.2 Saran 39
DAFTAR PUSTAKA **40**

DAFTAR TABEL

Table 2.1 Perbedaan *Machine Learning* dan *Deep Learning* *6*
Table 2.2 Penelitian Terdahulu..... 13
Table 3.1 Kematangan Buah Sawit 14
Table 4.1 Pengimporan Library Phyton dan Inisialisasi Sesi TensorFlow Pada Google Colab 20
Table 4.2 Proses Mounting Google Drive dan Penentuan Path Direktori Dataset Buah Sawit..... 21
Table 4.3 Tahab Preprocessing Data 22
Table 4.4 Tahap Image Processing..... 23
Table 4.5 Proses Implementasi Transfer Learning Menggunakan InceptionV3..... 25
Table 4.6 Proses Augmentasi dan Normalisasi Data Gambar Menggunakan ImageDataGenerator..... 26
Table 4.7 Proses Pelatihan Model CNN Menggunakan Fungsi .fit() 28
Table 4.8 Proses Visualisasi Akurasi dan Loss Model..... 29
Table 4.9 Proses Prediksi Tingkat Kematangan Buah Sawit..... 31
Table 4.10 Proses Prediksi Data Uji..... 32
Table 4.11 Prediksi Tingkat Kematangan Buah Sawit Dengan UI Gradio 34

DAFTAR GAMBAR

Gambar 2.1 Buah sawit 4
Gambar 2.2 Struktur *Conbolution Neural Network* (CNN) 8
Gambar 2.3 Convolution Layer 9
Gambar 2.4 Max Pooling 10
Gambar 2.5 Tampilan awal Google Collab 10
Gambar 2.6 Tampilan TensorFlow 11
Gambar 4.1 Dataset Buah Sawit Belum Masak Yang Digunakan Untuk Train, Test, Validation Pada Sistem 18
Gambar 4.2 Dataset Buah Sawit Masak Yang Digunakan Untuk Train, Test, Validation Pada Sistem 18
Gambar 4.3 Dataset Buah Sawit Terlalu matang Yang Digunakan Untuk Train, Test, Validation Pada Sistem 19
Gambar 4.4 Hasil Dari Penerapan Image Processing 20
Gambar 4.5 Hasil Pelatihan Model CNN 28
Gambar 4.6 Hasil Visualisasi Akurasi dan Loss Model CNN 30
Gambar 4.7 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas “Masak” 35
Gambar 4.8 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas “Belum Masak” 36
Gambar 4.9 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas “Terlalu Masak” 37

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Kelapa sawit (Elaeis guineensis) adalah salah satu komoditas pertanian yang sangat penting di dunia, terutama di negara-negara tropis seperti Indonesia dan Malaysia. Minyak kelapa sawit (CPO) yang dihasilkan dari buah kelapa sawit memiliki berbagai aplikasi industry, mulai dari makanan hingga energi terbarukan (Alkusma et al., 2016). Kualitas CPO sangat dipengaruhi oleh tingkat kematangan buah saat dipanen, yang menentukan kandungan minyak dan asam lemak bebas (FFA) dalam CPO. Oleh karena itu, indentifikasi tingkat kematangan buah kelapa sawit secara akurat dan efisiensi sangat penting dalam industry kelapa sawit.

Secara tradisional, indentifikasi kematangan buah kelapa sawit dilakukan secara manual oleh petani, berdasarkan pengamatan visual terhadap warna, ukuran, dan tekstur buah. Dalam hal persepsi warna, ada pendekatan ilmiah untuk gambar digital yang melibatkan pemprosesan intensitas *collor*.

Pengolaha citra digital sudah digunakan untuk mempermudah kinerja kebanyakan orang dalam meringkas pekerjaan. Sangat mudah menemukannya dalam lingkup berdekatan dengan kehidupan kita. Seiring berkembangnya teknologi pengolahan citra mulai digunakan untuk mengindentifikasi dalam segala hal termasuk kematangan. Dalam tugas akhir penulis, indentifikasi tingkat kematangan buah kelapa sawit selalu dilakukan dengan kasat mata, lebih baik jika kita mengenali buah kelapa sawit secara digital dari gambar tanamannya sendiri. Untuk mendapatkan tingkat kematangan dan perkembangan ilmu yang digarap oleh peneliti.

Perkembangan ilmu pengetahuan dan teknologi pengolahan citra digital memungkinkan untuk memilah produk pertanian dan Perkebunan secara otomatis. Pengolahan citra merupakan alternatif untuk mengatasi masalah ini. Kedepannya,

pengolahan citra diharapkan akan menjadi salah satu pilihan dalam mengkasifikasi tingkang kematangan buah sawit tanpa merusak sempel (objek).

Ada banyaknya metode yang digunakan untuk melakukan pengolahan citra, salah satunya adalah menggunakan teknik Deep Learning (DL) yaitu metode Convolution Neural Network (CNN) yang saat ini memilki hasil paling signifikan dalam pengenalan citra pada visual cortex manusia sehingga memilki kemampuan mengolah informasi citra.

Penelitian ini bertujuan untuk mengoptimalkan algoritma CNN dalam mengindentifikasi tingkat kematangan buah kelapa sawit. Data yang digunakan dalam penelitian ini diperoleh dari KAGGLE, sebuah platform online yang menyediakan dataset untuk penelitian dan pengembangan kecerdasan buatan. Penggunaan dataset dari Kaggle memungkinkan penelitian ini dilakukan tanpa perlu melakukan observasi langsung ke Perusahaan atau lahan pertanian, sehingga menghemat waktu dan biaya.

1.2 Rumusan Masalah

Bagaimana algoritma Convulotion Neural Network dapat melakukan deteksi tingkat kematangan buah tomat dari citra gambar dan bisa membedakan mana buah kelapa sawit yang matang dan mana buah kelapa sawit yang belum matang.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini sebagai berikut:

1. Penelitian ini hanya mengindentifikasi setiap tingkat perkembangan buah kelapa sawit hanya dari warnanya.
2. Data buah kelapa sawit sebanyak 429 sempel terdiri dari 135 buah sawit mentah, 150 sawit matang, dan 144 buah terlalu matang
3. Bagaimana metode CNN (*Convulotion Neural Network)* bisa mengindentifikasi tingkat kematangan pada buah kelapa sawit.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengembangkan model CNN yang optimal untuk mengindentifikasi tingkat kematangan buah kelapa sawit.
2. Menganalisis kinerja model CNN yang dikembangkan menggunakan metrik evaluasi yang relevan.
3. Menganalisis tingkat akurasi algoritma CNN dalam mengindentifikasi kematangan buah sawit.

1.5 Manfaat Penelitian

Manfaat dalam penelitian ini adalah:

1. Menanbah pemahaman mahasiswa serta akademis mengenai penerapan algoritma CNN dalam mengindentifikasi kematangan buah sawit.
2. Meningkatkan efisiensi proses panen dengan pemanfaatan teknologi berbasis kecerdasan buatan.
3. Mengurangi kesalahan dalam mengenali kematangan buah, sehingga meningkatkan kualitas dan kuantitas panen.
4. Mengurangi potensi kerugian akibat kesalahan dalam penentuan waktu panen yang dapat menyebabkan rendahnya rendemen minyak.

BAB II
LANDASAN TEORI

2.1 Buah Kelapa Sawit (CPO)

CPO merupakan merupakan produk yang ditransaksikan di pasar komoditas atau hasil sumber daya alam yang biasa diolah menjadi berbagai produk turunan, baik itu berupa barang atau bahan baku industri (Al-Khowarizmi et al., 2021). Hasil perkebunan kelapa sawit berupa minyak sawit dan turunannya menjadi komoditas ekspor yang terus memberikan sumbangan yang sangat besar bagi perekonomian nasional. Selain penghasil minyak sawit terbesar, pada tahun 2018, Indonesia menjadi pengekspor minyak terbesar di dunia (Saragih & Rahayu, 2022).

Gambar 2.1 Buah sawit
Sumber: (*Mengenal-Kriteria-Matang-Buah-Kelapa-Sawit*, n.d.)

Keberhasilan budidaya kelapa sawit salah satunya ditentukan oleh pengelolaan atau manajemen panen yang baik dan benar. Ketepatan dalam menentukan kematangan buah sawit yang dipanen sangat diperlukan. Kelapa sawit menghasilkan CPO (*Crude Palm Oil)* yang merupakan minyak kelapa sawit yang dikomsumsi di seluruh dunia (Ermawati & Septia, 2013). Kualitas dan kuantitas CPO dipengaruhi oleh tingkat kematangan buah sawit. Kematangan buah sawit biasanya dapat dilihat dari jumlah buah yang gugur dari tandan buah. Buah dengan kondisi mentah adalah buah yang masih berwarna kuning muda bahkan ujung buah berwarna masih hitam dan tidak terdapat unsur merah atau warna cerah merah pada buah (Setyamidjaja, 2006).

2.2 Pengolahan Citra Digital

Pengolahan citra adalah suatu teknik yang digunakan untuk memproses citra atau gambar dengan cara memanipulasinya dengan data gambar yang di inginkan untuk mendapatkan informasi tertentu dari gambar yang diamati. Banyak data yang diperoleh apabila data yang diamati menampilkan suatu keadaan yang sesuai dengan data awal (Putri, 2016).

Pengolahan citra merupakan suatu metode atau teknik yang dapat digunakan untuk proses citra atau gambar dengan cara memanipulasinya menjadi data citra yang diinginkan untuk mendapatkan informasi tertentu. Sistem pengolahan citra memberikan kemudahan untuk proses citra (Marpaung et al., 2022).

Pengolahan citra bertujuan untuk lebih mengembangkan kualitas gambar dengan tujuan agar dapat diuraikan secara efektif oleh orang atau mesi (untuk hal ini komputer). Teknik-teknik pengolahan citra mengubah citra menjadi citra yang berbeda. Jadi, masukannya adalah citra dan hasilnya juga citra, namun citra yang dihasilkan memiliki kualitas yang lebih baik daripada citra masukan. Selain itu yang diingat untuk bidang ini adalah tekanan citra (image compression) (Sugiyono, 2022). Teknik pengolahan citra juga dapat dilakukan dengan mengindentifikasi tekstur pada suatu objek dengan metode statistic secara jelas.

2.3 Deep Learning

Deep learning adalah suatu cabang dari *machine learning* yang menggunakan algoritma *neural network* yang sangat kompleks dan terdiri dari banyak lapisan (*deep*). Tujuan dari *deep learning* adalah untuk mempelajari suatu representasi fitur dari data dengan menggunakan beberapa lapisan pemprosesan data secara bertahap. Dalam *deep learning*, setiap lapisan *neural network* terdiri dari sejumlah besar unit pengolahan sinyal (*neuron*) yang terhubung satu sama lain dalam pola tertentu (Suhardin et al., 2021).

Secara umum, deep learning digunakan untuk memecahkan masalah yang sangat kompleks dan memerlukan representasi data yang sangat abstrak, seperti pengenalan

gambar dan suara, penegenalan bahasa alami, dan tugas-tugas lain yang terkait dengan data kompleks. Misalnya, dalam pengenalan gambar, *deep learning* dapat digunakan untuk memperbaiki kinerja pengenalan gambar dengan cara memperkenalkan banyak lapisan pemprosesan gambar yang semakin kompleks dan mampu mengenali fitur yang semakin abstrak. Sebagai hasilnya, *deep learning* mampu memecahkan masalah yang sebelumnya dianggap tidak mungking di pecahkan oleh mesin (Jaya & Risma, 2022).

