

**DETEKSI PENYAKIT TANAMAN MERICA MENGGUNAKAN
ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* (CNN)
BERBASIS CITRA DAUN**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana Komputer
(S.Kom) Program Studi Informatika



Reski Anugrah Sari

105841104421

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

**DETEKSI PENYAKIT TANAMAN MERICA
MENGUNAKAN ALGORITMA CONVOLUTIONAL
NEURAL NETWORK (CNN) BERBASIS CITRA DAUN**

Diajukan sebagai Salah Satu Syarat Untuk Mendapat Gelar Sarjana Komputer
(S.Kom) Program Studi Informatika

Disusun dan Diajukan oleh :

Reski Anugrah Sari

105841104421

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR**

2025

PENGESAHAN

Skripsi atas nama Reski Anugrah Sari dengan nomor induk Mahasiswa 105841104421, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng

2. Penguji

a. Ketua

: Dr. Ir. ZAHIR ZAINUDDIN, M.Sc.

b. Sekertaris

: MUHYIDDIN A M HAYAT, S.Kom, MT

3. Anggota

1. RIZKI YUSLIANA BAKTI, S.T., M.T.

2. TITIN WAHYUNI, S.Pd, M.T

3. DARNIATI, S.Kom., M.T

Mengetahui:

Pembimbing I

Pembimbing II

Chyquitha Danuputri, S.Kom, M.Kom

LUKMAN, S.Kom, M.T

Dekan

Dr. Muh. Syafar S. Kuba, S.T., M.T.

NPM : 795 283



HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi :DETEKSI PENYAKIT TANAMAN MERICA MENGGUNAKAN
ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)
BERBASIS CITRA DAUN

Nama : Reski Anugrah Sari

Stambuk : 105 841104421

Makassar, 09 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Chyquitha Danuputri, S.Kom, M.Kom

Pembimbing II

LUKMAN, S.Kom, M.T

Mengetahui,

Ketua Prodi Informatika



Rizki Kusliana Bakti, S.T., M.T.

NBM: 1307 284



MOTTO DAN PERSEMBAHAN

MOTTO

“Bukan Sekadar Gambar, Tapi Data. Bukan Sekadar Prediksi, Tapi Solusi”

“Melatih Mesin untuk 'Melihat', Membantu Manusia untuk Bertindak”

PERSEMBAHAN

Tiada kata yang mampu melukiskan betapa besar pengorbanan dan kasih sayang yang telah dicurahkan oleh kedua orang tua tercinta, Bapak Suma dan Ibu Mirawati. Persembahan ini adalah wujud bakti dan terima kasih atas setiap tetes keringat, doa tulus, serta dukungan tanpa henti yang tak pernah putus. Semoga Allah SWT senantiasa melimpahkan kesehatan, kebahagiaan, dan keberkahan kepada kalian berdua. Untuk saudara-saudaraku tersayang, Mirpan, Mirwan, Refa, dan Hafis, terima kasih atas kebersamaan, canda tawa, dan dukungan yang selalu ada. Kalian adalah bagian tak terpisahkan dari perjalanan hidup ini, yang selalu memberikan semangat dan motivasi. Semoga kita semua selalu dalam lindungan dan rahmat-Nya. Kepada Muh. Ulil Amri, terima kasih atas pengertian, kesabaran, dan dukungan yang tak pernah pudar selama proses penyusunan skripsi ini. Kehadiranmu adalah pelengkap dan penyemangat dalam setiap langkah. Tak lupa, kepada seluruh teman-teman seperjuangan kelas B, yang telah menjadi keluarga kedua selama masa perkuliahan. Terima kasih atas kebersamaan, diskusi, dan kenangan indah yang tak akan terlupakan. Semoga persahabatan kita tetap terjalin erat. Akhir kata, Kepada diriku sendiri, ini adalah bukti bahwa setiap tantangan adalah kesempatan untuk tumbuh. Terima kasih telah belajar dari setiap kesalahan, bangkit dari setiap kegagalan, dan terus melangkah maju menjadi versi terbaik.

ABSTRAK

Tanaman merica (*Piper nigrum*) merupakan komoditas perkebunan bernilai tinggi yang produktivitasnya sering terancam oleh serangan penyakit seperti busuk pangkal batang, penyakit kuning, dan bercak daun. Identifikasi penyakit secara manual oleh petani seringkali tidak efisien, memakan waktu, dan rentan terhadap kesalahan, sehingga berisiko menurunkan hasil panen secara signifikan. Penelitian ini bertujuan untuk menerapkan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2 untuk membangun sistem deteksi penyakit pada daun merica secara otomatis berbasis citra. Selain itu, penelitian ini juga bertujuan untuk menganalisis pengaruh teknik augmentasi data terhadap performa model. Metode penelitian melibatkan pengumpulan 750 citra daun dari perkebunan di Desa Matano, Luwu Timur, yang dibagi menjadi lima kelas: tiga kelas penyakit, satu kelas daun sehat, dan satu kelas bukan daun merica. Dua skenario pengujian dilakukan: Model A dilatih menggunakan data asli, sedangkan Model B menggunakan data yang telah melalui augmentasi. Hasil pengujian menunjukkan bahwa penerapan augmentasi data secara signifikan meningkatkan kinerja model. Akurasi Model B mencapai 91%, lebih unggul dibandingkan Model A yang hanya mencapai 85%. Augmentasi juga terbukti efektif menekan overfitting dan mengurangi kesalahan klasifikasi fatal (*False Negative*), di mana daun sakit terdeteksi sebagai daun sehat. Penelitian ini menyimpulkan bahwa CNN dengan arsitektur MobileNetV2 dan teknik augmentasi data merupakan solusi yang efektif dan andal untuk identifikasi penyakit tanaman merica, serta berpotensi untuk dikembangkan menjadi aplikasi praktis bagi petani.

Kata Kunci: Deteksi Penyakit, Tanaman Merica, CNN, MobileNet, Augmentasi Data

ABSTRACT

Pepper (Piper nigrum) is a high-value plantation commodity whose productivity is often threatened by diseases such as stem rot, yellow disease, and leaf spot. Manual disease identification by farmers is often inefficient, time-consuming, and prone to errors, risking a significant reduction in harvest yields. This research aims to implement the Convolutional Neural Network (CNN) algorithm with the MobileNetV2 architecture to build an image-based automatic detection system for pepper leaf diseases. Furthermore, this study also aims to analyze the effect of data augmentation techniques on model performance. The research method involved collecting 750 leaf images from a plantation in Matano Village, East Luwu, which were divided into five classes: three disease classes, one healthy leaf class, and one non-pepper leaf class. Two testing scenarios were conducted: Model A was trained using the original data, while Model B used augmented data. The test results show that the application of data augmentation significantly improved the model's performance. Model B's accuracy reached 91%, outperforming Model A, which only achieved 85%. Augmentation also proved effective in suppressing overfitting and reducing fatal misclassifications (False Negatives), where diseased leaves were detected as healthy. This study concludes that CNN with the MobileNetV2 architecture and data augmentation techniques is an effective and reliable solution for pepper plant disease identification, with the potential for development into a practical application for farmers.

Keywords: *Disease Detection, Pepper Plant, CNN, MobileNet, Data Augmentation*

KATA PENGATAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamualaikum wr.wb

Segala puji dan syukur penulis panjatkan kehadirat Allah Subhanallahu Wa Ta'ala, atas segala limpahan rahmat, karunia, dan kesehatan yang tak terhingga. Hanya dengan pertolongan-Nya, penulis dapat menyelesaikan skripsi yang berjudul **“DETEKSI PENYAKIT TANAMAN MERICA MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS CITRA DAUN”**. Shalawat serta salam tak lupa penulis kirimkan kepada baginda Rasulullah, Nabi Muhammad Shallallahu Alayhi Wasallam sebagai uswatun hasanah dan rahmatan lil alamin.

Dalam penyusunan Skripsi ini, penulis menghaturkan ucapan terima kasih yang mendalam kepada seluruh pihak yang telah memberikan kontribusi, dukungan, dan bimbingan selama proses penyusunan skripsi ini, terutama kepada:

1. Kedua orang tua tercinta, Bapak Suma dan Ibu Mirawati, atas segala doa, kasih sayang, pengorbanan, serta dukungan moril maupun materil yang tiada henti. Tanpa restu dan ridha kalian, penulis tidak akan mampu berada pada tahap ini.
2. Bapak Dr. Ir. H. Abd. Rakhim Nanda, S.T., M.T., IPU, selaku Rektor Universitas Muhammadiyah Makassar.
3. Bapak Ir. Muhammad Syafaat S. Kuba, S.T., M.T, dalam kapasitas beliau sebagai Dekan Fakultas Teknik.
4. Ibu Rizki Yusliana Bakti, S.T., M.T, selaku Ketua Program Studi Informatika.
5. Ibu Chyquitha Danuputri, S.Kom., M.Kom., selaku Dosen Pembimbing I yang dengan penuh kesabaran telah memberikan arahan, bimbingan, dan ilmu yang sangat berharga dalam proses penyusunan skripsi ini.

6. Bapak Lukman, S.Kom., M.T., selaku Dosen Pembimbing II yang juga memberikan saran, masukan, serta bimbingan yang sangat membantu dalam penyusunan skripsi ini.
7. Ibu Rizki Yusliana Bakti, S.T., M.T., selaku Pembimbing Akademi yang senantiasa memberikan arahan dan motivasi kepada mahasiswa.
8. Seluruh jajaran Dosen dan Tenaga Kependidikan di lingkungan Fakultas Teknik Universitas Muhammadiyah Makassar.
9. Bapak Suma selaku pemilik lokasi penelitian yang telah memberikan izin, fasilitas, dan kesempatan bagi penulis untuk melakukan penelitian.
10. Teman-teman dari Kelas B Angkatan 2021 Program Studi Informatika Universitas Muhammadiyah Makassar, atas kebersamaan, diskusi, bantuan, serta dukungan yang selalu menjadi penguat dalam perjalanan akademik ini.
11. Rekan-rekan mahasiswa Angkatan 2021 Fakultas Teknik Universitas Muhammadiyah Makassar, atas semangat, kerjasama, dan kebersamaan yang telah terjalin selama masa perkuliahan.
12. Untuk seseorang yang tidak dapat penulis sebutkan namanya di sini, namun kehadirannya sangat berarti. Terima kasih atas doa, perhatian, semangat, dan pengertian yang selalu diberikan. Kehadiranmu menjadi bagian penting dalam perjalanan penulis menyelesaikan skripsi ini.
13. Kepada diri sendiri, penulis menyampaikan penghargaan atas keteguhan, kedisiplinan, dan konsistensi dalam menghadapi berbagai tantangan serta keterbatasan selama proses penyusunan skripsi ini. Semoga ketekunan dan semangat tersebut senantiasa terjaga sebagai bekal berharga dalam menapaki tahap kehidupan dan perjalanan akademik selanjutnya.
14. Kepada semua pihak yang tidak dapat disebutkan satu per satu, penulis tetap menyampaikan rasa terima kasih yang sebesar-besarnya atas segala bantuan, doa, dan dukungan yang diberikan.

Penulis menyadari bahwa Laporan skripsi ini masih jauh dari kata sempurna. Untuk itu, penulis mengharapkan kritik dan saran guna penyempurnaan di masa mendatang. Harapan penulis, semoga Laporan skripsi ini dapat memberikan sumbangsih ilmu pengetahuan dan manfaat praktis, khususnya dalam bidang pertanian terkait identifikasi penyakit tanaman Merica. Akhir kata, penulis memohon maaf atas segala keterbatasan dan kekhilafan yang terdapat dalam penulisan Laporan Tugas Akhir ini.

Billahi fisabililhaq, fastabiqul khairat.

Wassalamualaikum Wr.Wb.

Makassar, Juni 2025

Penulis,

Reski Anugrah Sari



DAFTAR ISI

MOTTO DAN PERSEMBAHAN	i
ABSTRAK.....	ii
ABSTRACT.....	iii
KATA PENGATAR.....	iv
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN.....	xi
DAFTAR ISTILAH	xii
BAB I PENDAHULUAN.....	1
A. LATAR BELAKANG	1
B. RUMUSAN MASALAH.....	3
C. TUJUAN PENELITIAN.....	3
D. MANFAAT PENELITIAN	3
E. RUANG LINGKUP PENELITIAN.....	4
F. SISTEMATIKA PENULISAN	5
BAB II TINJAUAN PUSTAKA.....	7
A. LANDASAN TEORI.....	7
B. PENELITIAN TERKAIT	14
C. KERANGKA PIKIR.....	18
BAB III METODE PENELITIAN.....	19
A. TEMPAT DAN WAKTU PENELITIAN	19
B. ALAT DAN BAHAN	19
C. PERANCANGAN SISTEM.....	21
D. TEKNIK PENGUJIAN SISTEM	27
E. TEKNIK ANALISIS DATA	29
BAB IV HASIL DAN PEMBAHASAN	36
A. IMPLEMENTASI SISTEM.....	36

B. HASIL PENGUJIAN DAN EVALUASI MODEL.....	39
C. PEMBAHASAN.....	55
BAB V PENUTUP.....	57
A. KESIMPULAN.....	57
B. SARAN.....	57
DAFTAR PUSTAKA.....	59



DAFTAR GAMBAR

Gambar 1. Penyakit Busuk Pangkal Batang (erlanardianarismansyah.blogspot.com) .	8
Gambar 2. Penyakit Kuning (Ir.Tri Juni Sasonko, 2023).....	8
Gambar 3. Penyakit Bercak Daun (ditjenbun, 2021)	9
Gambar 4. Kerangka Pikir.....	18
Gambar 5. Flowchart Sistem.....	21
Gambar 6. Flowchart CNN dengan arsitektur MobileNet	24
Gambar 7. Flowchart Prosedur analisis data.....	33
Gambar 8. Dataset perkelas.....	37
Gambar 9. Hasil Augmentasi Citra	38
Gambar 10. Grafik Perbandingan Akurasi dan Loss Pelatihan & Validasi.....	40
Gambar 11. confusion matrix Model A.....	43
Gambar 12. confusion matrix Model B.....	47
Gambar 13. Tampilan Hasil Prediksi untuk Penyakit Bercak Daun	51
Gambar 14. Tampilan Hasil Prediksi untuk Penyakit Busuk Pangkal Batang.....	52
Gambar 15. Tampilan Hasil Prediksi untuk Penyakit Kuning	53
Gambar 16. Tampilan Hasil Prediksi untuk Daun Sehat.....	54
Gambar 17. Tampilan Hasil Prediksi Bukan daun merica	54

DAFTAR TABEL

Tabel 1. Penelitian Terkait.....	16
Tabel 2. Pembagian dataset	37
Tabel 3. Konfigurasi hyperparameter Pelatihan.....	39
Tabel 4. Perbandingan Head-to-Head Kedua Model	41
Tabel 5. Metrik Evaluasi Kinerja Model A pada Data Uji	44
Tabel 6. Metrik Evaluasi Kinerja Model B pada Data Uji.....	49



DAFTAR LAMPIRAN

Lampiran 1. Surat Izin Penelitian.....	62
Lampiran 2. Dokumentasi Penelitian.....	63
Lampiran 3. Dataset Citra Tiap Kelas	64
Lampiran 4. Source Code.....	71
Lampiran 5. Hasil Turnitin.....	86



DAFTAR ISTILAH

Akurasi (*Accuracy*)

Persentase total prediksi yang benar dari keseluruhan data uji. Ini mengukur seberapa sering model memberikan tebakan yang benar secara keseluruhan.

Algoritma

Serangkaian instruksi atau aturan yang didefinisikan dengan baik untuk memecahkan masalah atau melakukan tugas. Dalam proposal ini, merujuk pada *Convolutional Neural Network* (CNN).

Arsitektur *MobileNet*

Jenis arsitektur CNN yang dikembangkan oleh Google, dirancang khusus agar efisien untuk aplikasi pada perangkat seluler atau perangkat dengan keterbatasan komputasi.

Artificial Intelligence (AI)

Cabang ilmu komputer yang bertujuan menciptakan mesin dengan kemampuan meniru kecerdasan manusia, memungkinkan komputer untuk belajar dari data, mengenali pola, dan membuat keputusan.

Augmentasi Data

Serangkaian teknik untuk meningkatkan jumlah dan keragaman data latih secara buatan dengan menerapkan transformasi pada gambar asli, seperti rotasi, *flipping*, dan penyesuaian kecerahan.

Batch Size

Jumlah sampel data yang diproses oleh model sebelum bobotnya diperbarui selama satu iterasi pelatihan.

Bercak Daun (*Colletotrichum gloeosporioides*)

Penyakit pada tanaman merica yang menimbulkan bercak cokelat atau hitam, yang menyebabkan daun menguning dan gugur sebelum waktunya.

Busuk Pangkal Batang (*Phytophthora capsici*)

Penyakit yang menyerang pangkal batang dan daun merica, ditandai dengan bintik-bintik kecil hitam dan coklat yang membesar menjadi bercak hitam.

Citra Digital

Representasi visual dari objek dunia nyata yang diubah ke dalam format digital dan tersusun dari elemen-elemen kecil bernama piksel.

Convolutional Neural Network (CNN)

Salah satu jenis jaringan saraf tiruan yang sangat efektif untuk tugas analisis visual dan pengenalan pola pada gambar.

Dataset

Kumpulan data, dalam hal ini adalah 600 citra daun merica, yang digunakan untuk melatih, memvalidasi, dan menguji model.

Deep Learning (DL)

Sub-bidang dalam *Machine Learning* yang menggunakan jaringan saraf tiruan berlapis untuk secara otomatis mengekstrak fitur-fitur rumit dari data mentah.

Depthwise Separable Convolution

Teknik konvolusi efisien yang digunakan dalam arsitektur MobileNet, yang memecah proses menjadi dua langkah: *Depthwise Convolution* dan *Pointwise Convolution* untuk mengurangi biaya komputasi.

Epoch

Satu siklus lengkap di mana seluruh dataset pelatihan telah melewati proses pelatihan maju (*forward pass*) dan mundur (*backward pass*) melalui jaringan saraf.

Evaluasi Model

Proses mengukur kinerja model menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score* pada data uji untuk menilai kemampuannya.

F1-Score

Rata-rata harmonik antara presisi dan *recall*, yang memberikan ukuran keseimbangan antara kedua metrik tersebut.

False Negative (FN)

Kesalahan klasifikasi di mana model memprediksi suatu kondisi sebagai negatif (misalnya, 'sehat') padahal kondisi sebenarnya adalah positif (misalnya, 'sakit').

False Positive (FP)

Kesalahan klasifikasi di mana model memprediksi suatu kondisi sebagai positif (misalnya, 'sakit') padahal kondisi

	sebenarnya adalah negatif (misalnya, 'sehat').
<i>Fully Connected Layer</i>	Lapisan dalam arsitektur CNN yang berfungsi sebagai pengklasifikasi akhir, di mana setiap neuron terhubung ke semua neuron di lapisan sebelumnya untuk membuat keputusan akhir.
Generalisasi	Kemampuan model <i>deep learning</i> untuk beradaptasi dan memberikan hasil yang akurat pada data baru yang belum pernah dilihat sebelumnya selama pelatihan.
Hyperparameter	Parameter konfigurasi yang nilainya diatur sebelum proses pelatihan model dimulai, seperti <i>learning rate</i> , <i>batch size</i> , dan jumlah <i>epoch</i> .
Implementasi	Proses penerapan rancangan atau model ke dalam sistem yang berfungsi. Dalam proposal ini, mencakup persiapan lingkungan, pengolahan data, hingga pembangunan model CNN.
Klasifikasi	Proses mengkategorikan suatu input (misalnya, citra daun) ke dalam salah satu dari beberapa kelas yang telah ditentukan (misalnya, 'sehat', 'penyakit kuning', dll.).
Confusion Matrix	Tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi

	dengan menunjukkan jumlah prediksi benar dan salah untuk setiap kelas.
Learning Rate	Hyperparameter yang mengontrol seberapa besar penyesuaian yang dilakukan pada bobot model selama proses pelatihan untuk mengurangi nilai <i>loss</i> .
Loss (Kerugian)	Ukuran yang menunjukkan seberapa besar kesalahan yang dibuat oleh model dalam memprediksi. Tujuannya adalah meminimalkan nilai ini selama pelatihan.
Machine Learning	Bidang studi AI yang memberikan kemampuan pada sistem komputer untuk belajar dari data tanpa diprogram secara eksplisit.
Metrik	Ukuran standar yang digunakan untuk mengevaluasi aspek tertentu dari kinerja model, seperti akurasi, presisi, dan <i>recall</i> .
MobileNetV2	Versi lanjutan dari arsitektur MobileNet yang menawarkan performa lebih baik dengan jumlah parameter yang lebih sedikit, sehingga lebih efisien secara komputasi.
Normalisasi	Proses mengubah rentang nilai piksel dalam sebuah citra, misalnya menjadi rentang $[0, 1]$, untuk meningkatkan

	stabilitas dan mempercepat pelatihan model Kondisi di mana model <i>deep learning</i> memiliki performa sangat baik pada data latih tetapi buruk pada data baru (data validasi atau uji). Ini terjadi karena model "menghafal" data latih, bukan belajar polanya
Overfitting	
Piksel	Elemen terkecil dari sebuah citra digital yang tersusun dalam matriks baris dan kolom. Setiap piksel memiliki nilai yang merepresentasikan warna atau intensitas cahaya.
Pooling Layer	Lapisan dalam CNN yang berfungsi untuk mengurangi dimensi spasial (lebar dan tinggi) dari peta fitur, sehingga mengurangi jumlah parameter dan komputasi.
Presisi (<i>Precision</i>)	Metrik yang mengukur proporsi prediksi positif yang benar dari total prediksi positif yang dibuat oleh model. Mengukur seberapa "tepat" prediksi positif dari model.
<i>Preprocessing</i> (Pra-pemrosesan)	Tahapan awal dalam pengolahan citra digital sebelum dimasukkan ke model, seperti <i>resize</i> dan normalisasi, untuk mengubah citra menjadi format yang sesuai.

Recall (Sensitivitas)

Metrik yang mengukur proporsi kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model. Mengukur seberapa "lengkap" jangkauan model dalam menemukan semua kasus positif.

Resize

Proses mengubah ukuran dimensi (lebar dan tinggi) sebuah citra ke ukuran yang seragam, misalnya 224x224 piksel, untuk memenuhi persyaratan input model.

Tensor

Struktur data inti yang digunakan dalam *deep learning*; merupakan array multidimensi yang digunakan untuk merepresentasikan data seperti citra (misalnya, dimensi 224x224x3).

TensorFlow

Sebuah *framework open-source* yang dikembangkan oleh Google untuk *machine learning* dan *deep learning*, digunakan untuk membangun dan melatih model CNN.

True Negative (TN)

Hasil prediksi yang benar di mana model memprediksi kondisi sebagai negatif (misalnya, 'sehat') dan kondisi sebenarnya memang negatif.

