

**KLASIFIKASI JENIS-JENIS TANAMAN HIAS DAUN
DENGAN ALGORITMA CONVOLUTIONAL NEURAL
NETWORK (CNN)**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Menyusun Skripsi Program Studi
Informatika



ANDI DIAN FAJARWATI 105841110620

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025**

**KLASIFIKASI JENIS-JENIS TANAMAN HIAS DAUN
DENGAN ALGORITMA CONVOLUTIONAL NEURAL
NETWORK (CNN)**

Diajukan Sebagai Salah Satu Syarat Untuk Menyusun Skripsi Program Studi
Informatika

Disusun dan Diajukan Oleh:

ANDI DIAN FAJARWATI 105841110620

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025**



FAKULTAS TEKNIK

PENGESAHAN

Skripsi atas nama Andi Dian Fajarwati dengan nomor induk Mahasiswa 105 84 11106 20, 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

Makassar,

6 Rabi'ul Awwal 1447 H

30 Agustus 2025 M

a. Rektor Universitas Muhammadiyah Makassar

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

b. Dekan Fakultas Teknik Universitas Hasanuddin

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

2. Penguji

a. Ketua : Prof. Dr. Ir. Hafsa Nirwana, M.T.

b. Sekretaris : Ruzal Rezkiawan B.S.Kom., M.T.

3. Anggota

1. Fahrini Ibrahimna Rachman, S.Kom., M.T.

2. Desi Anggreani, S.Kom., M.T.

3. Lukman, S.Kom., M.T.

Mengetahui

Pembimbing I

Pembimbing II

Muhyiddin A M Hayat, S.Kom., M.T.

Titik Wahyuni, Spd., M.T.

Dekan



Dr. M. Syafaat S. Rupa, S.T., M.T.

NBM : 796 288





FAKULTAS TEKNIK

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 : **KLASIFIKASI JENIS TANAMAN HIAS DAUN DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)**

Nama : **ANDI DIAN FAJARWATI**

Stambuk : **105 84 11106 20**

Makassar, 09 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Muhyiddin A M Hayat, S.Kom., M.T

Titin Wahyuni, S.Pd., M.T

Mengetahui,
Ketua Prodi Informatika



Rizki Yushiana Bakti, S.T., M.T.

NPM : 1807 284



MOTTO DAN PERSEMBAHAN

Motto

"Dan bersabarlah, sesungguhnya Allah tidak menyalah-nyaikan pahala orang-orang yang berbuat kebaikan."

(Q.S Al-Hadid:4)

Persembahan

Setiap lembar dalam skripsi ini adalah wujud ikhtiar dan ketekunan, namun izinkan penulis menghadirkan lembar khusus ini sebagai ungkapan terima kasih dan penghargaan. Bismillahirrahmanirrahim, skripsi ini dengan tulus dipersembahkan untuk Baso pasenuangi dan Andi hartawati, dua sosok yang tak pernah berhenti melimpahkan kasih sayang, doa tulus, dan dukungan tanpa batas yang menjadi fondasi utama dalam setiap langkah dan penyelesaian skripsi ini. kepada teman-teman seperjuangan: Sri asmi, Nurfadillah, dan fatimah, terima kasih atas tawa, dukungan, dan kebersamaan yang berarti. Untuk keluarga besar informatika 20 khususnya kelas D, terima kasih atas semangat dan perjalanan yang dijalani bersama. Tak lupa, terima kasih kepada seluruh pihak yang tidak dapat disebutkan satu per satu, atas segala dukungan dan doa yang tulus. Dan akhir, untuk diriku sendiri, terima kasih telah bertahan, terus belajar, dan berjuang hingga titik ini, percaya bahwa setiap niat baik dan harapan akan selalu diberikan kemudahan oleh-Nya. Semoga persembahan ini bermanfaat bagi pengembangan ilmu pengetahuan dan menjadi bagian dari amal kebaikan. Selamat bergelar S.Kom.

ABSTRAK

ANDI DIAN FAJARWATI, Klasifikasi jenis-jenis tanaman hias daun dengan algoritma convolutional neural network (dibimbing oleh Muhyiddin A M Hayat, S.Kom., M.T dan Titin Wahyuni, S.Pd., M.T).

Tanaman hias daun memiliki nilai estetika sekaligus manfaat ekologis, seperti meningkatkan kualitas udara dan menciptakan suasana lingkungan yang lebih asri, namun klasifikasinya secara manual masih cukup sulit dilakukan karena adanya kemiripan visual antarspesies dari segi bentuk, warna, maupun pola daun. Penelitian ini bertujuan mengembangkan sistem klasifikasi tanaman hias daun menggunakan Convolutional Neural Network (CNN) berbasis arsitektur InceptionV3 dengan dataset berisi 2.133 citra dari delapan kelas yang dibagi menjadi data latih, validasi, dan uji dengan proporsi 80:10:10. Pelatihan dilakukan dalam dua tahap, yaitu Stage-1 dengan backbone model dibekukan untuk melatih lapisan atas, dan Stage-2 dengan fine-tuning pada sebagian layer agar model dapat mempelajari fitur yang lebih mendalam. Hasil eksperimen menunjukkan akurasi validasi sebesar 93,3% pada Stage-1 dan meningkat menjadi 95,1% setelah fine-tuning pada Stage-2, sementara evaluasi pada data uji menghasilkan akurasi 94,6% dengan precision, recall, dan F1-score rata-rata di atas 0,94. Confusion matrix memperlihatkan sebagian besar kelas dapat teridentifikasi dengan benar meskipun masih terdapat kesalahan kecil pada kelas dengan kemiripan visual tinggi, sehingga secara keseluruhan CNN berbasis InceptionV3 terbukti andal dalam mendukung identifikasi tanaman hias daun secara otomatis dan akurat.

Kata kunci: Tanaman hias daun, CNN, Inception V3, Transfer learning, Klasifikasi citra

ABSTRACT

Foliage ornamental plants have both aesthetic value and ecological benefits, such as improving air quality and creating a greener environment, but their classification is still difficult to perform manually due to visual similarities between species in terms of shape, color, and leaf patterns. This study aims to develop a classification system for foliage ornamental plants using a Convolutional Neural Network (CNN) based on the InceptionV3 architecture with a dataset of 2,133 images from eight classes, divided into training, validation, and testing sets with a ratio of 80:10:10. The training was carried out in two stages, namely Stage-1 with the backbone frozen to train the top layers, and Stage-2 with fine-tuning on selected layers to enable deeper feature learning. The experimental results show that the model achieved a validation accuracy of 93.3% in Stage-1 and increased to 95.1% after fine-tuning in Stage-2, while evaluation on the test data achieved an accuracy of 94.6% with average precision, recall, and F1-score values above 0.94. The confusion matrix indicates that most classes were correctly identified, although minor errors occurred in visually similar categories, thus overall the InceptionV3-based CNN proves to be a reliable approach for supporting automatic and accurate classification of foliage ornamental plants.

Keywords :

Ornamental foliage plants, CNN, InceptionV3, image classification, deep learning.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabaraktuh

Dengan penuh rasa syukur ke hadirat Allah Subhanallahu Wa Ta'ala, atas nikmat iman, Islam, dan kesehatan yang senantiasa terlimpahkan, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **"Klasifikasi Jenis-Jenis Tanaman Hias Daun Dengan Algoritam Convolutional Neural Network "**. Shalawat serta salam teriring kepada junjungan Nabi Muhammad SAW, pembawa rahmat bagi semesta alam, yang telah membawa kita dari Zaman jahiliah menuju Zaman yang serba modern seperti saat ini.

Ucapan terima kasih yang tidak terhingga penulis sampaikan kepada semua pihak yang telah membantu dan memberikan dukungan dalam penyusunan Laporan Tugas Akhir ini, khususnya:

1. Ibu **Dr.Ir.Hj Nurnawati, S.T., M.T., I.P.M.**, selaku Dekan Fakultas Teknik.
2. Bapak **Muh. Syafaat S Kuba, S.T., M.T.**, selaku Wakil Dekan Fakultas Teknik.
3. Ibu **Rizki Yusliana Bakti,S.T.,M.T.**, selaku Ketua Prodi Informatika.
4. Pak **Muhyiddin A M Hayat S.Kom., M.T** selaku Dosen Pembimbing 1.
5. Ibu **Titin Wahyuni, S. Pd., MT.** , selaku Dosen Pembimbing 2 .
6. Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar.
7. Kepada kedua orang tua tercinta saya, Bapak **Baso Pasennangi.A** dan Ibu **Andi Hartawati**, persembahkan terima kasih yang tak terhingga atas segala pengorbanan, kasih sayang, dan bimbingan, yang telah diberikan.
8. Teman-teman Khususnya Angkatan 2020 Fakultas Teknik, Universitas Muhammadiyah Makassar, terima kasih atas dukungan dan doanya.
9. Teman-teman kelas D angkatan 2020 Program Studi Informatika Universitas Muhammadiyah Makassar

Penulis menyadari bahwa Laporan Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan laporan ini di masa depan.

Harapan penulis, semoga Laporan Tugas Akhir ini dapat memberikan manfaat bagi penyandang disabilitas tunanetra dalam meningkatkan kemandirian dan kualitas hidup mereka.

Akhir kata, penulis mohon maaf atas segala kekurangan dan kekhilafan yang terdapat dalam Laporan Tugas Akhir ini.

Billahi fisabililhaq, fastabiqul khairat. Wassalamualaikum Wr. Wb.

Makassar, 10 September 2025

Penulis,

ANDI DIAN FAJARWATI



DAFTAR ISI

MOTTO DAN PERSEMBAHAN	v
ABSTRAK	vi
<i>ABSTRACT</i>	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
DAFTAR ISTILAH	xvi
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan penelitian	3
D. Manfaat penelitian	3
E. Ruang lingkup penelitian	3
F. Sistematika penulisan	4
BAB II TINJAUAN PUSTAKA	5
A. Landasan teori	5
B. Penelitian terkait	19
C. Kerangka Pikir	21
BAB III METODE PENELITIAN	22
A. Tempat dan waktu penelitian	22
B. Alat dan bahan	22
C. Arsitektur Convolutional Neural Network	22
D. Perancangan system	23
E. Teknik pengujian system	25
F. Teknik Analisis data	26

BAB IV HASIL DAN PEMBAHASAN	27
A. Pengumpulan Dataset	27
B. Pelabelan	28
C. Proses pengolahan data	29
D. Model convolutional neural network	31
E. Proses training	33
G. Hasil pengujian/Evaluasi model	37
BAB V PENUTUP	46
A. Kesimpulan	46
B. Saran	47
DAFTAR PUSTAKA	48
LAMPIRAN	50



DAFTAR GAMBAR

Gambar 1. Calathea white fusion (<i>Hamdani et al., 2025</i>)	6
Gambar 2. Dieffenbachia brazil. (Jardineries Botanix). (n.d.).....	7
Gambar 3. Dieffenbachia Camille (PictureThis,n.d.).	8
Gambar 4. Dieffenbachia white etna. (Zahra, 2023).....	8
Gambar 5. Dieffenbachia Rebecca.(Zahra, 2023).....	9
Gambar 6. Mangkokan Variegata. (Dreamstime, 2025)	10
Gambar 7. Polyscias Guilfoylei. (<i>New World Wildflowers</i> , 2025).....	10
Gambar 8. Syngonium Wendlandii (PictureThis,n.d.).	11
Gambar 9. Arsitektur Convolutional Neural Network. (Putri Navia Rena, 2019).14	
Gambar 10. Convolution layer (Medium Samuel Sena,2017).....	17
Gambar 11. Max pooling (Suartika, n.d.)	18
Gambar 12. Diagram alur InceptionV3.....	24
Gambar 13. Dataset Calathea white fusion	27
Gambar 14. dataset diven Rebecca	27
Gambar 15. Dataset Polyscias gulfoylei	27
Gambar 16. Contoh pelabelan dataset	28
Gambar 17. Pembuatan folder	29
Gambar 18. Sebelum dan sesudah augmentasi	30
Gambar 19. Augmentasi data pada setiap operasi	30
Gambar 20. Tahapan CNN InceptionV3	32
Gambar 21. Grafik Training Stage 1.....	35
Gambar 22. Grafik Training Stage 2.....	36
Gambar 23. Confusion Matrix	38

Gambar 24. Prediksi Kesalahan	40
Gambar 25. Confusion matrix CNN dan InceptionV3	44



DAFTAR TABEL

Tabel 1.Dataset tanaman hias	28
Tabel 2.Stage-1 epoch 25	34
Tabel 3. Stage-2 epoch 10	36
Tabel 4. Prediksi Kesalahan	39
Tabel 5. Classification Report	43
Tabel 6.Perbandingan CNN dan Inception	43



DAFTAR LAMPIRAN

Lampiran 1 Source codeSource code.....	50
Lampiran 2. Keterangan bebas plagiasi	61
Lampiran 3.Uji plagiasi	62



DAFTAR ISTILAH



CNN	Convolutional Neural Network (CNN) adalah bentuk pengembangan dari <i>multilayer perceptron</i> (MLP) yang dirancang untuk mengelolah data dua dimensi seperti citra.
Machine Learning	Bidang ilmu yang memungkinkan komputer untuk belajar dari data tanpa diprogram secara eksplisit, digunakan untuk membangun model prediktif.
Pra-pemrosesan	Tahapan penting dalam machine learning di mana data mentah diolah agar siap digunakan dalam pembangunan model. Ini mencakup penghapusan data yang tidak relevan, penanganan nilai yang hilang, dan penyesuaian format data.
Dataset	Kumpulan data yang digunakan untuk melatih dan menguji model machine learning.
Akurasi	Metrik evaluasi yang mengukur seberapa benar model dalam membuat prediksi, dihitung sebagai rasio prediksi yang benar terhadap total prediksi.
Presisi	Metrik evaluasi yang mengukur seberapa banyak dari kasus yang diprediksi positif oleh model benar-benar positif.
Recall	Metrik evaluasi yang mengukur seberapa banyak dari kasus positif yang sebenarnya dapat ditemukan oleh model.

F1-score

Rata-rata harmonik dari presisi dan recall, memberikan gambaran tunggal tentang kinerja model.

Confusion Matrix

Sebuah tabel yang digunakan untuk memvisualisasikan kinerja model klasifikasi, menampilkan jumlah prediksi benar dan salah.

Multilayer Perceptron

Salah satu jenis jaringan saraf tiruan (neural network) yang digunakan untuk tugas-tugas seperti prediksi dan klasifikasi

Augmentasi data

Proses memperbanyak variasi data latih dengan cara transformasi citra, seperti rotasi, flipping, zoom, atau perubahan cahaya

Batch Size

Jumlah sampel data yang diproses sekaligus dalam satu iterasi sebelum parameter model diperbarui.