Deep learning juga memiliki beberapa keunggulan lain, seperti kemampuan untuk mengenali pola yang sangat kompleks dan abstrak, kemampuan untuk mengektraksi fitur yang relevan secara otomatis, dan kemampuan untuk memperbaiki pemforma dengan data yang lebih banyak. Namun, *deep learning* juga memiliki beberapa kelemahan, seperti kebutuhan akan data yang sangat besar dan kemampuan komputasi yang sangat tinggi (Roy, 2024).

Beberapa contoh aplikasi *deep learning* yang popular saat ini adalah pengenalan wajah seperti pada sistem keamanan atau pembuka kuci ponsel, pendeteksi objek seperti pada sistem keamanan atau kendaraan otonom, prediksi harga saham dengan menggunakan data pasar yang akan dikenali pola-pola harga sahamnya dan memberikan prediksi yang akurat, pemprosesan Bahasa alami seperti asisten virtual atau sistem terjemah otomatis yang dilatih dengan menggunakan data teks, pengolahan citra medis, pengenalan suara, dan pembelajaran penguasaan permainan seperti permainan *Go*. Aplikasi lain dari deep learning meliputi bidang Kesehatan, keuangan, dan teknologi informasi.

Perbedaan antara machine learning dan deep learning dapat dilihat pada table berikut ini:

Table 2.1 Perbedaan *Machine Learning* dan *Deep Learning*
Sumber : (Amazon, 2023)

Machine Learning	***Deep Learning***
Hasil bagus dengan dataset sedikit	Membutuhkan dataset yang banyak

Melatih model dengan cepat	Komputasi secara intensif
Perlu mencoba berbagai fitur dan klasifikasi manual untuk mendapatkan hasil terbaik	Mempelajari fitur dan pengklasifikasian otomatis
Tingkat akurasi tinggi	Akurasi tidak terbatas

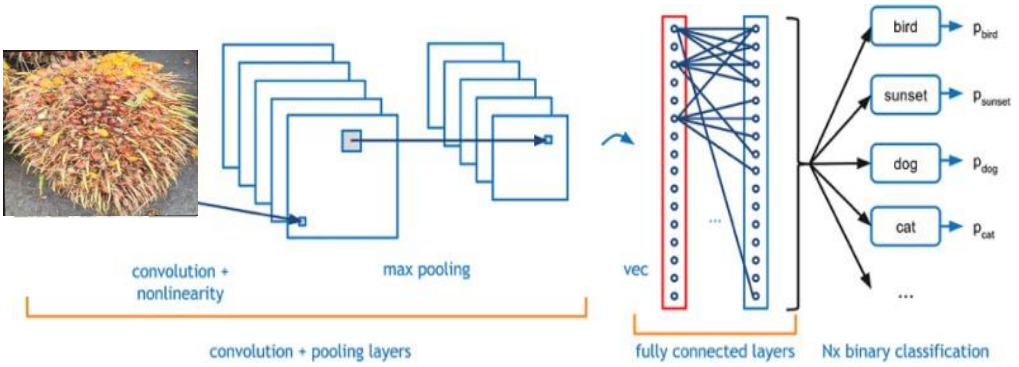
2.4 Convolutional Neural Network (CNN)

Convolution Neural Network (CNN) adalah tipe *neural network* yang khusus dirancang untuk memproses data berupa gambar dan citra. CNN juga merupakan salah satu jenis jaringan syaraf tiruan yang biasanya dimanfaatkan pada data gambar, yang bisa dimanfaatkan untuk mengindentifikasi serta mengkasifikasi objek pada sebuah gambar (Ilahiyah & Nilogiri, 2018) . CNN menggunakan operasi konvolusi untuk mengekstrak fitur atau pola dari gambar, dan lapisan-lapisan yang berbeda pada CNN dapat mempelajari fitur-fitur yang semakin kompleks pada gambar.

CNN terdiri dari lapisan *input*, lapisan konvolusi, lapisan pooling, dan lapisan *fully connected.* Lapisan konvolusi adalah lapisan yang paling penting pada CNN karena itulah tempat dimana proses pengenalan pola dan fitur terjadi. Sedangkan, lapisan pooling berfungsi untuk mengurangi dimensi dari data dan mengurangi pengaruh dari variabilitas lokasi pada gambar.

Beberapa contoh penggunaan CNN adalah dalam pengenalan objek, klasifikasi gambar, dan deteksi wajah. Dalam pengenalan objek, CNN dapat mengenali objek pada gambar dengan akurasi yang tinggi berdasarkan fitur-fitur yang dikenalinnya.

Convolution Neural Network (CNN) dalam *Deep Learning* tergolong kedalam *deep neural network* yang banyak digunakan dalam analisis citra (*visual imagery*). CNN merupakan versi *regularization* dari *multi-player perceptron* dan dibangun oleh banyak *convolution layer* dengan fungsi aktivasi yang kebanyakan adalah *ReLU*.

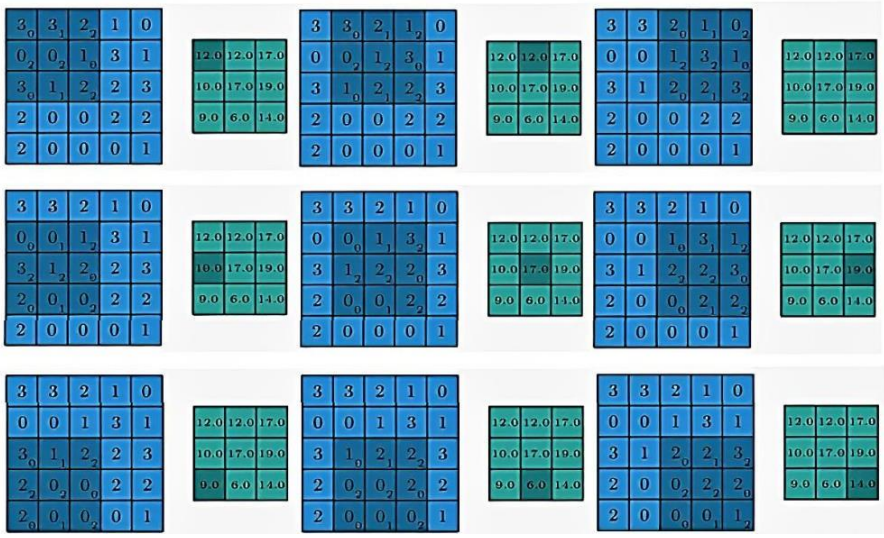


Gambar 2.2 Struktur *Conbolution Neural Network* (CNN)
Sumber: (*Penerapan-Convolutional-Neural-Network-Cnn-Terhadap-Jenis-Unggas-Ordo-Ansariformes-Menggunakan-5d0b408da20e*, n.d.)

Convolution Neural Network (CNN) merupakan jaringan saraf yang digunakan khusus untuk mengolah data berstruktur grid, salah satunya berupa citra dua dimensi. Proses konvolusi merupakan operasi aljabar linear yang menkalikan matriks dari filter pada citra yang akan diproses (Triwijoyo, 2019). Proses tersebut dinamakan dengan lapisan konvolution dan merupakan salah satu jenis dari banyak lapisan yang bisa dimiliki dalam satu jaringan.

2.1.1. Convolution Layer

Convolution Layer pada CNN bekerja dengan melakukan operasi konvolusi pada gambar input atau *feature map* dengan *Set Filter* atau *kernel*. *Filter* atau *Kernel* adalah matriks kecil yang digunakan untuk mengekstrak fitur-fitur penting dari gambar, seperti garis, sudut dan pola-pola kompleks lainnya. Proses konvolusi menghasilkan output berupa *feature map* yang merupakan representasi dari gambar input yang telah dilakukan ekstraksi fitur.

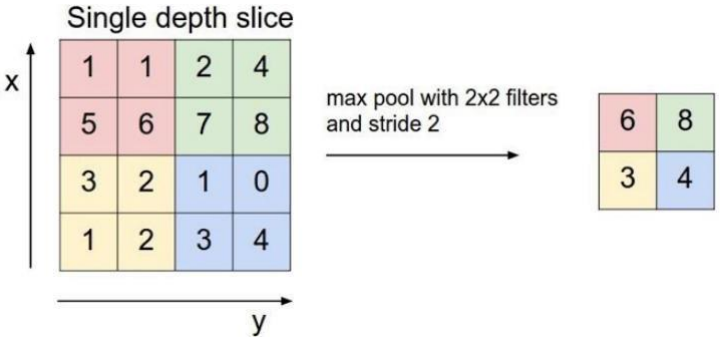


Gambar 2.3 Convolution Layer

Pada gamber 2.3, filter bergerak diatas gambar input dan melakukan operasi konvolusi untuk menghasilkan *feature map*. Contoh filter yang digunakan pada gambar adalah filter *edge detection* (menemukan tepi gambar). Setelah itu, *feature map* dihasilkan dan selanjutnya dapat digunakan sebagai *input* untuk *layer* berikutnya pada CNN.

2.1.2. Pooling Layer

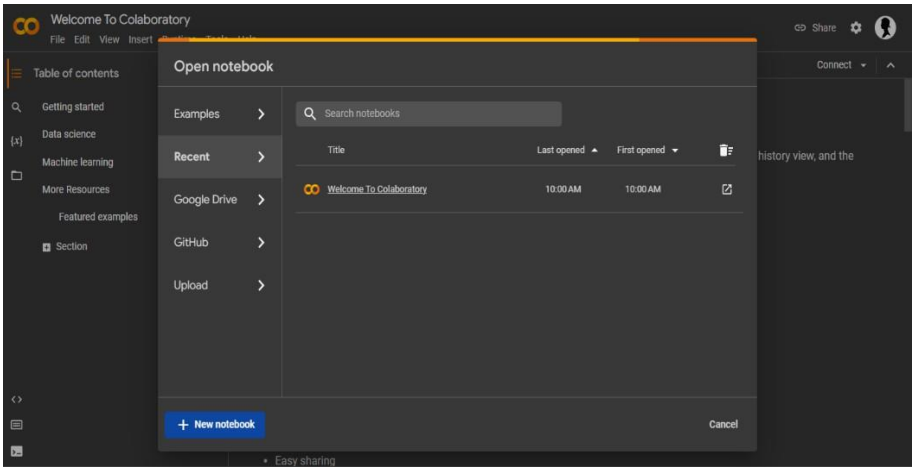
Pooling layer ada setelah convolution layer. Terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh *activation map*. Pooling yang sering digunakan adalah max pooling dan *average pooling*. Contoh penggunaanya apabila *max pooling* yang digunakan 2 x 2 dengan stride 2, maka setiap pergeseran filter, nilai maksimum pada area 2 x 2 pixel tersebut yang akan dipilih, sedangkan *average pooling* akan memilih nilai rata-ratanya. *Pooling layer* digunakan untuk mempercepat komputasi karena parameter yang harus di update semakin sedikit dan mengatasi overfitting (Kusuma Putra et al., 2020).