True Positive (TP)

Hasil prediksi yang benar di mana model memprediksi kondisi sebagai positif (misalnya, 'sakit') dan kondisi sebenarnya memang positif.

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Sebagai salah satu komoditas perkebunan bernilai tinggi, tanaman merica banyak dibudidayakan di Indonesia. Komoditas ini merupakan bagian dari sektor pertanian, yang menjadi pilar fundamental bagi ekonomi nasional, baik sebagai penyedia bahan baku industri maupun sumber pangan (Yazid et al., 2023). Namun, keberlangsungan produksi merica sangat dipengaruhi oleh faktor-faktor dan kondisi kesehatan tanaman, khususnya serangan penyakit yang dapat menurunkan hasil panen secara drastis. Oleh karena itu, peningkatan produktivitas dan kualitas tanaman merica menjadi perhatian utama (Sah et al., 2025).

Salah satu kendala serius dalam budidaya merica adalah serangan penyakit tanaman, khususnya yang menyerang daun. (Manohara et al., 2020) Tiga di antara penyakit utama yang sering menyerang pada daun tanaman merica adalah *Phytophthora capsici* (busuk pangkal batang), penyakit kuning, dan *Colletotrichum gloeosporioides* (bercak daun). Penyakit-penyakit ini dapat menyebar cepat melalui daun yang terinfeksi, serta dipicu oleh kondisi lingkungan yang buruk (Denaneera Waardhana & Sarianti, 2024). Identifikasi penyakit yang tidak akurat dapat menyebabkan kerusakan permanen, penurunan hasil panen secara drastis, bahkan kematian tanaman.

Saat ini, identifikasi penyakit pada tanaman merica masih dilakukan secara manual melalui tahapan pengamatan langsung oleh petani (Anam Alidrus et al., 2021). Cara ini memiliki beberapa kekurangan diantaranya membutuhkan keahlian khusus, waktu yang cukup lama, serta berisiko tinggi terhadap kesalahan manusia (*human error*). Selain itu, pada kondisi skala kebun yang luas, pendekatan manual menjadi tidak efisien dan mahal secara operasional. Oleh karena itu, diperlukan solusi berbasis

teknologi untuk membantu proses identifikasi penyakit tanaman merica secara akurat dan konsisten.

Kemajuan teknologi informasi dan kecerdasan buatan (*Artificial Intelligence/AI*) menjadi peluang baru untuk mengatasi keterbatasan tersebut. Salah satu pendekatan yang cocok digunakan adalah dengan CNN, yang merupakan bagian dari teknik *deep learning* (Julyani, 2025). Di mana CNN menjadi salah satu algoritma paling efektif untuk identifikasi penyakit tanaman melalui citra daun. CNN dikenal unggul dalam klasifikasi dan analisis pola visual pada gambar. CNN dapat secara otomatis mengenali ciri-ciri visual pada daun tanaman yang terinfeksi penyakit hanya dari gambar, sehingga menghasilkan prediksi dengan tingkat akurasi tinggi (Rijal et al., 2024) .

Namun, untuk memastikan model dapat bekerja secara efisien dan berpotensi diimplementasikan pada perangkat dengan sumber daya terbatas, dibutuhkan arsitektur CNN yang ringan namun tetap andal. Oleh karena itu, penelitian ini mengadopsi MobileNet, arsitektur CNN yang dirancang efisien untuk tugas seperti deteksi objek, klasifikasi gambar, dan geo-lokalisasi (M. FARHAN-FST, 2024). Dengan desain ringan dan kompleksitas rendah, MobileNet tetap menjaga akurasi yang baik. MobileNetV2, sebagai versi lanjutan dari MobileNet, menawarkan performa lebih baik dengan jumlah parameter pelatihan yang lebih sedikit, sehingga lebih hemat komputasi namun tetap akurat (Fitriatuzzahra et al., 2024).

Dalam penerapan model CNN, ketersediaan data latih yang cukup dan bervariasi sangat penting untuk mencapai akurasi tinggi. Karena pengumpulan citra daun secara manual terbatas dan sering tidak seragam pencahayaannya, digunakan teknik augmentasi data sebagai solusi. Teknik ini memperbanyak data secara buatan melalui transformasi seperti rotasi, flipping, zoom, dan penyesuaian *brightness*, tanpa mengubah label aslinya. Augmentasi membantu memperkaya variasi data, meningkatkan kemampuan generalisasi model, serta mengurangi risiko *overfitting* (Fadli Gunardi, 2023) .

Beberapa penelitian sebelumnya telah berhasil mengimplementasikan CNN untuk klasifikasi penyakit pada tanaman padi (Jinan & Hayadi, 2022), Apel (Pamungkas & Suhendar, 2024), Tomat (Kotta et al., 2022), dan Selada (Rahman & Pambudi (2023). Namun, penggunaan CNN khusus untuk tanaman merica masih terbatas. Deteksi penyakit pada tanaman rempah khususnya merica sangat penting karena bernilai ekonomi tinggi dan rentan terhadap kerusakan kualitas akibat infeksi. Oleh karena itu, penelitian ini berfokus pada penerapan sistem deteksi penyakit tanaman merica berbasis citra daun menggunakan algoritma CNN.

Dengan memanfaatkan pengolahan citra digital berbasis CNN, petani dapat mengidentifikasi kondisi tanaman secara cepat dan akurat tanpa memerlukan keahlian khusus. Penelitian ini diharapkan menjadi acuan awal dalam penerapan metode deteksi penyakit tanaman merica yang efisien, serta melanjutkan studi terdahulu yang mengintegrasikan CNN dalam bidang pertanian.

B. RUMUSAN MASALAH

1. Bagaimana cara menerapkan model CNN dengan arsitektur MobileNet untuk mendeteksi penyakit pada daun tanaman merica ?
2. Bagaimana perbandingan performa model CNN dengan dan tanpa penerapan teknik augmentasi data?

C. TUJUAN PENELITIAN

1. Menerapkan model CNN menggunakan arsitektur MobileNet untuk mendeteksi penyakit pada daun tanaman merica.
2. Untuk mengetahui perbandingan performa model CNN yang diuji dengan dan tanpa penerapan teknik augmentasi data.

D. MANFAAT PENELITIAN

1. Bagi perkembangan ilmu pengetahuan
Penelitian ini memberikan kontribusi dalam bidang pengolahan citra digital dan kecerdasan buatan, khususnya dalam penerapan algoritma CNN untuk klasifikasi

citra daun tanaman merica. Penelitian ini tidak hanya menerapkan CNN, tetapi juga memberikan bukti kuantitatif mengenai efektivitas teknik augmentasi data pada studi kasus penyakit tanaman merica. Hasil perbandingan ini dapat menjadi referensi penting bagi peneliti lain dalam memutuskan strategi persiapan data untuk masalah serupa.

2. Bagi petani

Hasil dari model klasifikasi ini dapat membantu petani, penyuluh pertanian, dan lembaga terkait dalam melakukan identifikasi penyakit tanaman merica secara lebih cepat, objektif, dan efisien.

3. Bagi peneliti

Penelitian ini memberikan pengalaman praktis dalam implementasi model deep learning untuk klasifikasi citra, serta memperluas wawasan peneliti dalam penerapan teknologi AI dalam sektor pertanian.

E. RUANG LINGKUP PENELITIAN

1. Penelitian ini berfokus pada penerapan model CNN arsitektur MobileNet untuk deteksi dan klasifikasi penyakit pada daun merica. Deteksi penyakit dalam penelitian ini didefinisikan sebagai kemampuan sistem untuk mengidentifikasi keberadaan penyakit (apakah daun sehat atau sakit serta jenis penyakitnya) berdasarkan bukti visual yang muncul, tanpa bertujuan untuk mengklasifikasikan tingkat keparahan (stadium awal atau parah) sebagai kelas keluaran terpisah. Cakupan penelitian ini terbatas pada pengembangan model, tanpa mencakup pembuatan aplikasi atau platform berbasis web.
2. Penelitian ini dibatasi pada penggunaan dataset citra daun tanaman merica yang tersedia di perkebunan merica yang berada Desa Matano, Kecamatan Nuha, Kabupaten Luwu Timur, Sulawesi Selatan dan memiliki anotasi label penyakit. Citra daun digunakan sebagai input tunggal, tanpa mempertimbangkan data lain seperti kondisi tanah, batang, atau lingkungan sekitar tanaman.

3. Penelitian ini akan menghasilkan dua model CNN dengan arsitektur MobileNet yang identik. Model pertama akan dilatih menggunakan dataset citra asli, sedangkan model kedua akan dilatih menggunakan dataset yang telah diperbanyak melalui teknik augmentasi. Evaluasi akhir akan berfokus pada perbandingan performa antara kedua model tersebut untuk menentukan pengaruh dari teknik augmentasi.
4. Hasil akhirnya akan menampilkan 5 kelas (1 kelas Bukan daun merica, 1 kelas daun sehat dan 3 kelas penyakit: busuk pangkal batang, penyakit kuning, dan bercak daun) yang akan menampilkan visual citra daun, serta metrik evaluasi model seperti akurasi, presisi, recall, dan F1-score. Sistem ini tidak mencakup tindakan lanjutan atau rekomendasi perawatan.

F. SISTEMATIKA PENULISAN

Sistem penulisan yang digunakan dalam penelitian ini adalah:

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian, dan sistematika penulisan yang digunakan dalam menyusun laporan penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini memuat teori-teori dan penelitian terdahulu yang relevan sebagai dasar dalam menerapkan sistem deteksi. Di dalamnya juga dibahas konsep tanaman merica, penyakit tanaman merica, pengertian dan pengolahan citra digital, *deep learning*, Teknik Augmentasi Data, algoritma *Convolutional Neural Network* (CNN), arsitektur MobileNet, *Artificial Intelligence*/AI serta kerangka berpikir penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini memuat tentang metode yang digunakan dalam penelitian, Tempat dan waktu penelitian, Alat dan bahan, Perancangan sistem, Teknik pengujian sistem, dan Teknik analisis data.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi arsitektur MobileNet dan pengujian kinerja dua model (dengan dan tanpa augmentasi) melalui metrik evaluasi, yang diakhiri dengan pembahasan mendalam mengenai perbandingan performa, dampak augmentasi data, dan efektivitas arsitektur yang digunakan.

BAB V PENUTUP

Bab ini memuat rangkuman kesimpulan yang diperoleh dari keseluruhan hasil penelitian sebagai jawaban atas rumusan masalah, serta memuat saran yang dapat dijadikan acuan dalam pengembangan penelitian di masa mendatang.



BAB II

TINJAUAN PUSTAKA

A. LANDASAN TEORI

1. Tanaman Merica

Merica, yang dikenal juga sebagai lada atau sahang dengan nama ilmiah *Piper nigrum*, adalah tanaman yang kaya akan senyawa kimia seperti minyak atsiri, minyak lemak, dan pati. Rasanya agak pahit, pedas, hangat, dan memiliki sifat penurun panas. Tanaman ini telah dikenal dan dimanfaatkan sejak ribuan tahun silam.

Sebagai komoditas ekspor, lada memegang peranan penting dalam perekonomian Indonesia. Lada berkontribusi sebagai sumber penyedia lapangan kerja, bahan baku industri, serta untuk konsumsi langsung. Lada mengandung berbagai senyawa aktif, termasuk minyak atsiri dan minyak lemak (ST NURJANNAH MJ, 2021).

Dibandingkan rempah-rempah lainnya, lada memiliki peran yang sangat vital dan sulit digantikan. Tanaman *Piper nigrum L.*, ini sudah sangat familiar di Indonesia, tidak hanya sebagai bumbu masakan tetapi juga sebagai komoditas bernilai ekonomi tinggi.

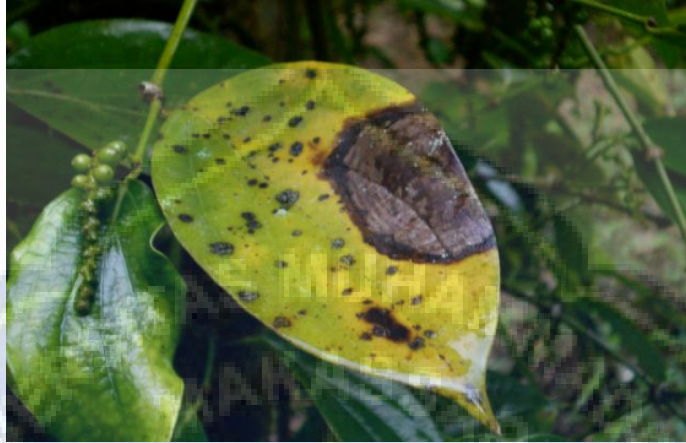
Lada tersebar luas di berbagai daerah di Indonesia, sehingga memiliki banyak sebutan lokal, seperti pada bagian (Jawa dan Sulawesi) dengan sebutan merica, pedes untuk bagian (Sunda), s'ang untuk (Madura), dan sahang untuk (Bangka Belitung).

2. Penyakit Tanaman Merica

Tanaman merica (*Piper nigrum*) merupakan salah satu komoditas perkebunan yang memiliki nilai ekonomi tinggi di Indonesia. Namun, tanaman ini rentan terhadap serangan berbagai penyakit yang dapat menurunkan produktivitas dan kualitas hasil panen. Beberapa penyakit utama yang sering menyerang tanaman merica antara lain:

- a. Busuk Pangkal Batang: Penyakit ini menyerang pangkal batang dan daun, diawali dengan munculnya bintik-bintik kecil berwarna hitam dan coklat pada permukaan daun. Bintik ini kemudian membesar menjadi bercak hitam di bagian tengah atau tepi daun. Seiring berjalannya waktu, bercak tersebut meluas dengan pinggiran yang tidak rata seperti renda, yang akan tampak lebih jelas jika daun dilihat dengan

latar belakang cahaya. Bercak ini dapat menyebar ke daun lain pada tanaman yang sama. Kerusakan pada sel-sel daun ini menyebabkan kematian jaringan (nekrosis), yang pada akhirnya berujung pada kerontokan daun (Lestari et al., 2023).



Gambar 1. Penyakit Busuk Pangkal Batang
(erlanardianarismansyah.blogspot.com)

- b. Penyakit kuning: Gejala diawali dengan perubahan warna daun menjadi kuning. Pada fase awal, daun yang terinfeksi tampak kaku dan menggantung tegak lurus, kemudian secara bertahap mulai melengkung ke arah batang. Daun menjadi sangat mudah patah dan akhirnya rontok dengan cepat (Sonya et al., 2023).



Gambar 2. Penyakit Kuning (Ir.Tri Juni Sasonko, 2023)

- c. Bercak Daun: Penyakit ini menimbulkan bercak daun dengan ciri khas bercak cokelat atau hitam yang semakin lama membuat daun menjadi kuning meluas dan menyebabkan daun gugur prematur (Sah et al., 2025)



Gambar 3. Penyakit Bercak Daun (ditjenbun, 2021)

3. Pengertian Citra Digital

Citra digital didefinisikan sebagai representasi wujud visual dari objek dunia nyata yang telah dikonversi ke dalam format digital agar dapat diolah oleh komputer. Gambar digital ini dibentuk oleh susunan elemen-elemen kecil bernama piksel dalam sebuah matriks baris dan kolom. Masing-masing piksel tersebut mempunyai nilai angka yang merepresentasikan intensitas cahaya atau warna pada titik spesifik. Sumber citra digital bisa berasal dari beragam perangkat, mulai dari kamera digital, alat pemindai (scanner), hingga simulasi yang dihasilkan oleh komputer. Oleh karena itu, dapat disimpulkan bahwa citra digital merupakan sebuah representasi visual berformat digital yang tersusun dari sekumpulan piksel dengan nilai-nilai numerik (R Dijaya, 2023).

4. Pengolahan Citra Digital

Pengolahan citra digital adalah proses mengubah citra asli menjadi bentuk yang lebih sesuai untuk analisis. Dalam penelitian deteksi penyakit tanaman, citra daun diproses melalui beberapa tahapan:

- a. **Resize:** Menyeragamkan ukuran citra dengan 224×224 piksel untuk memudahkan pemrosesan lebih lanjut.

- b. Normalisasi: Mengubah nilai piksel ke rentang tertentu $[0,1]$ untuk meningkatkan stabilitas numerik selama pelatihan model.
- c. Augmentasi: Teknik seperti rotasi, flip, dan penyesuaian kecerahan digunakan untuk memperbanyak data pelatihan secara buatan, sehingga meningkatkan generalisasi model,

Pengolahan citra digital memungkinkan ekstraksi fitur-fitur penting dari gambar daun, seperti tekstur, warna, dan pola kerusakan, yang menjadi dasar untuk klasifikasi penyakit.

5. *Deep Learning* (DL)

Deep Learning (DL) merupakan sebuah sub-bidang vital dalam *Machine Learning* yang memanfaatkan jaringan saraf tiruan berlapis (*multi-layered neural networks*) untuk mengolah data secara bertingkat. Ciri khas teknologi ini adalah kemampuannya untuk secara otomatis mengekstrak fitur-fitur rumit dari data mentah. Proses ini terjadi melalui serangkaian transformasi non-linear yang diorganisir dalam beberapa lapisan, memungkinkan sistem untuk belajar representasi data dari level yang sederhana hingga ke level yang sangat abstrak.

Dalam proses pembelajarannya, *deep learning* mengadopsi dua strategi utama: eksplorasi dan eksploitasi. Strategi eksploitasi digunakan untuk mendalami dan mengoptimalkan fitur-fitur spesifik yang sudah teridentifikasi dalam data. Sementara itu, strategi eksplorasi berfokus pada penemuan variasi fitur baru yang lebih luas. Perpaduan harmonis antara kedua strategi ini memungkinkan model *deep learning* untuk membangun representasi fitur yang kaya dan efektif untuk beragam tugas.

Ada tiga pilar utama yang mendorong kemajuan pesat *deep learning* dalam beberapa tahun terakhir. Pertama adalah ketersediaan volume data yang masif (*Big Data*). Kedua, peningkatan signifikan dalam kekuatan komputasi, khususnya dengan adopsi unit pemrosesan grafis (GPU) untuk komputasi paralel. Ketiga, inovasi dan penyempurnaan berkelanjutan pada algoritma jaringan saraf itu sendiri. Arsitektur

berlapis-lapis yang menjadi fondasi *deep learning* umumnya lebih dari tiga lapisan sangat andal dalam memodelkan hubungan non-linear yang kompleks, menjadikannya unggul untuk tugas-tugas seperti klasifikasi, prediksi, dan pengenalan pola.

Kesuksesan *deep learning* dalam menyelesaikan berbagai persoalan kompleks sangat bergantung pada kemampuannya melakukan ekstraksi fitur secara otomatis, sebuah metode yang jauh melampaui teknik-teknik konvensional. Didukung oleh infrastruktur komputasi modern dan melimpahnya data, kemampuan inilah yang memposisikan *deep learning* sebagai motor penggerak utama dalam evolusi kecerdasan buatan masa kini (Soebroto, 2021).

6. Teknik Augmentasi Data

Augmentasi data merupakan serangkaian teknik yang digunakan untuk meningkatkan jumlah dan keragaman data latih secara artifisial. Dalam konteks visi komputer dan pembelajaran mendalam (*deep learning*), augmentasi data menjadi strategi penting, terutama ketika jumlah dataset yang tersedia terbatas. Dengan menerapkan berbagai transformasi pada gambar asli, model dapat mempelajari fitur-fitur yang lebih kuat dan invarian, sehingga meningkatkan kemampuannya untuk melakukan generalisasi pada data baru yang belum pernah dilihat sebelumnya dan mengurangi risiko overfitting (M. FARHAN-FST, 2024).

Dalam implementasi model CNN untuk klasifikasi citra daun, beberapa teknik augmentasi yang umum digunakan meliputi:

- a. Rotasi (*Rotation*): Memutar gambar daun dalam rentang derajat tertentu, misalnya antara -30 hingga 30 derajat. Teknik ini membantu model untuk mengenali daun terlepas dari orientasinya. Hal ini penting karena posisi daun di dunia nyata sangat bervariasi.
- b. *Flipping* (Pembalikan): Membalik gambar secara horizontal atau vertikal. Pembalikan horizontal adalah teknik yang sangat umum dan efektif karena citra daun yang dibalik secara horizontal tetap merepresentasikan objek yang valid.

- c. *Zooming* (Pembesaran): Memperbesar atau memperkecil sebagian area pada gambar secara acak. Teknik ini mensimulasikan pengambilan gambar dari jarak yang berbeda-beda, melatih model untuk mengenali gejala penyakit baik dari jarak dekat maupun jauh.
- d. Penyesuaian Kecerahan (*Brightness Adjustment*): Mengubah tingkat kecerahan gambar. Variasi ini mensimulasikan kondisi pencahayaan yang berbeda di lapangan, seperti saat cuaca cerah, berawan, atau di bawah bayangan, sehingga membuat model lebih tangguh terhadap perubahan iluminasi.

Dengan menerapkan kombinasi dari teknik-teknik ini, dataset yang digunakan untuk melatih model CNN akan menjadi jauh lebih beragam. Setiap gambar asli dapat menghasilkan beberapa variasi gambar baru, yang semuanya tetap mempertahankan label kelas yang sama. Paparan terhadap variasi yang lebih luas ini akan meningkatkan ketangguhan (*robustness*) dan akurasi model dalam mendeteksi penyakit pada tanaman merica dalam kondisi dunia nyata yang beragam.

7. CNN

CNN merupakan salah satu jenis jaringan saraf tiruan yang strukturnya mirip dengan jaringan saraf standar, terdiri dari berbagai neuron atau node yang saling terhubung dalam susunan graf asiklik, artinya tidak ada siklus atau loop dalam aliran informasinya. Yang membedakan CNN adalah adanya lapisan tersembunyi dengan koneksi yang tidak lengkap ke lapisan sebelumnya, di mana setiap neuron hanya terhubung ke sebagian kecil neuron di lapisan sebelumnya. Konektivitas yang terbatas ini memungkinkan CNN untuk secara otomatis mempelajari fitur-fitur penting dari data.

Secara arsitektur, CNN bekerja dengan mengekstraksi fitur secara bertahap, dimulai dari fitur yang sederhana hingga yang lebih kompleks. Pada lapisan awal, filter biasanya dilatih untuk mengenali pola dasar seperti tepian objek atau perubahan warna. Lapisan berikutnya mulai mengidentifikasi bentuk yang lebih rumit dengan menggabungkan informasi dari lapisan sebelumnya. Semakin dalam lapisannya, CNN

mampu mempelajari bagian-bagian yang lebih spesifik dari objek, baik yang terlihat sebagian maupun utuh. Akhirnya, lapisan terakhir bertugas mengintegrasikan semua informasi yang telah dipelajari untuk mengenali objek secara keseluruhan. Dengan pendekatan hierarkis ini, CNN menjadi sangat efektif dalam tugas-tugas pengenalan pola, terutama dalam pemrosesan gambar (Soebroto, 2021).

