

**ANALISIS PERBANDINGAN KINERJA ARSITEKTUR
CONVOLUTIONAL NEURAL NETWORK (VGG16, RESNET50,
DAN MOBILENETV2) DALAM DETEKSI PENYAKIT PADA
DAUN MELON**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar
Strata Satu Program Studi Informatika**



**DAFFA IMANULLOH GUNAWAN
22EO10016**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026**

**ANALISIS PERBANDINGAN KINERJA ARSITEKTUR
CONVOLUTIONAL NEURAL NETWORK (VGG16, RESNET50,
DAN MOBILENETV2) DALAM DETEKSI PENYAKIT PADA
DAUN MELON**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar
Strata Satu Program Studi Informatika**



**DAFFA IMANULLOH GUNAWAN
22EO10016**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026**

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa penulisan skripsi dengan judul “Analisis Perbandingan Kinerja Arsitektur *Convolutional Neural Network* (VGG16, ResNet50, dan MobileNetV2) dalam Deteksi Penyakit pada Daun Melon” adalah hasil karya saya dengan arahan dari pembimbing dan belum diajukan kepada pihak manapun. Sumber informasi yang dikutip dalam skripsi ini telah dicantumkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenarnya. Apabila di kemudian hari terdapat ketidaksesuaian dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Cilacap, 05 Agustus 2026



Daffa Imanulloh Gunawan
22EO10016

LEMBAR PENGESAHAN

Skripsi Saudara

Nama : Daffa Imanulloh Gunawan
NIM : 22EO10016
Fakultas/Prodi : Fakultas MIKOM / Informatika
Judul : Analisis Perbandingan Kinerja Arsitektur *Convolutional Neural Network* (VGG16, ResNet50, dan MobileNetV2) dalam Deteksi Penyakit pada Daun Melon

Telah disidangkan oleh Dewan Penguji Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap pada hari / tanggal :

Senin, 27 Juli 2026

Dan dapat diterima sebagai pemenuhan tugas akhir mahasiswa Program Strata Satu Program Studi Informatika (S.Kom.) Fakultas Matematika dan Ilmu Komputer (FMIKOM) pada Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap.

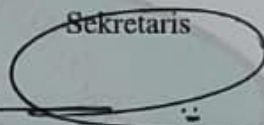
Cilacap, 05 Agustus 2026

Dewan Sidang

Ketua


Abdul Haq, SE., M.Cs.
NIDN. 0606067701

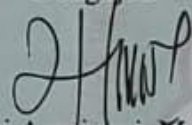
Sekretaris


Tri Anggoro, M.Kom.
NIDN. 0601128103

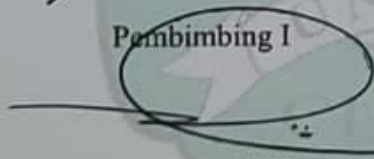
Penguji 1


Abdul Haq, SE., M.Cs.
NIDN. 0606067701

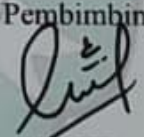
Penguji 2


Riski Aspriyani, M.Pd.
NIDN. 0616118901

Pembimbing I


Tri Anggoro, M.Kom.
NIDN. 0601128103

Pembimbing II


Eri Nur Shofi'i, S.H., M.A.
NIDN. 0602029501

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Komputer


Mochamad Taufiqurrochman Abdul Aziz Zein, S.Si, M.Kom.
NIDN. 2125098601



HALAMAN NOTA KONSULTAN

Abdul Haq, SE., M.Cs.

Dosen Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap

NOTA KONSULTAN

Hal : Skripsi Saudara Daffa Imanulloh Gunawan
Lampiran : -

Kepada
Yth. Dekan Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.