Gambar 2.4 Max Pooling
Sumber: (*Apa-Itu-Convolutional-Neural-Network-836f70b193a4*, n.d.)

2.5 Google Colaboratory

Google Colaboratory atau *Goole colab* adalah sebuah platform gratis berbasis *cloud* yang menyediakan lingkungan pengembangan *Jupyter Notebook (Python). Collab* memungkinkan pengguna untuk melatih dan menguji model *Machine Learning* di CPU, GPU, dan TPU tanpa memerlukan konfigurasi apapun. Ada 3 keunggulan utama *collab* yaitu tidak memerlukan konfigurasi, akses gratis ke GPU, dan mudah dibagikan dengan rekan penelitian (Shohibuddin et al., n.d.). *Collab* hanya memerlukan akun *Google* dan *Internet Browser*, serta menyediakan waktu penggunaan *notebook* maksimal 12 jam/hari dengan RAM 12GB dan disk 100GB untuk opsi gratis/tanpa berbayar. Adapun tampilan dari *Google Collab* dapat dilihat sebagai berikut:



Gambar 2.5 Tampilan awal Google Collab

Gambar 2.4 menunjukkan bahwa tampilan *Google Collab* mirip dengan *Jupyter Notebook*. Salah satu keuntungan pengguna *Google Collab* adalah tidak perlu melakukan konfigurasi karena menggunakan teknologi *cloud computing*, memberikan akses gratis untuk menggunakan mesin dengan kecepatan tinggi seperti GPU, dan mudah terhubung dengan *Google Drive*. Dalam penelitian ini, Google Collab digunakan untuk melatih dan menguji model *Convolution Neural Network* dalam mengindentifikasi kematangan buah sawit.

2.6 TensorFlow

Tensorflow merupakan open source framework yang dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek. Sistem ini sudah banyak diterapkan pada berbagai produk *Google* antara lain pencarian *image*, deteksi wajah, dan plat nomor kendaraan pada *Google street, view, google assistant, way mo* atau *self draving car*, dan lain-lain.

Tensorflow bekerja dengan *computational* untuk membuat model *machine learning*. *Tensorflow* menyediakan berbagai *toolkit* yang memungkinkan membuat model pada tingkat yang lebih rendah untuk membuat model dengan menentukan serangkaian matematis. Sebagai alternatif, dapat menggunakan API dengan tingkat yang lebih rendah untuk membuat model dengan menentukan API dengan tingkat yang lebih tinggi (seperti tf.estimator) untuk menentukan arsitektur yang telah ditetapkan, seperti regresi linear atau *neural network*.

Gambar 2.6 Tampilan TensorFlow
Sumber: (*TensorFlow*, n.d.)

Framework Tensorflow digunakan pada proses pembuatan system deteksi objek agar memudahkan implementasi algoritma dan penggunaan bahasa pemprograman, terdapat juga GPU untuk mempercepat proses *training*. *Tensorflow* sebagai kerangka machine learning yang dapat digunakan untuk mengolah banyak data atau ingin mempelajari kecerdasan buatan (artificial intelligence) secara mendalam.

Tensorflow memungkinkan pengguna untuk membangu model *machine learning* dengan mudah dan cepat dengan menggunakan API yang intuitif. Pengguna dapat memilih antara menggunakan API *high-level* seperti *Keras* atau API *low-level* yang lebih fleksibel untuk membangun model sesuai dengan kebutuhan. Selain itu, *tensorflow* juga menyediakan *tools* untuk memvisualisasikan di multiple GPUs atau mesin (Jaya & Risma, 2022)

Berikut adalah kelebihan dari *tensorflow*:

1. Kompatibilitas *platform* yang luas : *tensorflow* dapat digunakan pada berbagai platform seperti *desktop, server, mobile*, dan *edge devices*.membuat tensorflow sangat fleksibel dan dapat diintegrasikan ke dalam berbagai aplikasi.
2. Kecepatan dan sklabilitas : *tensorflow* dirancang untuk dapat memproses data yang sangat besar dan kompleks dengan kecepatan yang tinggi. Menjadikan *tensorflow* sebagai salah satu *framework* tercepat yang tersedia saat ini.
3. Dukungan komunitas yang luas : *tensorflow* memiliki komunitas yang besar dan aktif. Hal ini berarti bahwa ada banyak sumber daya dan dukungan yang tersedia saat ini.
4. Memiliki banyak fitur : tensorflow menyediakan berbagai jenis layer dan model yang dapat digunakan untuk membangun berbagai jenis model machine learning, seperti model deep learning dan reinforcement learning. Tensorflow menyediakan berbagai algoritma optimasi yang dapat digunakan untuk meningkatkan performa model (Maulana et al., 2024).

2.7 Penelitian Terdahulu

Penelitian ini didukung oleh beberapa referensi yang Dimana dapat memperkuat penelitian dalam Menyusun skripsi ini. Beberapa referensi mengenai system dan akan menjadi perbandingan dengan penelitian yang akan dijalankan, sebegai berikut:

Table 2.2 Penelitian Terdahulu

No.	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1.	(Maya et al., 2024)	Klasifikasi Tingkat Kematangan Buah Tomat Menggunakan CNN	Hasil penelitian ini dapat disimpulkan bahwa algoritma CNN berhasil mengkasifikasikan kematangan tomat dengan akurasi sebesar 89%
2.	(Dayera, Musa Bundaris Palungan, 2024)	Indentifikasi Kematangan Buah Pisang Menggunakan Model CNN	Hasil penelitian yang didapatkan ialah dengan menggunakan 2 sample buah, model memperoleh akurasi sebesar 91% setelah augmentasi citra
3.	("Klasifikasi Tingkat Kematangan Buah Sawit Berdasarkan Ekstraksi Fitur RGB Dan GLCM Menggunakan Algoritma K-NN," 2023)	Klasifikasi Tingkat Kematangan Buah Sawit Dengan Ekstraksi Warna RGB Dan GLCM Menggunakan K-Nearest Neighbor (KNN)	Akurasi klasifikasi mencapai 78%, namun pemforma menurun pada pencahayan rendah.
4.	(Nandaputra et al., 2024)	Klasifikasi Buah Tropis Menggunaka Transfer Learning InceptionV3 (CNN Pretrained)	Hasil penelitian yang didapatkan ialah, transfer learning berhasil meningkatkan akurasi hingga 98%.
5.	(Arkadia et al., 2021)	Deteksi Kematangan Buah Mangga Berbasis CNN Dengan 3 Conv Layer	Model mendeteksi kematangan dengan akurasu 87% dan visualisasi featured map.

BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang dilakukan adalah penelitian kuantitatif. Penelitian ilmiah yang dilakukan secara sistematis dengan menggunakan model-model matematis dan teori-teori yang berkaitan. Penelitian yang dilakukan menggunakan salah satu teknik pada *Deep Learning* yaitu algoritma Convolution Neural Network (CNN)

3.2 Dataset

Dataset yang diguanakan pada penelitian ini adalah data kematangan buah sawit. Data diambil dalam bentuk image dengan format file *JPG. Pengambilan data dilakukan pada tanggal 17 februari 2025. Jumlah keseluruhan data 429 dataset yang terdiri dari 135 sawit mentah, 150 sawit matang, dan 144 sawit terlalu matang.

Teknik pengumpulan data dalam penelitian ini:

1. Pengumpulan dataset dilakukan secara online melalui website KAGGLE.
2. Data berasal dari ripeness oil palm fruit/kematangan buah kelapa sawit.

Table 3.1 Kematangan Buah Sawit

No.	Buah Sawit (Gambar)	Label
1		Mentah
2		Mentah

3		Mentah
...	**...**	**...**
136		Matang
137		Matang
138		Matang
...	**.....**	**.....**
286		Terlalu Matang
287		Terlalu Matang

288		Terlalu Matang
....	.....	.....

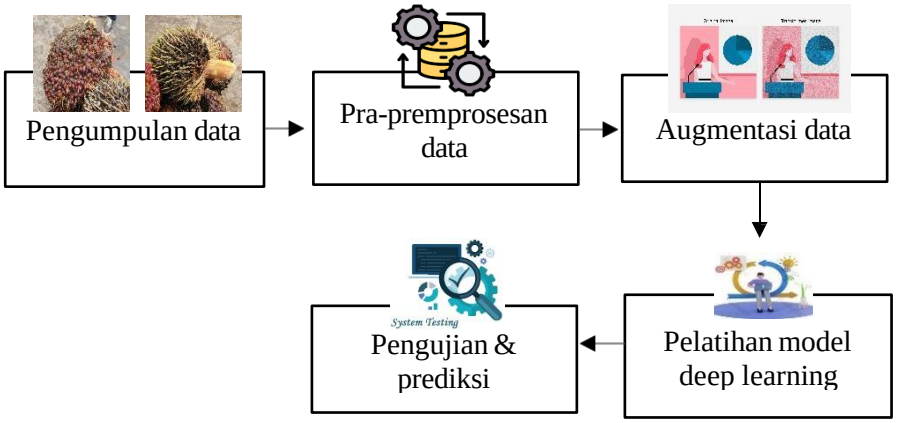
Dataset yang digunakan dalam penelitian ini merupakan kumpulan citra buah kelapa sawit yang telah dikategorikan berdasarkan tingkat kematangannya kedalam tiga kelas, yaitu belum masak, masak, dan terlalu masak. Dataset ini bersumber dari file gambar yang telah dikumpulkan dan disimpan dalam folder direktori Google Drive, kemudian diakses melalui Google Colab untuk proses pengolahan dan pelatihan model.

Seluruh gambar telah dikelompokkan secara manual ke dalam folder sesuai dengan kategori kematangannya. Selanjutnya, dataset dibagi menjadi tiga bagian utama, yaitu data pelatihan (training), data validasi (validation), dan data pengujian (testing). Pembagian ini dilakukan untuk memastikan model dapat dilatih, divalidasi, dan diuji dengan data yang berbeda agar hasil pelatihannya lebih objektif dan tidak overfitting.

Jumlah data dalam masing-masing kelas relative seimbang, dengan total ratusan gambar pada dataset memiliki variasi dalam hal pencahayaan, sudut pengambilan, serta latar belakang, sehingga sangat mendukung proses pelatihan model agar mampu menegenali fitur kematangan buah sawit secara lebih umum.