Epoch

Satu kali perulangan seluruh dataset latih melewati model pada saat pelatihan.

Backbone

Istilah untuk arsitektur dasar dari model pralatih (misalnya InceptionV3, ResNet, VGG, EfficientNet) yang berfungsi mengekstraksi fitur utama dari data

Fine-tuning

Teknik melatih model dengan cara menyesuaikan bobot dari model pralatih (pre-trained model) pada dataset baru

Pooling

Operasi dalam CNN yang berfungsi untuk mengurangi ukuran (dimensi) fitur hasil konvolusi tanpa menghilangkan informasi penting

Average pooling

merata-ratakan nilai pada setiap *feature map*, sehingga tiap channel menghasilkan 1 angka saja.

Dense Layer

Dense Layer (juga disebut Fully Connected Layer) adalah lapisan di jaringan saraf di mana setiap neuron terhubung ke semua neuron di lapisan sebelumnya.

Fungsi Aktivitas

Fungsi matematika yang menentukan output dari sebuah neuron, memperkenalkan non-linearitas ke dalam model

Fungsi Loss

Fungsi yang mengukur perbedaan antara nilai yang diprediksi oleh model dan nilai yang sebenarnya. Tujuannya adalah untuk meminimalkan nilai fungsi ini.

Learning Rate

Parameter yang menentukan seberapa besar langkah yang diambil oleh algoritma optimasi, seperti gradient descent, saat menyesuaikan bobot model.

InceptionV3

adalah salah satu arsitektur Convolutional Neural Network (CNN) yang terkenal, dikembangkan oleh Google.

BAB I

PENDAHULUAN

A. Latar Belakang

Tanaman Hias Daun merupakan salah satu komoditi jenis tanaman yang banyak diminati oleh banyak orang terutama kalangan pecinta tanaman, baik untuk keperluan estetika, dekorasi interior maupun eksterior. Selain nilai estetika tanaman hias daun juga memberikan manfaat ekologi seperti menjaga kelembapan udara, meningkatkan kualitas udara dan menciptakan lingkungan yang lebih sehat. Beriringan dengan meningkatnya ketertarikan pada tanaman hias daun penting bagi masyarakat, kolektor, penjual dan peneliti untuk dapat mengklasifikasikan berbagai jenis tanaman hias daun tersebut. Oleh karena itu, proses klasifikasi memerlukan pengetahuan botanikal sehingga menjadi tantangan bagi orang biasa. Namun mengklasifikasikan tanaman hias daun tidaklah mudah, berbagai jenis tanaman hias daun dengan karakteristik dan variasi dari bentuk daun, warna, tekstur menambah kerumitan dalam proses klasifikasi.

Dengan berkembangnya teknologi pembelajaran mesin khususnya pengolahan citra, sehingga membuka peluang baru untuk memudahkan dalam proses klasifikasi tanaman hias daun. Mengembangkan sistem yang mampu mengklasifikasikan jenis tanaman hias secara akurat, menjadi tantangan bagi para pengembang untuk membuat sistem klasifikasi jenis tanaman hias daun yang dapat memudahkan masyarakat mempelajari jenis tanaman hias daun. (Mareta Tama & Candra Noor Santi, 2023).

Penelitian sebelumnya yang menggunakan CNN di bidang pertanian lebih banyak berfokus pada klasifikasi penyakit tanaman, contohnya pada penelitian klasifikasi jenis mangga berdasarkan bentuk daun dan terstur daun menggunakan CNN (Hidayat, 2022) Dan penelitian Analisis dan implementasi diagnosis penyakit sawit dengan metode CNN (Rasywir et al., 2020a). Kedua penelitian ini menunjukkan efektifitas CNN dalam

mengenali pola citra daun yang terinfeksi penyakit, namun terbatas pada identifikasi penyakit tanaman.

Penelitian yang secara khusus membahas klasifikasi tanaman hias daun masih kurang, padahal tanaman hias daun memiliki karakteristik yang beragam dari segi bentuk dan warna. Keragaman pola dan bentuk tanaman hias daun menjadikan klasifikasi sulit dilakukan secara manual. Variasi visual pada daun tanaman hias yang sangat banyak, baik dari segi bentuk maupun warna, membuat identifikasi manual sering tidak konsisten, sehingga diperlukan pendekatan berbasis CNN yang lebih reliabel.

Dalam membangun sistem klasifikasi penulis membutuhkan metode yang tepat, metode Convolution Neural Network (CNN) merupakan pilihan yang tepat diterapkan pada penelitian ini. Metode Convolution Neural Network adalah salah satu jenis neural network yang biasa digunakan untuk mendeteksi dan mengenali objek. (Ramadhani et al., 2023). Beberapa penelitian terdahulu menunjukkan keberhasilan metode CNN dalam klasifikasi tanaman. Misalnya, penelitian oleh Fadila Huda dan M. Pajar Kharisma putra, (2023), mengembangkan model klasifikasi jenis buah pisang menggunakan Algoritma Convolution Neural Network dengan akurasi 78%.

Berdasarkan hasil penelitian sebelumnya, dapat disimpulkan bahwa metode Convolution Neural Network (CNN) mampu dan sangat baik dalam mengklasifikasikan objek. Akan tetapi, sedikit penelitian yang secara khusus bertujuan untuk mengklasifikasikan jenis tanaman hias daun, sehingga penelitian ini relevan dan penting untuk dilakukan. Oleh karena itu tujuan dari penelitian ini adalah untuk mengklasifikasi jenis tanaman hias daun menggunakan algoritma Convolution Neural Network.

B. Rumusan Masalah

1. Bagaimana cara mengklasifikasikan jenis tanaman hias daun menggunakan convolution neural network (CNN)?

2. Bagaimana akurasi metode Convolutional neural network dalam mengklasifikasikan tanaman hias daun?

C. Tujuan penelitian

1. Mengembangkan dan menerapkan model klasifikasi Convolutional Neural Network yang dapat mengidentifikasi jenis tanaman hias daun
2. Untuk mengetahui akurasi metode Convolutional neural network dalam mengklasifikasikan tanaman hias daun

D. Manfaat penelitian

1. Bagi ilmu pengetahuan
 - a. Penelitian ini dapat menjadi dasar pengembangan lebih lanjut dalam klasifikasi tanaman hias daun berbasis machine learning, khususnya dalam pengolahan citra
 - b. Penelitian ini dapat memperkaya literatur akademis mengenai penerapan metode Convolutional Neural Network (CNN) dalam klasifikasi tanaman.
2. Bagi masyarakat dan kolektor tanaman

Sistem klasifikasi yang dikembangkan melalui penelitian ini akan memudahkan masyarakat umum, penjual, kolektor dalam mengidentifikasi dan mengkategorikan jenis tanaman hias daun tanpa perlu pengetahuan yang mendalam tentang botani
3. Bagi pengembang sistem teknologi

Penelitian ini memberikan contoh penerapan klasifikasi objek visual CNN yang dapat menjadi referensi bagi para pengembang sistem teknologi yang bekerja di bidang serupa, baik untuk aplikasi komersial maupun penelitian lanjutan.

E. Ruang lingkup penelitian

Penelitian ini dibatasi ruang lingkup penelitian agar penelitian tetap pada jalur yang telah ditentukan diantaranya yaitu:

1. Data didapatkan dengan memotret 8 jenis tanaman hias daun yang memiliki warna dan karakteristik sejenis

2. Penelitian ini berfokus pada penerapan Algoritma Convolution Neural Network (CNN) untuk klasifikasi tanaman hias daun berdasarkan karakter visual seperti bentuk, warna, dan daun.
3. Menggunakan InceptionV3 untuk arsitektur Convolution Neural Network (CNN)

F. Sistematika penulisan

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bab yang tersusun yaitu:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang bagaimana pentingnya klasifikasi pada tanaman hias daun dan bagaimana metode CNN dapat memfasilitasi masalah tersebut. Pada bab ini juga dijelaskan juga masalah dan tujuan penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang jenis-jenis tanaman hias daun, convolution neural network yang merupakan algoritma yang digunakan untuk proses klasifikasi.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan waktu penelitian dan alat yang digunakan peneliti dalam melakukan penelitian. Serta menjelaskan pendekatan yang akan digunakan dalam proses klasifikasi menggunakan convolution neural network

BAB IV HASIL DAN PEMBAHAS

Pada bab ini menjelaskan tentang pengumpulan data dan bagaimana data itu diolah. Dan menjelaskan bagaimana hasil dari proses klasifikasi menggunakan convolution neural network

BAB V

Bab ini menjelaskan secara singkat kesimpulan dari hasil penelitian ini.

BAB II

TINJAUAN PUSTAKA

A. Landasan teori

1. Tanaman hias daun

Tanaman hias adalah tanaman dengan bentuk yang unik dan khas yang digunakan sebagai hiasan atau dekorasi baik di dalam maupun di luar ruangan. Pada awalnya, tanaman hias hanya merujuk pada tanaman berbunga. Namun seiring berjalannya waktu, definisi tanaman hias meluas hingga mencakup semua jenis tanaman yang memiliki nilai estetika, baik daun, bunga, batang, buah, ranting, aroma dan akar, serta memiliki nilai seni. Saat ini, tanaman hias telah menjadi populer karena selain mempercantik suatu tempat, tanaman hias juga memberikan udara segar di sekitarnya dan dapat digunakan sebagai sumber pewarna alami. (Kartika Sari et al., n.d.).

Tanaman hias secara garis besar dapat dikategorikan menjadi dua jenis utama: tanaman hias berdaun dan tanaman hias bunga. Tanaman hias daun memukau mata dengan bentuk dan warna daunnya yang unik, seperti *anthurium*, *aglaonema*, *adiantum*, dan *bromelia*. Sementara itu, tanaman hias bunga memikat hati dengan keindahan bunganya, baik dari segi bentuk, warna, maupun aromanya, seperti mawar, anggrek, lili, dan sukulen.

Berikut adalah beberapa jenis tanaman hias daun yang umum dijumpai di Makassar, Sulawesi Selatan:

1. Calathea white fusion

Tanaman hias *Calathea White Fusion* memiliki daun tipis seperti tisu dengan rata-rata tinggi berkisar antara 30- 50 cm. Daya tarik utama yang dimiliki oleh *Calathea White Fusion* adalah perpaduan corak warna pada daunnya. Perpaduan tersebut meliputi warna putih, warna hijau, dan warna ungu yang memberikan kesan elegan. Selain itu, tanaman ini juga dikenal mampu menyegarkan udara di sekitarnya (Hamdani et al., 2025).



Gambar 1. *Calathea white fusion* (Hamdani et al., 2025)

2. *Diefenbachia* Brazil

Diefenbachia Brazil adalah salah satu jenis tanaman hias dari keluarga *Araceae* yang populer digunakan sebagai tanaman indoor. Tanaman ini memiliki daun yang lebar, mengkilap, dan bercorak khas dengan warna dasar hijau muda hingga hijau tua yang dipadukan dengan bercak-bercak krem atau putih kehijauan, sehingga memberikan efek artistik dan menarik.



Gambar 2. Dieffenbachia brazil. (Jardineries Botanix). (n.d.)

3. Dieffenbachia Camille

Dieffenbachia Camille merupakan salah satu varietas tanaman hias daun dari keluarga Araceae. Tanaman ini memiliki ciri khas berupa daun berbentuk oval lebar dengan ujung meruncing. Warna daunnya didominasi oleh putih krem hingga kuning pucat di bagian tengah, yang kemudian dikelilingi tegas oleh pinggiran daun berwarna hijau tua sehingga menampilkan kontras yang mencolok. Permukaan daunnya halus dan mengkilap, menjadikannya menarik sebagai tanaman hias indoor. Selain keindahan coraknya, Dieffenbachia Camille juga tumbuh cukup rimbun dengan batang tegak, sehingga sering digunakan untuk memperindah ruangan sekaligus memberikan kesan segar pada lingkungan sekitar.



Gambar 3. Dieffenbachia Camille (PictureThis, n.d.).

4. Dieffenbachia white etna

Dieffenbachia White Etna merupakan tanaman hias daun yang populer karena memiliki corak unik dengan perpaduan warna hijau dan putih kekuningan. Bentuk daunnya lebar dengan tekstur halus, sehingga memberikan kesan segar dan elegan. Tanaman ini sering dijadikan penghias ruangan maupun taman karena mampu menambah nuansa alami dan menenangkan.



Gambar 4. Dieffenbachia white etna. (Zahra, 2023)

5. Dieffenbachia Rebecca

Dieffenbachia Rebecca dikenal dengan keindahan daunnya yang bercorak belang hijau dan putih. Daunnya berbentuk lonjong dengan urat yang tegas, menjadikannya terlihat eksotis dan menarik. Tanaman ini cocok dijadikan tanaman hias pot di dalam maupun luar ruangan karena daya tarik visualnya yang kuat dan mudah dipadukan dengan tanaman lain.



Gambar 5. Dieffenbachia Rebecca (Zahra, 2023)

6. Mangkokan variegata

Mangkokan Variegata adalah tanaman hias yang dikenal karena variasi warna daunnya yang indah, mulai dari hijau, merah, ungu, hingga kombinasi menarik di antaranya. Bentuk daunnya beragam, ada yang bulat maupun oval dengan tepian bergerigi sehingga menambah keunikan tampilannya. Selain mempercantik taman atau halaman, tanaman ini juga sering dijadikan penghias pot karena corak warnanya yang cerah dan mampu menciptakan suasana segar.



Gambar 6. Mangkokan Variegata. (Dreamstime, 2025)

7. Polyscias Guilfoylei

Polyscias Guilfoylei adalah tanaman hias daun yang memiliki tampilan menarik dengan bentuk daun yang beragam. Daunnya bercorak hijau tua dengan kombinasi kuning cerah pada bagian tepinya, memberikan kesan segar dan indah. Bentuk daun cenderung lonjong dan bergerigi halus, membuatnya terlihat elegan serta cocok dijadikan tanaman penghias halaman maupun pot di dalam rumah. Tanaman ini banyak digemari karena mudah perawatannya dan mampu tumbuh subur dalam berbagai kondisi.



Gambar 7. *Polyscias Guilfoylei*. (New World Wildflowers, 2025)

8. *Syngonium wendlandii*

Syngonium Wendlandii merupakan tanaman hias merambat yang populer karena bentuk daunnya yang unik. Daunnya berwarna hijau gelap dengan pola garis putih mencolok di bagian tengah, sehingga menambah daya tarik visualnya. Bentuk daunnya memanjang dengan ujung meruncing, memberi kesan eksotis dan elegan. Tanaman ini sering digunakan sebagai tanaman hias gantung maupun penutup area tertentu karena pertumbuhannya yang menjalar dan mampu memberikan kesan alami pada ruangan atau taman.