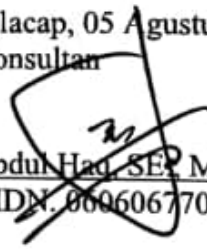
Setelah membaca, memeriksa dan melakukan perbaikan seperlunya maka skripsi saudara:

Nama : Daffa Imanulloh Gunawan
NIM : 22EO10016
Prodi : Informatika
Judul : Analisis Perbandingan Kinerja Arsitektur *Convolutional Neural Network* (VGG16, ResNet50, dan MobileNetV2) dalam Deteksi Penyakit pada Daun Melon

Dapat diajukan ke Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap untuk memenuhi syarat memperoleh gelar Strata Satu (S1).

Wassalamu'alaikum Wr. Wb.

Cilacap, 05 Agustus 2026
Konsultan


Abdul Haq, SE., M.Cs.
NIDN. 0006067701

NOTA PEMBIMBING

Cilacap, 17 Juli 2026

Kepada Yth :
Fakultas Matematika dan Komputer (FMIKOM)
UNUGHA Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.

Setelah melakukan bimbingan, telaah, arahan dan koreksi tahap penulisan skripsi saudara:

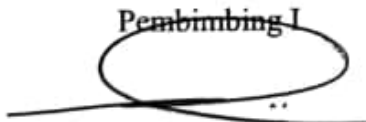
Nama : Daffa Imanulloh Gunawan
NIM : 22EO10016
Fakultas : Matematika dan Ilmu Komputer
Prodi : Informatika
Judul : Analisis Perbandingan Kinerja Arsitektur *Convolutional Neural Network* (VGG16, ResNet50, dan MobileNetV2) dalam Deteksi Penyakit pada Daun Melon

Kami berpendapat bahwa skripsi tersebut sudah dapat diajukan ke sidang skripsi. Bersama ini kami kirimkan skripsi tersebut, semoga dapat segera disidangkan.

Atas perhatiannya kami ucapkan terima kasih.
Wassalamu'alaikum Wr. Wb.

Mengetahui,

Pembimbing I



Tri Anggoro, M.Kom.
NIDN. 0601128103

Pembimbing II



Eri Nur Shofi'i, S.H., M.A.
NIDN. 0602029501

HALAMAN MOTO

“Napas pertamaku dibayar dengan pertarungan nyawa seorang Ibu, maka mustahil
tidak ada makna di balik itu”
(Penulis)

*“I’m gonna start a revolution from my bed”
(Don’t Look Back in Anger - Oasis)*

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, Rabb semesta alam yang senantiasa memberikan karunia sehingga penulis mampu menyelesaikan penulisan skripsi ini. Karya ini saya persembahkan kepada :

1. Kedua orang tua tercinta, Bapak Gunawan dan Ibu Mufi, yang senantiasa memberikan doa, kasih sayang, dukungan, dan pengorbanan tanpa batas. Semoga karya sederhana ini dapat menjadi salah satu bentuk kebahagiaan dan kebanggaan bagi Bapak dan Ibu.
2. Saudara tercinta, yang selalu menjadi tempat berbagi cerita, memberikan semangat, serta mengingatkan penulis untuk tetap kuat dalam menyelesaikan setiap proses.
3. Keluarga besar FMIKOM Angkatan 2022, atas kebersamaan, perjuangan, tawa, dan kenangan yang telah menjadi bagian berharga selama menempuh pendidikan.
4. Keluarga besar Program Studi Informatika, Sistem Informasi, dan Matematika, atas ilmu, pengalaman, arahan, dan pembelajaran yang telah diberikan selama masa perkuliahan.
5. Seluruh teman-teman Universitas Nahdlatul Ulama Al Ghazali Cilacap, yang telah memberikan doa, dukungan, motivasi, dan kebersamaan hingga penulis mampu menyelesaikan skripsi ini.
6. Bapak Muhtar Yusuf serta seluruh pihak Green House Labiq Farm, atas izin, bantuan, dan kesempatan yang diberikan kepada penulis dalam melakukan observasi dan pengumpulan data penelitian.

KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala limpahan nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi. Sholawat dan salam senantiasa penulis haturkan kepada Nabi Muhammad SAW sebagai pembimbing seluruh umat manusia.