3.3 Alur penelitian

Dalam pelaksanaan penelitian ini, ada serangkaian langkah yang akan diambil untuk mencapai Kesimpulan dari penelitian ini. Rincian mengenai langkah-langkah penelitian dapat ditemukan dalam diagram berikut:



Alur penelitian ini menggambarkan tahapan-tahapan yang dilakukan dalam proses klasifikasi tingkat kematangan buah sawit menggunakan algoritma Convolutional Neural Network (CNN). Berikut adalah penjelasannya :

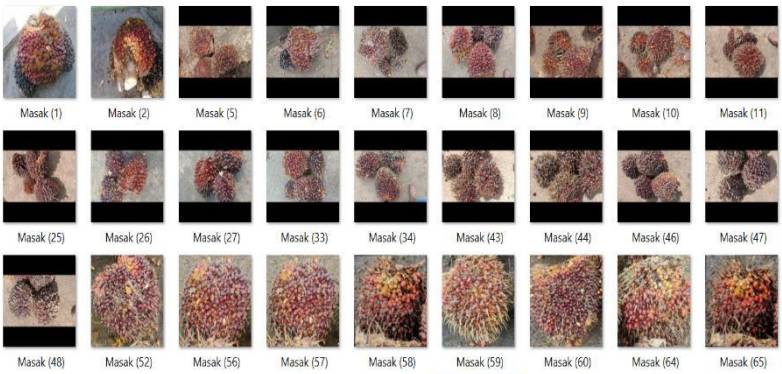
1. Pengumpulan data, mengumpulkan dataset berupa gamnbar buah sawit dengan berbagai tingkat kematangan. Gambar diklasifikasikan ke dalam tiga kategori, yaitu belum masak, masak, dan terlalu masak.
2. Pra-pemprosesan data, tahap selanjutnya melakukan pra-pemprosesan. Gambar diubah ukurannya menjadi 244 x 244 piksel, kemudian dilakukan normalisasi nilai piksel dari 0-255 menjadi 0-1.
3. Augmentasi data, teknik augmentasi pada penelitian ini meliputi rotasi, zoom, flip horizontal, dan pergeseran gambar.
4. Pelatihan model deep learning, gambar yang telah diproses kemudian digunakan untuk melatih model deep learning berbasis CNN. Dalam penelitian ini menggunakan arsitektur InceptionV3
5. Pengujian model dan prediksi, setelah model dilatih, dilakukan pengujian menggunakan data testing. Model mengklasifikasi gambar uji dan hasil prediksinya dibandingkan dengan label asli untuk mengukur akurasi. Ditahap ini juga dilakukan prediksi citra menggunakan UI Gradio

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Dataset

Dataset dalam setiap penelitian menjadi faktor utama. Dataset bisa berupa gambar, video, maupun angka dalam sebuah penelitian. Dataset yang digunakan dalam penelitian indentifikasi kematangan buah sawit bersumber dari Kaggle yang berjudul "Ripeness oil palm fruit". Jumlah keseluruhan dataset dari Kaggle sebanyak 429 data dengan data berupa gambar. Data gambar pada dataset ini berisi gambar buah sawit yang belum masak, masak, dan terlalu masak. Jumlah data gambar tersebut termasuk dalam data yang relatif kecil. Klasifikasi data terbagi menjadi tiga yakni buah sawit belum masak, masak, dan buah sawit yang terlalu masak.

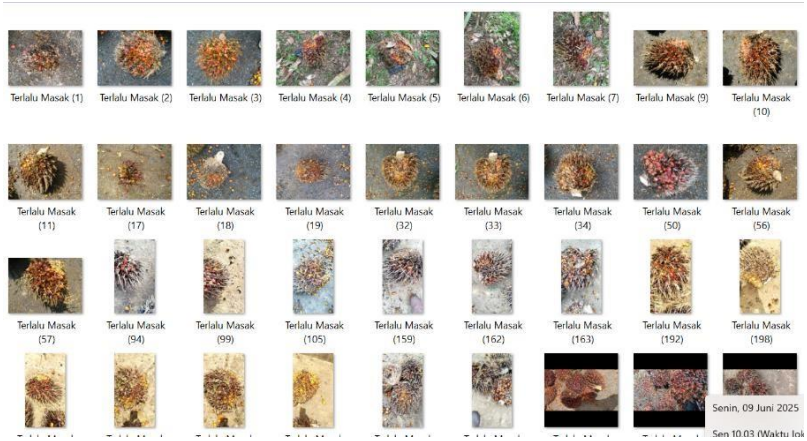
Gambar 4.1 Dataset Buah Sawit Belum Masak Yang Digunakan Untuk Train, Test, Validation Pada Sistem.



Gambar 4.2 Dataset Buah Sawit Masak Yang Digunakan Untuk Train, Test, Validation Pada Sistem.

Dataset yang digunakan pada penelitian ini tidak hanya yang belum masak dan masak saja tetapi yang terlalu masak akan digunakan. Perlu adanya 3 kelas dalam sebuah

sistem untuk menghasilkan tingkat perbandingan dan juga akurasi yang tinggi dan relevan.



Gambar 4.3 Dataset Buah Sawit Terlalu matang Yang Digunakan Untuk Train, Test, Validation Pada Sistem

4.2 Implementasi CNN

Tahap implementasi ini menjelaskan mekanisme Convolutional Neural Network (CNN) menggunkan Google Colab sebagai code editor. Tahap awal dalam proses implentasi model CNN ini dimulai dengan proses pengimporan berbagai pustakan (library) dan modul penting yang digunakan dalam pengolahan data citra serta pembangunakan arsitektur CNN. Proses ini dilakukan pada lingkungan Google Colab sebagai code editor berbasis cloud, yang mendukung eksekusi komputasi.

Dalam implementasi ini digunakan pustaka seperti Numpy, pandas, dan Matplotlib untuk manipulasi data dan visualisasi, skimage serta PIL untuk pemprosesan citra, serta TensorFlow dan Keras sebagai pustaka utama dalam membangun model deep learning. Selain itu, modul sklearn.metrics digunakan untuk evaluasi pemforma model melalui confusion matrix.

Table 4.1 Pengimporan Library Phyton dan Inisialisasi Sesi TensorFlow Pada Google Colab

Import Library Phyton dan Inisialisasi Sesi TensorFlow Pada Google Colab

```
import numpy as np
import pandas as pd
import seaborn as sns
from PIL import Image
from PIL import ImageEnhance
from skimage.io import imread
import matplotlib.pyplot as plt
import os, random, pathlib, warnings, itertools, math
import tensorflow as tf
import tensorflow.keras.backend as K
from sklearn.metrics import confusion_matrix
from tensorflow.keras import models
from tensorflow.keras.models import Model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import InceptionV3,
preprocess_input
from tensorflow.keras.layers import Dense, Flatten,
GlobalAveragePooling2D, Dropout

warnings.filterwarnings("ignore")
K.clear_session()
```

Langkah inisialisasi ini penting karena Pustaka-pustaka tersebut menyediakan berbagai fungsi penting dalam pengolahan dataset, pemodelan arsitektur CNN, serta evaluasi pemforma sistem. Khususnya penggunaan InceptionV3, yang merupakan arsitektur CNN pretrained, memungkinkan pemanfaatan transfer learning untuk meningkatkan akurasi citra buah sawit berdasarkan tingkat kematangan.

Selain pengimporan library, fungsi K.clear_session() digunakan untuk membersihkan sesi Tensorflow sebelumnya agar tidak terjadi konflik saat proses pelatihan model berlangsung Kembali. Sangat penting untuk menjaga kestabilan saat proses pelatihan ulang dilakukan.

Table 4.2 Proses Mounting Google Drive dan Penentuan Path Direktori Dataset Buah Sawit

Upload dataset ke dalam google colab dan membuat direktori

`from google.colab import drive`
`drive.mount('/content/drive')`
 	
`import os`
`dataset = '/content/drive/MyDrive/dataset_buahsawit1'`
`train_folder = os.path.join(dataset, "train")`
`test_folder = os.path.join(dataset, "validation")`
`validation_folder = os.path.join(dataset, "test")`

Table 4.2 menunjukkan proses mengunggah dataset ke Google Colab dan menentukan struktur direktori yang akan digunakan untuk proses pelatihan, validasi, dan pengujian model. Dataset yang digunakan telah disimpan di Google drive dan diakses melalui integrasi dengan Google Colab. Untuk mengakses file dari Google Drive , digunakan fungsi drive.mount() dari library Google.colab. dengan ini, penyimpanan Google Drive dapat dihubungkan ke Colab, sehingga file seperti dataset citra dapat digunakan langsung dalam proses pelatihan model.

Setelah berhasil menghubungkan Google Drive, Langkah selanjutnya adalah menentukan path atau direktori dari dataset yang telah diunggah. Dataset telah diatur kedalam tiga folder utama yaitu train, validation, dan test. Penentuan direktori ini sangat penting untuk mengarahkan model ke Lokasi dataset yang sesuai. Struktur direktori yang rapu akan mempermudah proses pemprosesan citra, terutama saat menggunakan metode image data generator atau data augmentation. Dengan demikian, sistem dapat secara otomatis memuat gambar dari setiap kategori yang telah dipisahkan dalam masing-masing folder, sesuai dengan tujuan pelatihan, validasi, dan pengujian.

Table 4.3 Tahab Preprocessing Data

Preprocessing Data

```
import os
import pathlib

def count_files(rootdir):
  '''Counts the number of files each subfolder in a directory'''

  #pastikan rootdir adalah path yang benar dan ada
  if not os.path.exists(rootdir):
    print(f"Error: Directory {rootdir} does not exist.")
    return

  #iterasi melalui subfolder dalam roodir
  for path in pathlib.Path(rootdir).iterdir():
    if path.is_dir():
      num_files = len([name for name in os.listdir(path)
      if os.path.isfile(os.path.join(path, name))])
      print(f"there are {num_files} files in {path.name}")

#mendefinisikan path dataset
dataset = '/content/drive/MyDrive/dataset_buahsawit1'

#membuat path untuk folder train, validation, dan test
train_folder = os.path.join(dataset, "train")
validation_folder = os.path.join(dataset, "validation")
test_folder = os.path.join(dataset, "test")

#memanggil fungsi untuk menghitung jumlah file di setiap folder
count_files(train_folder)
```

Pada table 4.3 merupakan tahap preprocessing data yang bertujuan untuk menyiapkan data agar dapat digunakan dalam proses pelatihan model CNN. Proses preprocessing dilakukan pada Google colab, memanfaatkan Pustaka os dan pathlib dalam Bahasa pempgraman phyton untuk membaca struktur direktori dataset yang telah dikelompokkan berdasarkan kelas kematangan buah sawit.

Dataset diklasifikasikan menjadi tiga kelas, yaitu belum masak, masak, dan terlalu masak, yang masing masing disimpan dalam subdirektori train, validation, dan test. Untuk memastikan bahwa jumlah data pada masing masing kelas merata dan valid, digunakan fungsi phyton untuk menghitung jumlah file gambar pada setiap subfoldel

dataset. Ini penting agar model yang akan dibangun tidak mengalami bias terhadap salah satu kelas tertentu.

Hasil eksekusi kode phyton menunjukkan jumlah file gambar di dalam folder train. Terlihat bahwa kelas belum masak memiliki 135 file, kelas masak memiliki 150 file, dan kelas terlalu masak memiliki 144 file. Ini menunjukkan bahwa distribusi data pada tahap pelatihan cukup seimbang, yang akan penting untuk menghindari ketimpangan dalam proses pelatihan model CNN.