Gambar 8. *Syngonium Wendlandii* (PictureThis,n.d.).

2. Klasifikasi Gambar

Klasifikasi adalah proses pengelompokan objek berdasarkan atribut atau karakteristiknya. Proses ini dapat dilakukan secara manual oleh manusia atau secara otomatis dengan bantuan algoritme komputer. Kategorisasi manual biasanya bersifat subjektif dan bergantung pada pengetahuan dan pengalaman pengklasifikasi. Sebaliknya, klasifikasi berbasis komputer menggunakan algoritma yang berbeda seperti algoritma Bayesian, support vector machine, decision tree, logika fuzzy, dan jaringan syaraf tiruan untuk mengklasifikasikan secara obyektif dan akurat. Pada penelitian ini menggunakan Algoritma Convolutional Neural Network.(Wibawa et al., 2018).

Klasifikasi gambar adalah proses mengelompokkan gambar ke dalam kategori tertentu. Ini adalah bagian dari ilmu komputer yang mempelajari bagaimana komputer bisa 'melihat' dan memahami gambar. Salah satu contohnya adalah mengenali tempat yang ada dalam sebuah foto.

Pada tahap pelatihan, setiap data gambar dipasangkan dengan label kelas yang benar. Ketika model membuat prediksi pada data baru, hasil prediksinya dibandingkan dengan label yang telah ditentukan. Perbedaan antara keduanya digunakan untuk meningkatkan kinerja model. Pendekatan pembelajaran yang diawasi, terutama dengan pembelajaran mendalam, telah terbukti sangat efektif untuk tugas klasifikasi gambar.

3. Pengolahan citra

Pengolahan citra digital adalah proses mengubah gambar asli menjadi gambar berkualitas lebih baik dengan menghilangkan gangguan, seperti noise dan meningkatkan ketajaman atau kejernihan gambar (Fadjeri et al., 2022). Proses pengolahan citra meliputi hal-hal sebagai berikut:

1. Image enhancement

Image enhancement atau peningkatan citra bertujuan meningkatkan kualitas gambar.

2. Image restoration

Image restoration atau restorasi citra yaitu pemulihan gambar yang mengalami kontaminasi oleh distorsi.

3. Image compression

Image compression atau Kompresi Citra adalah cara untuk membuat ukuran file gambar menjadi lebih kecil tanpa menghilangkan terlalu banyak detail penting.

4. Image segmentation

Image segmentation atau Segmentasi Citra proses membagi gambar menjadi bagian-bagian yang berbeda berdasarkan ciri khasnya seperti warna atau bentuk.

5. Image analysis

Image analys atau analisis citra adalah proses pengolahan gambar untuk mengekstrak informasi kuantitatif dan kualitatif, seperti fitur, segmen, atau karakteristik objek tertentu dalam sebuah citra

6. Image reconstruction

Image reconstruction rekontruksi citra atau proses menciptakan citra baru dengan menggabungkan informasi dari beberapa citra sumber untuk menghasilkan citra yang memiliki karakteristik yang diinginkan.

4. Convolutional Neural Network

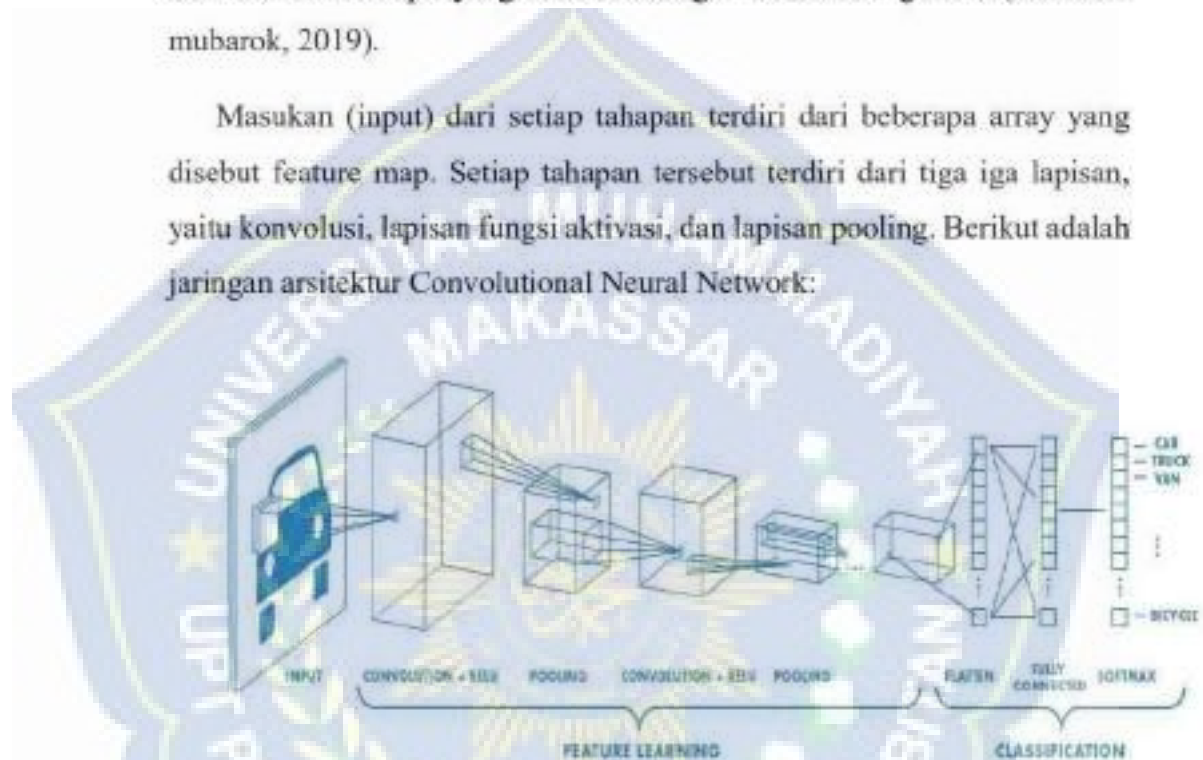
Algoritma CNN, yang termasuk dalam kategori *deep learning*, adalah yang paling umum digunakan untuk memahami dan memproses informasi visual dalam gambar. Berbagai aplikasi telah memanfaatkan keunggulan CNN, mulai dari pengenalan gambar dan video, sistem rekomendasi, hingga klasifikasi citra medis dan pemrosesan bahasa alami. (Rasywir et al., 2020)

Convolutional Neural Network (CNN) adalah bentuk pengembangan dari *multilayer perceptron* (MLP) yang dirancang untuk mengelolah data dua dimensi seperti citra. CNN tergolong sebagai *Deep Neural Network* karena arsitekturnya yang dalam dan berlapis-lapis dan banyak digunakan dalam berbagai aplikasi pengolahan citra. (Putri Navia Rena, 2019)

Convolutional Neural Network (CNN) adalah bentuk pengembangan dari *multilayer perceptron* (MLP) yang dirancang untuk mengelolah data dua dimensi seperti citra. CNN tergolong sebagai *Deep Neural Network* karena arsitekturnya yang dalam dan berlapis-lapis dan banyak digunakan dalam berbagai aplikasi pengolahan citra. Jika dibandingkan dengan arsitektur MLP, arsitektur CNN lebih kompleks dan membutuhkan tahapan yang lebih panjang sebelum klasifikasi. Algoritma CNN melakukan

pemrosesan melalui dua tahap, yaitu feature learning dan klasifikasi. Tahap feature learning sendiri merupakan proses ekstraksi gambar yang diinputkan untuk mempelajari value dari gambar tersebut. Proses inilah yang membedakan algoritma CNN dengan MLP. MLP hanya dapat mengekstrak sekali untuk setiap input yang diberikan, sedangkan CNN dapat melakukan ekstraksi berkali-kali. Hasil dari berbagai ekstraksi ini kemudian disimpan dan membentuk apa yang disebut sebagai kedalaman gambar.(Hamdani mubarak, 2019).

Masukan (input) dari setiap tahapan terdiri dari beberapa array yang disebut feature map. Setiap tahapan tersebut terdiri dari tiga iga lapisan, yaitu konvolusi, lapisan fungsi aktivasi, dan lapisan pooling. Berikut adalah jaringan arsitektur Convolutional Neural Network:



Gambar 9. Arsitektur Convolutional Neural Network. (Putri Navia Rena, 2019)

Berdasarkan gambar diatas, Tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*). Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

1. Convolution Layer

Convolution layer merupakan bagian dari tahap pada arsitektur CNN. Tahap ini melakukan operasi konvolusi pada *output* dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari input citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Operasi konvolusi dapat dituliskan sebagai berikut:

$$s(t) = (x * t)(t) = \sum_{\alpha = -\infty}^{\infty} x(\alpha) * w(t - \alpha)$$

Keterangan:

S(t) – Fungsi hasil operasi konvolusi

X= Input

W= bobot (kernel)

Fungsi $s(t)$ memberikan *output* tunggal berupa *feature Map*. Argumen pertama adalah *input* yang merupakan x dan argumen kedua w sebagai kernel atau filter. Apabila dilihat *input* sebagai citra dua dimensi, maka bisa dikatakan t sebagai piksel dan menggantinya dengan i dan j . Maka dari itu, operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat menulis sebagai berikut:

$$s(i,j) = (K * I) (i,j) = \sum_{\infty} \sum_{\infty} I(i - m, j - n) K(m, n)$$

$$s(i,j) = (K * I) (i,j) = \sum_{\infty} \sum_{\infty} I(i + m, j + n) K(m, n)$$

Kedua persamaan diatas menunjukkan perhitungan dasar dalam operasi konvolusi dimana, i dan j merepresentasikan posisi piksel dalam citra. Perhitungan ini bersifat kumulatif dan terjadi ketika K sebagai kernel, sedangkan I sebagai input. Dalam operasi konvolusi kernel dapat dibalik relatif terhadap input. Sebagai alternatif, operasi konvolusi dapat

dilihat sebagai perkalian matriks antar input dan kernel, dimana outputnya dihitung dengan *dot product*. Selain itu volume output dari operasi konvolusi dapat ditentukan menggunakan hyperparameter yang digunakan dalam persamaan berikut berfungsi untuk menghitung jumlah neuron aktivasi dalam satu kali output.

Perhatikan persamaan berikut

$$(w - f + 2P) / (S + 1)$$

Keterangan:

- W = Ukuran volume gambar
- F = Ukuran Filter
- P = Nilai *Padding* yang digunakan
- S = Ukuran Pergeseran (*Stride*)

Berdasarkan persamaan diatas, ukuran spasial dari output dapat ditentukan dengan mempertimbangkan beberapa hyperparameter, yaitu ukuran *volume* citra (W), ukuran *filter* (F), jumlah *padding nol* (P), serta *stride* (S) berfungsi sebagai parameter yang menentukan seberapa jauh filter bergeser setiap kali operasi konvolusi dilakukan pada citra input. Sementara itu, zero padding adalah Teknik yang digunakan untuk menambahkan nilai nol disekitar tepi citra guna mempertahankan dimensi atau mengontrol ukuran output yang dihasilkan.

Convolution layer terdiri dari neuron yang disusun sedemikian rupa sehingga membentuk filter dengan dimensi tertentu dalam hal Panjang dan tinggi (*pixels*). Sebagai contoh, pada tahap awal *feature extraction*, lapisan konvolusi biasanya memiliki ukuran 5x5x3, dimana Panjang 5 pixels, tinggi 5 pixels, dan tebal/ jumlahnya 3, sesuai dengan *channel* dan *image* tersebut. Filter – filter ini kemudian digeser keseluruhan bagian citra. Pada setiap pergeseran, dilakukan operasi "*dot*" antara nilai input dan nilai filter. Hasil dari operasi ini membentuk output yang

dikenal sebagai *activation map* atau *feature map*. Berikut ini adalah ilustrasi dari proses tersebut:



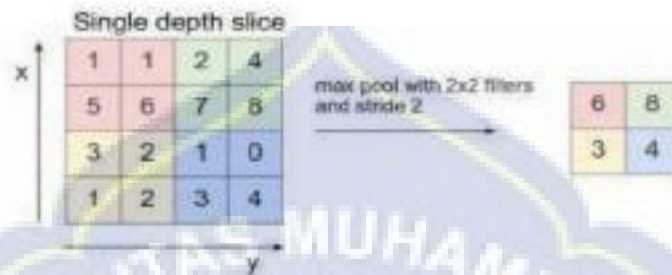
Gambar 10. Convolution layer (Medium Samuel Sena, 2017)

2. Operasi pooling

Setelah proses konvolusi selesai, hasil yang diperoleh akan melewati tahap pooling. Pooling berfungsi untuk mengurangi dimensi spasial dari input sehingga jumlah parameter yang di proses dapat dikurangi dengan menggunakan teknik down-sampling. Secara umum, metode pooling yang paling sering digunakan adalah max pooling, dimana nilai tertinggi dari suatu area ditentukan sebagai hasil akhir. (Suartika, n.d.)

Max Pooling adalah teknik yang digunakan untuk meningkatkan ketahanan fitur terhadap perubahan posisi dengan menerapkan operasi maksimum. Dalam proses ini, output dari Convolutional Layer dibagi menjadi beberapa area kecil, di mana setiap area diambil nilai maksimumnya untuk membentuk matriks citra yang telah dikompresi,

seperti yang ditunjukkan pada Gambar. Grid berwarna merah, hijau, kuning, dan biru mewakili kelompok area yang akan dipilih nilai tertingginya. Hasil akhir dari proses ini dapat dilihat pada kumpulan grid di sebelah kanan. Dengan metode ini, fitur yang diekstraksi tetap konsisten meskipun terjadi pergeseran atau translasi pada objek dalam citra.



Gambar 11. Max pooling (Suartika, n.d.)

Gambar diatas menunjukkan proses max-pooling, dimana hasil dari pooling adalah matriks dengan dimensi yang lebih kecil dibandingkan citra aslinya. Lapisan pooling ini bekerja dengan memproses setiap irisan kedalaman volume input secara bergantian. Dari gambar diatas, oprasi max-pooling dilakukan menggunakan filter berukuran 2x2. Input yang digunakan memiliki ukuran 4x4, menghasilkan output dengan ukuran 2x2.

3. Fully-Connected Layer

Lapisan yang terhubung penuh (Fully-Connected Layer) adalah komponen penting dalam jaringan saraf tiruan, terutama dalam Multi-Layer Perceptron (MLP). Lapisan ini bertugas menghubungkan setiap neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya, menciptakan jaringan yang saling terhubung secara penuh. Tujuannya adalah untuk mengubah dimensi data yang kompleks menjadi format yang lebih sederhana, sehingga data tersebut dapat diklasifikasikan secara linear. Proses ini melibatkan transformasi data multidimensional menjadi vektor satu dimensi, yang kemudian diolah melalui perkalian matriks dan penambahan bias.