Skripsi ini tidak akan selesai tanpa adanya bantuan dari banyak pihak, karena itu penulis menyampaikan terima kasih kepada :

1. Bapak Mochamad Taufiqurrochman Abdul Aziz Zein, S.Si, M.Kom., Dekan FMIKOM UNUGHA
2. Bapak Tri Anggoro, M.Kom., pembimbing I yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
3. Bapak Eri Nur Shofi'i, S.H., M.A., pembimbing II yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
4. Ibu Riski Aspriyani, M.Pd, sebagai Pembimbing Akademis
5. Bapak-Ibu dosen Program Studi Informatika FMIKOM UNUGHA
6. Semua pihak yang tidak dapat kami sebutkan satu persatu

Semoga Allah SWT membalas jerih payah dan pengorbanan yang telah diberikan dengan balasan yang lebih baik. Amiin. Penulis berharap semoga karya kecil ini bermanfaat bagi pembaca.

Cilacap, 17 Juli 2026



Daffa Imanulloh Gunawan

ABSTRAK

DAFFA IMANULLOH GUNAWAN. Analisis Perbandingan Kinerja Arsitektur *Convolutional Neural Network* VGG16, ResNet50, dan MobileNetV2 dalam Deteksi Penyakit Daun Melon. Dibimbing oleh TRI ANGGORO, M.Kom. dan ERI NUR SHOFI, S.H., M.A.

Penyakit daun pada melon hidroponik, seperti *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*, dapat menurunkan kualitas serta hasil panen. Identifikasi manual masih bersifat subjektif dan membutuhkan waktu, sehingga diperlukan pendekatan otomatis berbasis citra digital.

Penelitian ini bertujuan membangun dan membandingkan kinerja VGG16, ResNet50, dan MobileNetV2 dalam mendeteksi penyakit daun melon berdasarkan citra digital ke dalam empat kelas, yaitu *Healthy*, *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*.

Metode yang digunakan adalah eksperimen komparatif dengan 361 citra. Dataset dibagi menjadi 251 data *Training*, 54 *validation*, dan 56 *testing*. Tahapan penelitian meliputi resize 224×224 piksel, normalisasi, augmentasi, *class weighting*, *transfer learning*, pelatihan model, dan evaluasi menggunakan *accuracy*, *precision*, *recall*, serta *F1-score*.

Hasil pengujian menunjukkan bahwa ResNet50 memperoleh performa terbaik dengan *accuracy* sebesar 0,9821, *precision* sebesar 0,9844, *recall* sebesar 0,9821, dan *F1-score* sebesar 0,9825. MobileNetV2 dan VGG16 masing-masing memperoleh *accuracy* sebesar 0,9643. Dengan demikian, ResNet50 menjadi model terbaik pada dataset dan skenario penelitian ini.

Kata kunci: deteksi penyakit daun melon, *Convolutional Neural Network*, MobileNetV2, ResNet50, *transfer learning*, VGG16.

ABSTRACT

DAFFA IMANULLOH GUNAWAN. *Comparative Performance Analysis of Convolutional Neural Network Architectures (VGG16, ResNet50, and MobileNetV2) for Melon Leaf Disease Detection*. Supervised by TRI ANGGORO, M.Kom. and ERI NUR SHOFTI, S.H., M.A.

Leaf diseases in hydroponic melon plants, such as Powdery Mildew, Cucumber Mosaic Virus, and Leaf Curl Disease, can reduce crop quality and yield. Manual identification is subjective and time-consuming, creating a need for an automated image-based detection approach.

This study aimed to build and compare VGG16, ResNet50, and MobileNetV2 for detecting melon leaf diseases by classifying images into four classes: Healthy, Powdery Mildew, Cucumber Mosaic Virus, and Leaf Curl Disease.

A comparative experimental method was used with 361 images divided into 251 training images, 54 validation images, and 56 testing images. The research stages included resizing the images to 224×224 pixels, normalization, augmentation, class weighting, transfer learning, model training, and evaluation using accuracy, precision, recall, and F1-score.