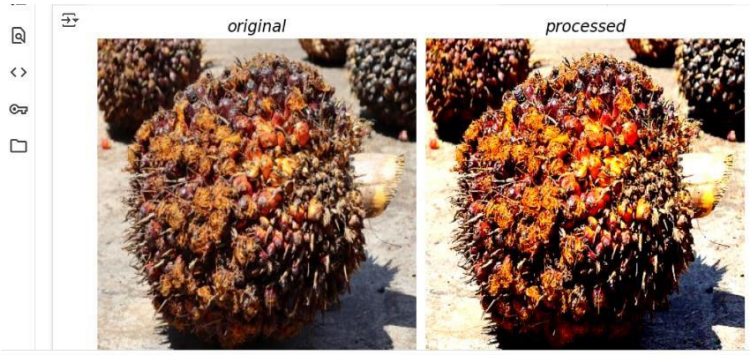
Table 4.4 Tahap Image Processing

Image Processing

```
import os
import random
from PIL import Image, ImageEnhance
import matplotlib.pyplot as plt

image_folder = "Masak"
number_of_image = 2 # jumlah gambar yang ingin ditampilkan

def preprocess():
j = 1 #untuk pengaturan suplot

for i in range(number_of_image):
folder = os.path.join(test_folder, image_folder)
a = random.choice(os.listdir(folder))

opened_image = Image.open(os.path.join(folder, a))
image_duplicate = opened_image.copy()

plt.figure(figsize=(10, 10))
plt.subplot(1, 2, 1)
plt.title(label='original', size=17, pad=7.0, loc="center",
fontstyle='italic')
plt.imshow(opened_image)
plt.axis('off')

#proses peningkatan gambar
image1 = ImageEnhance.Color(image_duplicate).enhance(1.35)
image1 = ImageEnhance.Contrast(image1).enhance(1.45)
image1 = ImageEnhance.Sharpness(image1).enhance(2.5)

plt.subplot(1, 2, 2)
plt.title(label='processed', size=17, pad=7.0, loc="center",
```

```
fontstyle='italic')
plt.imshow(image1)
plt.axis('off')

plt.tight_layout()
plt.show()
preprocess()
```

Table 4.4 merupakan tahap image preprocessing yang menjelaskan bagaimana proses peningkatan kualitas citra dilakukan sebelum digunakan dalam pelatihan model CNN. Tujuan proses ini untuk meningkatkan kualitas visual dari gambar buah sawit agar fitur penting dapat dikenali dengan lebih baik oleh model. Gambar yang diambil dari folder dataset akan ditampilkan dalam dua kondisi, yaitu kondisi asli (original) dan hasil pemprosesan (processed). Peningkatan kualitas gambar dilakukan melalui tiga tahapan utama yaitu peningkatan warna (color enhancement), peningkatan kontras (contrast enhancement), dan peningkatan ketajaman (sharpness enhancement).



Gambar 4.4 Hasil Dari Penerapan Image Processing

Pada gambar 4.4 menunjukkan hasil dari penerapan image processing pada dua buah citra sawit dari kelas “Masak”. Gambar sebelah kiri merupakan kondisi asli, sedangkan gambar sebelah kanan adalah hasil setelah dilakukan peningkatan warna sebesar 1.35, kontras sebesar 1.45, dan ketajaman sebesar 2.5. proses ini bertujuan untuk memperjelas fitur visual seperti tekstur dan warna buah sawit yang nantinya akan digunakan oleh model Convulutional Neural Network dalam proses pelatihan.

Table 4.5 Proses Implementasi Transfer Learning Menggunakan InceptionV3

Implementasi Transfer Learning Menggunakan InceptionV3

```
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense,
Dropout
from tensorflow.keras.optimizers import Adam

IMAGE_SIZE = (224, 224, 3)

# load base model inceptionV3 tampa bagian top (fully connected layer)
inception = InceptionV3(input_shape=IMAGE_SIZE, weights='imagenet',
include_top=False)

# membekukan semua layer inceptionV3 agar tidak dilatih ulang
for layer in inception.layers:
    layer.trainable = False

# menambahkan custom top layer
x = inception.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x) #perbaikan: aktivasi harus dengan
tanda '='
x = Dropout(0.2)(x)
prediction = Dense(3, activation='softmax')(x)

# membangun model akhir
model = Model(inputs=inception.input, outputs=prediction)

# menyusun model
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(), # atau bisa cukup 'adam'
    metrics=['accuracy']
)
```

Pada table 4.5 ini memperlihatkan proses Pembangunan model klasifikasi menggunakan metode transfer learning dengan arsitektur inceptionV3. InceptionV3 merupaka salah satu model deep learning yang telah dilatih sebelumnya menggunakan dataset ImageNet, sehingga memilki kemampuan dalam mengenali berbagai pola visual. Dengan memanfaatkan model ini, proses pelatihan dapat lebih efisien karena tidak perlu melatih selurug jaringan dari awal.

Tahap implementasi dimulai dengan melakukan import terhadap library TensorFlow dan Keras, yang memungkinkan penggunaan arsitektur InceptionV3 dan berbagai layer tambahan. Kemudian, model InceptionV3 di-load tanpa bagian atasnya (include_top=false) agar dapat disesuaikan dengan dataset yang digunakan, yaitu dataset klasifikasi kematangan buah sawit. Setelah itu, ditambahkan beberapa layer baru seperti GlobalAveragePooling2D, Dense, Dropout, dan Dense dengan aktivasi softmax yang disesuaikan untuk klasifikasi ke dalam tiga kelas, yaitu Belum Masak, Masak, dan Terlalu Masak. Langkah akhir adalah menyusun dan menyusun model akhir menggunakan metode compile, dengan loss function categorical_crossentropy, optimizer Adam, serta metrik accuracy. Kombinasi ini sangat umum digunakan dalam klasifikasi multi-kelas dan memungkinkan evaluasi performa model yang efisien.

Table 4.6 Proses Augmentasi dan Normalisasi Data Gambar Menggunakan ImageDataGenerator

Augmentasi dan Normalisasi Data Gambar

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# Augmentasi data untuk training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
)
# normalisasi data untuk testing
test_datagen = ImageDataGenerator(rescale=1./255)
# membaca gambar dari folder training
training_set = train_datagen.flow_from_directory(
    train_folder,
    target_size=(224, 224),
    batch_size=64,
    class_mode='categorical'
)
# membaca gambar dari folder testing
test_set = test_datagen.flow_from_directory(
    test_folder,
    target_size=(224, 224),
    batch_size=64,
    class_mode='categorical'
)
```

Pada table 4.6 menampilkan proses awal dari pelatihan model CNN, yaitu tahap data preprocessing menggunakan ImageDataGenerator dari library tensorflow.keras.preprocessing.image. Tahap ini sangat penting untuk mempersiapkan data gambar agar model dapat belajar secara optimal, terutama dalam mencegah overfitting. Untuk data pelatihan (tainning), diterapkan beberapa Teknik augmentasi seperti shear_range, zoom_range, dan horizontal_flip. Teknik augmentasi ini berfungsi untuk memperkaya variasi gambar yang digunakan saat pelatihan, meskipun jumlah gambar terbatas. Dengan menambahkan variasi ini, model akan lebih Tangguh dan mampu melakukan generalisasi yang lebih baik terhadap data baru.

Sementara itu, untuk data pengujian (testing), hanya dilakukan proses normalisasi menggunakan rescale=1./255. Normalisasi ini mengubah nilai piksel dari 0-255 menjadi 0-1, agar lebih sesuai dengan format input jaringan saraf tiruan dan mempercepat proses pelatihan. Selanjutnya, gambar dibaca dari folder menggunakan fungsi flow_from_directory() baik untuk training maupun testing. Pada proses ini, ditentukan parameter target_size sebesar (244, 244), bath_size sebanyak 64, dan class_mode=’categorical’ karena data diklasifikasikan ke dalam tiga kelas.

Berdasarkan hasil output yang ditampilkan, sistem mendeteksi 429 gambar yang terbagi ke dalam 3 kelas, baik pada data pelatihan maupun data pengujian. Hal ini menunjukkan bahwa struktur folder dataset sudah dikenali dengan baik dan siap digunakan dalam proses pelatihan model.

Table 4.7 Proses Pelatihan Model CNN Menggunakan Fungsi .fit()

Pelatihan Model CNN

`r = model.fit(`
`  x=training_set,`
`  steps_per_epoch=len(training_set),`
`  validation_data=test_set,`
`  validation_steps=len(test_set),`
`  epochs=10,`
`  verbose=1`
`)`

Table 4.7 merupakan proses pelatihan moden Convulutional Neural Network (CNN) menggunakan fungsi model.fit() dari Pustaka Keras pada TensorFlow. Tahapan ini merupakan bagian inti dari implementasi CNN, Dimana model dilatih menggunakan dataset yang telah dipersiapkan sebelumnya. Kode dalam gambar tersebut menyertakan beberapa parameter penting, seperti x=training_set yang menunjukkan data pelatihan, dan validation_data=test_set untuk proses validasi model secara bersamaan. Selain itu, parameter steps_per_epoch=len(training_set) dan validation_steps=len(test_set) digunakan untuk menentukan jumlah batch yang akan dieksekusi dalam setiap epoch dan selama validasi. Pelatihan dilakukan selama 10 epoch seperti yang ditentukan oleh epochs=10, sedangkan verbose=1 berfungsi untuk menampilkan proses pelatihan secara rinci di layer, termasuk metrik akurasi dan nilai loss pada setiap epoch.

```
Epoch 1/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 198s 29s/step - accuracy: 0.4058 - loss: 1.6002 - val_accuracy: 0.6107 - val_loss: 0.7875
Epoch 2/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 120s 18s/step - accuracy: 0.5883 - loss: 0.8082 - val_accuracy: 0.6923 - val_loss: 0.6294
Epoch 3/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 137s 17s/step - accuracy: 0.7095 - loss: 0.5914 - val_accuracy: 0.7552 - val_loss: 0.5513
Epoch 4/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 142s 17s/step - accuracy: 0.7969 - loss: 0.4954 - val_accuracy: 0.8159 - val_loss: 0.4436
Epoch 5/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 117s 18s/step - accuracy: 0.7697 - loss: 0.4979 - val_accuracy: 0.8275 - val_loss: 0.4107
Epoch 6/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 166s 22s/step - accuracy: 0.8029 - loss: 0.4678 - val_accuracy: 0.8392 - val_loss: 0.3817
Epoch 7/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 123s 17s/step - accuracy: 0.8299 - loss: 0.4214 - val_accuracy: 0.8438 - val_loss: 0.3842
Epoch 8/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 117s 17s/step - accuracy: 0.7943 - loss: 0.4576 - val_accuracy: 0.8601 - val_loss: 0.3378
Epoch 9/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 139s 17s/step - accuracy: 0.8405 - loss: 0.3647 - val_accuracy: 0.8741 - val_loss: 0.3237
Epoch 10/10
7/7 ━━━━━━━━━━━━━━━━━━━━ 171s 22s/step - accuracy: 0.8555 - loss: 0.3545 - val_accuracy: 0.8765 - val_loss: 0.3017
```

Gambar 4.5 Hasil Pelatihan Model CNN

Pada gambar diatas merupakan hasil pelatihan model CNN selama 10 epoch. Proses pelatihan ini mencakup metrik penting yaitu accuracy, loss, validation accuracy,

dan validation loss, yang diukur pada setiap epoch untuk mengevaluasi pemforma model dalam mengenali tingkat kematangan buah sawit.

Pada epoch pertama, nilai akurasi model masih rendah, yaitu 0.4058 dengan loss sebesar 1.6002. Namun, seiiring bertambahnya jumlah epoch, model menunjukkan peningkatan pemforma secara konsisten. Hal ini terlihat pada epoch ke-10, Dimana akurasi model mencapai 0.8555 dan nilai loss menurun signifika menjadi 0.3545. Demikian pula, validation accuracy meningkat dari 0.6107 di awal menjadi 0.8765 pada epoch ke-10, sedangkan validation loss menurun dari 0.7875 menjadi 0.3017.

Perubahan nilai-nilai ini menunjukkan bahwa model belajar secara efektif dari data pelatihan, dengan kemampuan generalisasi yang baik terhadap data pengujian. Penurunan loss yang stabil dan kenaikan accuracy yang signifikan mengindikasikan bahwa proses pelatihan berjalan optimal tanpa overfitting.

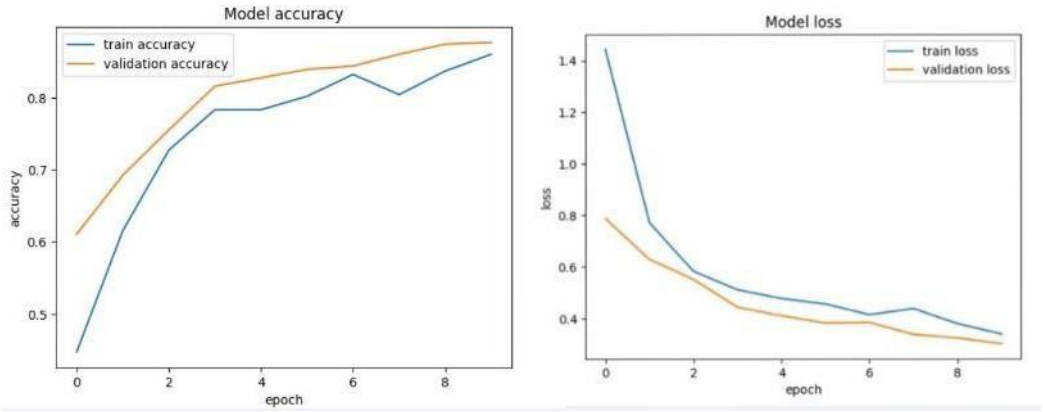
Table 4.8 Proses Visualisasi Akurasi dan Loss Model

Visualisasi Akurasi dan Loss Model