Perbedaan mendasar antara Fully-Connected Layer dan Convolutional Layer terletak pada pola konektivitasnya. Dalam Convolutional Layer, neuron hanya terhubung ke sebagian kecil area input, sementara Fully-Connected Layer menghubungkan setiap neuron ke seluruh neuron di lapisan sebelumnya. Meskipun demikian, kedua jenis lapisan ini menggunakan operasi perkalian titik (dot product) dalam pengolahannya. Data yang dihasilkan dari lapisan ekstraksi fitur (feature extraction layer) yang berbentuk multidimensional harus diubah menjadi vektor agar dapat diproses oleh Fully-Connected Layer.

Transformasi data menjadi vektor dalam Fully-Connected Layer menyebabkan hilangnya informasi spasial, sehingga lapisan ini umumnya ditempatkan di akhir arsitektur jaringan. Namun, Convolutional Layer dengan kernel 1×1 dapat melakukan fungsi serupa tanpa menghilangkan informasi spasial, sehingga penggunaan Fully-Connected Layer dalam Convolutional Neural Networks (CNN) saat ini semakin berkurang. Selain itu, masih banyak arsitektur jaringan saraf lain yang dapat digunakan, tergantung pada karakteristik data dan permasalahan yang dihadapi. (Suartika, n.d.)

B. Penelitian terkait

Penelitian yang dilakukan oleh Muhammad Baihaqy, Agung Toto Wibowo dan Dody Qori Utaman (2022) yang berjudul “Klasifikasi Tanaman Anggrek jenis Phalaenopsis berdasarkan Citra Labellum Bunga Menggunakan Metode Convolutional Neural Network (CNN)” Hasil penelitian ini menunjukkan akurasi rata-rata 99,2%; f1 score rata-rata 99,2%, nilai precision rata-rata 99,89% dan recal rata-rata 98,6%. Hasil ini menunjukkan bahwa model Convolutional Neural Network (CNN) sangat efektif dalam mengklasifikasikan gambar.

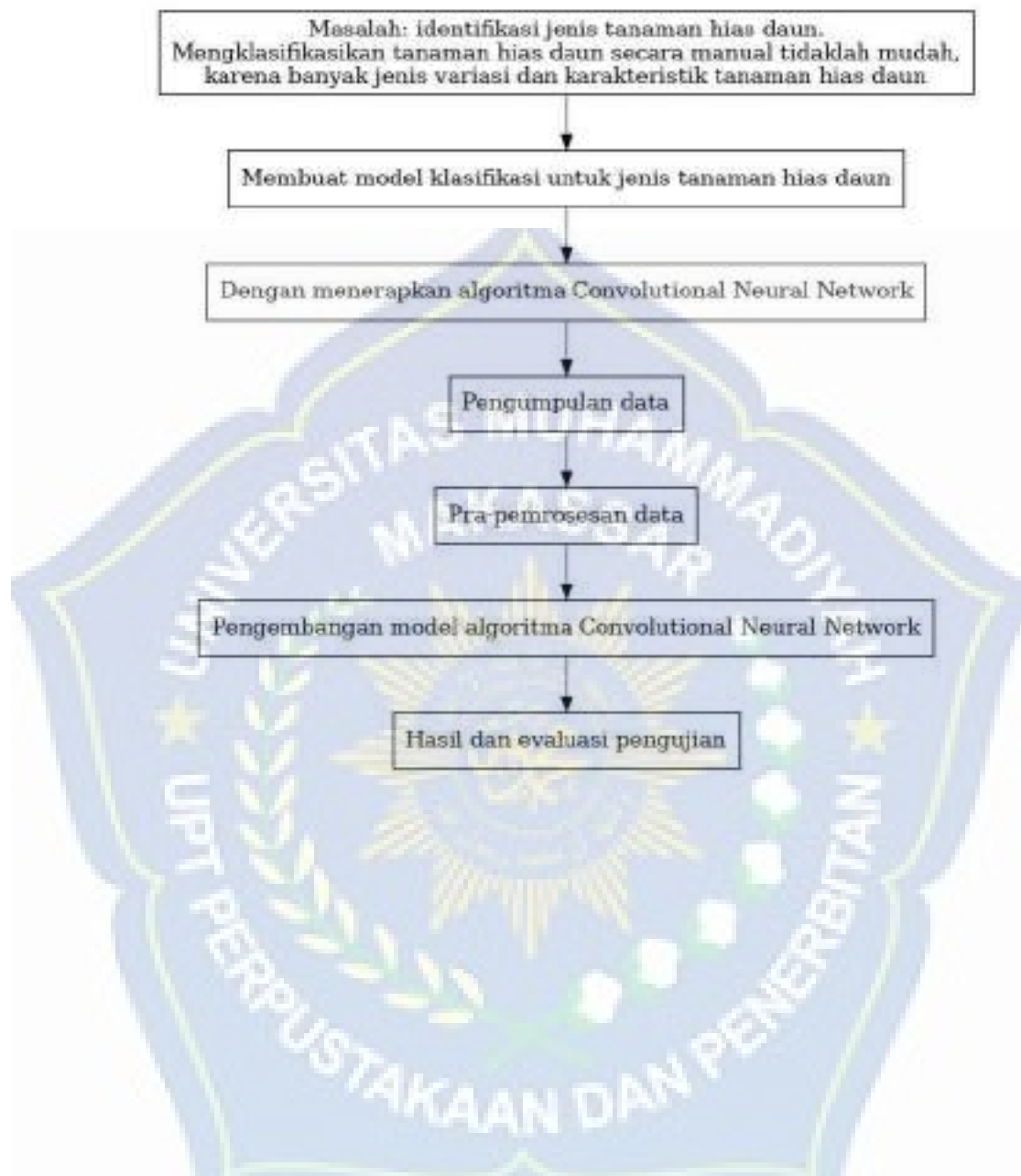
Penelitian yang dilakukan oleh Hanissa Anggraini, Margi Cahyanti, dan Missa Lamsani (2021) yang berjudul “Implementasi Deep Learning Flower Scanner Menggunakan Metode Convolutional Neural Network” Penelitian ini memperoleh hasil final test akurasi bernilai 99,30%.

Penelitian yang dilakukan oleh Intan Gilang Perwati, Nana Suarna, Tati Suprapti (2024) yang berjudul “Analisis Klasifikasi Gambar Bunga Lily Menggunakan Metode Convolutional Neural Network (CNN) Dalam Pengolahan Citra” Hasil penelitian ini menunjukkan bahwa metode Deep Learning dengan CNN dapat menghasilkan klasifikasi yang sangat baik pada gambar bunga lily. Nilai akurasi terhadap klasifikasi jenis bunga lily adalah sebesar 87%.

Penelitian yang dilakukan oleh Yoga Purna Irawan dan Indah Susilawati (2021) yang berjudul “Klasifikasi Jenis Aglaonema Berdasarkan Citra Daun Menggunakan Convolutional Neural Network (CNN)” Berdasarkan 4 percobaan menggunakan optimizer serta konfigurasi nilai epoch yang berbeda-beda, diperoleh nilai akurasi validasi training tertinggi yakni sebesar 98,00 %. Sistem yang dibangun juga dapat mengklasifikasikan citra aglaonema dengan baik, dengan tingkat keberhasilan akurasi sebesar 96% dari 50 citra yang diuji.

Penelitian yang dilakukan oleh Fadlan Sayyidul Anam, Muhammad Rafi Muttaqin dan Yudhi Raymond Ramadhan (2023) yang berjudul “Klasifikasi Penyakit Pada daun dan Buah Jambu Menggunakan Convolutional Neural Network” hasil model daun jambu memiliki performa yang sangat baik dengan hasil evaluasi model training akurasi sebesar 99,6%, validasi akurasi sebesar 100%, training loss sebesar 3,2% dan validation loss sebesar 3,1%. Hasil confusion matrix memiliki akurasi 100% dari jumlah data validasi sebanyak 63 data.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Tempat dan waktu penelitian

1. Tempat penelitian

Dalam suatu penelitian, tempat penelitian adalah lokasi atau objek yang akan diteliti. Penelitian ini dilakukan dengan mengambil data gambar secara langsung di salah satu toko tanaman hias daun

2. Waktu penelitian

Waktu penelitian ini akan dilakukan dalam jangka waktu kurang lebih 3 bulan, yaitu dimulai pada bulan Mei 2025 sampai dengan Agustus 2025.

B. Alat dan bahan

Alat penelitian yang digunakan untuk klasifikasi tanaman hias daun menggunakan algoritma CNN berupa laptop. Dalam penelitian ini, peneliti menggunakan perangkat keras dan perangkat lunak :

1. perangkat keras

- a. processor AMD Ryzen 3 3250U processor 2,6 GHz
- b. Besar memori 512 Gb dan Ram 8 Gb

2. perangkat lunak

- a. Windows 10
- b. Google colaboratory

C. Arsitektur Convolutional Neural Network

Model Inception V3 digunakan pada penelitian ini yang merupakan salah satu arsitektur CNN, Inception V3 dipilih karena mampu mengekstraksi fitur citra dengan baik. Berikut tahapan-tahapan arsitektur inceptionV3 yang digunakan pada penelitian ini:

1. Lapisan input

Citrان dimasukkan dan ditetapkan dengan ukuran 299x299 piksel. Sebelum memasuki tahap selanjutnya citra melewati proses augmentasi data dan pra pemrosesan data.

2. Lapisan Global Average pooling

Global Average pooling Tahap ini bertujuan untuk mereduksi dimensi fitur tanpa kehilangan informasi penting.

3. Lapisan Regularisasi (Dropout)

Tahap berikutnya adalah regularisasi dengan dropout sebesar 0,3 yang berfungsi untuk mengurangi risiko overfitting selama proses pelatihan.

4. Lapisan Klasifikasi (Dense Layer)

Lapisan terakhir adalah Dense layer yang menghasilkan probabilitas untuk setiap kelas tanaman hias daun. Jumlah neuron pada lapisan ini menyesuaikan dengan jumlah kelas pada dataset

5. Skema Training

Skema pelatihan dilakukan dalam dua tahap. Stage-1 (Head Training) menggunakan optimizer Adam dengan learning rate 0,001, di mana backbone dibekukan dan hanya classifier yang dilatih. Stage-2 (Fine-tuning) menggunakan learning rate lebih kecil yaitu 0,00001, dengan membuka 60 lapisan terakhir dari backbone untuk disesuaikan kembali dengan dataset.

D. Perancangan system

Dalam upaya mempermudah proses perancangan, peneliti merancang alur sistem menggunakan diagram alir (flowchart), sebagaimana diilustrasikan pada gambar berikut:



Gambar 12. Diagram alur InceptionV3

Dataset pada penelitian ini dikumpulkan dengan memotret 8 jenis tanaman hias daun. Pada tahap pre-processing yaitu flip, rotasi dan zoom dan rescaling.

Arsitektur CNN berbasis InceptionV3 dengan bobot awal dari dataset ImageNet. Pada tahap ini, citra yang telah diproses masuk ke backbone InceptionV3 untuk ekstraksi fitur. Setelah itu, hasil ekstraksi diperkecil dimensinya dengan Global Average Pooling (GAP), lalu ditambahkan Dropout sebagai regularisasi untuk mencegah overfitting. Lapisan klasifikasi Dense layer dengan aktivasi Softmax digunakan untuk menghasilkan prediksi probabilitas.

Proses pelatihan dilakukan dengan dua tahap yaitu Stage-1 frozen backbone dan stage -2 Fine-tuning. Setelah pelatihan selesai, model diuji menggunakan data uji untuk mengukur performa. Evaluasi dilakukan menggunakan akurasi, confusion matrix, dan classification report untuk melihat kemampuan model dalam mengklasifikasikan setiap kelas.

E. Teknik pengujian system

Pengujian sistem dalam klasifikasi tanaman hias daun menggunakan CNN berfokus pada evaluasi model sebagai sistem secara keseluruhan, dengan penekanan pada kinerja dan kemampuan generalisasi.

Metrik evaluasi utama yang digunakan sebagai berikut:

1. *Accuracy*, nilai akurasi menunjukkan berapa banyak prediksi model.
2. *Loss (Sparse Categorical Crossentropy)*, digunakan untuk mengukur selisih antara distribusi prediksi dan label sebenarnya.
3. *Confusion Matrix*, memberikan hasil benar atau salahnya prediksi.
4. *Classification Report (Precision, Recall, F1-Score)*

Adapun formula yang digunakan untuk menghitung nilai *Precision*, *Accuracy*, *Recall* dan *F1-Score* secara berturut-turut adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall}$$

Keterangan:

TP = True Positive

TN = True Negative

FP = False Positif

FN = False Negative

F. Teknik Analisis data

Dalam penelitian ini, saya menggunakan algoritma Convolutional neural network (CNN) untuk melakukan klasifikasi jenis tanaman hias daun. Oleh karena itu, analisis data saya akan berfokus pada evaluasi kinerja model CNN yang telah saya rancang. Berikut adalah Teknik analisis data yang berfokus pada kinerja model CNN:

Data citra tanaman hias daun dikumpulkan melalui observasi langsung dan dokumentasi visual. Dalam proses klasifikasi tanaman hias daun menggunakan CNN, tahap pra-pemrosesan citra sangat penting untuk memastikan data gambar memiliki kualitas dan keseragaman yang optimal. Langkah ini mencakup penyesuaian ukuran semua gambar agar konsisten, normalisasi nilai piksel untuk mempercepat proses pembelajaran model, dan augmentasi data untuk meningkatkan variasi dataset.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini ini, disajikan hasil dan pembahasan mengenai penerapapan algoritma Convolutional Neural Network (CNN) untuk klasifikasi tanamn hias daun terdiri dari 8 kelas berjumlah 2131 gambar.

A. Pengumpulan Dataset

Pada penelitian ini pengambilan gambar untuk dataset diambil menggunakan kamera handphone.



Gambar 13. Dataset Calathea white fusion



Gambar 14. dataset diven Rebecca



Gambar 15. Dataset Polyscias gulfoylei

Berikut distribusi jumlah dataset per kelas.