The results showed that ResNet50 achieved the highest performance, with an accuracy of 0,9821, precision of 0,9844, recall of 0,9821, and F1-score of 0,9825. MobileNetV2 and VGG16 each achieved an accuracy of 0,9643. Therefore, ResNet50 was the best-performing model for melon leaf disease detection under the dataset and experimental scenario used in this study.

Keywords: Convolutional Neural Network, melon leaf disease detection, MobileNetV2, ResNet50, transfer learning, VGG16.

DAFTAR ISI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI	iii
LEMBAR PENGESAHAN	iv
HALAMAN NOTA KONSULTAN.....	v
HALAMAN NOTA PEMBIMBING.....	vi
HALAMAN MOTO	vii
HALAMAN PERSEMBAHAN	viii
KATA PENGANTAR	ix
ABSTRAK	x
<i>ABSTRACT</i>	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah.....	3
C. Batasan Masalah	3
D. Tujuan Penelitian	4
E. Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	6
A. Penelitian Terkait	6
B. Landasan Teori.....	14
BAB III METODOLOGI.....	25
A. Waktu dan Tempat Penelitian.....	25
B. Alat dan Bahan.....	25
C. Prosedur Penelitian	26
D. Analisis Data.....	32
E. Jadwal Penelitian	33
BAB IV HASIL DAN PEMBAHASAN.....	34
A. Hasil	34
B. Pembahasan.....	53
BAB V KESIMPULAN.....	58

A.	Kesimpulan	58
B.	Saran/Rekomendasi.....	59
	DAFTAR PUSTAKA	60
	LAMPIRAN.....	63
A.	Surat Permohonan Observasi	63
B.	Pengambilan Dataset.....	64
C.	Kode MobileNetV2.....	65
D.	Kode <i>Training</i>	67
E.	Kode Grafik <i>Accuracy Training</i> dan <i>Validation</i>	68
F.	Kode Grafik <i>Loss Training</i> dan <i>Validation</i>	69
G.	Kode Perbandingan MobileNetV2, VGG16, dan ResNet50	70
H.	Kode <i>Confusion matrix</i>	71
I.	Kode grafik perbandingan <i>Accuracy</i>	72

DAFTAR GAMBAR

Gambar 1 Mind Map Penelitian.....	13
Gambar 2 Daun melon sehat berdasarkan literatur	19
Gambar 3 Gejala Powdery Mildew pada daun melon berdasarkan literatur	20
Gambar 4 Gejala Cucumber Mosaic Virus pada daun melon.....	20
Gambar 5 Tanaman melon tanpa gejala dan dengan gejala Leaf Curl Disease	21
Gambar 6 Flowchart Penelitian.....	27
Gambar 7 Daun Healthy.....	36
Gambar 8 Daun Leaf Curl Disease	37
Gambar 9 Daun Powdery Mildew.....	37
Gambar 10 Daun Cucumber Mosaic Virus.....	38
Gambar 11 Sebelum resize citra	39
Gambar 12 Setelah resize citra.....	40
Gambar 13 Hasil augmentasi	41
Gambar 14 Grafik Accuracy MobileNetV2	43
Gambar 15 Grafik Loss MobileNetV2.....	43
Gambar 16 Hasil Training MobileNetV2	44
Gambar 17 Grafik Accuracy Model VGG16	45
Gambar 18 Grafik Loss Model VGG16.....	45
Gambar 19 Hasil Training VGG16	46
Gambar 20 Grafik Accuracy Model ResNet50.....	47
Gambar 21 Grafik Loss Model ResNet50.....	47
Gambar 22 Hasil Training ResNet50	48
Gambar 23 Confusion matrix Model MobileNetV2	51
Gambar 24 Confusion matrix Model VGG16.....	52
Gambar 25 Confusion matrix Model ResNet50	53