```
import matplotlib.pyplot as plt
# Definisikan fungsi plot_accuracy
def plot_accuracy(history):
  plt.plot(history.history['accuracy'], label='train accuracy')
  plt.plot(history.history['val_accuracy'], label='validation
accuracy')
  plt.title('Model accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(loc='best')
  plt.savefig('accuracy_v1_inceptionV3.jpg')
  plt.show()

# Definisikan fungsi plot_loss sekarang berada di luar plot_accuracy
def plot_loss(history):
    plt.plot(history.history['loss'], label='train loss')
    plt.plot(history.history['val_loss'], label='validation loss')
    plt.title('Model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(loc='best')
    plt.savefig('loss_v1_inceptionV3.jpg')
    plt.show()
```

```
# memanggil fungsi untuk plot akurasi dan loss
plot_accuracy(r)
plot_loss(r)
```

Untuk mengevaluasi kinerja model selama proses pelatihan, dilakukan visualisasi terhadap dua metrik utama yaitu akurasi dan loss. Visualisasi ini sangat penting untuk memahami sejauh mana model belajar dari data latih dan bagaimana pemformanya terhadap data validasi, sehingga dapat digunakan untuk mendeteksi kemungkinan overfitting. Visualisasi akurasi dilakukan dengan mendefinisikan fungsi plot_accuracy(history) yang menerima objek history dari hasil pelatihan model menggunakan Keras. Fungsi kedua adalah plot_loss(history) yang digunakan untuk menvisualisasikan nilai loss selama proses pelatihan. Sama seperti fungsi akurasi, fungsi ini menvisualisasikan nilai loss (data latih) dan val_loss (data validasi), lalu menyimpannya sebagai file loss_v1_inceptionV3.jpg. Pemanggilan kedua fungsi dilakukan pada bagian akhir dengan menggunakan argumen r, yang merupakan objek hasil dari training model.



Gambar 4.6 Hasil Visualisasi Akurasi dan Loss Model CNN

Gambar 4.6 menunjukkan hasil visualisasi dari proses pelatihan model CNN dalam bentuk grafik akurasi dan loss. Grafik ini dihasilkan menggunakan library matplotlib pada phyton, yang menvisualisasikan pemforma model selama 10 epoch. Gambar kiri memperlihatkan perubahan nilai akurasi model terhadap data pelatihan dan data validasi. Terlihat bahwa akurasi meningkat secara konsisten seiiring bertambahnya

epoch. Pada akhir epoch ke-9, akurasi pelatihan mendekati 0,88 sementara akurasi validasi sedikit lebih tinggi, yaitu 0,90. Hal ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang tidak terlihat sebelumnya.

Gambar kanan menampilkan grafik nilai loss pada proses pelatihan dan validasi. Loss pada data pelatihan awalnya cukup tinggi sekitar 1,45, namun mengalami penurunan signifikan hingga mencapai nilai sekitar 0,35 pada akhir epoch. Demikian pula, loss pada data validasi menurun dari 0,80 menjadi 0,30. Penurunan loss yang stabil pada kedua data menunjukkan bahwa model berhasil mempelajari pola secara efektif dan tidak mengalami overfitting.

4.3 Prediksi Kematangan Buah Sawit Menggunakan UI Gradio

Sistem prediksi kematangan buah sawit dibangun menggunakan algoritma Convolutional Neural Network (CNN) yang telah dilatih sebelumnya dengan dataset buah sawit pada berbagai tingkat kematangan. Setelah proses pelatihan selesai dan model memperoleh akurasi yang cukup baik, Langkah selanjutnya adalah menguji model dengan citra baru menggunakan antarmuka berbasis Python dan Gradio.

Table 4.9 Proses Prediksi Tingkat Kematangan Buah Sawit

Proses Prediksi sistem

```
!pip install gradio
import os
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
import gradio as gr

category = {0: 'Belum Masak', 1: 'Masak', 2: 'Terlalu Masak'}
def predic_image(filename, model):
  if not os.path.exists(filename):
    print(f"Error: File not found at {filename}")
    return
  img_ = image.load_img(filename, target_size=(224, 224))
  img_array = image.img_to_array(img_)
  img_processed = np.expand_dims(img_array, axis=0)
  img_processed /=255

  prediction = model.predict(img_processed)
```

```
index = np.argmax(prediction)

plt.title("prediction: {}". format(category[index]))
plt.imshow(img_array)
plt.show()
```

Pada table 4.9 menunjukkan proses prediksi tingkat kematangan buah sawit menggunakan model CNN yang telah dilatih sebelumnya. Kode ini ditulis dalam bahasa pemprograman python dan menggunakan berberapa Pustaka utama yaitu, TensorFlow untuk memuat dan memproses gambar, NumPy untuk manipulasi array, serta Matplotlib untuk visualisasi hasil prediksi. Pertama adalah melakukan impor Pustaka yang dibutuhkan dan mendifisinikan label klasifikasi dalam bentuk dictionary, yaitu 0: Belum Masak, 1:Masak, dan 2:Terlalu Masak. Fungsi predic_image dibuat untuk proses citra yang diberikan melalui filename dan melakukan prediksi menggunakan model yang diberikan sebagai parameter.

Citra buah sawit terlebih dahulu dimuat dan diubah ukurannya menjadi resolusi 244x244 piksel untuk menyesuaikan dengan input layer model CNN. Setelah itu, citra dikonversi menjadi array numerik dan didistandarisasi ke rentang [0, 1] agar dapat dikenali oleh model. Citra yang telah diproses ini kemudian diberikan ke model untuk menghasilkan prediksi.

Table 4.10 Proses Prediksi Data Uji

Prediksi Data uji

`# Fungsi predic_dir sebagai fungsi terpisah dengan indentasi yang benar`
`def predic_dir(filedir, model, true_label):`
`  cols = 3`
`  pos = 0`
`  images = []`
 	
`  if not os.path.exists(filedir):`
`  print(f"Error: Directory not found at {filedir}")`
`  return`
 	
`  for i in sorted(os.listdir(filedir)):`

```
    filepath = os.path.join(filedir, i)

    if os.path.isfile(filepath):
        images.append(filepath)

total_images = len(images)

if total_images == 0:
    print(f"No images found in directory: {filedir}")
    return
print(f"Predicting images in directory: {filedir}")

for imggg in images:
  img = image.load_img(imggg, target_size=(224, 224))
  img_array = image.img_to_array(img)
  img_processed = np.expand_dims(img_array, axis=0)
  img_processed /= 255

  prediction = model.predict(img_processed)
  index = np.argmax(prediction)
  pred = category.get(index)

  if pred == true_label:
    pos += 1

acc = pos / total_images
print(f"Accuracy for {true_label}: {acc:.2f} ({pos}/{total_images})")
```

Table diatas menampilkan kode proses prediksi yang digunakan untuk menguji pemforma model CNN terhadap data uji yang disusun berdasarkan kategori kematangan buah sawit. Fungsi predic_dir() bertugas membaca seluruh gambar dalam sebuah kategori. Setiap gambar diprediksi kelasnya oleh model, kemudian hasil prediksi dibandingkan dengan label sebenarnya (true_label). Jika prediksi sesuai maka sistem akan mencatat sebagai prediksi yang benar. Setelah seluruh gambar direktori selesai diproses, sistem akan menghitung akurasi dengan membagi jumlah prediksi yang benar dengan jumlah total gambar.

Table 4.11 Prediksi Tingkat Kematangan Buah Sawit Dengan UI Gradio

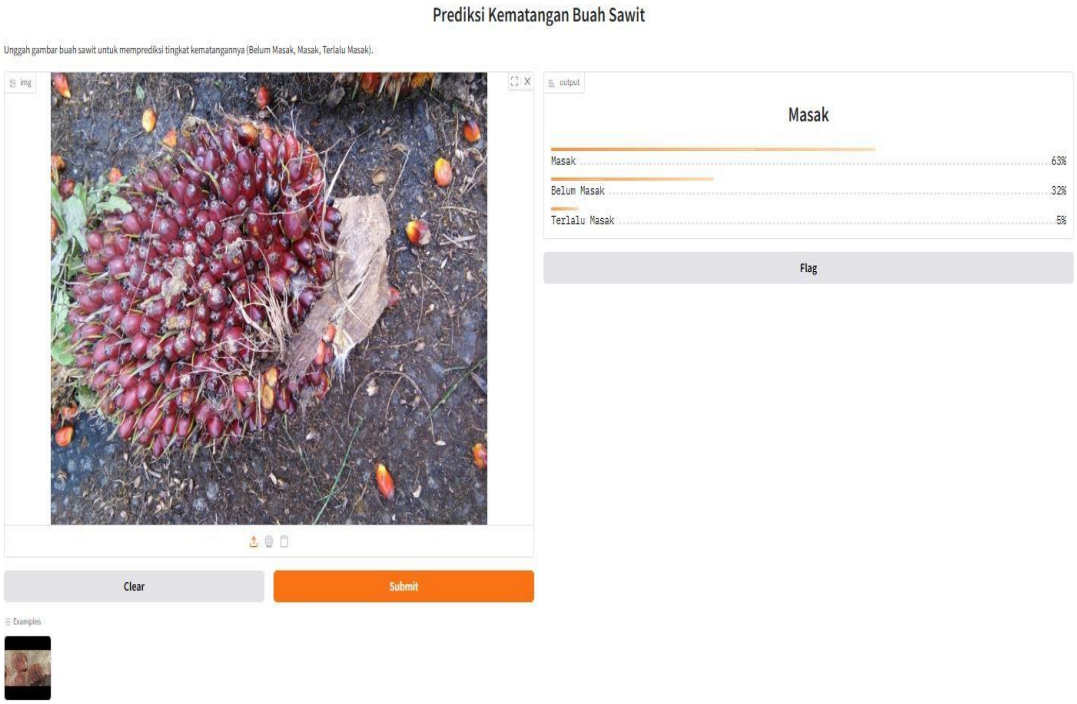
Prediksi Tingkat Kematangan Buah Sawit Dengan UI Gradio