Tabel 1. Dataset tanaman hias

No	Jenis tanaman hias daun	Jumlah
1	Calathea_White_Fusion	255 Gambar
2	Dieffenbachia Brazil	265 Gambar
3	Dieffenbachia Camille	274 Gambar
4	Dieffenbachia White Etna	288 Gambar
5	Dieffenbachia Rebecca	259 Gambar
6	Mangkoka Variegata	267 Gambar
7	Polyscias Guilfoylei	262 Gambar
8	Syngonium Wedlandii	263 Gambar

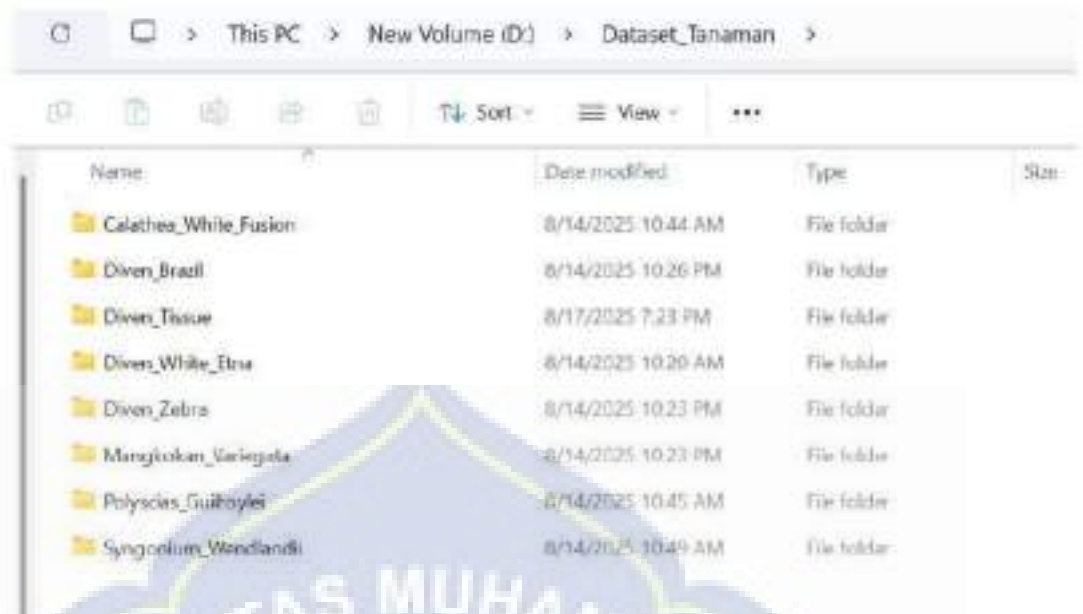
B. Pelabelan

Tahap selanjutnya dalam penelitian saya adalah pelabelan data. Dataset yang berupa kumpulan gambar, dimulai dengan menentukan label yang akan digunakan. Setiap gambar dalam dataset diberi label sesuai dengan jenis tanaman hias daunnya.



Gambar 16. Contoh pelabelan dataset

Selanjutnya membuat folder yang berisi masing-masing jenis tanaman hias daun.



Gambar 17. Pembuatan folder

C. Proses pengolahan data

1. Akses dataset

Pengolahan data dimulai dari menghubungkan googledrive dengan google colab dan memuat dataset yang berisi 8 kelas dengan jumlah 2133 gambar dengan setiap kelas berisi 255-288 gambar. Dataset di simpan di google drive.

2. Memuat dataset

Memuat dataset dilakukan menggunakan utilitas tf. keras. `utils.image_dataset_from_directory`, yang berfungsi untuk membaca gambar yang otomatis memberi label berdasarkan nama folder. Kemudian dataset dibagi menjadi 80% data latih 20% data validasi lalu bagian validasi dibagi lagi menjadi validation set dan validation tes set. Setiap gambar di resize ke 299x299.

3. Pra-pemrosesan dan augmentasi data

pada tahap ini pre-processing data menggunakan resize ke 299x299 kemudia citra di normalisasi dengan layer rescaling untuk mengubah rentang piksel [0, 255] ke [-1, 1] yang sesuai untuk inception v3 dan rescaling.

Selanjutnya, untuk meningkatkan keragaman dataset, penelitian ini menerapkan augmentasi data secara on-the-fly. Augmentasi hanya aktif saat pelatihan dan otomatis nonaktif pada tahap validasi maupun pengujian.



Gambar 18. Sebelum dan sesudah augmentasi
Operasi yang digunakan:

- Random flip ("horizontal")*: Digunakan untuk membalik gambar secara horizontal agar mendapatkan variasi arah/posisi daun, sehingga model tidak bergantung pada orientasi kiri-kanan.
- Random rotation (0,1)*: untuk memutar citra secara acak dalam rentang kecil ($\pm 10^\circ$) agar model lebih robust terhadap kemiringan pengambilan gambar
- Random zoom*: Melakukan pembesaran atau pengecilan acak hingga $\pm 15\%$ untuk meniru jarak kamera yang bervariasi.
- Random Contrast*: Mengubah contrast gambar, membantu model tetap akurat dengan pencahayaan terang atau redup.



Gambar 19. Augmentasi data pada setiap operasi

Pada penelitian ini augmentasi data hanya digunakan pada data training, data validasi dan dan test tidak menggunakan augmentasi data. Hal ini dilakukan untuk menghindari data *leakage*, yaitu ketika data evaluasi ikut terpengaruh oleh proses augmentasi yang hanya untuk data latih. Untuk memastikan tidak terjadinya data leakage dataset dibagi dari awal dengan 80% data train, 20% data validasi serta data uji.

D. Model convolutional neural network

Convolutional neural Network adalah metode jaringan saraf mendalam, yang paling umum diterapkan untuk menganalisis gambar. Pada penelitian ini menggunakan arsitektur Inception V3 sebagai backbone untuk mengekstraksi fitur dan melakukan klasifikasi. Dengan mengimport from tensorflow.keras.applications.inception_v3 import InceptionV3 yang digunakan untuk memanggil arsitektur inception V3 yang sudah disediakan oleh TensorFlow Keras.

InceptionV3 merupakan salah satu arsitektur CNN yang telah dilatih sebelumnya menggunakan dataset besar ImageNet, sehingga memiliki kemampuan yang baik dalam mengenali pola pada gambar. Pada tahap awal, parameter `include_top=False` digunakan untuk menghapus lapisan klasifikasi bawaan InceptionV3, karena pada penelitian ini lapisan klasifikasi tersebut diganti dengan lapisan baru yang sesuai dengan jumlah kelas tanaman hias. Selain itu, dilakukan pengaturan `base.trainable = False` yang berarti bobot InceptionV3 dibekukan sementara. Hal ini bertujuan agar model hanya berfungsi sebagai ekstraktor fitur tanpa mengubah bobot bawaan, sehingga proses pelatihan menjadi lebih cepat dan stabil.

Setelah tahap ekstraksi fitur, `GlobalAveragePooling2D` merangkum hasil fitur dari backbone menjadi satu vektor per channel. Teknik ini digunakan untuk mereduksi ukuran data fitur menjadi lebih sederhana. Selanjutnya ditambahkan lapisan Dropout sebesar 30% untuk mematikan sebagian neuron secara acak pada saat pelatihan, yang bertujuan untuk mencegah overfitting. Selanjutnya menggunakan `Dense (num_classes, activation="softmax")` merupakan lapisan terakhir yang berfungsi sebagai lapisan klasifikasi. Pada penelitian ini jumlah neuron disesuaikan dengan

jumlah kelas tanaman hias, yaitu delapan neuron. Lapisan ini menghasilkan output berupa probabilitas untuk setiap kelas, sehingga model dapat memberikan prediksi dengan memilih kelas yang memiliki nilai probabilitas tertinggi sebagai hasil akhir.

Berikut tahap CNN dengan arsitektur InceptionV3



Gambar 20. Tahapan CNN InceptionV3

Input berupa gambar tanaman hias daun dari dataset dimasukkan dengan ukuran standar 299x299 piksel dan 3 channel warna RGB. Format ini disesuaikan dengan kebutuhan arsitektur InceptionV3. selanjutnya memasuki tahap pre-processing, yaitu Data Augmentasi untuk memperkaya dataset dengan operasi random flip, rotation, zoom, contrast. Dan rescaling normalisasi ke piksel rentang $[-1,1]$, sesuai processing standar InceptionV3, agar proses pelatihan lebih stabil. selanjutnya memasuki tahap model cnn yaitu Inceptionv3 sebagai backbone yang merupakan inti dari cnn inceptionV3. pada tahap ini pelatihan dibagi menjadi dua tahap, yaitu

1. Tahap awal (Stage-1), layer CNN dibekukan, sehingga hanya lapisan atas yang dilatih.

2. Tahap lanjut (Stage-2), 60layer terakhir dibuka untuk fine-tuning

Setelah melewati dua tahap diatas, hasil keluaran fiturnya berupa feature map 3D agar bias memasuki tahap selanjutnya kita memerlukan Global Average Pooling untuk mengubah 3D menjadi vektor 1 D. Kemudian langkah berikutnya adalah melakukan regulasi dengan lapisan dropout dengan rasio 0,3. Selanjutnya memasuki lapisan dense (Fully Connected) yang berfungsi sebagai *classifier*. Pada penelitian ini, jumlah neuron disesukain dengan kelas tanaman hias daun yaitu ada 8 kelas. Aktivasi yang digunakan adalah *softmax*, sehingga setiap keluaran neuron menghasilkan probabilitas.

E. Proses training

Pada penelitian ini, proses training dilakukan dalam dua tahap menggunakan arsitektur InceptionV3 sebagai backbone:

1. Stage-1 (Feature Extraction)

Backbone InceptionV3 dibekukan (frozen) sehingga hanya lapisan atas (classifier/head) yang dilatih. Tujuannya untuk memanfaatkan bobot pretrained tanpa mengubah representasi awal.

2. Stage -2 (Fine-tuning)

Backbone dibuka kembali, namun hanya 60 lapisan terakhir yang dilatih, sedangkan lapisan awal tetap dibekukan. Hal ini dilakukan agar model dapat menyesuaikan fitur tingkat tinggi dengan dataset tanaman hias, tanpa merusak representasi dasar dari pretrained model.

Untuk memastikan proses training berjalan optimal, beberapa callback digunakan yaitu, *Model Checkpoint*, untuk menyimpan model terbaik secara otomatis berdasarkan peningkatan validation accuracy, sehingga hanya bobot terbaik yang dipertahankan dan *EarlyStopping* untuk menghentikan pelatihan lebih awal apabila validation loss tidak membaik selama 8 epoch berturut-turut, dengan opsi mengembalikan bobot terbaik. Serta *ReduceLROnPlateau* untuk menurunkan learning rate apabila validation loss stagnan, sehingga model lebih mudah mencapai konvergensi.

Setelah konfigurasi callback ditetapkan, proses pelatihan dilakukan dalam dua tahap (Stage-1 dan Stage-2) dengan jumlah epoch yang berbeda pada setiap eksperimen. Pada stage-1 dilakukan dengan epoch 15 dan stage-2 dilakukan

F. Hasil training

Setelah proses training model CNN menggunakan arsitektur InceptionV3 dengan dua tahap (Stage-1 dan Stage-2), dilakukan dua kali eksperimen dengan jumlah epoch yang berbeda. Tujuannya adalah untuk membandingkan pengaruh variasi jumlah epoch terhadap performa model pada data uji. Hasil akhir dari masing-masing epoch dapat dilihat pada tabel – tabel berikut:

Tabel 2.Stage-1 epoch 25

Epoch	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Learning Rate
1	0.8560	0.4607	0.7232	3.7361	0.0010
2	0.9544	0.1370	0.8616	0.7547	0.0010
3	0.9657	0.1041	0.9107	0.3220	0.0010
4	0.9839	0.0656	0.9509	0.1873	0.0010
5	0.9861	0.0507	0.9554	0.1471	0.0010
6	0.9923	0.0236	0.9241	0.2784	0.0010
7	0.9789	0.0578	0.9688	0.1191	0.0010
8	0.9845	0.0510	0.9464	0.4243	0.0010
9	0.9893	0.0377	0.9732	0.0911	0.0010
10	0.9961	0.0111	0.9732	0.0497	0.0010
11	0.9974	0.0096	0.9598	0.0747	0.0010
12	0.9960	0.0164	0.9554	0.1314	0.0010
13	0.9935	0.0209	0.9777	0.0735	0.0005
14	0.9999	0.0037	0.9821	0.0570	0.0005
15	0.9997	0.0066	0.9866	0.0500	0.0005



Gambar 21. Grafik Training Stage 1

Pada epoch pertama, akurasi data training mencapai 85,60% dengan nilai loss yang masih cukup tinggi sebesar 0,46, sedangkan akurasi data validasi baru berada pada 73,32% dengan loss sebesar 3,73. Seiring bertambahnya epoch, performa model terus meningkat hingga mencapai titik optimal. Akurasi terbaik pada data training diperoleh pada epoch ke-14 dengan capaian 99,99% dan loss yang sangat rendah, menandakan model hampir sempurna mengenali pola data latih. Sementara itu, akurasi tertinggi pada data validasi dicapai pada epoch ke-15 dengan nilai 98,66% dan loss rendah, yang menunjukkan kemampuan model melakukan generalisasi dengan sangat baik. Model terbaik ditentukan berdasarkan hasil validasi, bukan hanya dari akurasi data latih. Pada proses training, meskipun akurasi validasi tertinggi sempat tercapai pada epoch ke-15 sebesar 98,66%, mekanisme *EarlyStopping* mengembalikan bobot model ke kondisi paling optimal yang terjadi pada epoch ke-10. Pada epoch ini, akurasi validasi mencapai 97,32% dengan nilai *loss* terendah dan paling stabil. Oleh karena itu, akurasi validasi 97,32% pada epoch ke-10 ditetapkan sebagai performa terbaik model.

Pada proses training ini, model awalnya menggunakan learning rate sebesar 0.001 yang cukup efektif untuk mempercepat pembelajaran di epoch-epoch awal. Namun pada epoch 11 hasil akurasi validasi

mengalami satganan sehingga pada epoch 13 learning rate diturunkan menjadi 0.0005.