Dalam penelitian ini, CNN sangat cocok untuk deteksi penyakit daun merica karena kemampuannya menganalisis detail visual, termasuk pola penyakit yang sulit dikenali manual. CNN juga tahan terhadap variasi gambar seperti pencahayaan, sudut, dan ukuran daun. Dengan augmentasi data dan normalisasi, model ini menjadi lebih andal terhadap gangguan kecil.

Prosesnya melibatkan pra-pemrosesan citra daun, ekstraksi fitur melalui lapisan konvolusi dan pooling, lalu klasifikasi oleh lapisan fully connected untuk menentukan kesehatan daun dan jenis penyakitnya. CNN belajar otomatis dengan menyesuaikan parameter filternya, meningkatkan akurasi deteksi seiring waktu. Dengan arsitektur yang tepat, CNN mencapai akurasi tinggi, menjadikannya solusi efektif untuk sistem deteksi penyakit tanaman berbasis citra. Untuk memenuhi kebutuhan akan model yang efisien, berbagai turunan arsitektur CNN telah dikembangkan, salah satunya yang paling populer adalah MobileNet.

8. Arsitektur MobileNet

MobileNet adalah arsitektur CNN yang dikembangkan oleh Google, dirancang khusus untuk aplikasi pada perangkat seluler dan perangkat lain dengan keterbatasan komputasi. Keunggulan utama MobileNet terletak pada efisiensinya yang tinggi, yang dicapai melalui penggunaan *Depthwise Separable Convolutions*. Tidak seperti konvolusi standar, *Depthwise Separable Convolutions* memecah proses menjadi dua langkah:

- a. *Depthwise Convolution*: Melakukan konvolusi pada setiap channel input secara terpisah. Langkah ini bertugas untuk menyaring fitur spasial.

- b. *Pointwise Convolution*: Menggunakan konvolusi 1x1 untuk menggabungkan hasil dari *depthwise convolution*. Langkah ini bertugas untuk menciptakan fitur-fitur baru dari kombinasi yang ada.

Pendekatan ini secara drastis mengurangi jumlah parameter dan biaya komputasi dibandingkan dengan arsitektur CNN konvensional seperti VGG, tanpa penurunan akurasi yang signifikan. Hal ini membuat MobileNet sangat cocok untuk tugas-tugas klasifikasi citra pada perangkat keras dengan spesifikasi terbatas, seperti laptop yang digunakan dalam penelitian ini, serta membuka potensi implementasi di lapangan pada masa depan (Puspita Putri et al., 2025).

9. *Artificial Intelligence/AI*

AI atau Kecerdasan Buatan, merupakan cabang ilmu komputer yang bertujuan untuk menciptakan mesin dengan kemampuan meniru kecerdasan manusia. Dengan memanfaatkan algoritma dan model matematika, AI memungkinkan komputer untuk belajar dari data, mengenali pola, dan membuat keputusan cerdas secara otonom.

Meskipun konsepnya sudah ada sejak lama dalam fiksi dan mitos, sejarah pengembangan AI secara ilmiah dimulai pada awal abad ke-20. Bidang ini mencakup berbagai konsep penting seperti *machine learning* (pembelajaran mesin), *neural networks* (jaringan saraf tiruan), dan *natural language processing* (pemrosesan bahasa alami). Dampak kemajuan AI sudah sangat terasa di berbagai sektor, mulai dari teknologi pengenalan suara dan wajah, mobil swakemudi, hingga terobosan dalam dunia pengobatan dan banyak bidang lainnya (Eriana et al., 2023).

B. PENELITIAN TERKAIT

Penelitian terkait ialah bagian yang membahas dan menganalisis penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan masalah yang diteliti. Berbagai studi telah dilakukan mengenai deteksi penyakit tanaman menggunakan teknik pengolahan citra, *deep learning* dengan menggunakan arsitektur MobileNet. (Munir et al., 2024) dalam penelitian berjudul "Penerapan Arsitektur Mobilenet Dalam

Cnn Pada Klasifikasi Penyakit Daun Cabai" mengembangkan sebuah sistem untuk mengklasifikasi tiga jenis kondisi daun cabai (sehat, keriting, dan bercak daun). Menggunakan dataset sebanyak 100 citra daun , model dengan arsitektur MobileNet ini berhasil mencapai akurasi sebesar 97.22% pada pelatihan data dengan 10 epoch. Penelitian ini menyimpulkan bahwa penambahan jumlah epoch dapat meningkatkan tingkat akurasi model secara signifikan.

Pada tanaman kentang,(Teresia Ompusunggu, 2022) melakukan penelitian berjudul "Klasifikasi Penyakit Tanaman Pada Daun Kentang Dengan Metode *Convolutional Neural Network* Arsitektur Mobilenet". Penelitian ini mengklasifikasikan tiga kondisi daun kentang: early blight, late blight, dan sehat. Dengan dataset sebanyak 3.000 citra dan melalui empat skenario pengujian hyperparameter , hasil terbaik mencapai akurasi 97,90% dan loss 0,0390. Performa optimal ini didapatkan menggunakan optimizer RMSprop, learning rate 0.0001, 100 epoch, dan batch size 32.

(Puspita Putri et al., 2025) juga meneliti daun cabai dalam "Klasifikasi Penyakit Daun Cabai Menggunakan Model MobileNet". Penelitian ini menggunakan metode CRISP-DM dan arsitektur MobileNet untuk mengklasifikasi enam jenis penyakit dari dataset berisi 1.286 gambar. Dengan 40 epoch, model mencapai akurasi training 94.18% dan akurasi pada data uji sebesar 84.84%. Keunggulan penelitian ini adalah keberhasilannya dalam menerapkan model ke dalam sebuah aplikasi mobile, menunjukkan bahwa model tidak mengalami overfitting yang parah dan memiliki generalisasi yang baik.

(Fuadi et al., 2022) Dalam penelitian "Perbandingan Arsitektur Mobilenet dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang" membandingkan dua arsitektur ringan untuk deteksi penyakit *Early Blight* dan *Late Blight*. Mereka menggunakan berbagai skema pembagian data dan menemukan bahwa arsitektur NASNetMobile memberikan hasil yang lebih unggul dibandingkan MobileNet. Dengan perbandingan data latih dan uji 90:10, NASNetMobile mencapai akurasi 90.96%, precision 90.86%, dan recall 91.11%.

Pada tanaman singkong, (Arafat et al., 2025) melakukan penelitian "Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital". Penelitian ini mengklasifikasikan lima kategori penyakit daun singkong menggunakan data yang dibagi 80% untuk data latih dan 20% untuk data uji. Hasilnya menunjukkan bahwa dengan 30 epoch, model mencapai akurasi training sebesar 88.28% dan akurasi pada data uji sebesar 84%. Studi ini juga menegaskan bahwa kinerja model meningkat secara konsisten seiring bertambahnya jumlah epoch tanpa mengalami overfitting yang signifikan.

Berbagai penelitian tersebut menunjukkan bahwa arsitektur CNN, khususnya MobileNet, sangat efektif untuk klasifikasi penyakit pada daun cabai, kentang, dan singkong dengan tingkat akurasi yang tinggi. Tantangan yang umum dihadapi adalah kebutuhan akan dataset yang representatif dan penanganan data yang tidak seimbang. Tren utama yang terlihat adalah optimasi hyperparameter seperti jumlah epoch, learning rate, dan optimizer untuk mencapai performa terbaik, serta keberhasilan implementasi model ke dalam aplikasi mobile yang praktis untuk membantu petani.

Tabel 1. Penelitian Terkait

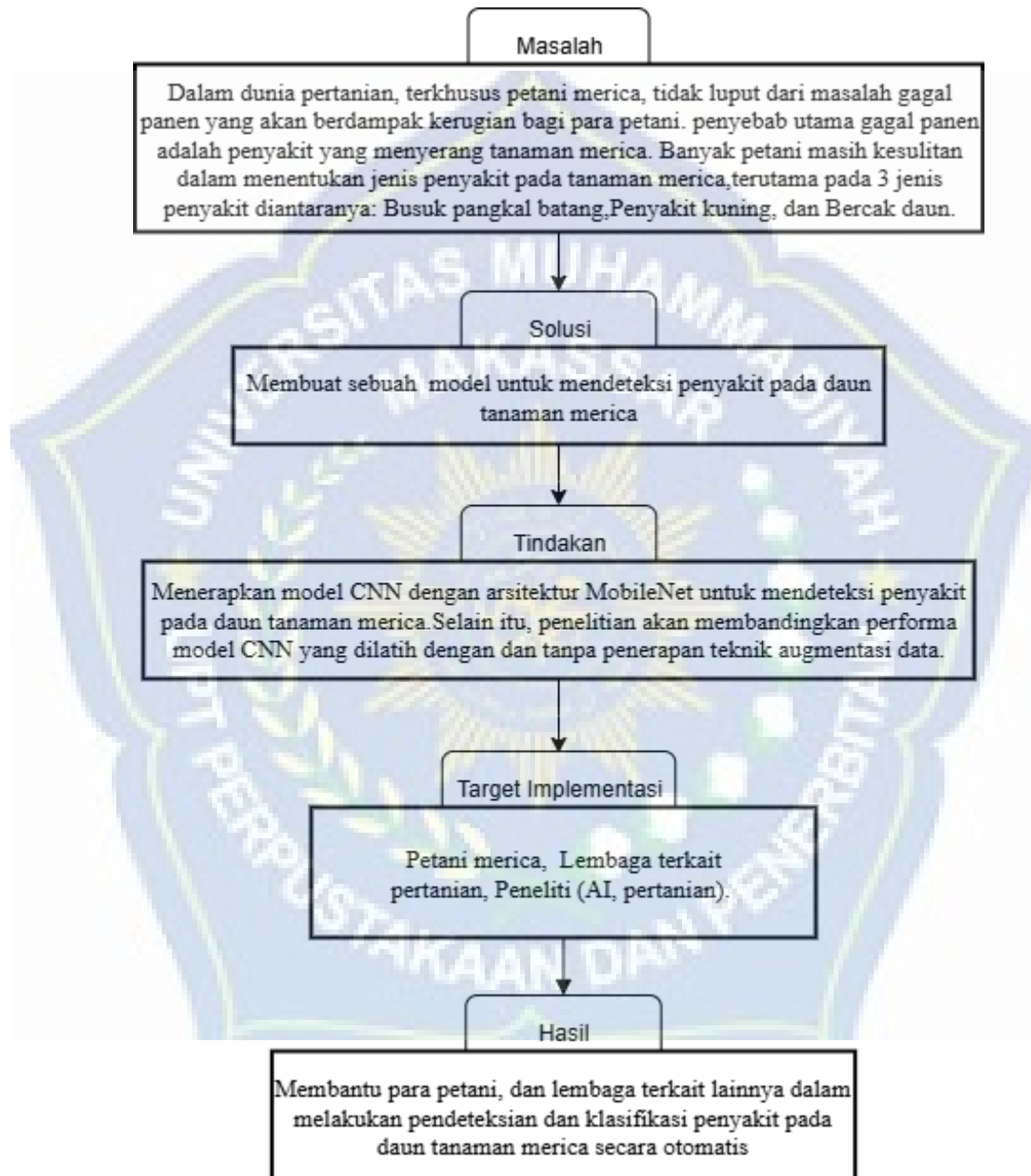
NO	Peneliti (Nama, Penelitian)	Judul	Kontribusi
1	(Munir et al., 2024; Teresia Ompusunggu, 2022) Penerapan Arsitektur Mobilenet Dalam Cnn Pada Klasifikasi Penyakit Daun Cabai.		Mengembangkan sistem klasifikasi untuk tiga kondisi daun cabai (sehat, keriting, dan bercak daun) menggunakan CNN dengan arsitektur MobileNet. Model ini berhasil mencapai akurasi 97,22% pada pelatihan data dengan 10 epoch, menggunakan dataset 100 citra daun
2	(Teresia Ompusunggu, 2022) Klasifikasi Penyakit Tanaman Pada Daun Kentang Dengan Metode Convolutional Neural Network Arsitektur Mobilenet.		Mengklasifikasikan tiga kondisi daun kentang (early blight, late blight, dan sehat) menggunakan arsitektur CNN MobileNet. Penelitian ini mencapai akurasi 97,90% setelah melakukan optimasi hyperparameter, dengan menggunakan optimizer RMSprop, learning rate 0,0001, 100 epoch, dan batch size 32 pada dataset 3.000 citra

- 3 (Puspita Putri et al., 2025) Klasifikasi Penyakit Daun Cabai Menggunakan Model MobileNet. Menerapkan metode CRISP-DM dengan arsitektur MobileNet untuk mengklasifikasi enam jenis penyakit pada daun cabai. Model yang dihasilkan mencapai akurasi 84,84% pada data uji dan berhasil diimplementasikan ke dalam aplikasi mobile berbasis Android

- | | | |
|---|---|--|
| 4 | (Fuadi et al., 2022) Perbandingan Arsitektur Mobilenet dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang. | Membandingkan kinerja arsitektur MobileNet dan NASNetMobile untuk klasifikasi penyakit daun kentang (Early Blight dan Late Blight). Penelitian menemukan bahwa NASNetMobile memberikan performa lebih baik, dengan akurasi mencapai 90,96% pada pembagian data 90:10 |
| 5 | (Arafat et al., 2025) Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital. | Merancang sistem untuk mendiagnosis lima kategori penyakit pada daun singkong menggunakan arsitektur MobileNet-CNN. Dengan pelatihan selama 30 epoch dan rasio data latih-uji 80%:20%, model mencapai akurasi training sebesar 88% dan akurasi pada data uji sebesar 84% |
| 6 | (Sah et al., 2025) Pengembangan Model Klasifikasi Citra Penyakit Daun Lada Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ). | Mengembangkan model klasifikasi penyakit daun lada (empat kelas: Daun Sehat, Leaf Blight, Yellow Mottle Virus, Anthracnose) menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ) dengan ekstraksi fitur warna (Mean Color pada ruang CIELAB) dan tekstur (GLCM). Model mencapai akurasi keseluruhan 90,83%. |

C. KERANGKA PIKIR

Kerangka pikir dalam penelitian ini dapat digambarkan seperti di bawah ini:



Gambar 4. Kerangka Pikir

BAB III

METODE PENELITIAN

A. TEMPAT DAN WAKTU PENELITIAN

1. Tempat Penelitian

Penelitian ini akan dilaksanakan di kebun merica yang terletak di Desa Matano, Kecamatan Nuha, Kabupaten Luwu Timur, Provinsi Sulawesi Selatan. Pemilihan lokasi ini cocok karena statusnya sebagai salah satu kawasan utama dalam budidaya tanaman merica di wilayah Sulawesi.

2. Waktu Penelitian

Pelaksanaan penelitian akan dimulai pada bulan Mei 2025 dan akan berlangsung hingga seluruh tahapan penelitian terselesaikan dengan baik.

B. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam penelitian ini diantaranya:

1. Kebutuhan Hardware (Perangkat Keras)

- a. Leptop Lenovo Ideapad Slim 1(AMD Ryzen 3 5300U with Radeon Graphics, RAM 8.00 GB, SSD 512 GB, OS Windows 11)
- b. *Smartphone Android* Readmi Not 13 (RAM 8 GB, Penyimpanan 256 GB, OS 2.0.4.0, Snapdragon 685)

2. Kebutuhan Software (Perangkat Lunak)

- a. Python (Bahasa pemrograman utama)
- b. TensorFlow (framework deep learning digunakan untuk mengimplementasikan dan melatih model CNN)
- c. Keras (API high-level yang berjalan di atas TensorFlow, digunakan untuk menyederhanakan implementasi dan pelatihan model CNN.
- d. Visual Studio Code (digunakan sebagai software utama)
- e. OpenCV untuk preprocessing citra (resize, normalisasi, augmentasi).
- f. Numpy dan Pandas untuk manipulasi array gambar (konversi citra ke tensor)

- g. Matplotlib dan Seaborn untuk visualisasi hasil.
- h. Scikit-learn untuk menghitung metrik klasifikasi (akurasi,presisi,recall,dan F1-score).

3. Kebutuhan Dataset

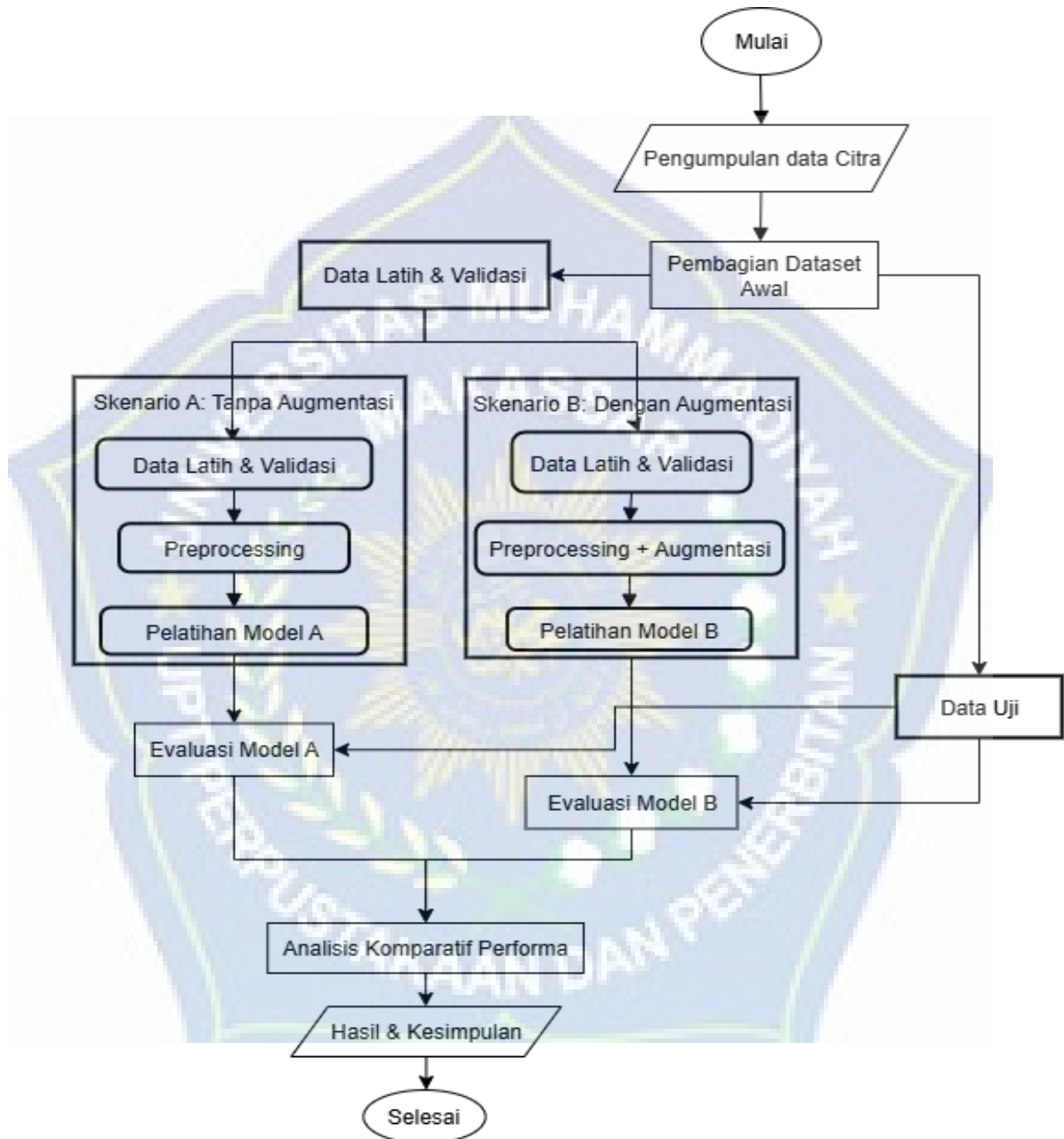
Data penelitian ini terdiri dari 750 citra daun merica yang diambil langsung dari lokasi budidaya menggunakan kamera digital. Dataset dibagi secara seimbang ke dalam 5 kelas, yaitu :

- a. [150] gambar daun sehat
- b. [150] gambar penyakit busuk pangkal batang
- c. [150] gambar penyakit kuning
- d. [150] gambar penyakit bercak daun
- e. [150] gambar bukan daun merica

Untuk membangun model yang andal, setiap kelas penyakit menggunakan campuran citra dengan tingkat gejala yang bervariasi, mulai dari yang ringan hingga parah. Strategi ini bertujuan untuk melatih model agar mampu mengenali penyakit dalam berbagai keadaan di lapangan, sehingga meningkatkan kemampuan generalisasinya.

C. PERANCANGAN SISTEM

Perancangan sistem menggunakan flowchart:



Gambar 5. Flowchart Sistem

Penjelasan gambar 5. Flowchart sistem:

1. Pengumpulan Data citra

Mengumpulkan gambar daun merica, baik yang sehat maupun yang terinfeksi penyakit (Busuk pangkal batang, Penyakit kuning, dan Bercak daun). Gambar diambil langsung dari lapangan menggunakan kamera digital/*smartphone*.

2. Pembagian Dataset Awal

Setelah data terkumpul, langsung bagi menjadi tiga set: Data Latih (70%), Data Validasi (20%), dan Data Uji (10%). Data Uji harus dipisahkan di awal dan tidak boleh di-augmentasi sama sekali agar bisa menjadi penguji yang adil bagi kedua model.

3. Skenario A: Tanpa Augmentasi (Grup Kontrol)

Model pertama dilatih dengan data asli tanpa penambahan variasi buatan. Tujuannya untuk menjadi pembanding dasar.

a. Preprocessing (Resize & Normalisasi)

- 1) Resize: Semua gambar diubah ukurannya menjadi 224x224 piksel.
- 2) Normalisasi: Nilai piksel diubah ke rentang [0, 1] agar memudahkan proses pelatihan dan mempercepat konvergensi model.

b. Pelatihan Model A

- 1.) Model CNN dengan arsitektur MobileNet dilatih dengan data latih hasil preprocessing.
- 2.) Tidak ada augmentasi.
- 3.) Model ini akan menunjukkan seberapa baik kinerja CNN dengan data asli saja.

4. Skenario B: Dengan Augmentasi (Grup Eksperimen)

Model kedua dilatih menggunakan data latih yang telah diaugmentasi untuk meningkatkan keragaman data dan ketahanan model.

a. Preprocessing + Augmentasi

- 1.) Resize dan normalisasi tetap dilakukan seperti pada Skenario A.
- 2.) Tambahan: Data Augmentasi, seperti Flip horizontal/vertikal, Rotasi, Zoom-in/out, *Brightness Adjustment*.

Tujuannya adalah membuat variasi buatan yang bisa memperkuat kemampuan generalisasi model terhadap data baru.

b. Pelatihan Model B

1) Model CNN dengan arsitektur MobileNet kedua dilatih dengan dataset yang sudah diaugmentasi.