```
# Fungsi prediksi untuk Gradio
def predict_fruit_ripeness(img):
    img_resized = img.resize((224, 224))
    img_array = image.img_to_array(img_resized)
    img_processed = np.expand_dims(img_array, axis=0)
    img_processed /= 255.0

    predictions = model.predict(img_processed)[0]

    results = {}
    for i, class_name in category.items():
        results[class_name] = float(predictions[i])

    return results
# Buat antarmuka Gradio
iface = gr.Interface(
    fn=predict_fruit_ripeness,
    inputs=gr.Image(type="pil"),
    outputs=gr.Label(),
    title="Prediksi Kematangan Buah Sawit",
    description="Unggah gambar buah sawit untuk memprediksi tingkat
kematangannya (Belum Masak, Masak, Terlalu Masak).", # Deskripsi UI
    examples=[
        '/content/drive/MyDrive/dataset_buahsawit1/test/Masak/Masak(5).jpg',
    ]
)
iface.launch(share=True)
```

Pada table 4.11 menunjukkan proses pembuatan user interface berbasis Gradio yang dirancang untuk klasifikasi tingkat kematangan buah sawit berdasarkan gambar yang diunggah pengguna. Fungsi predic_fruit_ripeness() merupakan fungsi prediksi utama yang menerima input gambar dalam formar PIL, lalu mengubah ukuran gambar menjadi 244 x 244 piksel, mengkonversinya menjadi array, dan menormalkan ke skala 0-1 agar sesuai dengan format input model CNN.

Setelah melaului proses normalisasi, gambar tersebut diprediksi menggunakan model, dan hasilnya dikembalikan dalam bentuk dictionary yang menunjukkan nilai probabilitas untuk setiap kelas (Belum Masak, Masak, dan Terlalu Masak). Probabilitas

ini kemudian ditampilkan sebagai output pada UI Gradio. Selanjutnya, UI dibangun menggunakan gr.Interface dengan fungsi predict_fruit_ripeness() sebagai inti prediksi. UI menerima input berupa gambar, menampilkan hasil prediksi berupa label, dan dilengkapi dengan deskripsi tugas klasifikasi. Contoh gambar (dari direktori Masak) disediakan untuk uji coba pengguna.

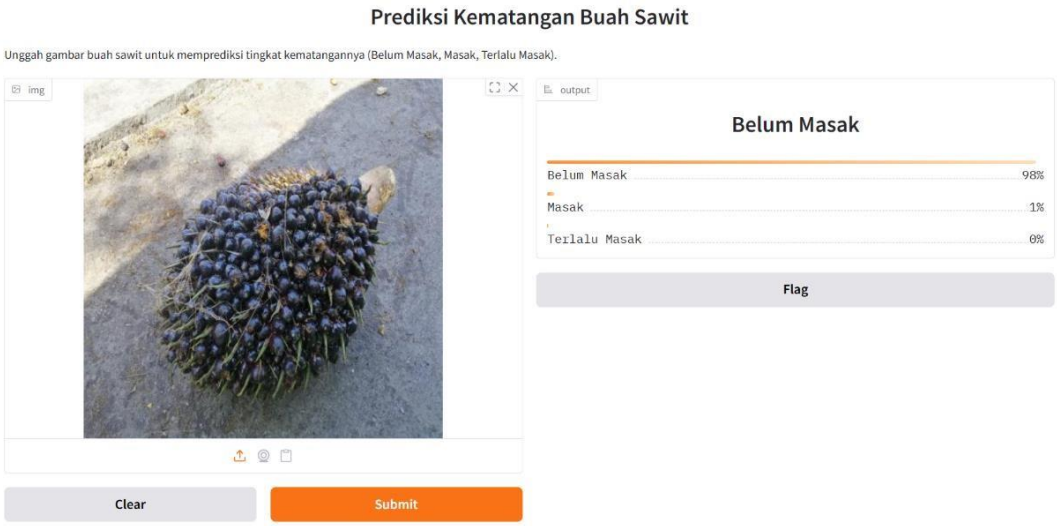


Gambar 4.7 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas “Masak”

Gambar diatas memperlihatkan user interface berbasis Gradio yang telah dibangun untuk memvisualisasikan hasil prediksi tingkat kematangan buah sawit menggunakan model CNN. UI ini memungkinkan pengguna untuk mengunggah gambar tandan buah sawit, yang kemudian akan diproses dan diprediksi oleh model kedalam tig akelas: Belum Masak, Masak, dan Terlalu Masak.

Pada contoh yang ditampilkan, sistem menampilkan hasil prediksi bahwa buah sawit dala gambar tergolong kedalam kelas Masak dengan probabilitas sebesar 63%, diikuti oleh kelas Belum Masak sebesar 32%, dan Terlalu Masak sebesar 5%. Hasil ini divisualisasikan dalam bentuk horizontal bar chart, sehingga memudahkan pengguna

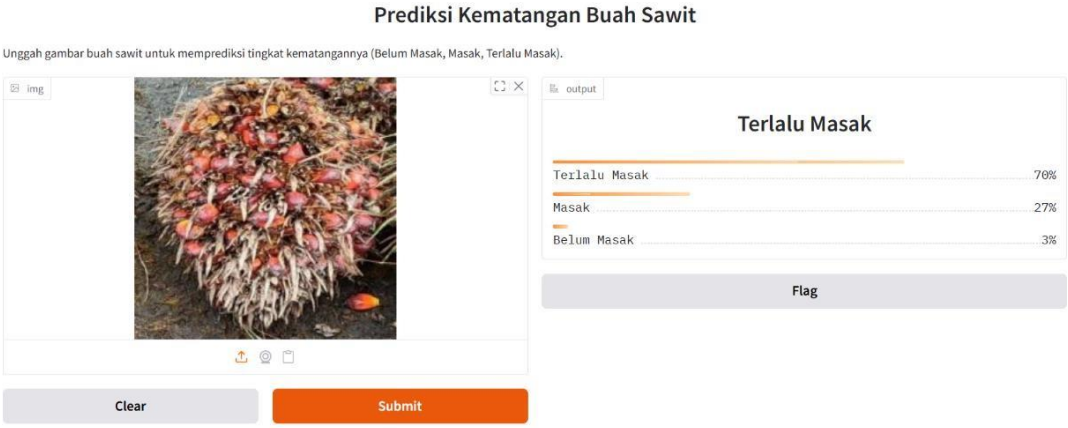
dalam memahami distribusi probabilitas antar kelas. Pengguna gradio sebagai antarmuka interaktif memberikan nilai tambah dalam hal kemudahan pengguna dan visualisasi. Tanpa perlu menulis ulang kode, pengguna dapat langsung mencoba sistem dengan hanya mengunggah gambar dari perangkat.



Gambar 4.8 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas "Belum Masak"

Gambar diatas menunjukkan tampilan user interface berbasis gradio yang dikembangkan untuk menvisualisasikan hasil prediksi tingkat kematangan buah sawit menggunkan model CNN. Antarmuka ini memungkinkan pengguna untuk menggunggah citra tandan buah sawit, yang kemudian akan diproses dan diprediksi oleh model ke dalam tiga kelas matang.

Pada gambar yang ditampilkan, sistem memberikan hasil prediksi bahwa buah sawit pada gambar termasuk ke dalam kelas Belum Masak dengan tingkat probabilitas sebesar 98%. Sementara itu, kelas Masak hanya mendapatkan nilai sebesar 1%, dan kelas Terlalu Masak sebesar 0%. Tampilan ini memberikan Gambaran nyata bagaimana sistem mampu mengidentifikasi kondisi kematangan buah sawit berdasarkan ciri visual. Dengan probabilitas yang tinggi pada kelas "Belum Masak", sistem ini menunjukkan pemforma yang baik dalam mengenali buah sawit yang belum matang sepenuhnya, ditandai dengan warna buah yang masih kehitaman dan permukaan tandan yang belum mengkilap.



Gambar 4.9 Tampilan User Interface Gradio Klasifikasi Kematangan Buah Sawit Untuk Kelas "Terlalu Masak"

Pada gambar 4.9 sistem memprediksi bahwa tanda buah sawit tersebut tergolong dalam kategori Terlalu Masak dengan tingkat keyakinan sebesar 70%. Selain itu, sistem juga memberikan probabilitas sebesar 27% untuk kategori Masak, dan hanya 3% untuk kategori Belum Masak. Prediksi ini memperlihatkan bahwa sistem mengenali karakteristik visual dari buah sawit yang sudah melewati masa panen optimal, seperti warna yang cenderung merah tua kecoklatan, kulit yang mulai mengering, dan tektur buah yang tidak lagi mengkilap.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis, implementasi, dan pengujian system indentifikasi tingkat kematangan buah kelapa sawit menggunakan algoritma Convolutional Neural Network (CNN) yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Model CNN yang dibangun berhasil melakukan klasifikasi kematangan buah sawit ke dalam tiga kelas yaitu Belum Masak, Masak, dan Terlalu Masak, dengan Tingkat akurasi tertinggi mencapai 87.65% pada data validasi setelah dilakukam pelatihan selama 10 epoch.
2. Proses augmentasi data dan preprocessing seperti normalisasi, resizing, dan peningkatan kualitas gambar (enhancement) terbukti efektif dalam meningkatkan kemampuan generalisasi model terhadap citra baru.
3. Penggunaan inceptionV3 sebagai metode transfer learning memberikan kontribusi besar terhadap akurasi dan efisiensi proses pelatihan model karena memanfaatkan bobot pre-trained yang sudah memiliki pemahaman terhadap vitur visual umum.
4. Antarmuka pengguna (UI) berbasis Gradio yang dibangun mempermudah pengguna dalam menggunggah gambar dan memperoleh hasil prediksi secara langsung dan intuitif, serta menampilkan hasil klasifikasi dalam bentuk persentase probabilitas dengan visualisasi bar chart.
5. System ini dapat membantu petani dan industry kelapa sawit dalam melakukan klasifikasi otomatis terhadap tingkat kematangan buah, sehingga dapat meningkatkan efisiensi panen dan mengurangi kesalahan manusia dalam indentifikasi visual.