Berikut ditampilkan grafik

Tabel 3. Stage-2 epoch 10

Epoch	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Learning Rate
1	1.0000	0.0045	0.9866	0.0378	1e-5
2	0.9998	0.0055	0.9866	0.0359	1e-5
3	0.9980	0.0076	0.9866	0.0334	1e-5
4	1.0000	0.0022	0.9866	0.0322	1e-5
5	1.0000	0.0020	0.9911	0.0319	1e-5
6	0.9993	0.0028	0.9866	0.0329	1e-5
7	1.0000	0.0025	0.9911	0.0335	1e-5
8	0.9995	0.0028	0.9911	0.0333	1e-5
9	0.9997	0.0033	0.9911	0.0342	1e-5 → 5e-6
10	1.0000	0.0022	0.9866	0.0348	5e-6



Gambar 22. Grafik Training Stage 2

Pada epoch pertama, akurasi data training langsung mencapai 100% dengan loss 0,0045, sedangkan akurasi validasi sudah tinggi di angka 98,66% dengan val_loss 0,0378. Pada epoch 2 hingga 4, akurasi training

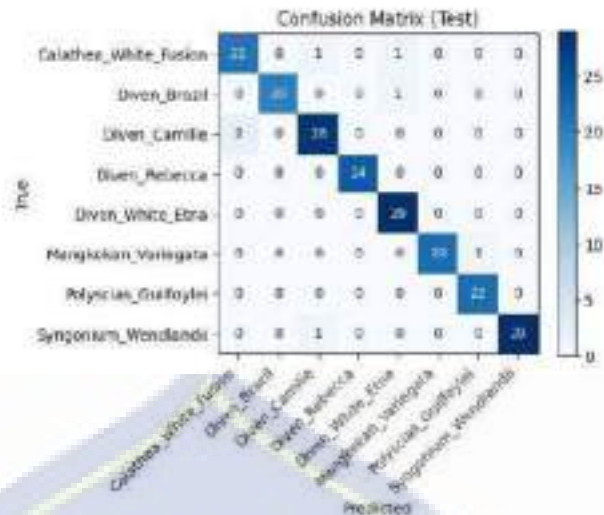
tetap sangat tinggi (99–100%) dengan nilai loss sangat kecil, namun akurasi validasi stagnan di 98,66% meskipun val_loss sedikit membaik. Performa terbaik dicapai pada epoch ke-5 dengan akurasi training 100% dan akurasi validasi 99,11% disertai val_loss terendah 0,0319, sehingga model pada epoch ini disimpan sebagai model terbaik.

Setelah itu, pada epoch 6 hingga 8, validasi tidak mengalami peningkatan, hanya menunjukkan fluktuasi kecil dengan akurasi tetap di 98,66–99,11% dan val_loss sekitar 0,033. Pada epoch ke-9, karena validasi tidak membaik, mekanisme ReduceLROnPlateau menurunkan learning rate dari $1e-5$ menjadi $5e-6$. Namun hingga epoch ke-10, performa validasi tidak lagi meningkat dan justru kembali turun ke 98,66% dengan val_loss sebesar 0,0348. Akhirnya, mekanisme EarlyStopping mengembalikan bobot model ke kondisi terbaik pada epoch ke-5 dengan akurasi validasi 99,11%.

Pada tahap training Stage 1, model sempat mencapai akurasi validasi tertinggi sebesar 98,66% pada epoch ke-15. Namun demikian, mekanisme EarlyStopping menetapkan model terbaik pada epoch ke-10 dengan akurasi validasi sebesar 97,32% karena val_loss lebih stabil. Pada tahap training Stage 2, performa model meningkat lebih baik lagi, di mana akurasi validasi tertinggi sekaligus model terbaik dicapai pada epoch ke-5 dengan nilai 99,11% dan val_loss terendah. Dengan demikian, dapat disimpulkan bahwa terjadi peningkatan performa dari Stage 1 ke Stage 2, dengan kenaikan akurasi validasi sebesar sekitar 1,79%, sehingga model hasil Stage 2 dipilih sebagai model akhir.

G. Hasil pengujian/Evaluasi model

1. Confusion matrix



Gambar 23. Confusion Matrix

Hasil evaluasi menggunakan data test ditunjukkan melalui confusion matrix. Dari delapan kelas yang diuji, sebagian besar gambar berhasil diprediksi dengan benar. Pada kelas *Calathea_White_Fusion* terdapat 21 prediksi benar dan 2 prediksi salah (masing-masing salah ke *Diven_Camilie* dan *Diven_Rebecca*). Kelas *Diven_Brazil* memiliki 20 prediksi benar dan 1 prediksi salah ke *Diven_Camilie*. Sementara itu, *Diven_Camilie* memperoleh 28 prediksi benar dengan 2 salah prediksi ke *Calathea_White_Fusion*. Kelas *Diven_Rebecca* diprediksi sempurna dengan 24 benar dan 0 salah, demikian pula *Polyscias_Guilfoylei* dengan 22 benar dan 0 salah. Pada kelas *Diven_White_Etna*, model mencapai 29 benar dan hanya 1 salah ke *Calathea_White_Fusion*. Untuk *Mangkokaan_Variegata* terdapat 22 benar dan 1 salah prediksi ke *Syngonium_Wendlandii*. Terakhir, kelas *Syngonium_Wendlandii* mencatat 29 benar dan 1 salah ke *Diven_Camilie*.

Secara keseluruhan, model menunjukkan performa yang sangat baik pada data test dengan sebagian besar kelas diprediksi benar hampir 100%. Kesalahan prediksi yang muncul hanya terjadi dalam jumlah kecil, dan umumnya disebabkan oleh kemiripan visual antar spesies tanaman. Hal ini menegaskan bahwa model memiliki kemampuan generalisasi yang tinggi pada data uji.

2. Evaluasi model

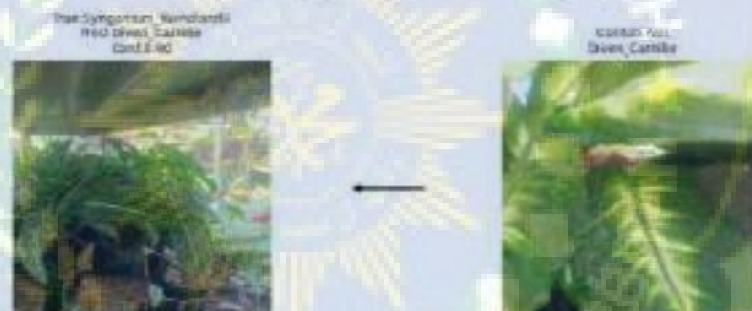
Tabel 4. Prediksi Kesalahan

Kelas	TP	TN	FP	FN
Calathea_White_Fusion	21	177	2	2
Diven_Brazil	20	181	0	1
Diven_Camilie	28	170	2	2
Diven_Rebecca	24	178	0	0
Diven_White_Etna	29	171	2	0
Mangkokaan_Variegata	22	179	0	1
Polyscias_Guilfoylei	22	179	1	0
Syngonium_Wendlandii	29	172	0	1

Tabel di atas menunjukkan rincian hasil prediksi model pada data test untuk setiap kelas dalam bentuk True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Pada kelas *Calathea_White_Fusion* terdapat 21 data yang terprediksi benar (TP), 2 data salah diprediksi ke kelas lain (FN), serta 2 data dari kelas lain yang keliru diprediksi sebagai *Calathea_White_Fusion* (FP), dengan 177 sisanya benar bukan *Calathea* (TN). Kelas *Diven_Brazil* memiliki 20 prediksi benar, 1 kesalahan ke kelas lain (FN), dan tidak ada data kelas lain yang salah diprediksi sebagai *Diven_Brazil* (FP=0), sehingga akurasinya relatif tinggi. Kelas *Diven_Camilie* mencatat 28 prediksi benar, dengan 2 kesalahan ke kelas lain (FN) dan 2 data dari kelas lain yang salah diprediksi sebagai *Diven_Camilie* (FP). Untuk kelas *Diven_Rebecca*, model berhasil memprediksi seluruh 24 sampel dengan benar (TP=24, FN=0, FP=0), menunjukkan performa sempurna. Kelas *Diven_White_Etna* memperoleh 29 prediksi benar, tanpa kesalahan ke kelas lain (FN=0), meskipun ada 2 data dari kelas lain yang salah diklasifikasikan sebagai *Diven_White_Etna* (FP=2). Kelas *Mangkokaan_Variegata* diprediksi dengan baik pada 22 sampel, namun

terdapat 1 sampel yang salah diprediksi (FN=1) tanpa adanya FP. *Polyscias_Guilfoylei* juga mencapai 22 prediksi benar, dengan 1 data dari kelas lain yang salah masuk ke kelas ini (FP=1). Terakhir, kelas *Syngonium_Wendlandii* memiliki 29 prediksi benar, dengan 1 data yang seharusnya kelas ini namun salah diklasifikasikan (FN=1), dan tidak ada FP.

Berikut ditunjukkan salah satu contoh kesalahan prediksi yang dilakukan oleh model. Pada kasus ini, gambar asli dari kelas *Syngonium_Wendlandii* diprediksi sebagai kelas *Diven_Camilie* dengan tingkat kepercayaan (confidence) 0,90. Hal ini terjadi karena bentuk dan corak daun *Syngonium_Wendlandii* pada gambar cukup mirip dengan pola visual daun *Diven_Camilie*, sehingga model mengalami kebingungan dalam melakukan klasifikasi. Kemiripan morfologi antar spesies tanaman inilah yang menjadi penyebab utama kesalahan prediksi pada model.



Gambar 24. Prediksi Kesalahan

Beberapa metrik evaluasi yang digunakan dalam penelitian ini adalah sebagai berikut:

a. *Calathea white fusion*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{21 + 177}{21 + 177 + 2 + 2} = 0.980 \text{ (98,0\%)}$$

$$Precision = \frac{TP}{TP + FP} = \frac{21}{21 + 2} = \frac{21}{23} = 0.913$$

$$Recall = \frac{TP}{TP + FN} = \frac{21}{21 + 2} = 0.913$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2 \times 0.913 \times 0.913}{0.913 + 0.913} = 0.913$$

b. Diven brazil

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{20+181}{20+181+0+1} = 0.995 \text{ (99,5\%)}$$

$$Precision = \frac{TP}{TP+FP} = \frac{20}{20+0} = 1.000$$

$$Recall = \frac{TP}{TP+FN} = \frac{20}{20+1} = 0.952$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision+recall} = \frac{2 \times 1.000 \times 0.952}{1.000+0.952} = 0.976$$

c. Diven camillie

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{28+170}{28+170+2+2} = 0.980 \text{ (98,0\%)}$$

$$Precision = \frac{TP}{TP+FP} = \frac{28}{28+2} = 0.93$$

$$Recall = \frac{TP}{TP+FN} = \frac{28}{28+2} = 0.933$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision+recall} = \frac{2 \times 0.933 \times 0.933}{0.933+0.933} = 0.933$$

d. Diven rebcca

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{24 + 178}{24 + 178 + 0 + 0} = 1.000 \text{ (100\%)}$$

$$Precision = \frac{TP}{TP + FP} = \frac{24}{24 + 0} = 1.000$$

$$Recall = \frac{TP}{TP + FN} = \frac{24}{24 + 0} = 1.000$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2 \times 1.000 \times 1.000}{1.000 + 1.000} = 1.000$$

e. Diven white etna

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{29+171}{29+171+2+0} = 0.990 \text{ (99,0\%)}$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{29}{29+2} = 0.935$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{29}{29+0} = 1.000$$

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times 0.935 \times 1.000}{0.935 + 1.000} = 0.967$$

f. Mangkokan variegata

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{22+179}{22+179+0+1} = 0.995(99,5\%)$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{22}{22+0} = 1.000$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{22}{22+1} = 0.956$$

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times 1.000 \times 0.956}{1.000 + 0.956} = 0.977$$

g. Polyscias guilfoylei

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{22+179}{22+179+1+0} = 0.995(99,5\%)$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{22}{22+1} = 0.956$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{22}{22+0} = 1.000$$

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times 0.956 \times 1.000}{0.956 + 1.000} = 0.977$$

h. Syngonium wedlandii

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{29+172}{29+172+0+1} = 0.995(99,5\%)$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{29}{29+0} = 1.000$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{29}{29+1} = 0.967$$

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \times 0.967 \times 1.000}{1.000 + 0.967} = 0.984$$

3. Classification report

Tabel 5. Classification Report

Kelas	Precision	Recall	F1-Score	Support
Calathea_White_Fusion	0.913	0.913	0.913	23
Diven_Brazil	1.000	0.952	0.976	21
Diven_Camilie	0.933	0.933	0.933	30
Diven_Rebecca	1.000	1.000	1.000	24
Diven_White_Etna	0.935	1.000	0.967	29
Mangkokaan_Variegata	1.000	0.957	0.978	23
Polyscias_Guilfoylei	0.957	1.000	0.978	22
Syngonium_Wendlandii	1.000	0.967	0.983	30

Berdasarkan hasil classification report, model mencapai akurasi keseluruhan sebesar 96,50% dengan nilai precision, recall, dan F1-score yang rata-rata konsisten tinggi di semua kelas (di atas 0,88). Kelas Polyscias Guilfoylei berhasil diprediksi sempurna dengan precision, recall, dan F1-score = 1,00, sedangkan kelas lain seperti Diven Camilie, Mangkokaan Variegata, dan Syngonium Wendlandii juga menunjukkan performa sangat baik dengan skor di atas 0,93. Setelah itu, hasil confusion matrix memperlihatkan pola kesalahan prediksi yang relatif kecil dan terjadi hanya pada kelas-kelas dengan kemiripan morfologi, misalnya Diven Brazil kadang tertukar dengan Diven Etna atau Diven Rebecca, serta Mangkokaan Variegata sesekali tertukar dengan Syngonium Wendlandii. Temuan ini menunjukkan bahwa secara umum model mampu melakukan klasifikasi dengan sangat baik, sementara kesalahan yang muncul masih dapat dijelaskan oleh kemiripan visual antarspesies tanaman.

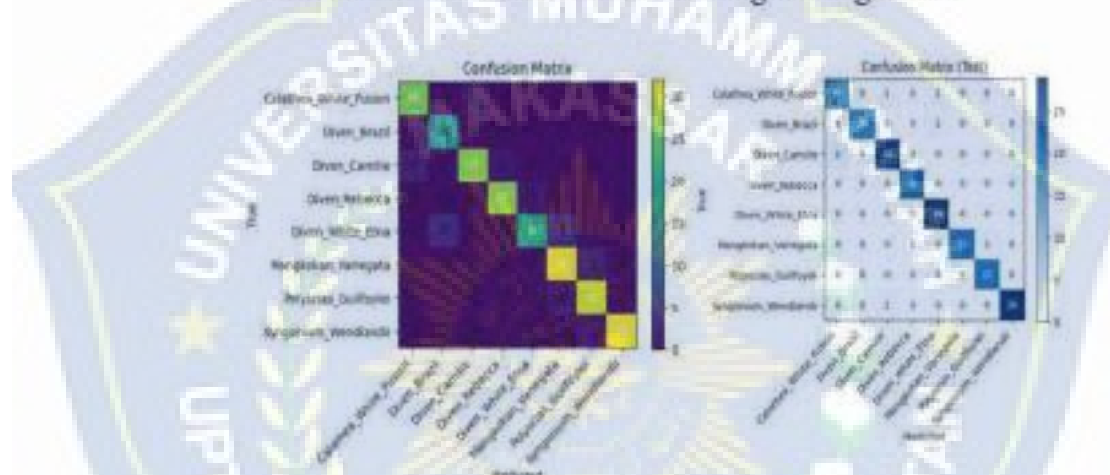
4. Evaluasi model CNN dan InceptionV3

Berikut perbandingan model CNN dan Inception V3 pada data uji
Sebagai berikut:

Tabel 6. Perbandingan CNN dan Inception

Model	Akurasi Test	Macro F1	Weighted F1	keterangan
CNN	93.30%	0.929	0.932	Masih terjadi kesalahan pada kelas dengan kemiripan daun, terutama <i>Diven_White_Etna</i> (recall 0.76).
InceptionV3	96.50%	0.966	0.965	Hampir semua kelas memiliki f1-score di atas 0.93; <i>Diven_Rebecca</i> bahkan mencapai 1.000.