2) Arsitektur model identik dengan Model A untuk menjaga validitas eksperimen (variabel bebas hanya data augmentasi).

5. Evaluasi Model A & Model B

Setelah kedua model selesai dilatih, dilakukan evaluasi menggunakan data uji yang belum pernah dilihat oleh model.

a. Data uji sama untuk kedua model.

b. Metrik yang digunakan:

1) Akurasi

2) Presisi

3) Recall

4) F1-score

c. Analisis dilakukan untuk menilai seberapa baik masing-masing model mengenali setiap kelas (jenis penyakit).

6. Analisis Komparatif Performa

Di sini dilakukan perbandingan langsung antara Model A dan Model B berdasarkan hasil evaluasi:

a. Apakah augmentasi data membuat model lebih akurat?

b. Confusion matrix dibandingkan untuk melihat jenis kesalahan yang dominan:

1) Apakah Model A lebih sering salah mendeteksi daun sakit sebagai sehat?

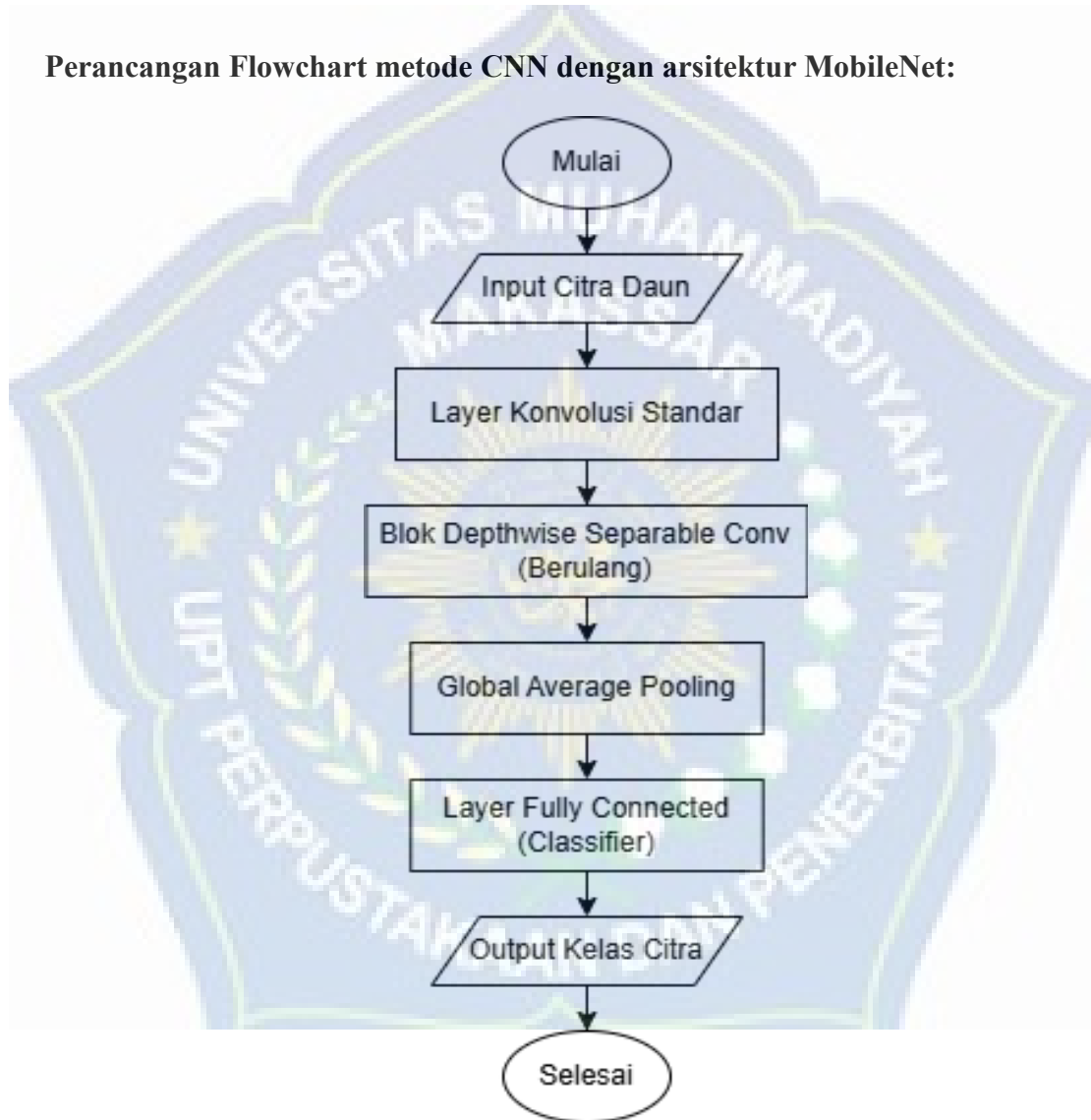
2) Apakah Model B lebih stabil mengenali semua kelas?

7. Hasil & Kesimpulan

Berdasarkan semua metrik evaluasi dan analisis komparatif:

- a. Ditentukan apakah augmentasi data secara signifikan meningkatkan performa model.
- b. Disimpulkan model mana yang paling efektif untuk mendeteksi penyakit daun merica.

Perancangan Flowchart metode CNN dengan arsitektur MobileNet:



Gambar 6. Flowchart CNN dengan arsitektur MobileNet

Penjelasan gambar 6. Flowchart CNN dengan arsitektur MobileNet:

1. Input Citra Daun

Pada tahap ini, model menerima data masukan berupa citra digital daun merica. Sesuai dengan rancangan, citra ini telah melalui tahap pra-pemrosesan di mana ukurannya diseragamkan menjadi 224x224 piksel. Nilai warna RGB (0-255) pada setiap piksel kemudian dinormalisasi ke dalam rentang [0, 1]. Input ini menjadi sebuah tensor berdimensi (224, 224, 3) yang siap diproses.

2. Layer Konvolusi Standar (*Convolution Layer*)

Setelah masuk ke dalam model, citra pertama kali akan melewati satu lapisan konvolusi standar. Tujuan dari lapisan awal ini adalah untuk mengekstraksi fitur-fitur visual paling dasar atau tingkat rendah, seperti deteksi tepi, sudut, dan perubahan gradasi warna pada citra daun. Lapisan ini berfungsi sebagai pintu masuk yang kuat sebelum fitur diolah lebih lanjut oleh blok-blok arsitektur MobileNet.

3. Blok *Depthwise Separable Conv* (Berulang)

Langkah selanjutnya adalah tahap ekstraksi fitur yang paling inti dalam arsitektur MobileNet. Blok Depthwise Separable Convolution bekerja dengan sangat efisien melalui dua langkah terpisah untuk mengolah informasi visual.

a. Tahap Pertama: Depthwise Convolution

Pada tahap ini, setiap channel warna (Merah, Hijau, Biru) dari citra input akan diolah secara terpisah menggunakan kernel khususnya masing-masing. Proses ini berfokus untuk menemukan pola spasial (seperti bentuk atau tekstur) di dalam setiap channel secara independen. Rumus matematis untuk operasi ini adalah sebagai berikut:

$$S_{(i,j)} = (K * I)_{(i,j)} = \sum_{i=1}^m \sum_{j=1}^n (I_{(i-m,j-n)} * K)_{(m,n)} \dots\dots\dots(1)$$

Dimana:

- 1.) $O_{DW}(i, j, k)$ adalah nilai output depthwise pada posisi (i, j) untuk channel k.
- 2.) I adalah tensor input *Global Average Pooling*
- 3.) K adalah tensor kernel

Sebagai contoh perhitungan manual untuk menghasilkan satu nilai pada output channel Merah (Depth_R.csv di posisi 0,0), prosesnya adalah perkalian elemen-per-elemen antara matriks input dan kernel-nya:

$$\begin{pmatrix} 10 & 40 & 70 \\ 15 & 45 & 75 \\ 20 & 50 & 80 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

Input Kernel

$$O_{DW}(i, j, k) = (10 \times 1) + (40 \times 0) + (70 \times -1) + (15 \times 1) + (45 \times 0) + (75 \times -1) + (20 \times 1) + (50 \times 0) + (80 \times -1) = -180$$

b. Tahap Kedua: Pointwise Convolution

Setelah fitur spasial diekstrak, Pointwise Convolution (menggunakan kernel berukuran 1x1) bertugas untuk menggabungkan informasi dari semua channel hasil depthwise (Depth_R, Depth_G, Depth_B). Tahap ini menciptakan representasi fitur baru yang lebih kaya. Rumus matematisnya adalah:

$$O_{PW}(i, j, l) = \sum_k O_{DW}(i, j, k) \cdot W(k, l) \quad \dots\dots\dots(2)$$

Dimana:

- a. $O_{PW}(i, j, l)$ adalah nilai output akhir pada posisi (i, j) untuk output channel l.
- b. $O_{DW}(i, j, k)$ adalah hasil dari Depthwise Convolution.
- c. $W(k, l)$ adalah bobot (weight) dari kernel pointwise.

Sebagai contoh perhitungan manual untuk menghasilkan satu nilai pada output Out_A.csv di posisi (0,0):

- a. Input (Hasil Depthwise): $O_{DW}(0,0,R) = -180, O_{DW}(0,0,G) = -30, O_{DW}(0,0,B) = 55$
- b. Bobot (Pointwise_W.csv untuk Out_A): $W_R = 0.5, W_G = 0.3, W_B = 0.2$
- c. Perhitungan: $O_{PW}(0,0,A) = (-180 \times 0.5) + (-30 \times 0.3) + (55 \times 0.2) = -88$

Label "Berulang" menandakan bahwa blok ini ditumpuk berkali-kali untuk membangun pemahaman fitur dari yang sederhana hingga yang sangat kompleks, seperti mengenali pola utuh dari suatu penyakit.

4. *Global Average Pooling (GAP).*

Setelah proses ekstraksi fitur selesai, hasilnya akan masuk ke lapisan Global Average Pooling (GAP). Lapisan ini berfungsi sebagai jembatan antara tahap ekstraksi fitur dan klasifikasi dengan cara menghitung nilai rata-rata dari setiap peta fitur, yang kemudian mengubahnya menjadi sebuah vektor fitur yang ringkas. Metode ini secara efektif mengurangi jumlah parameter model, yang sangat membantu dalam mencegah overfitting.

5. *Layer Fully Connected (Classifier)*

Vektor fitur yang ringkas dari GAP kemudian diteruskan ke Layer Fully Connected, yang berfungsi sebagai "otak" atau pengklasifikasi dalam model. Lapisan ini menghitung skor untuk setiap kelas (daun sehat, penyakit kuning, dll.) dan mengubahnya menjadi probabilitas menggunakan fungsi *softmax*. Kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi akhir.

6. Output kelas Citra

Tahap ini merupakan hasil akhir dari keseluruhan proses. Berdasarkan skor probabilitas tertinggi dari lapisan sebelumnya, sistem akan mengeluarkan sebuah label prediksi tunggal. Contohnya, output dapat berupa "Daun Sehat" atau "Terinfeksi Penyakit Kuning", yang kemudian akan dievaluasi untuk mengukur performa model.

D. TEKNIK PENGUJIAN SISTEM

Pengujian sistem dalam penelitian ini menggunakan *confusion matrix* dan evaluasi model yang difokuskan untuk membandingkan performa antara dua model (CNN). Tujuan utamanya adalah untuk mengevaluasi secara kuantitatif dampak dari penerapan teknik augmentasi data. Pengujian akan dilakukan dengan mengevaluasi performa

Model A (tanpa augmentasi) dan Model B (dengan augmentasi) menggunakan set data uji yang sama dan belum pernah digunakan selama pelatihan.

Komponen-komponen yang akan diuji meliputi:

1. Perbandingan Performa Klasifikasi
 - a. Kemampuan masing-masing model dalam membedakan daun sehat dari daun yang terinfeksi penyakit (Busuk pangkal batang, Penyakit kuning, dan bercak daun) akan diukur.
 - b. Pengujian menggunakan metrik evaluasi: akurasi, presisi, recall, dan F1-score untuk kedua model. Hasilnya akan disajikan dalam tabel perbandingan.
2. Analisis Stabilitas Model
 - a. Kedua model akan diuji dengan berbagai variasi input citra daun pada data uji untuk memastikan konsistensi performa.
 - b. Ketahanan masing-masing model terhadap noise atau variasi yang tidak ada dalam data latih akan dianalisis secara kualitatif.
3. Perbandingan Proses Pelatihan
 - a. Kurva akurasi dan loss dari proses pelatihan dan validasi kedua model akan divisualisasikan dan dibandingkan.
 - b. Tujuannya adalah untuk menganalisis apakah Model B menunjukkan tanda-tanda overfitting yang lebih rendah dibandingkan Model A.

Metodologi Pengujian:

1. Pengujian Performa
 - a. Model A dan Model B akan diuji menggunakan dataset uji yang sama.
 - b. Kinerja kedua model diukur dan dibandingkan menggunakan confusion matrix untuk setiap model
2. Pengujian Validasi

Penelitian menggunakan skema pembagian dataset 70% untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian. Pengujian akhir dilakukan pada 10% data uji yang sama untuk kedua model

Alat Pengujian:

1. Library Python (sklearn.metrics) untuk perhitungan metrik evaluasi
2. Tools visualisasi (Matplotlib, Seaborn) untuk menampilkan hasil pengujian
3. Dataset citra daun merica yang telah dilabeli

E. TEKNIK ANALISIS DATA

Teknik analisis data dalam penelitian ini dirancang untuk mengevaluasi dan membandingkan secara sistematis kinerja dari dua model CNN berbasis arsitektur MobileNet: Model A (dilatih tanpa augmentasi) dan Model B (dilatih dengan augmentasi). Analisis dilakukan secara kuantitatif untuk membuktikan pengaruh teknik augmentasi terhadap kemampuan model dalam mendeteksi dan klasifikasi penyakit daun tanaman merica.

1. Pengujian Model Menggunakan Data Uji

Setelah kedua model selesai dilatih, keduanya akan diuji menggunakan set data uji yang sama, yang belum pernah dilihat oleh kedua model sebelumnya. Tujuannya adalah untuk mengukur kemampuan generalisasi masing-masing model terhadap data baru secara adil. Data uji terdiri dari gambar daun merica dalam kondisi sehat dan terinfeksi penyakit (Busuk pangkal batang, Penyakit kuning, dan Bercak daun).

2. Evaluasi Kinerja Klasifikasi

Evaluasi performa model dilakukan dengan menghitung beberapa metrik klasifikasi, antara lain:

- a. Akurasi (*Accuracy*): Presentase total prediksi yang benar dari keseluruhan data uji. Dari semua daun yang diuji, berapa persen yang berhasil ditebak dengan benar oleh model.

Rumus:

$$Akurasi = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) \times 100\% \quad \dots\dots\dots(3)$$

Keterangan:

- 1) TP (*True Positive*): Prediksi positif yang benar.
- 2) TN (*True Negative*): Prediksi negatif yang benar.
- 3) FP (*False Positive*): Prediksi positif yang salah.
- 4) FN (*False Negative*): Prediksi negatif yang salah.

Penjelasan dalam penelitian sebagai berikut:

1.) *True Positif* (TP)

Daun merica yang benar benar sakit dan terdeteksi sebagai sakit oleh sistem

2.) *False Negatif* (FN)

Daun merica yang sakit, tetapi tidak terdeteksi oleh sistem.

3.) *False Positif* (FP)

Daun merica yang sehat tetapi salah terdeteksi sakit oleh sistem

4.) *True Negatif* (TN)

Daun merica yang sehat dan terdeteksi dengan benar sebagai sehat oleh sistem.

- b. Presisi (*Precision*): Mengukur seberapa tepat model dalam mengklasifikasikan citra sebagai positif (terinfeksi) dari semua prediksi positif yang dibuat.

Rumus:

$$Presisi = \frac{TP}{TP+FP} \quad \dots\dots\dots(4)$$

Artinya: Dari semua daun yang diprediksi oleh model sebagai 'Penyakit Kuning', berapa banyak yang benar-benar 'Penyakit Kuning'

- c. Recall (*Sensitivitas*): Mengukur seberapa baik model dalam menemukan semua data positif yang sebenarnya.

Rumus:

$$Recall = \frac{TP}{TP+FN} \quad \dots\dots\dots(5)$$

Artinya: Dari semua daun yang sebenarnya menderita 'Penyakit Kuning' di dataset, berapa banyak yang berhasil ditemukan oleh model

- d. F1-Score: Rata-rata harmonik antara presisi dan recall, cocok digunakan saat data tidak seimbang antar kelas.

Rumus:

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \dots\dots\dots(6)$$

Artinya: Seberapa baik keseimbangan antara kemampuan model untuk tidak salah menebak (Presisi) dan kemampuannya untuk tidak melewatkan kasus yang ada (Recall).

3. Analisis *Confusion Matrix*

Confusion matrix digunakan untuk memberikan informasi detail mengenai prediksi model, dengan memperlihatkan jumlah prediksi benar dan salah pada setiap kelas. Analisis ini membantu mengidentifikasi kesalahan yang terjadi, seperti:

- a. *False Positive* (FP): Daun sehat diklasifikasikan sebagai sakit.
- b. *False Negative* (FN): Daun sakit diklasifikasikan sebagai sehat.

4. Analisis Komparatif Hasil

Analisis utama difokuskan pada perbandingan langsung antara Model A dan Model B.

- a. Perbandingan Metrik Kinerja: Hasil prediksi dari kedua model pada data uji akan digunakan untuk menghitung metrik performa masing-masing. Metrik tersebut (akurasi, presisi, recall, F1-score) akan disajikan dalam bentuk tabel perbandingan untuk melihat model mana yang lebih unggul secara kuantitatif.
- b. Analisis *Confusion Matrix*: *Confusion matrix* akan dibuat untuk kedua model. Analisis akan difokuskan pada perbandingan jenis kesalahan yang dibuat, terutama untuk melihat apakah augmentasi berhasil mengurangi kasus False Positive (daun sehat diklasifikasikan sebagai sakit) atau False Negative (daun sakit diklasifikasikan sebagai sehat).

- c. Visualisasi Kurva Pelatihan: Grafik akurasi dan loss selama proses pelatihan dan validasi dari kedua model akan ditampilkan dalam satu plot yang sama. Ini bertujuan untuk membandingkan kecepatan konvergensi dan melihat secara visual apakah teknik augmentasi efektif dalam mengurangi overfitting (celah antara kurva latih dan validasi).

5. Evaluasi Faktor-Faktor yang Mempengaruhi Hasil

Analisis dilakukan terhadap faktor-faktor eksternal dan internal yang dapat mempengaruhi hasil klasifikasi, antara lain:

- a. Kualitas pencahayaan saat pengambilan gambar daun.
- b. Resolusi citra dan ukuran input yang digunakan.
- c. Parameter pelatihan model, seperti jumlah epoch, batch size, dan learning rate.
- d. Konfigurasi dan hyperparameter dari model MobileNet, seperti learning rate, batch size, dan fungsi optimizer yang digunakan.

6. Visualisasi Hasil Evaluasi

Hasil dari pelatihan dan pengujian model divisualisasikan dalam bentuk:

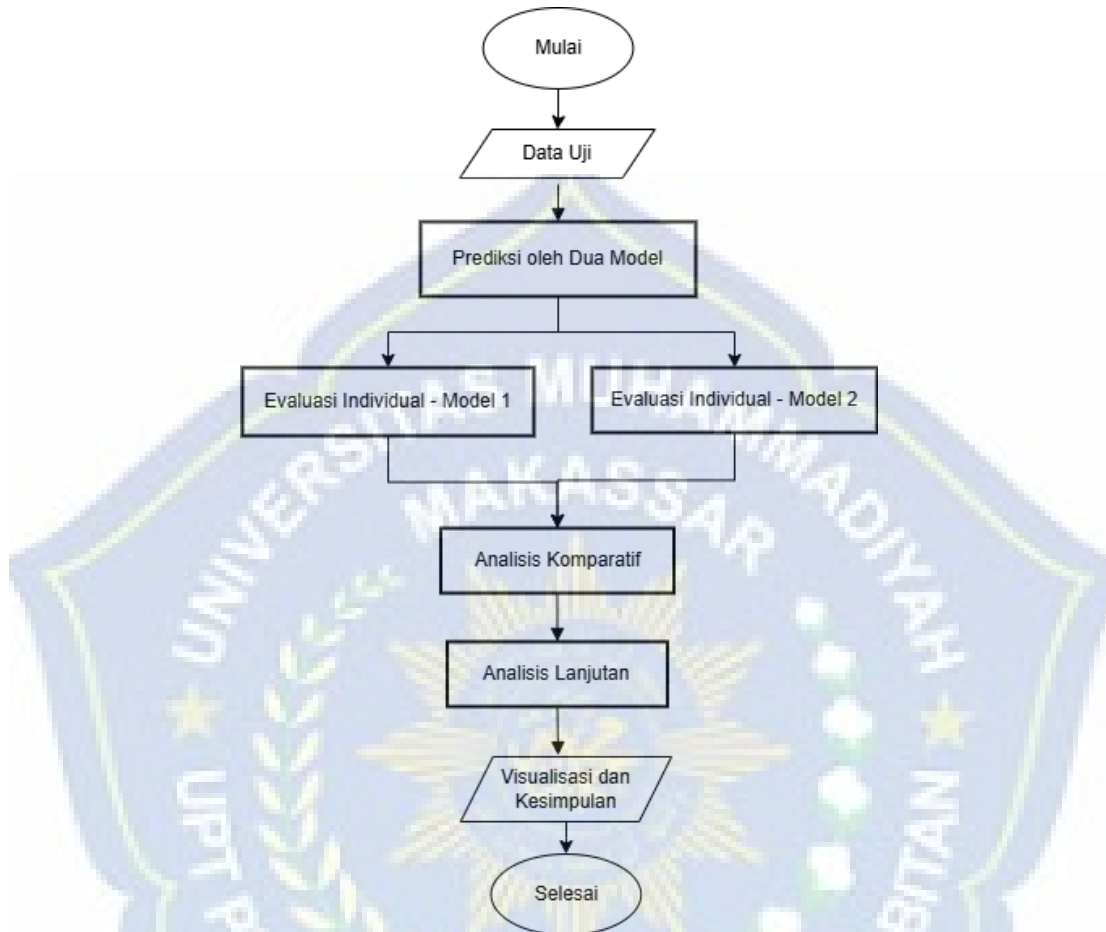
- a. Grafik akurasi dan loss per epoch untuk melihat proses konvergensi model.
- b. Visualisasi confusion matrix untuk interpretasi klasifikasi.
- c. Grafik precision, recall, dan F1-score per kelas, jika jumlah data per kelas bervariasi.

Visualisasi ini dibuat menggunakan Matplotlib dan Seaborn.