5.2 Saran

Berdasarkan hasil yang didapatkan dan diperoleh pada penelitian ini, terdapat beberapa saran yang dapat diberikan untuk penelitian selanjutnya :

1. Jumlah data perlu ditingkatkan, terutama dengan variasi pencahayaan, latar belakang, dan posi buah sawit untuk meningkatkan kemampuan model dalam mengenali citra yang lebih kompleks dan realistik.
2. Penggunaan arsitektur model yang lebih kompleks atau kombinasi metode deep learning seperti ResNet atau EfficienNet dapat dieksplorasi untuk meningkatkan akurasi lebih lanjut.
3. System dapat dikembangkan ke dalam bentuk aplikasi mobile atau desktop agar dapat digunakan secara langsung oleh petani di lapangan tanpa memerlukan akses Google Colab.

DAFTAR PUSTAKA

Al-Khowarizmi, Syah, R., Nasution, M. K. M., & Elveny, M. (2021). Sensitivity of MAPE using detection rate for big data forecasting crude palm oil on k-nearest neighbor. *International Journal of Electrical and Computer Engineering*, *11*(3), 2696–2703. https://doi.org/10.11591/ijece.v11i3.pp2696-2703

Alkusma, Y. M., Hermawan, H., & Hadiyanto, H. (2016). Pengembangan Potensi Energi Alternatif Dengan Pemanfaatan Limbah Cair Kelapa Sawit Sebagai Sumber Energi Baru Terbarukan Di Kabupaten Kotawaringin Timur. *Jurnal Ilmu Lingkungan*, *14*(2), 96. https://doi.org/10.14710/jil.14.2.96-102

Amazon. (2023). *Deep Learning vs Machine Learning - Perbedaan Antara Berbagai Teknologi Data - AWS*. https://aws.amazon.com/id/compare/the-difference-between-machine-learning-and-deep-learning/
apa-itu-convolutional-neural-network-836f70b193a4. (n.d.).

Arkadia, A., Ayu Damayanti, S., & Sandya Prasvita, D. (2021). Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, *2*(2), 158–165. https://conference.upnvj.ac.id/index.php/senamika/article/view/1813

Dayera, Musa Bundaris Palungan, F. O. (2024). G-Tech : Jurnal Teknologi Terapan. *G-Tech : Jurnal Teknologi Terapan*, *8*(1), 186–195. https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/1823/1229

Ermawati, T., & Septia, Y. (2013). Kinerja Ekspor Minyak Kelapa Sawit Indonesia. *Buletin Ilmiah Litbang Perdagangan*, *7*(10), 129–148. http://jurnal.kemendag.go.id/bilp/article/view/104

Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, *3*(2), 49–56.

Jaya, & Risma, A. (2022). *Repository _ Universitas Hasanuddin*.

Klasifikasi Tingkat Kematangan Buah Sawit Berdasarkan Ekstraksi Fitur RGB dan GLCM Menggunakan Algoritma K-NN. (2023). *Jurnal Ilmiah Komputasi*, *22*(4), 457–466. https://doi.org/10.32409/jikstik.22.4.3402

Kusuma Putra, A., Bunyamin, H., & Maranatha Jl drg Surya Sumantri No, K. (2020). Pengenalan Simbol Matematika dengan Metode Convolutional Neural Network (CNN). *Jurnal Strategi*, *2*(November), 426.

Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer Vision Dan Pengolahan Citra Digital*.

Maulana, I., Khairunisa, N., & Mufidah, R. (2024). Deteksi Bentuk Wajah Menggunakan Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(6), 3348–3355. https://doi.org/10.36040/jati.v7i6.8171

Maya, A., Putri, K., Rozi, A. F., Informasi, S., Mercu, U., & Yogyakarta, B. (2024). *IMPLEMENTASI CONVUTIONAL NEURAL NETWORK DALAM*. *8*(5), 10388–10394. *mengenal-kriteria-matang-buah-kelapa-sawit*. (n.d.).

Nandaputra, R., Studi, P., Informatika, T., Sains, F., Teknologi, D. A. N., Islam, U., & Syarif, N. (2024). *IMPLEMENTASI TRANSFER LEARNING PADA KLASIFIKASI KUALITAS KELAPA MENGGUNAKAN PRE-TRAINED MODEL EFFICIENTNET KLASIFIKASI KUALITAS KELAPA MENGGUNAKAN PRE-TRAINED MODEL*. *penerapan-convolutional-neural-network-cnn-terhadap-jenis-unggas-ordo-ansariformes-menggunakan-5d0b408da20e*. (n.d.).

Putri, A. R. (2016). Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *1*(01), 1–6. https://doi.org/10.29100/jipi.v1i01.18

Roy, R. R. (2024). Deep Learning. *Networked Artificial Intelligence*, *November*, 83–98. https://doi.org/10.1201/9781003499466-10

Saragih, H. M., & Rahayu, H. (2022). Pengaruh kebijakan Uni Eropa terhadap ekspor kelapa sawit Indonesia. *JPPI (Jurnal Penelitian Pendidikan Indonesia)*, *8*(2), 296. https://doi.org/10.29210/020221377

Setyamidjaja, D. (2006). Budi Daya Kelapa Sawit. In *Kanisius*. http://repository.pertanian.go.id/handle/123456789/14851%0Ahttp://repository.pertanian.go.id/bitstream/handle/123456789/14851/BUDI DAYA KELAPA SAWIT.pdf?sequence=1&isAllowed=y

Shohibuddin, M., Utami, T. H., Listiana, H., Munawwarah, F., Mahendra, Y., Timamah, I., Jannah, F., & Permatasari, I. D. (n.d.). *DEEP LEARNING DALAM PENDIDIKAN DAN ARTIFICIAL*.

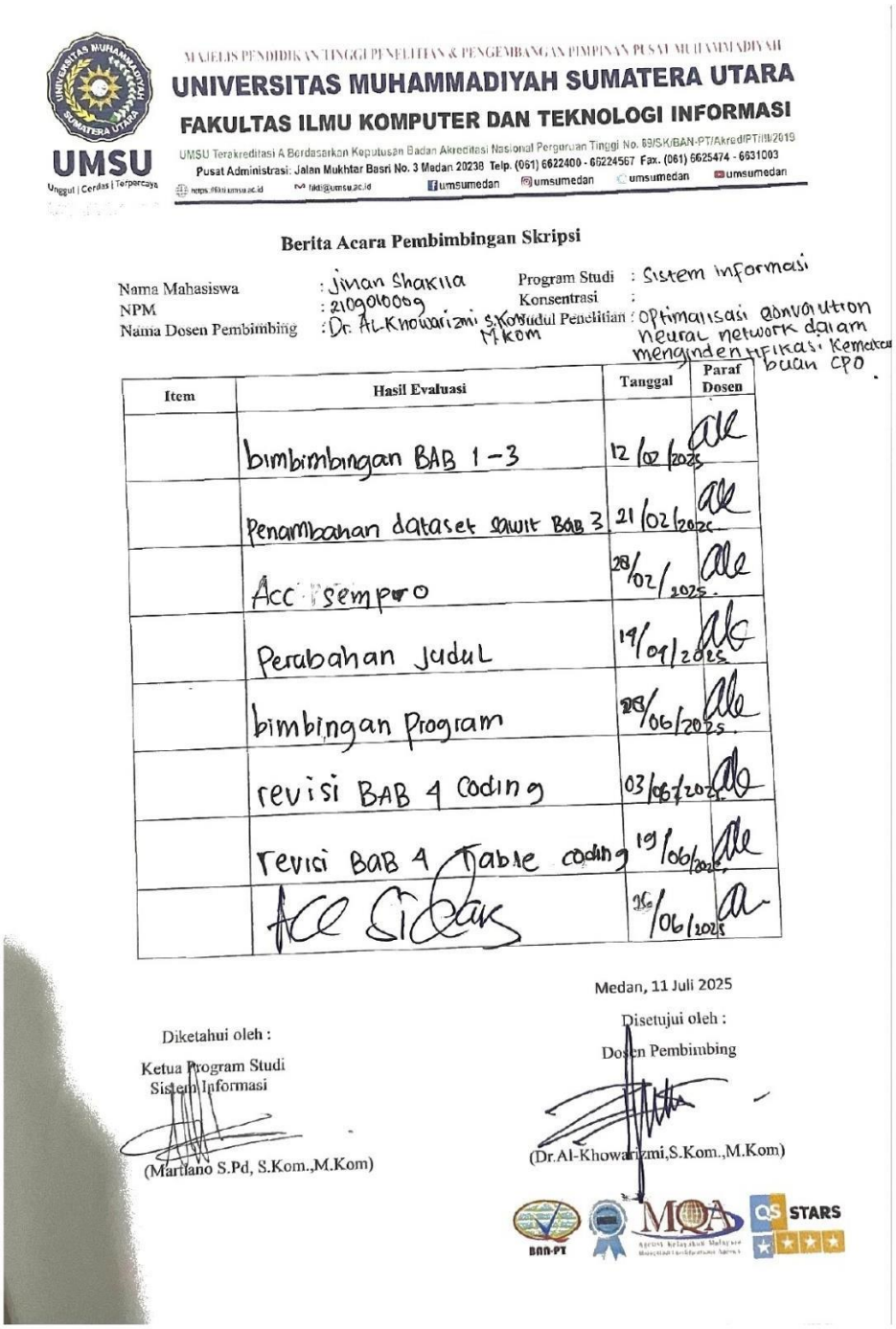
Sugiyono, R. A. (2022). Implemantasi Kompresi Citra Digital Mengatur Kualitas Citra Digital. *Jurnal Pendidikan Dan Konseling*, *4*(5), 1349–1358.

Suhardin, I., Patombongi, A., & Islah, A. M. (2021). MENGIDENTIFIKASI JENIS TANAMAN BERDASARKAN CITRA DAUN MENGGUNAKAN AlGORITMA CONVOLUTIONAL NEURAL NETWORK. *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, *6*(2), 100–108. https://doi.org/10.51876/simtek.v6i2.101*TensorFlow*. (n.d.).

Triwijoyo, B. K. (2019). Model Fast Tansfer Learning pada Jaringan Syaraf Tiruan Konvolusional untuk Klasifikasi Gender Berdasarkan Citra Wajah. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *18*(2), 211–221. https://doi.org/10.30812/matrik.v18i2.376

LAMPIRAN

1. Lampiran surat pembimbing



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

UMSU Unggul | Cerdas | Terpercaya

https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

Berita Acara Pembimbingan Skripsi

Nama Mahasiswa : Jihan Shakila
NPM : 2109010009
Nama Dosen Pembimbing : Dr. AL-Khowarizmi S.Kom M.Kom
Program Studi : Sistem Informasi
Konsentrasi :
Judul Penelitian : Optimalisasi convolution neural network dalam mengidentifikasi kematangan buah CPO.

Item	Hasil Evaluasi	Tanggal	Paraf Dosen
	bimbimbingan BAB 1-3	12/02/2025	
	Penambahan dataset sawit BAB 3	21/02/2025	
	Acc sempro	28/02/2025	
	Perubahan judul	19/04/2025	
	bimbingan program	28/06/2025	
	revisi BAB 4 coding	03/06/2025	
	revisi BAB 4 tabel coding	19/06/2025	
	Acc Sidang	26/06/2025	

Medan, 11 Juli 2025

Diketahui oleh :
Ketua Program Studi
Sistem Informasi

(Martiano S.Pd, S.Kom.,M.Kom)

Disetujui oleh :
Dosen Pembimbing

(Dr.Al-Khowarizmi,S.Kom.,M.Kom)

BAN-PT MQA QS STARS