Berikut confusion matrix untuk masing-masing model



Gambar 25. Confusion matrix CNN dan InceptionV3

Hasil evaluasi menunjukkan bahwa model CNN mampu mencapai akurasi sebesar 93,30% dengan nilai macro F1 sebesar 0,929. Nilai ini menandakan bahwa CNN cukup baik dalam mengenali pola daun pada sebagian besar kelas tanaman hias, meskipun performanya tidak merata. Salah satu kelemahan terlihat pada kelas *Diven_White_Etna* yang hanya memiliki recall 0,76, sehingga model masih sering salah mengklasifikasikan kelas ini ke tanaman lain yang memiliki kemiripan visual. Hal ini menunjukkan bahwa CNN yang dibangun dari nol masih memiliki keterbatasan dalam mengekstraksi fitur kompleks pada kelas yang serupa.

Sedangkan, model InceptionV3 dengan transfer learning mampu memberikan hasil yang lebih unggul, dengan akurasi test mencapai 96,50% dan macro F1 sebesar 0,966. Hampir seluruh kelas memperoleh nilai precision, recall, dan f1-score di atas 0,93, bahkan pada kelas Diven_Rebecca model ini mampu mencapai performa sempurna (precision, recall, dan f1-score = 1,000). Konsistensi ini juga terlihat pada confusion matrix, di mana jumlah salah prediksi jauh lebih sedikit dibanding CNN.

Maka InceptionV3 memiliki performa yang lebih stabil dan unggul dibanding CNN, baik dari sisi akurasi keseluruhan maupun pemerataan performa per kelas.



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian mengenai klasifikasi tanaman hias daun menggunakan Convolutional Neural Network (CNN) dengan arsitektur InceptionV3, dapat diambil beberapa kesimpulan sebagai berikut:

1. Model Convolutional Neural Network (CNN)

CNN mengklasifikasikan tanaman hias daun dengan cara mengekstraksi pola visual dari citra secara bertahap melalui lapisan konvolusi untuk mendeteksi tepi, tekstur, dan corak daun. Fitur-fitur penting tersebut kemudian dipadatkan melalui pooling layer dan diteruskan ke fully connected layer dengan softmax untuk menghasilkan probabilitas kelas. Kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi, sehingga model mampu membedakan delapan jenis tanaman hias daun, meskipun masih ada beberapa kesalahan pada kelas yang memiliki morfologi mirip.

2. Hasil akurasi Convolutional Neural Network (CNN)

Proses training menunjukkan bahwa model mampu belajar dengan sangat baik pada data latih (training) dengan akurasi mencapai lebih dari 99% disertai nilai loss yang semakin kecil. Pada data validasi, akurasi terbaik yang ditetapkan oleh mekanisme EarlyStopping terjadi pada epoch ke-10 dengan nilai sebesar 97,32%, meskipun akurasi validasi tertinggi sempat mencapai 98,66% pada epoch ke-15. Hasil ini menunjukkan bahwa model tidak hanya kuat dalam menghafal data latih, tetapi juga memiliki kemampuan generalisasi yang baik pada data yang belum pernah dilatih.

Selanjutnya, evaluasi menggunakan data test menghasilkan akurasi sebesar 96,5% dengan precision, recall, dan f1-score rata-rata yang tinggi pada hampir semua kelas. Hal ini menegaskan bahwa model tidak mengalami overfitting berlebihan, karena performa pada data test tetap konsisten dengan hasil validasi. Dengan demikian, model dapat dikatakan andal dan efektif dalam melakukan klasifikasi gambar tanaman hias berdasarkan delapan kelas yang diuji.

B. Saran

Untuk pengembangan lebih lanjut peneliti memberikan beberapa saran sebagai berikut.

1. Mencoba model convolutional neural network yang lain seperti, EfficientNet, ResNet, atau MobileNet.
2. Melakukan penambahan jumlah dataset
3. Melakukan uji coba pada dataset baru atau gambar dari sumber berbeda untuk memastikan model tidak hanya bekerja baik pada dataset penelitian ini



DAFTAR PUSTAKA

- Fadjeri, A., Saputra, B. A., Adri Ariyanto, D. K., & Kurniatin, L. (2022). Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital. *Jurnal Ilmiah SINUS*, 20(2), 1.
<https://doi.org/10.30646/sinus.v20i2.601>
- Hamdani mubarak. (2019). *IDENTIFIKASI EKSPRESI WAJAH BERBASIS CITRA MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)*.
- Hamdani, S. S., Rahmadi, A., & Subagja, R. (2025). Budidaya Tanaman Hias Calathea White Fusion Menggunakan Teknik Perbanyakan Vegetatif Split di Art Garden, Cigugur Girang, Kabupaten Bandung Barat Cultivation of Calathea White Fusion Ornamental Planis Using Split Vegetative Propagation Technique at Art Garden, Cigugur Girang, West Bandung Regency. *Gunung Djati Conference Series*, 49.
- Hidayat, D. (2022). KLASIFIKASI JENIS MANGGA BERDASARKAN BENTUK DAN TEKSTUR DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) CLASSIFICATION OF TYPES OF MANGO BASED ON LEAVE SHAPE AND TEXTURE USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD. *Journal of Information Technology and Computer Science (INTECOMS)*, 5(1).
- Kartika Sari, P., Rosanti, D., Panca Putri, Y., Studi Biologi, P., & Sains dan Teknologi, F. (n.d.). KARAKTERISTIK MORFOLOGI JENIS TANAMAN HIAS PEKARANGAN RUMAH DI KELURAHAN PLAJU ULU KOTA PALEMBANG. In *Jurnal Indobiosains* (Vol. 4).
- Mareta Tama, A., & Candra Noor Santi, R. (2023). KLASIFIKASI JENIS TANAMAN HIAS MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) ORNAMENTAL PLANT CLASSIFICATION USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD. *Journal of Information Technology and Computer Science (INTECOMS)*, 6(2).
- Putri Navia Rena. (2019). *PENERAPAN METODE CONVOLUTIONAL NEURAL*.
- Ramadhani, F., Satria, A., & Salamah, S. (2023). Implementasi Algoritma Convolutional Neural Network dalam Mengidentifikasi Dini Penyakit pada Mata Katarak. *Sudo Jurnal Teknik Informatika*, 2(4), 167–175.
<https://doi.org/10.56211/sudo.v2i4.408>

Rasywir, E., Sinaga, R., Pratama, Y., Dinamika, U., & Jambi, B. (2020a). *Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)*. 22(2). <https://doi.org/10.31294/p.v2i1i2>

Suartika. (n.d.). *KLASIFIKASI CITRA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CALTECH 101*.

Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, 3(1).

Dreamstime. (2025). *Polyscias scutellaria variegata – ornamental plant known as mangkokan leaf* [Fotografi]. Dreamstime.

New World Wildflowers. (2025). *Polyscias guilfoylei*. Retrieved August 29, 2025, from <https://nwwildflowers.com>



LAMPIRAN

Lampiran 1 Source codeSource code

```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)

import os, math, itertools, random, numpy as np, pandas as pd
from pathlib import Path
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.layers import Rescaling,
GlobalAveragePooling2D, Dropout, Dense, Input
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras import Model
from sklearn.metrics import classification_report,
confusion_matrix

MYDRIVE = Path("/content/gdrive") / "MyDrive"
DATASET_FOLDER = "Dataset_Tanaman_Hias" # <<-- ganti jika
perlu
ROOT_DIR = MYDRIVE / DATASET_FOLDER
SAVE_DIR = MYDRIVE / "models_inceptionv3"
IMG_SIZE, BATCH_SIZE, SEED = (299, 299), 32, 42

SAVE_DIR.mkdir(parents=True, exist_ok=True)
tf.keras.utils.set_random_seed(SEED)
try:
    tf.config.experimental.enable_op_determinism(True)
except Exception:
    pass

print("ROOT_DIR:", ROOT_DIR)
assert ROOT_DIR.exists(), "Folder dataset tidak ditemukan. Cek
nama DATASET_FOLDER!"
%matplotlib inline

print("Memuat dataset...")

# 80% - train
train_ds = tf.keras.utils.image_dataset_from_directory(
    ROOT_DIR, image_size=IMG_SIZE, batch_size=BATCH_SIZE,
```

```

        validation_split=0.2, subset="training", seed=SEED,
        shuffle=True
    )

    # 20% → val+test sementara
    val_test_ds = tf.keras.utils.image_dataset_from_directory(
        ROOT_DIR, image_size=IMG_SIZE, batch_size=BATCH_SIZE,
        validation_split=0.2, subset="validation", seed=SEED,
        shuffle=True
    )

    # 10% val + 10% test
    vt_batches = val_test_ds.cardinality().numpy()
    val_batches = vt_batches // 2
    val_ds = val_test_ds.take(val_batches)
    test_ds = val_test_ds.skip(val_batches)

    class_names = train_ds.class_names
    print("Classes:", class_names)

    def count_images(ds): return sum(1 for _ in ds.unbatch())
    n_train, n_val, n_test = map(count_images, (train_ds, val_ds,
        test_ds))
    print(f"Images → Train:{n_train} | Val:{n_val} | Test:{n_test}
        | Total:{n_train+n_val+n_test}")

    AUTOTUNE = tf.data.AUTOTUNE
    train_ds = train_ds.cache().prefetch(AUTOTUNE)
    val_ds = val_ds.cache().prefetch(AUTOTUNE)
    test_ds = test_ds.cache().prefetch(AUTOTUNE)

    images, labels = next(iter(train_ds))

    data_augmentation = tf.keras.Sequential([
        layers.RandomFlip("horizontal", seed=SEED),
        layers.RandomRotation(0.2, seed=SEED),
        layers.RandomZoom(0.2, seed=SEED),
        layers.RandomContrast(0.2, seed=SEED),
    ], name="augment")

    aug_images = data_augmentation(images, training=True)

    plt.figure(figsize=(12,5))
    for i in range(8):

```

```

plt.subplot(2,8,i+1);
plt.imshow(images[i].numpy().astype("uint8")); plt.axis("off")
plt.subplot(2,8,8+i+1);
plt.imshow(tf.cast(tf.clip_by_value(aug_images[i],0,255),
tf.uint8)); plt.axis("off")
plt.suptitle("Top: Original | Bottom: Augmented");
plt.tight_layout(); plt.show()

```

```

# ambil satu contoh dari batch
sample_uint8 = images[0] #
(299,299,3) uint8
sample_f32 = tf.cast(sample_uint8, tf.float32) / 255. #
float32 [0,1]
sample_f32 = tf.expand_dims(sample_f32, 0) #
tambah batch dim

# definisikan layer satuan
aug_layers = {
    "Original": None,
    "Flip": layers.RandomFlip("horizontal", seed=SEED),
    "Rotation": layers.RandomRotation(0.2, seed=SEED),
    "Zoom": layers.RandomZoom(0.2, seed=SEED),
    "Contrast": layers.RandomContrast(0.2, seed=SEED),
}

def to_uint8_for_show(x):
    # x float32 [0,1] -> uint8 [0,255]
    x = tf.clip_by_value(x, 0.0, 1.0)
    x = tf.cast(x * 255.0, tf.uint8)
    return x.numpy()

plt.figure(figsize=(12,3))
for i, (name, layer) in enumerate(aug_layers.items()):
    if layer is None:
        out = to_uint8_for_show(sample_f32[0])
    else:
        # apply layer pada data float32 [0,1]
        aug_out = layer(sample_f32, training=True)[0]
        out = to_uint8_for_show(aug_out)

plt.subplot(1, len(aug_layers), i+1)
plt.imshow(out); plt.axis("off"); plt.title(name)

```

```

plt.suptitle("Contoh Augmentasi Per-Layer", y=1.05)
plt.tight_layout(); plt.show()

x_test, _ = next(iter(test_ds)) # <- tambahkan baris ini
kalau belum ada

x0 = tf.cast(x_test, tf.float32)/255.0
off = tf.cast(data_augmentation(x_test, training=False),
tf.float32)/255.0
on = tf.cast(data_augmentation(x_test, training=True ),
tf.float32)/255.0

print("Max diff (False, 0-1):",
float(tf.reduce_max(tf.abs(off-x0))))
print("Max diff (True, 0-1):", float(tf.reduce_max(tf.abs(on
-x0))))

from tensorflow.keras.layers import Rescaling,
GlobalAveragePooling2D, Dropout, Dense, Input
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras import Model

# pre-processing [-1,1] khusus InceptionV3
preprocess_layer = Rescaling(1./127.5, offset=-1,
name="inception_preprocess")

# backbone
base = InceptionV3(weights="imagenet", include_top=False,
input_shape=IMG_SIZE+(3,))
base.trainable = False # Stage-1: freeze

# head
inputs = Input(shape=IMG_SIZE+(3,), name="image")
x = data_augmentation(inputs) # <- sudah
didefinisikan di Cell 2/2a
x = preprocess_layer(x)
x = base(x, training=False)
x = GlobalAveragePooling2D(name="gap")(x)
x = Dropout(0.3, name="dropout")(x)
outputs = Dense(len(class_names), activation="softmax",
name="classifier")(x)

model = Model(inputs, outputs, name="inceptionv3_plant")
model.summary()