DAFTAR TABEL

Tabel 1 Tabel Penelitian Terkait	6
Tabel 2 Tabel kriteria visual penyakit daun tanaman	21
Tabel 3 Alat dan Bahan Penelitian.....	25
Tabel 4 Parameter Pelatihan Model	30
Tabel 5 Rancangan Hasil Perbandingan Model	32
Tabel 6 Jadwal Penelitian.....	33
Tabel 7 Jumlah Citra Dataset Setiap Kelas	34
Tabel 8 Karakteristik Pengambilan Citra Dataset Daun Melon.....	35
Tabel 9 Hasil pembagian dataset	39
Tabel 10 Hasil Pengujian dan Perbandingan Model.....	49
Tabel 11 Classification report MobileNetV2	49
Tabel 12 Classification report VGG16	50
Tabel 13 Classification report ResNet50	50
Tabel 14 Karakteristik Arsitektur CNN	56

BAB I PENDAHULUAN

A. Latar Belakang

Peningkatan produksi melon di Indonesia mendorong kebutuhan terhadap sistem budidaya yang mampu menghasilkan buah berkualitas. Berdasarkan data Badan Pusat Statistik dan Direktorat Jenderal Hortikultura, tanaman melon (*Cucumis melo* L.) merupakan salah satu komoditas hortikultura yang memiliki nilai ekonomi tinggi dan banyak dibudidayakan di Indonesia [1], [2]. Seiring berkembangnya teknologi budidaya, sistem hidroponik semakin banyak diterapkan karena mampu meningkatkan efisiensi penggunaan air, nutrisi, dan lahan serta menghasilkan buah dengan kualitas yang lebih baik. Namun demikian, budidaya melon hidroponik masih menghadapi berbagai kendala, salah satunya adalah serangan penyakit daun yang dapat menghambat pertumbuhan tanaman, menurunkan kualitas buah, bahkan menyebabkan kerugian ekonomi apabila tidak ditangani secara tepat. Penyakit seperti *Powdery Mildew* (embun tepung), *Cucumber Mosaic Virus* (penyakit kuning), dan *Leaf Curl Disease* (daun keriting) merupakan beberapa penyakit yang sering ditemukan pada tanaman melon dan menyebabkan perubahan warna, tekstur, maupun bentuk daun sehingga mengganggu proses fotosintesis dan pertumbuhan tanaman. Serangan penyakit yang tidak terdeteksi sejak dini berpotensi menyebar ke bagian tanaman lain sehingga menyebabkan penurunan hasil panen dan meningkatkan biaya pengendalian penyakit. Oleh karena itu, identifikasi penyakit daun sejak dini menjadi langkah penting dalam mendukung pengendalian penyakit secara efektif [3], [4].

Identifikasi penyakit daun pada budidaya melon umumnya masih dilakukan secara manual melalui pengamatan visual oleh petani atau tenaga ahli. Pendekatan tersebut memiliki beberapa keterbatasan, antara lain bergantung pada pengalaman pengamat, bersifat subjektif, membutuhkan waktu yang relatif lama, serta berpotensi menghasilkan kesalahan diagnosis ketika gejala beberapa penyakit memiliki karakteristik yang serupa. Selain itu, proses identifikasi manual menjadi kurang efisien apabila diterapkan pada jumlah tanaman yang banyak. Kondisi tersebut menunjukkan perlunya suatu pendekatan berbasis teknologi yang mampu melakukan identifikasi penyakit secara otomatis, cepat, objektif, dan konsisten sehingga dapat membantu proses pengambilan keputusan dalam budidaya melon hidroponik [3], [5].

Perkembangan *Artificial Intelligence* (AI), khususnya *Deep learning*, telah memberikan kontribusi yang signifikan dalam bidang pengolahan citra digital. Salah satu metode *Deep learning* yang paling banyak digunakan untuk deteksi citra adalah *Convolutional Neural Network* (CNN). CNN memiliki kemampuan mempelajari karakteristik visual suatu citra secara otomatis

melalui proses ekstraksi fitur tanpa memerlukan perancangan fitur secara manual sebagaimana pada metode *machine learning* konvensional. Kemampuan tersebut menjadikan CNN banyak diterapkan pada berbagai penelitian deteksi penyakit tanaman karena mampu mengenali pola warna, tekstur, bentuk, maupun bercak penyakit secara efektif [3], [4], [6].