7. Representasi Data dalam Bentuk Tensor

Semua citra yang telah diproses (di-resize, dinormalisasi, dan diaugmentasi) dikonversi ke dalam bentuk tensor, yaitu array multidimensi (seperti: $224 \times 224 \times 3$) yang menyimpan informasi nilai piksel gambar. Tensor ini adalah format standar yang digunakan dalam deep learning dan diproses oleh framework seperti TensorFlow dan Keras. Dengan tensor, proses pelatihan dan inferensi pada CNN dapat dilakukan secara efisien

Flowchart Prosedur Analisis Data



Gambar 7. Flowchart Prosedur analisis data

Penjelasan gambar 7. Flowchart prosedur analisis data:

1. Data Uji

Ini adalah langkah awal dari prosedur analisis. Data Uji merupakan sekumpulan citra daun merica (baik yang sehat maupun yang terinfeksi) yang sama sekali belum pernah dilihat oleh kedua model selama fase pelatihan atau validasi. Penggunaan data yang benar-benar baru ini bertujuan untuk mendapatkan evaluasi yang objektif dan tidak bias mengenai seberapa baik kemampuan generalisasi masing-masing model pada kondisi dunia nyata.

2. Prediksi Oleh Dua Model

Pada tahap ini, Data Uji yang sama dimasukkan ke dalam dua model yang telah selesai dilatih:

- a. Model 1: Model CNN yang dilatih tanpa teknik augmentasi data.
- b. Model 2: Model CNN yang dilatih dengan teknik augmentasi data.

Setiap model akan membuat prediksi kelas (seperti: “Daun Sehat”, “Terinfeksi Busuk pangkal batang”, “Penyakit kuning” atau “Terinfeksi Bercak daun”.) untuk setiap gambar dalam set data uji.

3. Evaluasi Individual

Setelah prediksi dibuat, performa masing-masing model dievaluasi secara terpisah. Proses evaluasi ini meliputi:

- a. Menghitung Metrik Kinerja: Metrik kuantitatif seperti akurasi, presisi, recall, dan F1-score dihitung untuk setiap model berdasarkan perbandingan hasil prediksi dengan label yang sebenarnya.
- b. Membuat *Confusion Matrix*: Sebuah matriks dibuat untuk menunjukkan rincian kesalahan klasifikasi (misalnya, berapa banyak daun sakit yang salah diklasifikasikan sebagai sehat, dan sebaliknya).

4. Analisis Komparatif

Ini adalah tahap inti dari analisis. Hasil dari evaluasi individual kedua model digabungkan dan dibandingkan secara langsung. Tujuannya adalah untuk menjawab pertanyaan-pertanyaan kunci seperti:

- a. Model manakah yang memiliki akurasi keseluruhan yang lebih tinggi?
- b. Apakah augmentasi data (pada Model 2) berhasil mengurangi jenis kesalahan tertentu (misalnya, false negative) dibandingkan Model 1?
- c. Model manakah yang menunjukkan performa lebih seimbang di semua kelas penyakit?

5. Analisis Lanjutan

Setelah perbandingan kuantitatif, analisis dilanjutkan ke level yang lebih dalam untuk memahami mengapa ada perbedaan performa. Ini bisa mencakup:

- a. Menganalisis faktor-faktor eksternal seperti kualitas pencahayaan atau resolusi gambar pada data uji yang mungkin lebih sulit dikenali oleh salah satu model.
- b. Meninjau kembali parameter pelatihan (seperti learning rate atau jumlah epoch) yang mungkin mempengaruhi hasil akhir.

6. Visualisasi dan Kesimpulan

Ini adalah tahap akhir dari prosedur analisis.

- a. Visualisasi: Hasil dari analisis komparatif disajikan dalam bentuk visual yang mudah dipahami, seperti grafik batang untuk perbandingan metrik atau menampilkan confusion matrix secara berdampingan.
- b. Kesimpulan: Berdasarkan semua analisis dan visualisasi, sebuah kesimpulan ditarik. Kesimpulan ini akan menyatakan secara tegas model mana yang lebih efektif dan mengapa, serta mengonfirmasi apakah penerapan teknik augmentasi data terbukti memberikan dampak positif yang signifikan pada performa model.

BAB IV

HASIL DAN PEMBAHASAN

A. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan berdasarkan kerangka kerja dan metodologi yang telah diuraikan pada BAB III. Tahapan implementasi mencakup persiapan lingkungan pengujian, pengolahan dataset, hingga pembangunan arsitektur model CNN.

1. Lingkungan Pengujian

Penelitian ini dilakukan menggunakan perangkat keras dan perangkat lunak dengan spesifikasi yang telah ditentukan untuk memastikan proses pelatihan dan pengujian model dapat berjalan secara optimal.

a. Perangkat keras

Proses pelatihan dan pengujian model menggunakan laptop Lenovo Ideapad Slim 1 dengan prosesor AMD Ryzen 3 5300U, RAM 8 GB, dan SSD 512 GB.

b. Perangkat lunak

Lingkungan pengembangan utama adalah Python 3.11.9 dan Implementasi model deep learning memanfaatkan framework TensorFlow dengan Keras sebagai API high-level. Untuk pemrosesan citra, digunakan library OpenCV, sedangkan untuk manipulasi data dan visualisasi digunakan NumPy, Matplotlib, dan Seaborn.

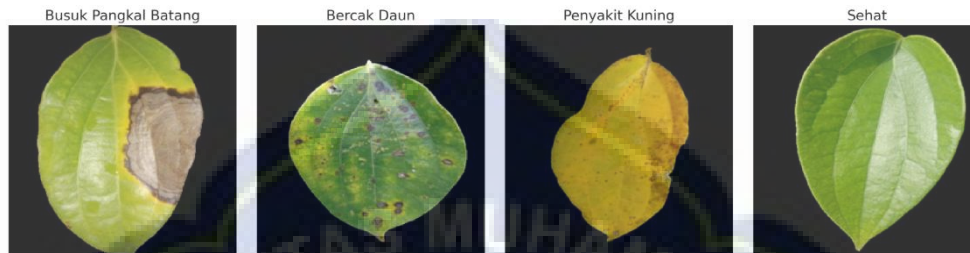
2. Persiapan dan Pengolahan Dataset

a. Dataset Akhir

Dataset yang digunakan dalam penelitian ini diperoleh melalui pengambilan gambar secara langsung di perkebunan merica Desa Matano, Kabupaten Luwu Timur. Total dataset yang berhasil dikumpulkan dan dilabeli adalah sebanyak 600 citra, yang terbagi rata ke dalam empat kelas, yaitu:

- 1) [150] gambar daun sehat
- 2) [150] gambar penyakit busuk pangkal batang
- 3) [150] gambar penyakit kuning

- 4) [150] gambar penyakit bercak daun
 - 5) [150] gambar bukan daun merica
- b. Visualisasi gambar dari setiap kelas



Gambar 8. Dataset perkelas

c. Pembagian Dataset

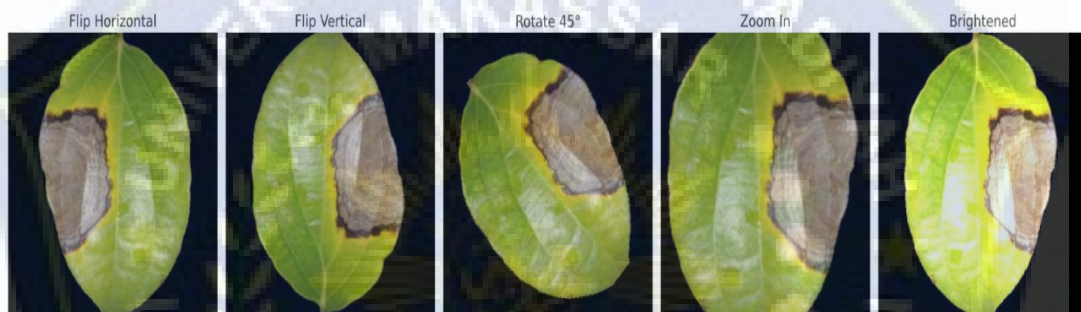
Dataset kemudian dibagi menjadi tiga bagian dengan proporsi 70% data latih, 20% data validasi, dan 10% data uji. Pembagian ini bertujuan untuk melatih model, memvalidasi performa selama pelatihan, dan menguji kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Rincian pembagian disajikan pada Tabel 2.

Tabel 2. Pembagian dataset

Kelas penyakit	Data latih 70%	Data Validasi 20%	Data uji 10%	Total perkelas
Daun sehat	105	30	15	150
Busuk pangkal batang	105	30	15	150
Bercak daun	105	30	15	150
Penyakit kuning	105	30	15	150
Bukan daun merica	105	30	15	150
Total keseluruhan	525	150	75	750

3. Implementasi Preprocessing dan Augmentasi Data

Setiap citra dalam dataset diproses dengan mengubah ukurannya menjadi 224x224 piksel dan dinormalisasi. Khusus untuk Model B, 525 citra latih diperbanyak melalui teknik augmentasi *on-the-fly* selama 30 epoch pelatihan, dengan transformasi meliputi rotasi, *flip* horizontal, dan penyesuaian kecerahan. Proses ini secara efektif membuat model belajar dari 15.750 variasi gambar (525 gambar $\times$ 30 epoch). Paparan data yang masif ini dirancang untuk memperkaya pembelajaran model, sehingga dapat meningkatkan akurasi dan kemampuan generalisasinya terhadap data baru. Hasil augmentasi pada sebuah citra dapat dilihat pada gambar 9 di bawah ini:



Gambar 9. Hasil Augmentasi Citra

4. Implementasi Arsitektur Model CNN MobileNet

Penelitian ini mengimplementasikan dua model CNN yang identik dari segi arsitektur, yaitu MobileNetV2, namun berbeda dalam perlakuan data latihnya.

- a. Model A (Kontrol): Dilatih menggunakan 525 citra latih asli yang hanya melalui preprocessing.
- b. Model B (Eksperimen): Dilatih menggunakan 525 citra latih yang sama, namun dengan penerapan teknik augmentasi.

Pengaturan hyperparameter untuk kedua model dibuat identik untuk memastikan bahwa satu-satunya variabel pembeda adalah penggunaan augmentasi. Konfigurasi pelatihan dirangkum pada Tabel dibawah ini.

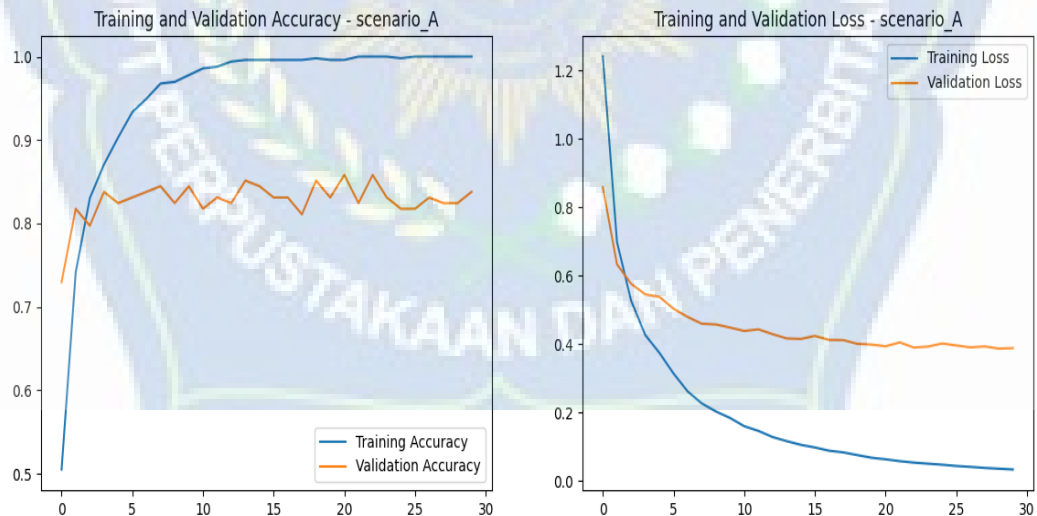
Tabel 3. Konfigurasi hyperparameter Pelatihan

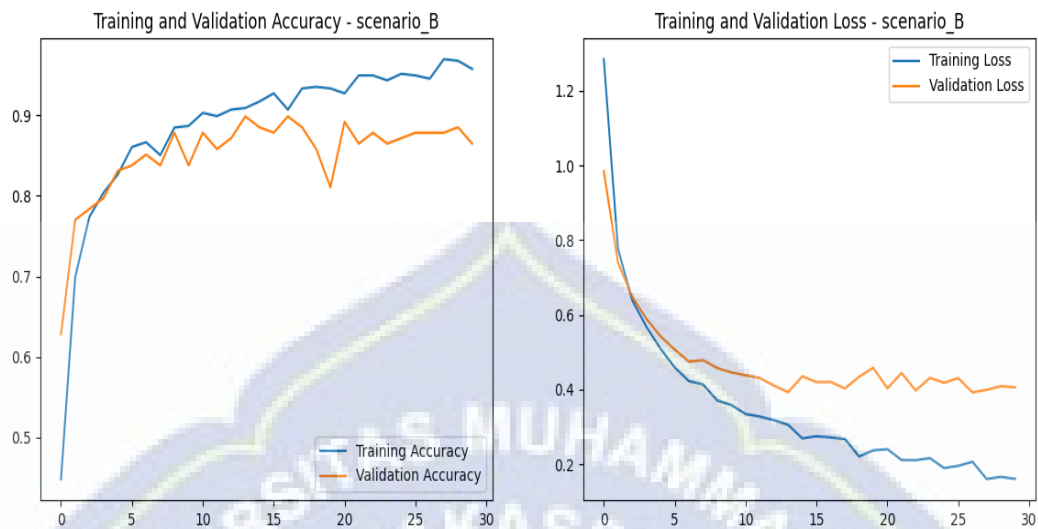
Parameter	Nilai yang digunakan
Arsitektur dasar	MobileNetV2
Optimizer	Adam
Learning Rate	0.0001
Fungsi Loss	Sparse categorical crossentropy
Metrik Pantauan	Accuracy
Jumlah Epoch	30
Batch Size	32

B. HASIL PENGUJIAN DAN EVALUASI MODEL

1. Kinerja Proses Pelatihan

Proses pelatihan selama 30 epoch menghasilkan kurva akurasi dan loss yang disajikan pada Gambar 10. Grafik ini membandingkan secara langsung kinerja Model A dan Model B pada data latih dan data validasi.





Gambar 10. Grafik Perbandingan Akurasi dan Loss Pelatihan & Validasi

Dari grafik di atas dapat dilihat perbedaan utama antara kedua model bahwa Scenario A menunjukkan model yang mengalami *overfitting* parah, sedangkan Scenario B menunjukkan model yang belajar dengan baik (*good fit*) dan mampu menggeneralisasi pengetahuannya. Grafik ini secara visual membuktikan dampak krusial dari penggunaan augmentasi data dalam pelatihan model. Perbandingan lebih lanjut dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan *Head-to-Head* Kedua Model

Aspek Perbandingan	Grafik Scenario A (Tanpa Augmentasi)	Grafik Scenario B (Dengan Augmentasi)
Kinerja Validasi	Stagnan & Tidak Stabil. Akurasi validasi yang direpresentasikan oleh kurva oranye berhenti mengalami peningkatan signifikan setelah beberapa epoch awal dan berfluktuasi di rentang 82-85%. Grafik yang sangat bergejolak dengan naik-turun yang tajam ini menunjukkan performa yang tidak konsisten saat dihadapkan pada data baru	Terus Membaik. Sebaliknya, akurasi validasi pada Skenario B menunjukkan tren peningkatan yang konsisten. Model ini berhasil mencapai tingkat kinerja yang lebih baik, yaitu di sekitar 85-88%.
Celah Antara Training & Validasi	Sangat Besar. Terdapat jurang yang sangat lebar antara kurva data training (biru) dan kurva data validasi (oranye). Hal ini mengindikasikan bahwa performa model sangat buruk pada data baru yang belum pernah dilihat sebelumnya.	Kecil dan Terkendali. Celah antara kedua kurva terlihat jauh lebih sempit. Ini menunjukkan bahwa kinerja model seimbang antara data latih dan data baru, menandakan kemampuan generalisasi yang baik.
Stabilitas Kurva	Sangat Tidak Stabil. Kurva validasi (oranye) yang naik-turun secara drastis menandakan adanya ketidakpastian pada model saat melakukan prediksi terhadap data baru.	Relatif Stabil. Kurva validasi pada skenario ini terlihat jauh lebih mulus. Hal ini menunjukkan bahwa proses belajar model berjalan lebih konsisten dan dapat diandalkan.
Indikasi Model	OVERFITTING PARAH. Model ini hanya menghafal data latihnya dan tidak benar-benar belajar pola yang mendasarinya.	GOOD FIT. Model ini berhasil belajar pola umum dari data yang ada dan mampu mengaplikasikan pengetahuannya tersebut pada data baru dengan baik.

Skenario A (Tanpa Augmentasi) adalah contoh model yang "rapuh". Ia menjadi sempurna (akurasi 100%) pada soal-soal yang sudah dihafalnya (data latih), tetapi

gagal total saat diberi soal baru (data validasi). Model seperti ini tidak berguna dalam aplikasi dunia nyata karena tidak dapat diandalkan.

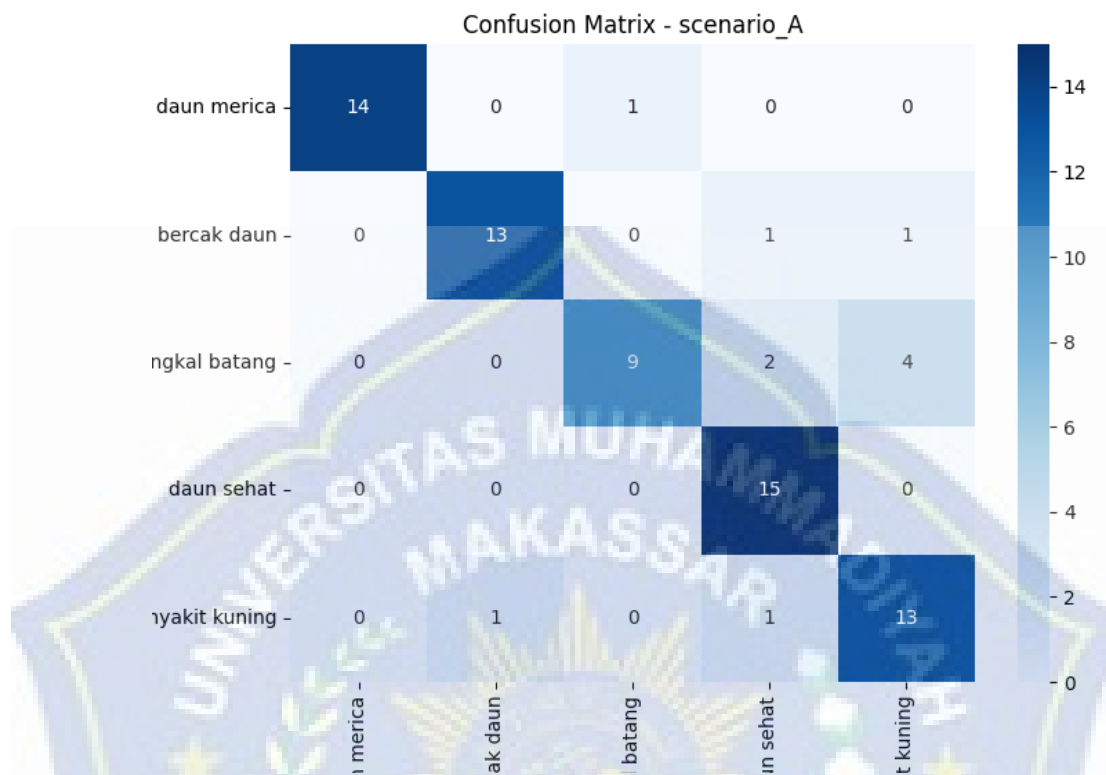
Skenario B (Dengan Augmentasi) adalah contoh model yang "tangguh". Dengan memberikan variasi gambar buatan (diputar, digeser, dicerahkan), augmentasi memaksa model untuk belajar fitur-fitur yang esensial dari setiap penyakit, bukan hanya menghafal posisi atau pencahayaan dari gambar asli. Hasilnya adalah model yang lebih "pintar", yang mampu mengenali penyakit bahkan dalam kondisi gambar yang sedikit berbeda.

2. Hasil Evaluasi Kinerja Model A (Tanpa Augmentasi)

Model A diuji pada 75 citra data uji dan menghasilkan akurasi keseluruhan sebesar 85%. Rincian kesalahan klasifikasi divisualisasikan dalam confusion matrix pada Gambar 11, dan metrik performa lengkapnya disajikan pada Tabel 5.

a. Evaluasi Kinerja Model Menggunakan Confusion Matrix

Untuk mengevaluasi performa klasifikasi dari Model A (Tanpa Augmentasi), digunakan metode confusion matrix. Metrik ini memberikan gambaran visual yang rinci mengenai kinerja model dalam memprediksi setiap kelas pada data uji. Hasil evaluasi disajikan pada Gambar 11.



Gambar 11. confusion matrix Model A

Grafik di atas menunjukkan hasil prediksi dari Model A (Tanpa Augmentasi) terhadap 75 data uji yang terdistribusi secara seimbang ke dalam lima kelas, yaitu bercak daun, busuk pangkal batang, penyakit kuning, daun sehat, dan Bukan daun merica. Masing-masing kelas diasumsikan memiliki 15 data uji. Sumbu vertikal (y-axis) merepresentasikan kelas aktual (*True Label*), sedangkan sumbu horizontal (x-axis) merepresentasikan kelas yang diprediksi oleh model (*Predicted Label*).

- 1.) Kelas Bukan daun merica: Dari total 15 gambar asli yang bukan daun merica, 14 gambar berhasil diprediksi dengan benar sebagai 'bukan daun merica'. 1 gambar salah diprediksi sebagai 'busuk pangkal batang'.
- 2.) Kelas Bercak daun: Dari total 15 gambar asli yang terkena bercak daun, 13 gambar berhasil diprediksi dengan benar sebagai 'bercak daun'. 1 gambar salah diprediksi sebagai 'daun sehat' dan 1 gambar salah diprediksi sebagai 'penyakit kuning'.

- 3.) Kelas Busuk pangkal batang: Dari total 15 gambar asli yang terkena busuk pangkal batang hanya 9 gambar yang berhasil diprediksi dengan benar. Ini adalah performa terendah. 2 gambar salah diprediksi sebagai 'daun sehat'. Ini adalah kesalahan serius karena penyakit dianggap sehat. 4 gambar salah diprediksi sebagai 'penyakit kuning'. Ini adalah kesalahan prediksi terbanyak untuk kelas ini.
- 4.) Kelas Sehat: Dari total 15 gambar asli yang merupakan daun sehat Semua 15 gambar berhasil diprediksi dengan sempurna sebagai 'daun sehat'. Tidak ada kesalahan sama sekali.
- 5.) Kelas Penyakit kuning: Dari total 15 gambar asli yang terkena penyakit kuning 13 gambar berhasil diprediksi dengan benar sebagai 'penyakit kuning', 1 gambar salah diprediksi sebagai 'bercak daun', 1 gambar salah diprediksi sebagai 'daun sehat'.