```

```

from tensorflow.keras.callbacks import EarlyStopping,
ReduceLROnPlateau, ModelCheckpoint

CKPT1 = str(SAVE_DIR / "best_stage1.keras")

callbacks_s1 = [
    EarlyStopping(monitor="val_loss", patience=5,
restore_best_weights=True, verbose=1),
    ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=3, verbose=1),
    ModelCheckpoint(CKPT1, monitor="val_accuracy", mode="max",
save_best_only=True, verbose=1),
]

model.compile(optimizer=tf.keras.optimizers.Adam(1e-3),
loss="sparse_categorical_crossentropy",
metrics=["accuracy"])

history1 = model.fit(train_ds, validation_data=val_ds,
epochs=25,
callbacks=callbacks_s1, verbose=1)

CKPT2 = str(SAVE_DIR / "best_stage2.keras")

base.trainable = True
for layer in base.layers[:-60]:
    layer.trainable = False

callbacks_s2 = [
    EarlyStopping(monitor="val_loss", patience=8,
restore_best_weights=True, verbose=1),
    ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=4, verbose=1),
    ModelCheckpoint(CKPT2, monitor="val_accuracy", mode="max",
save_best_only=True, verbose=1),
]

model.compile(optimizer=tf.keras.optimizers.Adam(1e-5),
loss="sparse_categorical_crossentropy",
metrics=["accuracy"])

history2 = model.fit(train_ds, validation_data=val_ds,
epochs=10,
callbacks=callbacks_s2, verbose=1)

```

```

from tensorflow import keras, io

best_ckpt = CKPT2 if Path(CKPT2).exists() else CKPT1
print("Using best checkpoint:", best_ckpt)

best_model = keras.models.load_model(best_ckpt, compile=False)
best_model.compile(optimizer=tf.keras.optimizers.Adam(1e-5),
                  loss="sparse_categorical_crossentropy",
                  metrics=["accuracy"])

FINAL_PATH = str(SAVE_DIR / "inceptionv3_final.keras")
best_model.save(FINAL_PATH); print("Saved:", FINAL_PATH)

test_loss, test_acc = best_model.evaluate(test_ds, verbose=0)
print(f"TEST → acc={test_acc:.4f} | loss={test_loss:.4f}")

# Cell 7 - Classification report + Confusion matrix (+ CSV)
from sklearn.metrics import classification_report,
confusion_matrix
import pandas as pd
import numpy as np
import itertools
import matplotlib.pyplot as plt

y_true, y_pred = [], []
for x, y in test_ds:
    p = best_model.predict(x, verbose=0)
    y_pred += list(np.argmax(p, axis=1))
    y_true += list(y.numpy())

# Report
report_txt = classification_report(y_true, y_pred,
target_names=class_names, digits=3)
print("\n=== Classification Report (Test) ===\n", report_txt)

# Simpan report & CM ke CSV
pd.DataFrame(classification_report(
    y_true, y_pred, target_names=class_names,
    output_dict=True)).T.to_csv(
    SAVE_DIR/"classification_report_test.csv")
cm = confusion_matrix(y_true, y_pred)
pd.DataFrame(cm, index=class_names,
columns=class_names).to_csv(
    SAVE_DIR/"confusion_matrix_test.csv")

```

```

# Plot CM
plt.figure(figsize=(6,5))
plt.imshow(cm, cmap="Blues"); plt.title("Confusion Matrix
(Test)"); plt.colorbar()
ticks = np.arange(len(class_names))
plt.xticks(ticks, class_names, rotation=45, ha="right");
plt.yticks(ticks, class_names)
th = cm.max()/2
for i, j in itertools.product(range(cm.shape[0]),
range(cm.shape[1])):
    plt.text(j, i, cm[i, j], ha="center", va="center",
            color="white" if cm[i, j] > th else "black",
            fontsize=9)
plt.ylabel("True"); plt.xlabel("Predicted");
plt.tight_layout(); plt.show()

```

```

import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix

# y_true = label asli
# y_pred = label prediksi
# pastikan kamu sudah punya kedua array ini
# contoh:
# y_true = [0,0,1,1,2,2,2]
# y_pred = [0,0,1,2,2,1,2]

# buat confusion matrix
cm = confusion_matrix(y_true, y_pred)
labels = class_names # isi sesuai nama kelas kamu

# hitung TP, FP, FN, TN per kelas
TP = np.diag(cm)
FN = cm.sum(axis=1) - TP
FP = cm.sum(axis=0) - TP
TN = cm.sum() - (TP + FN + FP)

# gabung ke tabel dataframe
df_conf = pd.DataFrame({
    'Kelas': labels,
    'TP': TP,
    'TN': TN,
    'FP': FP,
    'FN': FN
})

```

```

print(df_conf)

# === Ringkasan Total Prediksi Benar vs Salah ===
import numpy as np

# Total benar = jumlah diagonal confusion matrix
total_benar = int(np.trace(cm))
# Total semua sample
total_sample = int(cm.sum())
# Total salah = total - benar
total_salah = total_sample - total_benar

print(f"Total sample : {total_sample}")
print(f"Prediksi benar: {total_benar}")
print(f"Prediksi salah: {total_salah}")
print(f"Akurasi manual: {total_benar/total_sample:.4f}")

# Cell 8 - Test no-shuffle & path sinkron
val_test_noshuf = tf.keras.utils.image_dataset_from_directory(
    ROOT_DIR, image_size=IMG_SIZE, batch_size=1,
    validation_split=0.2, subset="validation", seed=SEED,
    shuffle=False
)
vt_batches = val_test_noshuf.cardinality().numpy()
val_batches = vt_batches // 2
val_noshuf = val_test_noshuf.take(val_batches)
test_noshuf = val_test_noshuf.skip(val_batches)

all_paths = val_test_noshuf.file_paths
val_count = sum(1 for _ in val_noshuf) # jumlah sampel val
(no-shuffle)
test_paths = all_paths[val_count:] # sisa = path test

# Prediksi untuk no-shuffle test
y_true_idx, y_pred_idx, y_conf = [], [], []
for x, y in test_noshuf:
    prob = best_model.predict(x, verbose=0)[0]
    y_pred_idx.append(int(np.argmax(prob)))
    y_conf.append(float(np.max(prob)))
    y_true_idx.append(int(y.numpy()[0]))
y_true_idx, y_pred_idx, y_conf = map(np.array, (y_true_idx,
y_pred_idx, y_conf))

print("Samples synced:", len(test_paths), len(y_true_idx),
len(y_pred_idx))

```

```

# === Cell 9-11 - df_mis, pair_counts, grid, dan side-by-side
===

import pandas as pd, matplotlib.pyplot as plt, tensorflow as
tf, math, glob, random

# --- Buat df_mis & pair_counts ---
mis_list = []
for i in range(len(y_true_idx)):
    if y_true_idx[i] != y_pred_idx[i]:
        mis_list.append({
            "idx": int(i),
            "true_idx": int(y_true_idx[i]),
            "pred_idx": int(y_pred_idx[i]),
            "true": class_names[int(y_true_idx[i])],
            "pred": class_names[int(y_pred_idx[i])],
            "conf": float(y_conf[i]),
            "path": test_paths[i],
        })
df_mis = pd.DataFrame(mis_list)

if df_mis.empty:
    print("Tidak ada salah prediksi di test set 🤖")
    pair_counts =
pd.DataFrame(columns=["true", "pred", "count"])
else:
    pair_counts = (df_mis.groupby(["true", "pred"])
                    .size().reset_index(name="count")
                    .sort_values(["count", "true", "pred"],
ascending=[False, True, True])
                    .reset_index(drop=True))
    print("\n=== Ringkasan Salah Prediksi (Top 10) ===")
    display(pair_counts.head(10))

# Simpan ke CSV
df_mis.to_csv(SAVE_DIR/"misclassified_test.csv", index=False)
pair_counts.to_csv(SAVE_DIR/"misclassified_pairs_test.csv",
index=False)
print("Saved misclassified details to:", SAVE_DIR)

# --- Fungsi grid salah prediksi ---
def show_misgrid(true_label: str, pred_label: str, k: int =
8):
    sub = df_mis[(df_mis['true']==true_label) &
(df_mis['pred']==pred_label)]
    if sub.empty:
        print(f"Tidak ada contoh: {true_label} ->
{pred_label}")

```

```

        return
    n = min(k, len(sub)); cols = min(4, n); rows =
math.ceil(n/cols)
    plt.figure(figsize=(cols*3, rows*3))
    for i, (_, r) in enumerate(sub.head(n).iterrows()):
        img =
tf.image.decode_image(tf.io.read_file(r['path']), 3)
        img = tf.image.resize(img, IMG_SIZE)
        plt.subplot(rows, cols, i+1)
        plt.imshow(tf.cast(img, tf.uint8)); plt.axis("off")
        plt.title(f"T: {r['true']} -
P: {r['pred']} \n Conf: {r['conf']:.2f}", fontsize=9)
        plt.suptitle(f"Salah Prediksi - {true_label} -
{pred_label} (n={len(sub)}", y=0.98)
        plt.tight_layout(); plt.show()

# --- Fungsi side-by-side ---
def show_pair_side_by_side(true_label: str, pred_label: str,
pick: int = 0, ref_random: bool = True):
    sub = df_mis[(df_mis['true']==true_label) &
(df_mis['pred']==pred_label)]
    if sub.empty:
        print(f"Tidak ada contoh: {true_label} -
{pred_label}")
        return
    pick = max(0, min(pick, len(sub)-1))
    r = sub.iloc[pick]

    left =
tf.image.resize(tf.image.decode_image(tf.io.read_file(r['path'
]), 3), IMG_SIZE)

    pred_dir = ROOT_DIR / pred_label
    cand = glob.glob(str(pred_dir/'*.jpg')) +
glob.glob(str(pred_dir/'*.jpeg')) +
glob.glob(str(pred_dir/'*.png'))
    if not cand:
        print("Folder contoh kosong:", pred_dir); return
    ref_path = random.choice(cand) if ref_random else cand[0]
    right =
tf.image.resize(tf.image.decode_image(tf.io.read_file(ref_path
), 3), IMG_SIZE)

    fig, ax = plt.subplots(1, 3, figsize=(10, 4),
gridspec_kw={'width_ratios': [1, 0.2, 1]})
    fig.suptitle(f"Salah Prediksi - {true_label} -
{pred_label}", y=0.97, fontsize=12, fontweight="bold")

```

```

ax[0].imshow(tf.cast(left, tf.uint8)); ax[0].axis("off")
ax[0].set_title(f"True:{true_label}\nPred:{pred_label}\nConf:{r['conf']:.2f}", fontsize=10)
ax[1].axis("off")
ax[1].annotate("", xy=(0.0,0.5), xytext=(1.0,0.5),
               xycoords="axes fraction", textcoords="axes
fraction",
               arrowprops=dict(arrowstyle="->", lw=3,
color="black"))
ax[2].imshow(tf.cast(right, tf.uint8)); ax[2].axis("off")
ax[2].set_title(f"Contoh Asli\n{pred_label}", fontsize=10)
plt.tight_layout(); plt.show()

# --- Contoh pakai otomatis salah terbanyak ---
if not df_mis.empty:
    top_true, top_pred, top_n = pair_counts.loc[0,
["true","pred","count"]]
    print(f"\nTampilkan salah terbanyak: {top_true} -
{top_pred} (n={top_n})")
    show_misgrid(top_true, top_pred, k=mn(8, top_n))
    show_pair_side_by_side(top_true, top_pred, pick=0)

```

Lampiran 2. Keterangan bebas plagiasi



**MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN**

Alamat Kantor : Jl. Sultan Alauddin No.259 Makassar 90221 Telp (0411) 864972, 881593, Fax (0411) 860588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

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

Nama : Andi Dian Fajarwati

Nim : 105841110620

Program Studi : Teknik Informatika

Dengan nilai

No	Bab	Nilai	Ambang Batas
1	Bab 1	9 %	10 %
2	Bab 2	25 %	25 %
3	Bab 3	8 %	15 %
4	Bab 4	0 %	10 %
5	Bab 5	5 %	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, 28 Agustus 2025

Mengetahui

Kepala UPT- Perpustakaan dan Penerbitan



Spesialis Sistem M.I.P
NBM. 964 591

Jl. Sultan Alauddin no 259 makassar 90122
Telepon (0411)864972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail: perpustakaan@unismuh.ac.id

Lampiran 3.Uji plagiasi



Andi dian fajarwati Nim : 105841110620 BAB 1

ORIGINALITY REPORT

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

3%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCE

1

journal.ipm2kpe.or.id

Internet Source

3%

2

digilibadmin.unismuh.ac.id

Internet Source

2%

3

journal.untar.ac.id

Internet Source

2%

4

www.researchgate.net

Internet Source

2%

Exclude quotes

Exclude bibliography

Exclude matches

Andi dian fajarwati Nim :
105841110620 BAB 2

by TutupTahap

Submission date: 26-Aug-2023 09:57PM (UTC+0700)
Submission ID: 2735622584
File name: TUTUP_BAB_2.docx (24.51M)
Word count: 3628
Character count: 14619

Andi dian fajarwati Nim : 105841110620 BAB 2

25%
UNPLAKED INDEX

27%
INTERNET SOURCES

14%
PUBLICATIONS

18%
STUDENT PAPERS

Excluded sources

1 pdfcoffee.com
Internet Source

13%

2 Submitted to Sriwijaya University
Student Paper

2%

3 publishing-widyagama.ac.id
Internet Source

2%

4 eprints.umm.ac.id
Internet Source

2%

5 ejournal.itn.ac.id
Internet Source

2%

6 Andrian Herbert Parsaoran Sitohang, Teguh Iman Hermanito, Candra Dewi Lestari, "KLASIFIKASI JENIS PENYAKIT PADA DAUN TUMBUHAN STROBERI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK ARSITEKTUR INCEPTIONV3", Jurnal Informatika dan Teknik Elektro Terapan, 2024
Publication

2%

7 Submitted to Rogers State University
Student Paper

2%

Exclude sources

0%

Exclude bibliography

0%

Exclude matches

0%

Andi dian fajarwati Nim :
105841110620 BAB 3

by TutupTahap

Submission date: 27-Aug-2025 03:06PM (UTC+0700)

Submission ID: 2736061176

File name: BAB_3_2.docx (63.97K)

Word count: 713

Character count: 4669

Andi dian fajarwati Nim : 105841110620 BAB 3

SIMILARITY REPORT

8%

SIMILARITY INDEX

8%

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

SEARCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

www.stuffspec.com

Internet Source

Exclude quotes

Exclude bibliography

Exclude matches





Submission date: 26-Aug-2025 10:01PM (UTC+0700)

Submission ID: 2735624483

File name: BAB_4_3.docx (3.44M)

Word count: 2356

Character count: 14343

Andi dian fajarwati Nim : 105841110620 BAB 4

ORIGINALITY REPORT

0%

SIMILARITY INDEX

PRIMARY SOURCES

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

Exclude quotes

Exclude bibliography

Exclude matches

Exclude





Submission date: 28-Aug-2025 10:26AM (UTC+0700)

Submission ID: 2736654668

File name: BAB_5_1.docx (15.97K)

Word count: 161

Character count: 1068

Andi Dian Fajarwati 105841110620 BAB V

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

REMISS SOURCES



repository.fe.unj.ac.id

Internet Source

5%