Dalam implementasinya, performa CNN dipengaruhi oleh arsitektur jaringan yang digunakan. Beberapa arsitektur yang banyak diterapkan pada deteksi citra adalah VGG16, ResNet50, dan MobileNetV2. VGG16 memiliki struktur jaringan yang sederhana dengan penggunaan filter konvolusi berukuran 3×3 sehingga mampu mempelajari fitur citra secara mendalam. ResNet50 menerapkan konsep *residual learning* yang memungkinkan pelatihan jaringan yang lebih dalam tanpa mengalami penurunan performa akibat degradasi jaringan. Sementara itu, MobileNetV2 menggunakan *depthwise separable convolution*, *inverted residual*, dan *linear bottleneck* sehingga menghasilkan model yang lebih ringan dan efisien untuk diterapkan pada perangkat dengan sumber daya komputasi terbatas [7], [8], [9].

Berbagai penelitian telah menunjukkan bahwa CNN mampu memberikan performa yang baik dalam deteksi penyakit tanaman berbasis citra digital. Tian *et al.* membuktikan bahwa CNN efektif digunakan untuk deteksi penyakit daun tomat. Muslih dan Krismawan menunjukkan bahwa penerapan *transfer learning* menggunakan ResNet50 mampu meningkatkan performa deteksi. Attallah mengembangkan pendekatan *Compact CNN* dengan *transfer learning* dan *feature selection* untuk meningkatkan akurasi deteksi penyakit daun. Penelitian lain yang dilakukan oleh Mamatha dan Raju, Sharma *et al.*, Hukkeri *et al.*, serta Utami *et al.* juga menunjukkan bahwa berbagai arsitektur CNN maupun model *pretrained* mampu memberikan performa deteksi yang tinggi pada berbagai jenis tanaman. Meskipun demikian, hasil penelitian tersebut juga menunjukkan bahwa performa setiap arsitektur CNN masih bervariasi bergantung pada karakteristik dataset, teknik *preprocessing*, parameter pelatihan, serta metode evaluasi yang digunakan [5], [6], [10], [11], [12], [13], [14], [15].

Meskipun penelitian mengenai deteksi penyakit tanaman menggunakan CNN telah berkembang pesat, sebagian besar penelitian masih berfokus pada tanaman tomat, padi, dan jagung. Penelitian yang secara khusus membahas deteksi penyakit daun melon, terutama pada sistem budidaya hidroponik, masih relatif terbatas. Selain itu, penelitian terdahulu menggunakan dataset, teknik *preprocessing*, pembagian data, parameter pelatihan, dan metode evaluasi yang berbeda sehingga hasil performanya sulit dibandingkan secara langsung. Oleh karena itu, penelitian ini membandingkan VGG16, ResNet50, dan MobileNetV2 menggunakan dataset penyakit daun melon dan skenario eksperimen yang sama agar diperoleh perbandingan performa yang lebih objektif dan komprehensif [11], [12], [14], [15].

Berdasarkan uraian tersebut, penelitian ini bertujuan membangun dan membandingkan kinerja tiga arsitektur *Convolutional Neural Network* (CNN), yaitu VGG16, ResNet50, dan MobileNetV2, dalam deteksi penyakit daun melon hidroponik berbasis citra digital. Dataset penelitian terdiri atas empat kelas, yaitu *Healthy*, *Powdery Mildew* (Embun Tepung), *Cucumber Mosaic Virus* (Penyakit Kuning), dan *Leaf Curl Disease* (Daun Keriting). Tahapan penelitian meliputi pengumpulan dataset, *preprocessing* citra, perancangan model, pelatihan model, pengujian model, serta evaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil penelitian diharapkan dapat memberikan informasi mengenai arsitektur CNN yang memiliki performa terbaik sehingga dapat menjadi referensi dalam pengembangan sistem deteksi penyakit daun melon berbasis kecerdasan buatan.

B. Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan perancangan model deteksi penyakit daun melon hidroponik berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2?
2. Bagaimana mengukur performa model deteksi penyakit daun melon hidroponik berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2 berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*?