Kesimpulannya: Model sangat andal dalam mengenali daun sehat, dengan akurasi 100% pada data uji namun model kesulitan mengenali busuk pangkal batang, di mana ia membuat 6 kesalahan dari 15 gambar, dan sering keliru menganggapnya sebagai penyakit kuning atau daun sehat.

Metrik Evaluasi

Tabel 5. Metrik Evaluasi Kinerja Model A pada Data Uji

	Precision	Recall	F1-Score	Support
Bukan Daun Merica	1.00	0.93	0.97	15
Bercak Daun	0.93	0.87	0.90	15
Busuk Pangkal Batang	0.90	0.60	0.72	15
Sehat	0.79	1.00	0.88	15
Penyakit Kuning	0.72	0.87	0.79	15

Accuracy			0.85	75
Macro Avg	0.87	0.85	0.85	75
Weighted Avg	0.87	0.85	0.85	75

Analisis kinerja per kelas:

1.) Bukan daun merica

- a.) Precision: 1.00 (100%), Sempurna. Setiap kali model menebak sebuah gambar sebagai 'Bukan daun merica', tebakannya selalu benar.
- b.) Recall: 0.93 (93%), Sangat baik. Model berhasil menemukan 93% dari semua gambar 'Bukan daun merica' yang ada di data uji.
- c.) F1-Score: 0.97, Skor keseluruhan yang sangat tinggi, menunjukkan performa yang sangat andal untuk kelas ini.

2.) Bercak daun

- a.) Precision: 0.93 (93%), Sangat baik. 93% dari tebakan model untuk 'bercak daun' adalah benar.
- b.) Recall: 0.87 (87%), Baik. Model berhasil mengidentifikasi 87% dari semua kasus 'bercak daun' yang sebenarnya.
- c.) F1-Score: 0.90, Skor yang sangat solid, menandakan model ini cukup baik dalam mendeteksi bercak daun.

3.) Busuk pangkal batang

- a.) Precision: 0.90 (90%), Baik. Saat model menebak 'busuk pangkal batang', tebakannya 90% benar.
- b.) Recall: 0.60 (60%), Rendah. Ini adalah kelemahan terbesar model. Model ini melewatkan 40% dari total kasus 'busuk pangkal batang' yang ada.
- c.) F1-Score: 0.72, Paling rendah di antara semua kelas, mengonfirmasi bahwa ini adalah kelas yang paling sulit dikenali oleh model.

4.) Daun sehat

- a.) Precision: 0.79 (79%), Cukup. Saat model menebak 'daun sehat', tingkat kebenarannya hanya 79%. Artinya, sekitar 21% dari yang dianggap sehat oleh model sebenarnya sakit.
- b.) Recall: 1.00 (100%) , Sempurna. Model berhasil menemukan semua gambar 'daun sehat' yang ada. Tidak ada satu pun daun sehat yang terlewatkan.
- c.) F1-Score: 0.88, Skor yang baik, meskipun ada masalah pada presisi.

5.) Penyakit kuning

- a.) Precision: 0.72 (72%), Rendah. Ini berarti model sering salah tebak saat memprediksi 'penyakit kuning'.
- b.) Recall: 0.87 (87%), Baik. Model ini cukup andal dalam menemukan sebagian besar kasus penyakit kuning.
- c.) F1-Score: 0.79 , Cukup, menunjukkan ada ruang untuk perbaikan, terutama pada presisi.

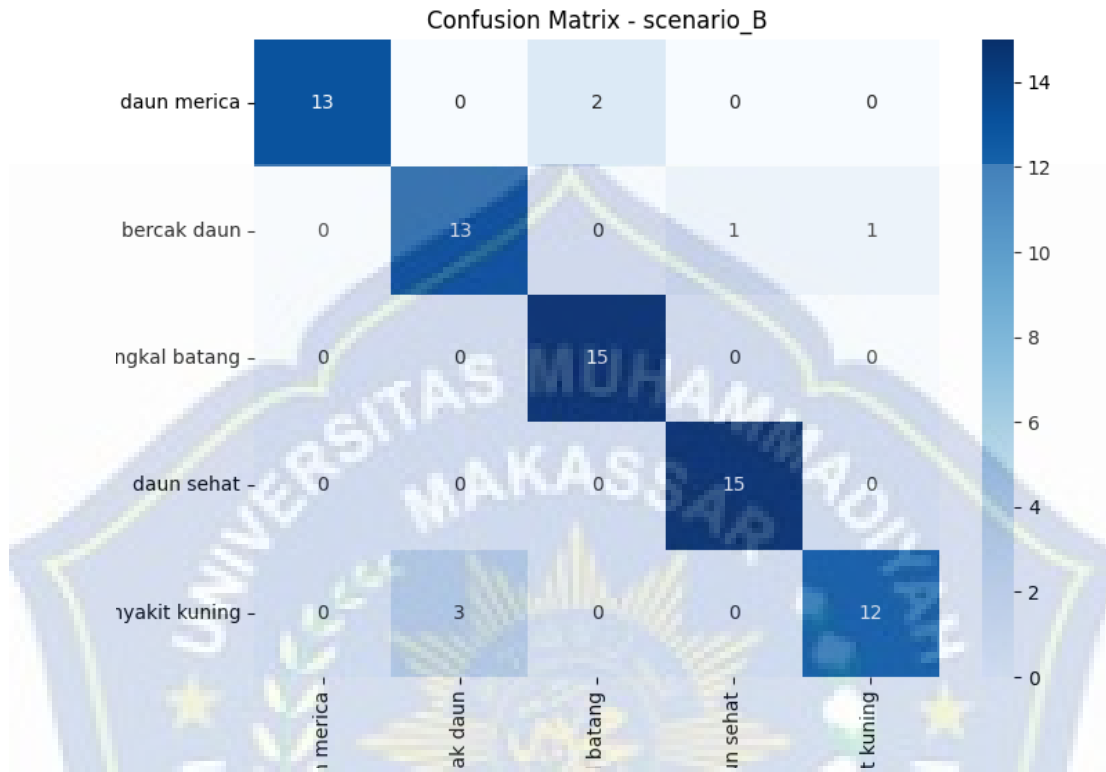
Analisis keseluruhan (Rata-Rata):

- 1.) Accuracy: 0.85 (85%), Secara keseluruhan, 85% dari total 75 prediksi yang dibuat oleh model adalah benar.
- 2.) Macro Avg: 0.87 (precision), 0.85 (recall), 0.85 (f1-score), Ini adalah rata-rata skor dari semua kelas, di mana setiap kelas diberi bobot yang sama.
- 3.) Weighted Avg: 0.87 (precision), 0.85 (recall), 0.85 (f1-score), Ini adalah rata-rata skor yang dibobot berdasarkan jumlah sampel per kelas (support). Karena semua kelas memiliki jumlah data yang sama (15), maka nilai macro avg dan weighted avg akan identik.

3. Hasil Evaluasi Kinerja Model B (Dengan Augmentasi)

Model B yang dilatih dengan augmentasi, diuji pada 75 citra data uji yang sama dan menghasilkan akurasi keseluruhan sebesar 92%. Rincian performanya disajikan pada Gambar 12 dan Tabel 6 .

a. Evaluasi Kinerja Model Menggunakan Confusion Matrix



Gambar 12.confusion matrix Model B

Grafik di atas menunjukkan hasil prediksi dari Model B (Dengan Augmentasi) terhadap 75 data uji yang terdistribusi secara seimbang ke dalam empat kelas, yaitu bercak daun, pangkal batang, penyakit kuning, daun sehat, dan Bukan daun merica. Masing-masing kelas memiliki 15 data uji. Sumbu vertikal (y-axis) merepresentasikan kelas aktual (*True Label*), sedangkan sumbu horizontal (x-axis) merepresentasikan kelas yang diprediksi oleh model (*Predicted Label*). Kinerja model untuk setiap kelas dapat dianalisis secara mendalam sebagai berikut:

- 1.) Kelas Bukan daun merica: Dari total 15 gambar asli yang bukan daun merica, 13 gambar berhasil diprediksi dengan benar sebagai 'bukan daun merica', 2 gambar salah diprediksi sebagai 'busuk pangkal batang'.

- 2.) Kelas Bercak daun: Dari total 15 gambar asli yang terkena bercak daun, 13 gambar berhasil diprediksi dengan benar sebagai 'bercak daun', 1 gambar salah diprediksi sebagai 'daun sehat', 1 gambar salah diprediksi sebagai 'penyakit kuning'.
- 3.) Kelas Busuk Pangkal Batang: Dari total 15 gambar asli yang terkena busuk pangkal batang, Semua 15 gambar berhasil diprediksi dengan benar. Tidak ada kesalahan sama sekali untuk kelas ini.
- 4.) Kelas Sehat: Dari total 15 gambar asli yang merupakan daun sehat: Semua 15 gambar juga berhasil diprediksi dengan benar tanpa kesalahan.
- 5.) Kelas Penyakit kuning: Dari total 15 gambar asli yang terkena penyakit kuning, 12 gambar berhasil diprediksi dengan benar, 3 gambar salah diprediksi sebagai 'bercak daun'.

kesimpulan perbandingan dengan Skenario A

Jika kita bandingkan dengan matriks Skenario A sebelumnya, terlihat jelas dampak positif dari augmentasi data:

- 1.) Peningkatan Drastis: Kelas busuk pangkal batang yang sebelumnya menjadi kelas terlemah (hanya 9 prediksi benar), kini menjadi sempurna (15 prediksi benar). Ini menunjukkan augmentasi sangat efektif membuat model mengenali fitur-fitur penyakit ini.
- 2.) Kinerja Stabil dan Sempurna: Model mempertahankan kinerja sempurna untuk kelas daun sehat dan juga mencapai kesempurnaan pada busuk pangkal batang.
- 3.) Pergeseran Kesalahan: Meskipun ada sedikit penurunan pada kelas penyakit kuning (dari 13 menjadi 12 prediksi benar), secara keseluruhan model Skenario B jauh lebih andal, terutama dalam mengatasi kelas yang paling sulit.

Secara keseluruhan, matriks ini membuktikan bahwa Skenario B (dengan augmentasi) secara signifikan lebih unggul daripada Skenario A.

Metrik Evaluasi

Tabel 6. Metrik Evaluasi Kinerja Model B pada Data Uji

	Precision	Recall	F1-Score	Support
Bukan daun merica	1.00	0.87	0.93	15
Bercak Daun	0.81	0.87	0.84	15
Busuk Pangkal Batang	0.88	1.00	0.94	15
Sehat	0.94	1.00	0.97	15
Penyakit Kuning	0.92	0.80	0.86	15
Accuracy			0.91	75
Macro Avg	0.91	0.91	0.91	75
Weighted Avg	0.91	0.91	0.91	75

Analisis kinerja per kelas:

Analisis ini diurutkan berdasarkan F1-Score untuk mengevaluasi kinerja model dari yang terbaik hingga yang terendah pada setiap kategori.

1.) Bukan daun merica

- Precision: 1.00 (100%), Sempurna. Setiap kali model ini menebak sebuah gambar adalah 'Bukan daun merica', tebakannya 100% benar.
- Recall: 0.87 (87%), Baik. Model berhasil menemukan 87% dari semua gambar 'Bukan daun merica', namun masih ada 13% yang terlewat.

- c.) F1-Score: 0.93, Skor keseluruhan yang sangat tinggi, menunjukkan keandalan yang luar biasa.

2.) Bercak daun

- a.) Precision: 0.81 (81%), Cukup baik. Saat model menebak 'bercak daun', tebakannya benar sebanyak 81% dari waktu.
- b.) Recall: 0.87 (87%), Baik. Model berhasil mengidentifikasi 87% dari semua kasus 'bercak daun' yang sebenarnya.
- c.) F1-Score: 0.84, Kinerja yang seimbang dan solid untuk kelas ini.

3.) Busuk pangkal batang

- a.) Precision: 0.88 (88%), Baik. Tebakan model untuk kelas ini memiliki tingkat akurasi 88%.
- b.) Recall: 1.00 (100%), Sempurna! Model ini menemukan semua kasus 'busuk pangkal batang' tanpa ada yang terlewat. Ini adalah peningkatan yang sangat signifikan dari model sebelumnya.
- c.) F1-Score: 0.94, Skor yang sangat tinggi, menunjukkan model ini sangat sensitif dan efektif untuk penyakit ini.

4.) Sehat

- a.) Precision: 0.94 (94%), Sangat baik. 94% dari tebakan model untuk 'daun sehat' adalah benar.
- b.) Recall: 1.00 (100%), Sempurna! Model juga berhasil mengidentifikasi setiap 'daun sehat' yang ada.
- c.) F1-Score: 0.97, Performa terbaik di antara semua kelas. Model ini sangat andal dalam mengenali daun yang sehat.

5.) Penyakit kuning

- a.) Precision: 0.92 (92%), Sangat baik. Tebakan model untuk 'penyakit kuning' sangat bisa dipercaya.
- b.) Recall: 0.80 (80%), Cukup baik. Model berhasil menemukan 80% kasus, namun masih melewatkan 20% dari total penyakit kuning yang ada.

c.) F1-Score: 0.86, Skor keseluruhan yang baik, meskipun ada sedikit kelemahan pada recall.

Analisis Keseluruhan:

- 1.) Accuracy: 0.91 (91%), Secara keseluruhan, 91% dari total 75 prediksi yang dibuat oleh model adalah benar. Ini adalah tingkat akurasi yang sangat tinggi.
- 2.) Macro Avg & Weighted Avg: 0.91, Rata-rata skor untuk semua kelas adalah 91%. Nilainya sama karena jumlah data (support) untuk setiap kelas seimbang (masing-masing 15).

4. Visualisasi Hasil Prediksi Sistem

Untuk memberikan gambaran kualitatif mengenai kinerja sistem, dilakukan pengujian pada sebuah citra daun yang terinfeksi penyakit Bercak Daun, Busuk pangkal batang, Penyakit kuning, dan Sehat. Gambar - gambar di bawah ini menampilkan hasil prediksi yang diberikan oleh Model A (tanpa augmentasi) dan Model B (dengan augmentasi).

a. Bercak Daun

Analisis Klasifikasi: IMG_20250511_090011-removebg-preview - Copy.png



Model A (Tanpa Augmentasi)
Identifikasi: Daun Merica (99.47%)
Status: bercak daun
Gejala:
• Menimbulkan bercak khas cokelat atau hitam pada daun, yang dapat meluas, menyebabkan daun menguning, dan gugur sebelum waktunya.

Model B (Dengan Augmentasi)
Identifikasi: Daun Merica (99.49%)
Status: bercak daun
Gejala:
• Menimbulkan bercak khas cokelat atau hitam pada daun, yang dapat meluas, menyebabkan daun menguning, dan gugur sebelum waktunya.

Gambar 13. Tampilan Hasil Prediksi untuk Penyakit Bercak Daun

Berdasarkan hasil pada Gambar 13, dapat dilihat bahwa kedua model berhasil melakukan identifikasi dan klasifikasi dengan benar dan dengan tingkat kepercayaan yang sangat tinggi. Model B menunjukkan skor kepercayaan yang sedikit lebih unggul (0.02%), namun perbedaannya tidak signifikan untuk kasus ini. Hal ini menunjukkan bahwa untuk contoh gambar dengan gejala yang jelas, kedua model mampu memberikan hasil yang akurat.

b. Busuk Pangkal Batang

Pengujian pada citra daun yang terinfeksi penyakit Busuk Pangkal Batang. Hasil dari pengujian tersebut ditampilkan pada gambar 14.

Analisis Klasifikasi: IMG-20250804-WA0012-removebg-preview.png



Gambar 14. Tampilan Hasil Prediksi untuk Penyakit Busuk Pangkal Batang

Dari hasil yang ditampilkan pada Gambar 14 dapat dianalisis bahwa secara jelas menunjukkan keunggulan signifikan Model B yang dilatih dengan augmentasi data, di mana perbedaan tingkat kepercayaan yang mencolok Model B yang jauh lebih yakin (91.66%) dibandingkan Model A yang ragu-ragu (64.34%) membuktikan bahwa Model B lebih andal karena telah mempelajari fitur-fitur yang lebih kuat, sehingga

menjadikannya tidak hanya lebih akurat, tetapi juga lebih "pintar" dan percaya diri dalam membuat keputusan.

c. Penyakit Kuning

Pengujian sistem menggunakan citra daun yang terinfeksi Penyakit Kuning. Tampilan antarmuka yang menampilkan hasil prediksi untuk citra ini disajikan pada gambar 15.

Analisis Klasifikasi: IMG-20250726-WA0088-removebg-preview.png



Gambar 15. Tampilan Hasil Prediksi untuk Penyakit Kuning

Analisis terhadap hasil pada Gambar 15 menunjukkan sebuah fenomena menarik di mana tingkat kepercayaan Model A yang lebih tinggi (93.04%) kemungkinan disebabkan oleh overfitting, sementara kepercayaan Model B yang lebih rendah (68.98%) menunjukkan proses yang lebih hati-hati dan terukur, membuktikan bahwa keunggulan sebuah model terletak pada akurasi tingginya di berbagai data uji, bukan pada skor kepercayaan di satu contoh saja.

d. Daun Sehat

Pengujian pada citra daun yang terinfeksi penyakit Busuk Pangkal Batang. Hasil dari pengujian tersebut ditampilkan pada gambar 16.

Analisis Klasifikasi: IMG-20250802-WA0184-removebg-preview.png



Model A (Tanpa Augmentasi)

Identifikasi: Daun Merica (99.43%)

Status: daun sehat

Gejala:

• -

Model B (Dengan Augmentasi)

Identifikasi: Daun Merica (99.24%)

Status: daun sehat

Gejala:

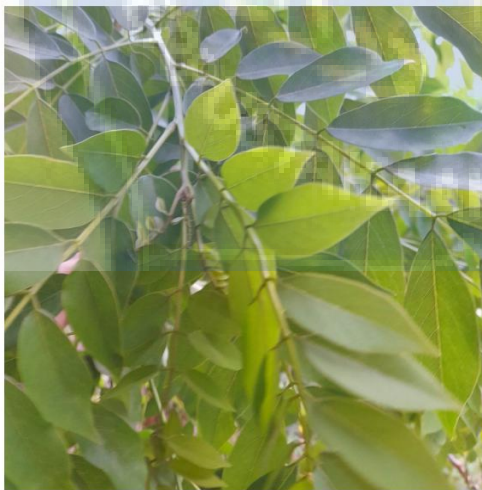
• -

Gambar 16. Tampilan Hasil Prediksi untuk Daun Sehat

Hasil prediksi gambar ini menunjukkan bahwa baik Model A (Tanpa Augmentasi) maupun Model B (Dengan Augmentasi) sama-sama berhasil mengklasifikasikan citra sebagai "daun sehat" dengan benar dan dengan tingkat kepercayaan yang sangat tinggi, masing-masing sebesar 99.43% dan 99.24%.

e. Bukan daun merica

Analisis Klasifikasi: Bukan daun merica_4.jpg



Model A (Tanpa Augmentasi)

Identifikasi: Bukan Daun Merica (99.80%)

Model B (Dengan Augmentasi)

Identifikasi: Bukan Daun Merica (98.91%)

Gambar 17. Tampilan Hasil Prediksi Bukan daun merica

C. PEMBAHASAN

Bagian ini membahas secara mendalam hasil pengujian, dengan fokus pada perbandingan kinerja Model A (tanpa augmentasi) dan Model B (dengan augmentasi), dampak teknik augmentasi, efektivitas arsitektur MobileNetV2, dan analisis kesalahan klasifikasi.

1. Analisis Perbandingan Kinerja

Perbandingan kinerja menunjukkan keunggulan signifikan Model B (dengan augmentasi). Akurasi total meningkat dari 85% pada Model A menjadi 91% pada Model B. Namun, dampak terpenting terlihat pada kemampuannya mengurangi kesalahan fatal. Berdasarkan confusion matrix (Gambar 11), Model A salah mengklasifikasikan total 4 citra daun sakit sebagai 'daun sehat'. Sementara pada Model B (Gambar 12), kesalahan fatal ini berhasil ditekan drastis menjadi hanya 1 citra. Ini membuktikan augmentasi secara spesifik meningkatkan keandalan model dalam situasi kritis.

2. Dampak Augmentasi Data terhadap Generalisasi Model

Efek positif dari augmentasi data paling jelas terlihat pada kemampuannya untuk menekan overfitting secara signifikan. Pada grafik kinerja pelatihan (Gambar 10), kurva akurasi latih Model A berhasil mencapai hampir 100%, namun akurasi validasinya stagnan pada level yang jauh lebih rendah dan tidak stabil. Celah yang lebar antara kurva latih dan validasi ini merupakan indikator klasik bahwa model hanya "menghafal" data latih dan tidak mampu menggeneralisasi polanya dengan baik.

Sebaliknya, kurva latih dan validasi pada Model B bergerak jauh lebih serasi dengan celah yang lebih sempit dan terkendali. Hal ini membuktikan bahwa dengan memberikan variasi gambar buatan (rotasi, flip, kecerahan), Model B dipaksa untuk belajar mengenali fitur-fitur penyakit yang lebih fundamental dan invarian (tidak terpengaruh oleh orientasi atau pencahayaan). Hasilnya adalah model yang memiliki kemampuan generalisasi yang jauh lebih baik saat dihadapkan pada data uji yang baru dan belum pernah dilihat sebelumnya.

3. Analisis Penerapan Arsitektur MobileNet

Hasil penelitian ini secara keseluruhan menegaskan bahwa arsitektur MobileNetV2 merupakan pilihan yang tepat dan efektif untuk tugas ini. Kemampuannya untuk mencapai akurasi tinggi di atas 91% pada Model B dengan arsitektur yang relatif ringan menunjukkan potensinya untuk diimplementasikan pada perangkat dengan sumber daya terbatas, seperti smartphone, yang akan memudahkan proses identifikasi penyakit secara cepat di lapangan. Model ini berhasil mengekstraksi fitur-fitur visual yang relevan dari citra daun merica untuk membedakan antara kondisi sehat dan tiga jenis penyakit yang berbeda.

4. Analisis Kesalahan Klasifikasi (Misclassification Analysis)

confusion matrix mengungkap kelemahan fundamental Model A yang bersumber dari overfitting, dengan kegagalan paling signifikan terjadi pada kelas ‘busuk pangkal batang’, di mana hanya 9 dari 15 citra yang berhasil diklasifikasikan dengan benar. Kesalahan fatal terjadi karena 2 citra ‘busuk pangkal batang’ keliru diidentifikasi sebagai ‘daun sehat’, sementara 4 citra lainnya keliru dianggap sebagai ‘penyakit kuning’, yang mengindikasikan kebingungan model akibat kemiripan gejala menguning. Secara total, Model A membuat 4 kesalahan fatal di mana daun sakit dianggap sehat di seluruh kelas penyakit.

Di sisi lain, sumber kesalahan pada Model B menunjukkan dampak positif dari augmentasi data, di mana masalah kini bersumber dari kompleksitas data itu sendiri, bukan lagi kelemahan model. Model B berhasil memperbaiki secara sempurna kelemahan terbesar Model A, dengan kelas ‘busuk pangkal batang’ kini mencapai akurasi 100% setelah 15 dari 15 citra teridentifikasi dengan benar. Tantangan utama kini bergeser ke kelas ‘penyakit kuning’, di mana 3 dari 15 citranya keliru diprediksi sebagai ‘bercak daun’, memperkuat hipotesis bahwa kesalahan bersumber dari kemiripan visual antar gejala. Jumlah kesalahan fatal (daun sakit dianggap sehat) berhasil berkurang drastis dari 4 kasus pada Model A menjadi hanya 1 kasus pada Model B, yang terjadi pada kelas 'bercak daun'.

BAB V

PENUTUP

A. KESIMPULAN

Berdasarkan hasil implementasi, pengujian, dan analisis perbandingan sebelumnya, dapat ditarik kesimpulan utama yang secara langsung menjawab rumusan masalah penelitian.

1. Metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2 terbukti efektif untuk mendeteksi dan mengklasifikasikan lima kelas kondisi daun, yaitu daun sehat, busuk pangkal batang, penyakit kuning, bercak daun, dan bukan daun merica. Efektivitas ini ditunjukkan dengan pencapaian akurasi keseluruhan sebesar 91% pada model yang menggunakan augmentasi data.
2. Penerapan teknik augmentasi data berperan signifikan dalam meningkatkan performa model. Hal ini dibuktikan oleh kenaikan akurasi dari 85% pada model tanpa augmentasi (Model A) menjadi 91% pada model dengan augmentasi (Model B). Selain itu, augmentasi juga terbukti mampu menekan overfitting terlihat dari kurva pelatihan yang lebih stabil dan selisih yang lebih kecil antara data latih dan validasi serta meningkatkan keandalan model dalam mendeteksi daun yang terinfeksi penyakit dengan mengurangi kesalahan klasifikasi fatal (False Negative).

B. SARAN

Meskipun penelitian ini telah mencapai tujuannya, untuk keberlanjutan dan penyempurnaan penelitian di masa depan, berikut adalah beberapa saran yang diajukan:

1. Untuk Penelitian Selanjutnya:

Untuk pengembangan di masa mendatang, disarankan agar penelitian ini dilanjutkan dengan validasi lapangan yang lebih komprehensif. Pengujian model pada dataset yang lebih besar dan beragam dari berbagai lokasi perkebunan akan sangat berguna untuk memastikan ketangguhan dan kemampuan adaptasi model terhadap

variasi geografis yang ada. Selain itu, perlu dilakukan studi komparatif antara arsitektur MobileNetV2 dengan model lain seperti EfficientNet-Lite atau ShuffleNet untuk menentukan arsitektur paling efisien yang menawarkan keseimbangan optimal antara akurasi dan kecepatan inferensi pada platform mobile. Terakhir, untuk meningkatkan nilai praktis penelitian, fungsionalitas model dapat diperkaya dengan kemampuan untuk mengklasifikasikan tingkat keparahan penyakit, sehingga dapat memberikan rekomendasi yang lebih spesifik dan dapat ditindaklanjuti oleh para petani.

2. Untuk Implementasi Praktis:

Hasil dari penelitian ini sangat potensial untuk diwujudkan dalam bentuk aplikasi *smartphone* yang *user-friendly*. Petani cukup memotret daun merica, dan aplikasi akan memberikan hasil diagnosis secara instan, menjembatani kesenjangan antara teknologi AI dan kebutuhan pertanian. Aplikasi tersebut dapat diperkaya dengan fitur yang memberikan rekomendasi penanganan awal (misalnya, jenis pestisida organik atau tindakan sanitasi kebun) berdasarkan penyakit yang terdeteksi.

DAFTAR PUSTAKA

- Anam Alidrus, S., Aziz, M., Virgantara Putra, O., & Darussalam Gontor Jl Raya Siman KecSiman KabPonorogo, U. (2021). Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network. In *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*.
- Arafat, F. A., Nur Ichsan, M., & Pramoedya, M. F. (2025). *Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital* (Vol. 4).
- Denaneera Waardhana, F., & Sarianti, T. (2024). FAKTOR-FAKTOR YANG MEMENGARUHI PRODUKSI LADA DI INDONESIA FACTORS AFFECTING PEPPER PRODUCTION IN INDONESIA. *Jurnal Ekonomi Pertanian Dan Agribisnis (JEPA)*, 8(2), 677–686. <https://doi.org/10.21776/ub.jepa.2024.008.02.23>
- Eriana, E. S., Kom, S., Kom, M., & Zein, D. A. (2023). *ARTIFICIAL INTELLIGENCE (AI) PENERBIT CV. EUREKA MEDIA AKSARA*.
- Fadli Gunardi, M. (2023). *Implementasi Augmentasi Citra pada Suatu Dataset*.
- Fitriatuzzahra, A., Zahira, A., Ihsani, F., Diah, T., Octavia, R., & Rizdafayi, S. (2024). *Deteksi Penyakit Tanaman Menggunakan Arsitektur MobileNetV2*.
- Fuadi, A., Suharso, A., Fakultas,), Komputer -Universitas, I., Karawang, S., Ronggo Waluyo, J. H., Timur, T., & Karawang, K. (2022). *PERBANDINGAN ARSITEKTUR MOBILENET DAN NASNETMOBILE UNTUK KLASIFIKASI PENYAKIT PADA CITRA DAUN KENTANG*. <https://jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/view/3026/1307>
- Jinan, A., & Hayadi, B. H. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). In *Journal of Computer and Engineering Science* (Vol. 1, Issue 2).

- Julyani, S. (2025). Pemanfaatan Pengolahan Citra Untuk Deteksi dan Identifikasi Hama pada Tanaman Secara Otomatis. *R2J*, 7(5). <https://doi.org/10.38035/rj.v7i5>
- Kotta, C. R., Paseru, D., Sumampouw, M., Informatika, T., Katolik De La Salle Manado, U., & Kombos Manado -, K. I. (2022). Implementasi Metode Convolutional Neural Network untuk Mendeteksi Penyakit pada Citra Daun Tomat Implementation of Convolutional Neural Network Method to Detect Diseases in Tomato Leaf Image. In *Jurnal_Pekommas_Vol._7_No* (Vol. 2).
- Lestari, N. I., Rianto, F., & Syahputra, E. (2023). IDENTIFIKASI SERANGAN *Phytophthora capsici* PENYEBAB PENYAKIT BUSUK PANGKAL BATANG PADA BIBIT LADA IDENTIFICATION OF *Phytophthora capsici* ATTACK CAUSES STEM ROT DISEASE ON PEPPER SEEDLINGS. In *Jurnal Pertanian Agros* (Vol. 25, Issue 3).
- M. FARHAN-FST. (2024). Analisis perbandingan pengaruh variasi data augmentasiterhadap kinerja mobilenetv2 dalam klasifikasi penyakit daun teh. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(4), 763-768.
- Manohara, D., Wahyuno, D., Penelitian, P., Perkebunan, P., Penelitian, B., Pertanian, P., Pertanian, K., Penelitian, B., Rempah, T., & Obat. (2020). *HAMA DAN PENYAKIT PADA TANAMAN LADA DI INDONESIA*. www.balittro.litbang.pertanian.go.id
- Munir, M. M., Kasih, P., & Sanjaya, A. (2024). Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) 1626. In *Agustus* (Vol. 8). Online.
- Pamungkas, N. B., & Suhendar, A. (2024). Penerapan Metode Convolutional Neural Network pada Sistem Klasifikasi Penyakit Tanaman Apel berdasarkan Citra Daun. *Edumatic: Jurnal Pendidikan Informatika*, 8(2), 675–684. <https://doi.org/10.29408/edumatic.v8i2.27958>
- Puspita Putri, L., Rafi Muttaqin, M., Raymond Ramadhan, Y., Informatika, T., & Wastukencana, S. (2025). Klasifikasi Penyakit Daun Cabai Menggunakan Model MobileNet. In *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains* (Vol. 15). <https://teknois.unbin.ac.id/JBS/>
- R Dijaya, H. S. (2023). 1364-Article Text-6433-3-10-20240305. *Umsida Press,Press.Umsida.Ac.Id*.

- Rahman, M., & Pambudi, A. (2023). IDENTIFIKASI CITRA DAUN SELADA DALAM MENENTUKAN KUALITAS TANAMAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN). *Jurnal Informatika Dan Teknik Elektro Terapan*, 11(3), 2830–7062. <https://doi.org/10.23960/jitet.v11i3%20s1.3438>
- Rijal, M., Muhammad, Y. A., & Rahman Abdul. (2024). DETEKSI CITRA DAUN UNTUK KLASIFIKASI PENYAKIT PADI MENGGUNAKAN PENDEKATAN DEEP LEARNING DENGAN MODEL CNN. *Jurnal Teknologi Terpadu Vol. 10 No. 1 2024*, 56-62.
- Sah, A., Korespondensi, P., Desi Alexander, A., & Tanniewa, A. M. (2025). *Pengembangan Model Klasifikasi Citra Penyakit Daun Lada Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ)*. <https://doi.org/10.58602/jima-ilkom.v4i1.53>
- Soebroto, A. A. (2021). *Buku Ajar AI, Machine Learning & Deep Learning*. <https://www.researchgate.net/publication/348003841>
- Sonya, P., Ropalia, dan, Biologi, J., Pertanian, F., dan Biologi, P., Bangka Belitung Jl Kampus Peradaban, U., Terpadu UBB, K., Bangka, K., Bangka Belitung, K., Agroteknologi Fakultas Pertanian, J., & Biologi Universitas Bangka Belitung Jl Kampus Peradaban, dan. (2023). Perbandingan Struktur Anatomi dan Kadar Klorofil antara Daun Terinfeksi dan Tidak Terinfeksi Penyakit Kuning pada Tanaman Lada. *Jurnal Agrikultura*, 2023(1), 58–65.
- ST NURJANNAH MJ. (2021). *PENGARUH PRODUKSI MERICA TERHADAP PENDAPATAN EKONOMI MASYARAKAT DI DESA SAOTENGNGA SKRIPSI*.
- Teresia Ompusunggu, P. (2022). Klasifikasi Penyakit Tanaman Pada Daun Kentang Dengan Metode Convolutional Neural Network Arsitektur Mobilenet. *Jurnal Syntax Fusion*, 2(09), 740–751. <https://doi.org/10.54543/fusion.v2i09.217>
- Yazid, M., Puteri, R., Ramayanti, S., Likuina, M. R., Simbolon, J., Mangara, B., & Simatupang, T. (2023). *SERI MANAJEMEN AGRIBISNIS AGRIBISNIS LADA*.

Lampiran 1. Surat Izin Penelitian


PEMERINTAH PROVINSI SULAWESI SELATAN
DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU
Jl. Bougenville No.5 Telp. (0411) 441077 Fax. (0411) 448936
Website : <http://simap-new.sulselprov.go.id> Email : ptsp@sulselprov.go.id
Makassar 90231

Nomor	: 17539/S.01/PTSP/2025	Kepada Yth.
Lampiran	: -	Kepala Dinas Tanaman Pangan, Hortikultura dan Perkebunan Provinsi Sulawesi Selatan
Perihal	: <u>Izin penelitian</u>	

di-
Tempat

Berdasarkan surat Ketua LP3M UNISMUH Makassar Nomor : 276/LP3M/05/C.4-VIII/VIII/1447/2025 tanggal 11 Agustus 2025 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

N a m a	: RESKI ANUGRAH SARI
Nomor Pokok	: 105841104421
Program Studi	: Informatika
Pekerjaan/Lembaga	: Mahasiswa (S1)
Alamat	: Jl. Sultan Alauddin No. 259 Makassar PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI, dengan judul :

" DETEKSI DINI PENYAKIT TANAMAN MERICA MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS CITRA DAUN "

Yang akan dilaksanakan dari : Tgl. **12 Agustus s/d 12 September 2025**

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami **menyetujui** kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar
Pada Tanggal 12 Agustus 2025

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU
SATU PINTU PROVINSI SULAWESI SELATAN

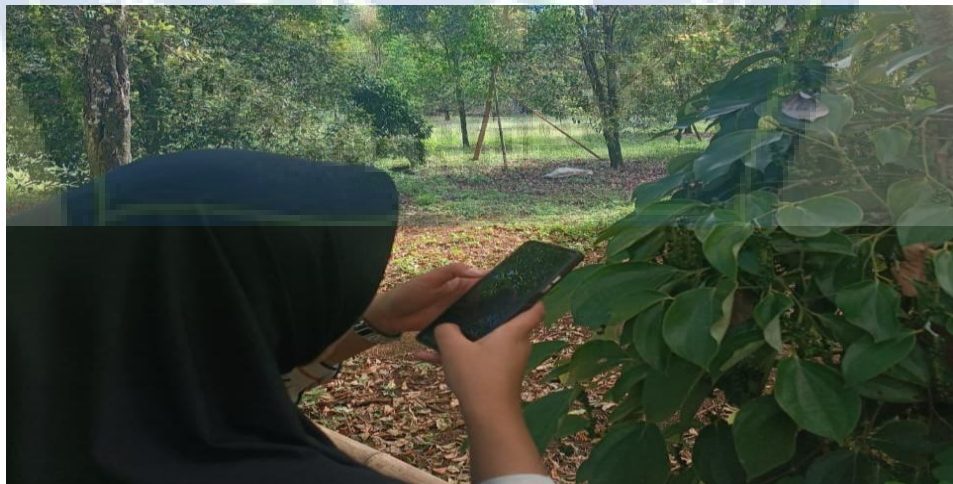
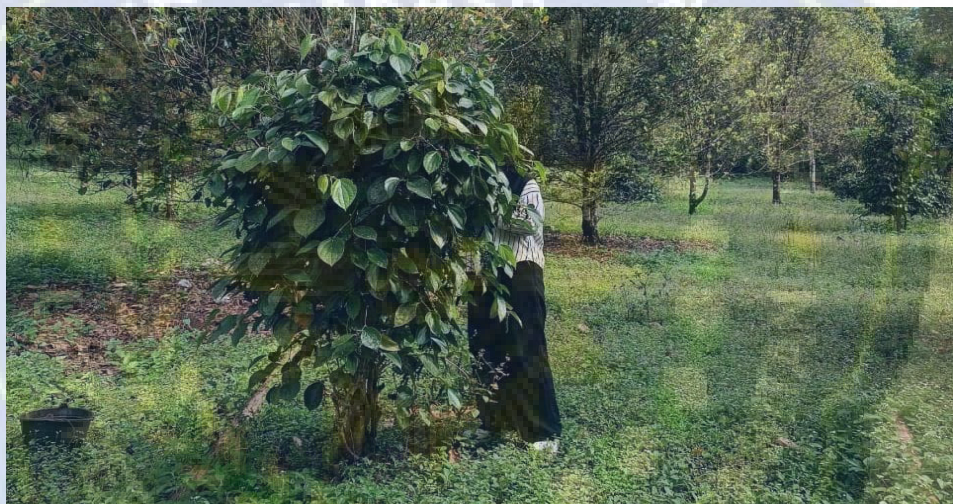


ASRUL SANI, S.H., M.Si.
Pangkat : PEMBINA UTAMA MUDA (IV/c)
Nip : 19750321 200312 1 008

Tembusan Yth

1. Ketua LP3M UNISMUH Makassar di Makassar;
2. Pertinggal.

Lampiran 2. Dokumentasi Penelitian



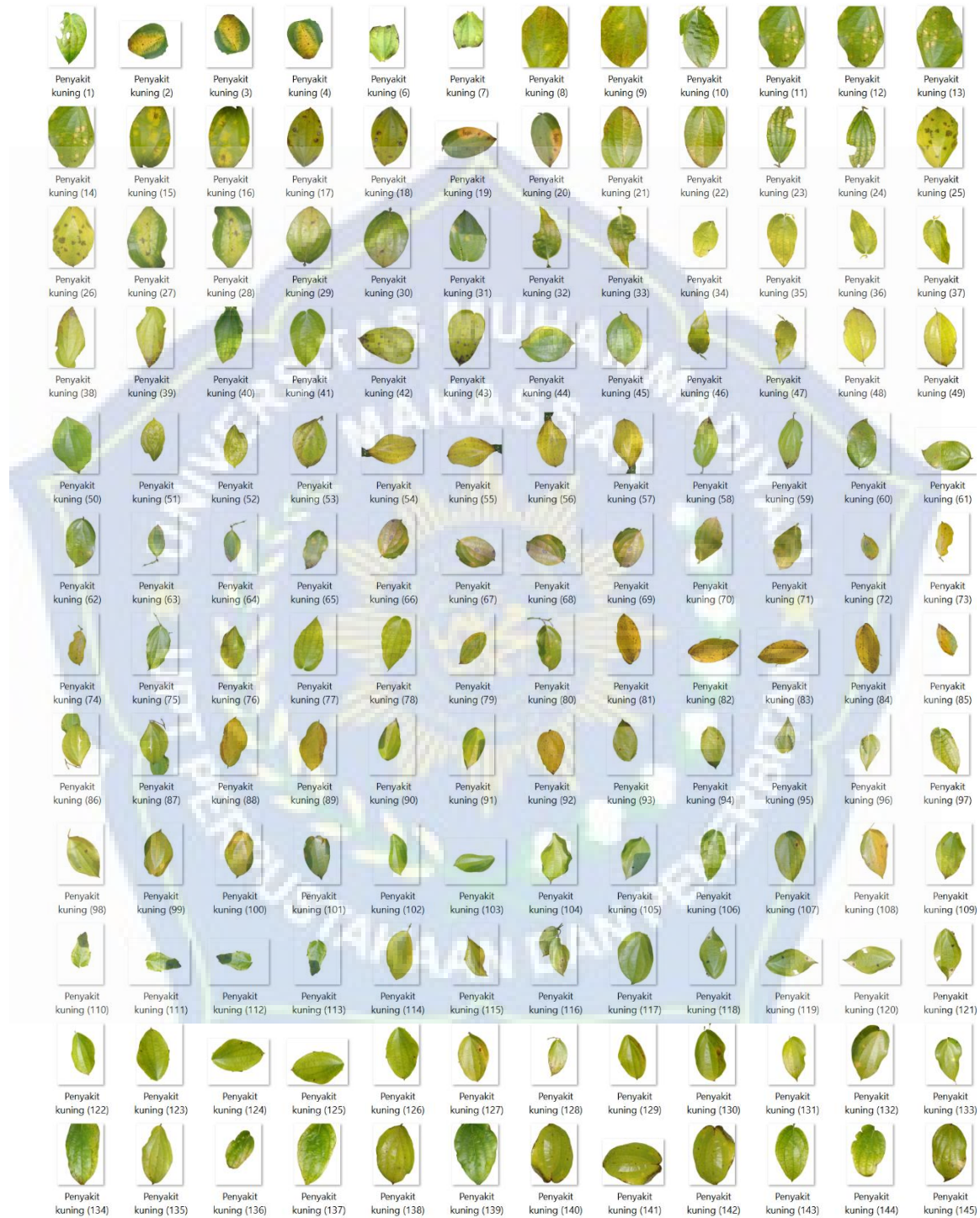
Lampiran 3. Dataset Citra Tiap Kelas

Kelas Bercak Daun



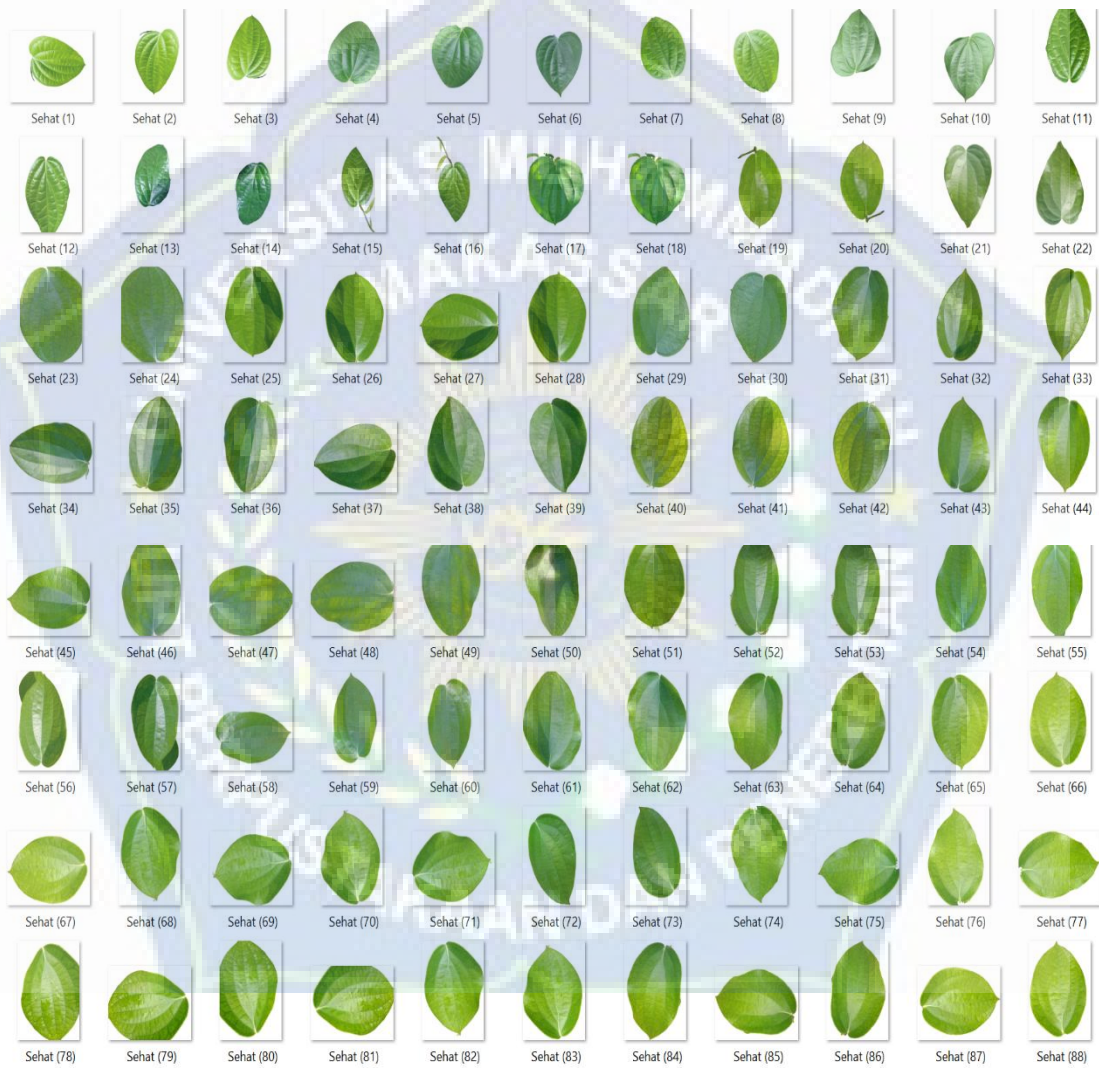


Kelas Penyakit Kuning





Kelas Sehat





Kelas Bukan daun merica



Lampiran 4. Source Code

Pengembangan Model

```
import os
import shutil
import random
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import json
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report, confusion_matrix

sumber_dir = 'data_sumber'
dataset_dir = 'dataset'

classes = ['Bercak daun', 'Bukan daun merica', 'Busuk Pangkal Batang',
'daun_sehat', 'penyakit kuning']

IMG_SIZE = (224, 224)
BATCH_SIZE = 32
EPOCHS = 30

def prepare_dataset():
    """
    Membuat struktur direktori train/validation/test dan membagi data
    dari sumber_dir ke direktori tersebut dengan rasio 70/20/10.
    Fungsi ini sekarang hanya berjalan jika direktori dataset_split
    belum ada atau kosong.
    """
    train_path = os.path.join(dataset_dir, 'train')
    if os.path.exists(dataset_dir) and os.path.exists(train_path) and
os.listdir(train_path):
        print(f"Direktori '{dataset_dir}' sudah ada dan berisi data.
Melewati tahap persiapan.")
```

```

        return

    print(f"Membuat dan membagi dataset ke '{dataset_dir}'...")
    if os.path.exists(dataset_dir):
        shutil.rmtree(dataset_dir)
    os.makedirs(dataset_dir)

    splits = ['train', 'validation', 'test']
    for split in splits:
        split_path = os.path.join(dataset_dir, split)
        os.makedirs(split_path)
        for cls in classes:
            os.makedirs(os.path.join(split_path, cls))

    for cls in classes:
        print(f"Memproses kelas: {cls}")
        cls_path = os.path.join(sumber_dir, cls)
        if not os.path.isdir(cls_path):
            print(f"Peringatan: Direktori untuk kelas '{cls}' tidak
ditemukan di '{sumber_dir}'.")
            continue

        images = [f for f in os.listdir(cls_path) if
os.path.isfile(os.path.join(cls_path, f))]
        random.shuffle(images)

        train_count = int(len(images) * 0.70)
        validation_count = int(len(images) * 0.20)

        for i, img in enumerate(images):
            sumber_file = os.path.join(cls_path, img)
            if i < train_count:
                tujuan_dir = os.path.join(dataset_dir, 'train', cls)
            elif i < train_count + validation_count:
                tujuan_dir = os.path.join(dataset_dir, 'validation',
cls)
            else:
                tujuan_dir = os.path.join(dataset_dir, 'test', cls)

            shutil.copy(sumber_file, os.path.join(tujuan_dir, img))

```

```
print("Pemisahan dataset selesai.")
```

```
def create_data_generators():
```

```
    """
```

```
    Membuat generator data untuk training (dengan augmentasi) dan
    validasi/testing (tanpa augmentasi).
```

```
    """
```

```
    train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=30,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        brightness_range=[0.8, 1.2],
        fill_mode='nearest'
    )
```

```
    test_val_datagen = ImageDataGenerator(rescale=1./255)
```

```
    train_generator = train_datagen.flow_from_directory(
        os.path.join(dataset_dir, 'train'),
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical'
    )
```

```
    validation_generator = test_val_datagen.flow_from_directory(
        os.path.join(dataset_dir, 'validation'),
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical',
        shuffle=False
    )
```

```
    test_generator = test_val_datagen.flow_from_directory(
        os.path.join(dataset_dir, 'test'),
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
```

```

        class_mode='categorical',
        shuffle=False
    )

    return train_generator, validation_generator, test_generator

```

```

def build_model(num_classes):
    """
    Membangun model CNN menggunakan arsitektur MobileNetV2 pre-trained.
    """
    base_model = MobileNetV2(weights='imagenet', include_top=False,
input_shape=(IMG_SIZE[0], IMG_SIZE[1], 3))
    base_model.trainable = False

    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dense(1024, activation='relu')(x)
    predictions = Dense(num_classes, activation='softmax')(x)

    model = Model(inputs=base_model.input, outputs=predictions)
    model.compile(optimizer=Adam(learning_rate=0.0001),
loss='categorical_crossentropy', metrics=['accuracy'])

    return model

def plot_training_history(history, model_name, result_dir):
    """
    Membuat plot akurasi dan loss selama training dan validasi.
    """
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs_range = range(len(acc))

    plt.figure(figsize=(14, 5))

    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')

```

```

plt.legend(loc='lower right')
plt.title(f'Training and Validation Accuracy - {model_name}')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title(f'Training and Validation Loss - {model_name}')

plt.savefig(os.path.join(result_dir,
f"accuracy_loss_plot_{model_name}.png"))
plt.show()

def plot_confusion_matrix(y_true, y_pred, class_names, model_name,
result_dir):
    """
    Membuat plot confusion matrix.
    """
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=class_names, yticklabels=class_names)
    plt.title(f'Confusion Matrix - {model_name}')
    plt.ylabel('Actual Class')
    plt.xlabel('Predicted Class')
    plt.savefig(os.path.join(result_dir,
f"confusion_matrix_{model_name}.png"))
    plt.show()

def plot_per_class_metrics(report_dict, model_name, result_dir):
    """
    Membuat plot untuk precision, recall, dan f1-score per kelas dari
    dictionary.
    """

    class_metrics = {k: v for k, v in report_dict.items() if
isinstance(v, dict)}

    class_metrics.pop('accuracy', None)
    class_metrics.pop('macro avg', None)
    class_metrics.pop('weighted avg', None)

```

```

    if not class_metrics:
        print("Tidak dapat mengekstrak data laporan klasifikasi untuk
diplot.")
        return

```

```

df = pd.DataFrame(class_metrics).T
df = df[['precision', 'recall', 'f1-score']]

df.plot(kind='bar', figsize=(12, 6))
plt.title(f'Per Class Metrics - {model_name}')
plt.ylabel('Score')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--')
plt.tight_layout()
plt.savefig(os.path.join(result_dir,
f"classification_metrics_plot_{model_name}.png"))
plt.show()

def visualize_predictions(model, test_generator, class_names,
model_name, result_dir):
    """
    Menampilkan beberapa gambar dari test set beserta prediksi model.
    """
    images, labels = next(test_generator)
    predictions = model.predict(images)

    plt.figure(figsize=(15, 10))
    for i in range(min(9, len(images))):
        plt.subplot(3, 3, i + 1)
        plt.imshow(images[i])
        true_label = class_names[np.argmax(labels[i])]
        pred_label = class_names[np.argmax(predictions[i])]
        plt.title(f"True: {true_label}\nPred: {pred_label}")
        plt.axis('off')
    plt.suptitle(f"Sample Predictions - {model_name}")
    plt.savefig(os.path.join(result_dir,
f"prediction_examples_{model_name}.png"))
    plt.show()

```

```

if __name__ == '__main__':
    result_dir = 'results'
    models_dir = 'models'

    if not os.path.exists(result_dir):
        os.makedirs(result_dir)
    if not os.path.exists(models_dir):
        os.makedirs(models_dir)

    prepare_dataset()

    train_dir_path = os.path.join(dataset_dir, 'train')
    if not os.path.exists(train_dir_path) or not
os.listdir(train_dir_path):
        print(f"Error: Direktori training '{train_dir_path}' kosong
atau tidak ada. Hentikan eksekusi.")
        exit()

    train_datagen_no_aug = ImageDataGenerator(rescale=1./255)
    train_gen_A = train_datagen_no_aug.flow_from_directory(
        train_dir_path,
        target_size=IMG_SIZE,
        batch_size=BATCH_SIZE,
        class_mode='categorical'
    )

    _, val_gen, test_gen = create_data_generators()

    train_gen_B, _, _ = create_data_generators()

    class_names = list(train_gen_A.class_indices.keys())
    num_classes = len(class_names)

    print("\n--- Memulai Skenario A: Tanpa Augmentasi Data ---")
    model_A = build_model(num_classes)
    history_A = model_A.fit(
        train_gen_A,
        epochs=EPOCHS,
        validation_data=val_gen,
        verbose=1

```

```

)

print("\n--- Memulai Skenario B: Dengan Augmentasi Data ---")
model_B = build_model(num_classes)
history_B = model_B.fit(
    train_gen_B,
    epochs=EPOCHS,
    validation_data=val_gen,
    verbose=1
)

print("\n--- Hasil Skenario A (Tanpa Augmentasi) ---")
loss_A, acc_A = model_A.evaluate(test_gen)
print(f"Test Accuracy (Model A): {acc_A*100:.2f}%")

y_pred_A = np.argmax(model_A.predict(test_gen), axis=1)
y_true_A = test_gen.classes

report_A_str = classification_report(y_true_A, y_pred_A,
target_names=class_names)
report_A_dict = classification_report(y_true_A, y_pred_A,
target_names=class_names, output_dict=True)
print("Classification Report (Model A):\n", report_A_str)

with open(os.path.join(result_dir,
"classification_report_scenario_A.txt"), "w") as f:
    f.write(f"Test Accuracy: {acc_A*100:.2f}%\n\n")
    f.write("Classification Report:\n")
    f.write(report_A_str)

model_name_A = "scenario_A"
plot_training_history(history_A, model_name_A, result_dir)
plot_confusion_matrix(y_true_A, y_pred_A, class_names,
model_name_A, result_dir)
plot_per_class_metrics(report_A_dict, model_name_A, result_dir)
visualize_predictions(model_A, test_gen, class_names, model_name_A,
result_dir)

print(f"Menyimpan Model A ke {models_dir}...")
model_A.save(os.path.join(models_dir,
"pepper_disease_model_scenario_A.h5"))

```

```

print("\n--- Hasil Skenario B (Dengan Augmentasi) ---")
loss_B, acc_B = model_B.evaluate(test_gen)
print(f"Test Accuracy (Model B): {acc_B*100:.2f}%")

y_pred_B = np.argmax(model_B.predict(test_gen), axis=1)
y_true_B = test_gen.classes

report_B_str = classification_report(y_true_B, y_pred_B,
target_names=class_names)
report_B_dict = classification_report(y_true_B, y_pred_B,
target_names=class_names, output_dict=True)
print("Classification Report (Model B):\n", report_B_str)

with open(os.path.join(result_dir,
"classification_report_scenario_B.txt"), "w") as f:
    f.write(f"Test Accuracy: {acc_B*100:.2f}%\n\n")
    f.write("Classification Report:\n")
    f.write(report_B_str)

model_name_B = "scenario_B"
plot_training_history(history_B, model_name_B, result_dir)
plot_confusion_matrix(y_true_B, y_pred_B, class_names,
model_name_B, result_dir)
plot_per_class_metrics(report_B_dict, model_name_B, result_dir)
visualize_predictions(model_B, test_gen, class_names, model_name_B,
result_dir)

print(f"Menyimpan Model B ke {models_dir}...")
model_B.save(os.path.join(models_dir,
"pepper_disease_model_scenario_B.h5"))

class_indices = train_gen_B.class_indices
with open(os.path.join(models_dir, 'class_indices.json'), 'w') as
f:
    json.dump(class_indices, f)

print(f"\nAnalisis Selesai. Gambar-gambar hasil visualisasi dan
laporan telah disimpan di direktori '{result_dir}'.")

```

Proses Klasifikasi

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
import os
import argparse
import json
import textwrap

MODEL_A_PATH = 'models/pepper_disease_model_scenario_A.h5'
MODEL_B_PATH = 'models/pepper_disease_model_scenario_B.h5'
IMG_SIZE = (224, 224)

def load_class_names(model_dir):
    """Memuat pemetaan kelas dari file json dan mengurutkannya
    berdasarkan indeks."""
    indices_path = os.path.join(model_dir, 'class_indices.json')
    try:
        with open(indices_path, 'r') as f:
            class_indices = json.load(f)
            return sorted(class_indices.keys(), key=class_indices.get)
    except FileNotFoundError:
        print(f"Error: File pemetaan kelas '{indices_path}' tidak ditemukan.")
        print("Pastikan Anda telah menjalankan skrip pelatihan (train.py) untuk membuatnya.")
        return None

CLASS_NAMES = load_class_names('models')
if CLASS_NAMES is None:
    exit()

SYMPTOMS = {
    "bercak daun": ["Menimbulkan bercak khas cokelat atau hitam pada daun, yang dapat meluas, menyebabkan daun menguning, dan gugur sebelum waktunya."],
    "busuk pangkal batang": [
```

"Gejala awal berupa bintik kecil hitam/coklat pada daun yang membesar menjadi bercak tidak beraturan, akhirnya menyebabkan kematian jaringan (nekrosis) dan kerontokan daun."

```
],
"penyakit kuning": [
    "Daun berubah warna menjadi kuning secara merata, menjadi kaku, menggantung, lalu melengkung ke arah batang, dan akhirnya rontok dengan cepat."
],
"daun_sehat": ["Daun tampak hijau segar, utuh, dan tidak menunjukkan tanda-tanda kelainan atau kerusakan."],
"bukan daun merica": ["Gambar tidak teridentifikasi sebagai daun merica atau tidak fokus pada objek daun."]
}
```

```
def load_and_preprocess_image(img_path):
    """Memuat dan memproses gambar untuk input model."""
    img = image.load_img(img_path, target_size=IMG_SIZE)
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0
    return img_array

def predict_and_display(img_path, model_a, model_b,
    output_filename=None):
    """Melakukan prediksi dan membuat visualisasi hasil yang cocok untuk skripsi."""
    if not os.path.exists(img_path):
        print(f"Error: File tidak ditemukan di '{img_path}'")
        return

    processed_img = load_and_preprocess_image(img_path)
    img = image.load_img(img_path)
```

```
predictions = {
    'model_a': model_a.predict(processed_img),
    'model_b': model_b.predict(processed_img)
}

results = {}
```

```

for model_key, pred in predictions.items():
    confidence = np.max(pred[0]) * 100
    class_idx = np.argmax(pred[0])
    class_name = CLASS_NAMES[class_idx]

    is_pepper_leaf = "bukan daun merica" not in class_name.lower()

    results[model_key] = {
        'confidence': confidence,
        'is_pepper_leaf': is_pepper_leaf,
        'identification': "Daun Merica" if is_pepper_leaf else
"Bukan Daun Merica",
        'status': class_name if is_pepper_leaf else "-",
        'symptoms': SYMPTOMS.get(class_name, ["-"])
    }

fig = plt.figure(figsize=(11, 7))
fig.suptitle(f"Analisis Klasifikasi: {os.path.basename(img_path)}",
fontsize=16, weight='bold')
gs_main = fig.add_gridspec(1, 2, width_ratios=[1.7, 1], wspace=0.3)

ax_img = fig.add_subplot(gs_main[0])
ax_img.imshow(img)
ax_img.axis('off')

gs_info = gs_main[1].subgridspec(2, 1, hspace=0.4)

def create_info_box(ax, model_name, result, color):
    ax.set_facecolor(color)
    ax.tick_params(axis='both', which='both', left=False,
bottom=False, labelleft=False, labelbottom=False)
    for spine in ax.spines.values():
        spine.set_edgecolor('gray')
        spine.set_linewidth(0.8)

    title = f"{model_name}"
    confidence_text = f"({result['confidence']:.2f})%"
    identification_text = f"Identifikasi:
{result['identification']}"
    status_text = f"Status: {result['status']}"

```

```

        ax.text(0.07, 0.90, title, ha='left', va='top', fontsize=11,
weight='bold', transform=ax.transAxes)
        ax.text(0.07, 0.75, identification_text, ha='left', va='top',
fontsize=10, transform=ax.transAxes)
        ax.text(0.93, 0.75, confidence_text, ha='right', va='top',
fontsize=9, color='gray', transform=ax.transAxes)

        if result['is_pepper_leaf']:
            ax.text(0.07, 0.60, status_text, ha='left', va='top',
fontsize=10, transform=ax.transAxes)
            ax.text(0.07, 0.45, "Gejala:", ha='left', va='top',
fontsize=10, style='italic', transform=ax.transAxes)
            symptoms_text =
"\n".join(textwrap.wrap(result['symptoms'][0], width=40,
initial_indent=' \u2022 ', subsequent_indent=' '))
            ax.text(0.07, 0.35, symptoms_text, ha='left', va='top',
fontsize=9, transform=ax.transAxes)

        ax_a = fig.add_subplot(gs_info[0])
        create_info_box(ax_a, "Model A (Tanpa Augmentasi)",
results['model_a'], '#eaf4ff')

        ax_b = fig.add_subplot(gs_info[1])
        create_info_box(ax_b, "Model B (Dengan Augmentasi)",
results['model_b'], '#f0fff0')

        plt.tight_layout(rect=[0, 0, 1, 0.95])

        if output_filename:
            try:
                fig.savefig(output_filename, dpi=300, bbox_inches='tight',
pad_inches=0.2)
                print(f"\n[+] Hasil visual disimpan ke: {output_filename}")
            except Exception as e:
                print(f"\n[!] Gagal menyimpan gambar: {e}")

        plt.show()

        print("\n" + "="*70)
        print("HASIL KLASIFIKASI GAMBAR".center(70))

```

```

print("="*70)
print(f"[*] Gambar Dianalisis: {os.path.basename(img_path)}")
print("-"*70)

def print_console_report(model_name, result):
    print(f"\n--- {model_name} ---")
    print(f"> Identifikasi : {result['identification']}")
    print(f"({result['confidence']:.2f}%)")
    if result['is_pepper_leaf']:
        print(f"> Status : {result['status']}")
        print("> Gejala Terkait:")
        wrapper = textwrap.TextWrapper(width=65,
initial_indent='    - ', subsequent_indent='    ')
        print(wrapper.fill(result['symptoms'][0]))

    print_console_report("MODEL A (Tanpa Augmentasi)",
results['model_a'])
    print_console_report("MODEL B (Dengan Augmentasi)",
results['model_b'])
    print("\n" + "="*70 + "\n")

if __name__ == '__main__':
    parser = argparse.ArgumentParser(
        description="Klasifikasi gambar penyakit daun merica dan
        hasilkan visual resolusi tinggi.",
        formatter_class=argparse.RawTextHelpFormatter
    )
    parser.add_argument("image_path", nargs='?', default=None,
help="Path ke gambar yang ingin diuji.")
    parser.add_argument("-o", "--output", default=None, help="Path
    untuk menyimpan gambar hasil (contoh: hasil/output.png).")
    args = parser.parse_args()

    print("Memuat model...")
    try:
        model_A = tf.keras.models.load_model(MODEL_A_PATH,
compile=False)
        model_B = tf.keras.models.load_model(MODEL_B_PATH,
compile=False)
        print("Model berhasil dimuat.")
    except Exception as e:

```

```

print(f"Error saat memuat model: {e}")
exit()

def get_output_path(img_path, output_arg):
    if output_arg:

        os.makedirs(os.path.dirname(output_arg), exist_ok=True)
        return output_arg

    os.makedirs("results", exist_ok=True)
    base, ext = os.path.splitext(os.path.basename(img_path))
    return os.path.join("results",
f"{base}_classification_result.png")

if args.image_path:
    output_path = get_output_path(args.image_path, args.output)
    predict_and_display(args.image_path, model_A, model_B,
output_filename=output_path)
else:
    while True:
        try:
            img_path_input = input("\nMasukkan path gambar (atau
'exit'): ").strip()
            if img_path_input.lower() == 'exit':
                break
            if not os.path.exists(img_path_input):
                print(f"Error: File tidak ditemukan di
'{img_path_input}'")
                continue

            output_path_loop = get_output_path(img_path_input,
args.output)
            predict_and_display(img_path_input, model_A, model_B,
output_filename=output_path_loop)
        except (KeyboardInterrupt, EOFError):
            print("\nKeluar dari program.")
            break

```

Lampiran 5. Hasil Turnitin

 **MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH**
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat Kantor : Jl. Sultan Alauddin No.259 Makassar 90221 Tlp. (0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Reski Anugrah Sari
Nim : 105841104421
Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	8%	10 %
2	Bab 2	2%	25 %
3	Bab 3	2%	10 %
4	Bab 4	0%	10 %
5	Bab 5	2%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 26 Agustus 2025
Mengetahui,
Kepala UPT- Perpustakaan dan Penerbitan,


Nursinah S. Hani, M.P.
NBM-964.591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id



BAB I Reski Anugrah Sari 105841104421

ORIGINALITY REPORT

6%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.its.ac.id
Internet Source

2%

2

digilibadmin.unismuh.ac.id
Internet Source

2%

3

Submitted to INTI Universal Holdings SDM
BHD
Student Paper

2%

Exclude quotes On

Exclude bibliography On

Exclude matches < 2%



BAB II Reski Anugrah Sari

105841104421

by Tahap Tutup

Submission date: 26-Aug-2025 11:13AM (UTC+0700)

Submission ID: 2735414894

File name: BAB_II_15.docx (3,24M)

Word count: 2521

Character count: 16679

BAB II Reski Anugrah Sari 105841104421

ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

0%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.penerbiteurka.com
Internet Source

2%

Exclude quotes

On

Exclude matches

< 2%

Exclude bibliography

On

BAB III Reski Anugrah Sari 105841104421

by Tahap Tutup

Submission date: 26-Aug-2025 11:15AM (UTC+0700)
Submission ID: 2735415556
File name: BAB_III_19.docx (271.04K)
Word count: 2529
Character count: 16280

BAB III Reski Anugrah Sari 105841104421

ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES



kc.umn.ac.id
Internet Source

2%

Exclude quotes

0%

Exclude bibliography

0%

Exclude matches

< 2%



BAB IV Reski Anugrah Sari

105841104421

by Tahap Tutup

Submission date: 26-Aug-2025 11:16AM (UTC+0700)

Submission ID: 2735416209

File name: BAB_IV_11.docx (2.59M)

Word count: 3477

Character count: 20893

BAB IV Reski Anugrah Sari 105841104421

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

Exclude bibliography

Exclude matches





BAB V Reski Anugrah Sari

105841104421

by Tahap Tutup

Submission date: 26-Aug-2025 11:17AM (UTC+0700)

Submission ID: 2735416416

File name: BAB_V_14.docx (120.9K)

Word count: 342

Character count: 2357

BAB V Reski Anugrah Sari 105841104421

ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

2%

★ e-journal.uajy.ac.id

Internet Source

Exclude quotes

Exclude bibliography

Exclude matches

2%