C. Batasan Masalah

Agar penelitian dapat dilakukan secara lebih terarah dan sesuai dengan ruang lingkup skripsi, maka penelitian ini memiliki beberapa batasan sebagai berikut:

1. Dataset penelitian terdiri atas empat kelas, yaitu *Healthy* (daun sehat), *Powdery Mildew* (Embun Tepung), *Cucumber Mosaic Virus* (Penyakit Kuning), dan *Leaf Curl Disease* (Daun Keriting).
2. Penelitian ini tidak membahas implementasi model ke dalam aplikasi mobile, website, atau perangkat IoT.
3. Penelitian ini tidak melakukan segmentasi objek daun secara khusus, sehingga citra daun diproses secara langsung sebagai input model.
4. Penelitian ini tidak membahas diagnosis penyakit tanaman secara agronomis, tetapi hanya melakukan deteksi berdasarkan pola visual pada citra daun.
5. Setiap arsitektur CNN diuji satu kali menggunakan konfigurasi parameter pelatihan yang sama. Penelitian ini tidak melakukan pengulangan eksperimen dengan variasi *random seed* maupun metode *cross-validation*.
6. Citra yang digunakan dibatasi pada daun melon yang telah berkembang dan menunjukkan karakteristik visual sesuai kelasnya. Daun pucuk yang belum berkembang secara penuh, daun sehat yang telah mengering, serta

citra yang buram, terpotong berlebihan, terlalu gelap, atau terlalu terang tidak digunakan dalam dataset.

D. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menyusun skenario pembangunan model deteksi penyakit daun melon hidroponik berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2.
2. Menganalisis performa model deteksi penyakit daun melon hidroponik berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2 berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.

E. Manfaat Penelitian

Manfaat penelitian merupakan dampak positif yang diharapkan dari hasil penyelesaian permasalahan yang diteliti, baik bagi pengembangan ilmu pengetahuan maupun bagi pihak-pihak yang terkait. Dalam penelitian ini, manfaat yang diharapkan dibagi menjadi dua, yaitu manfaat teoritis dan manfaat praktis.

1. Manfaat Teoritis

Secara teoritis, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan, khususnya pada bidang *Artificial Intelligence*, *Computer Vision*, dan *Deep learning*. Penelitian ini dapat menambah pemahaman mengenai penerapan *Convolutional Neural Network* (CNN) dalam deteksi citra penyakit daun tanaman. Selain itu, penelitian ini juga dapat memberikan wawasan mengenai perbandingan performa model deteksi berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2 berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.

2. Manfaat Praktis

a. Bagi Peneliti

Penelitian ini dapat meningkatkan pengetahuan dan keterampilan peneliti dalam menerapkan metode *Deep learning*, khususnya pada proses pembangunan model deteksi citra penyakit daun tanaman menggunakan arsitektur CNN.

b. Bagi Akademisi

Hasil penelitian ini dapat digunakan sebagai referensi tambahan dalam kegiatan pembelajaran maupun penelitian di bidang kecerdasan buatan, khususnya terkait deteksi citra dan perbandingan performa arsitektur CNN.

c. Bagi Pengembang Sistem

Penelitian ini dapat menjadi acuan awal dalam menentukan arsitektur CNN yang sesuai untuk pengembangan sistem deteksi penyakit daun tanaman berbasis citra digital.

d. Bagi Praktisi Pertanian

Hasil penelitian dapat membantu petani atau pengelola budidaya dalam memilih pendekatan deteksi penyakit daun yang memiliki tingkat akurasi tinggi sehingga proses identifikasi penyakit dapat dilakukan lebih cepat.

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Tabel 1 Tabel Penelitian Terkait

No	Penulis (Tahun)	Keterangan
(1)	(2)	(3)
1	Rodríguez-Lira <i>et al.</i> (2024) [3]	Judul: <i>Trends in Machine and Deep learning Techniques for Plant Disease Identification: A Systematic Review.</i> Metode: Systematic literature review terhadap penerapan <i>machine learning</i> dan <i>Deep learning</i> untuk identifikasi penyakit tanaman. Hasil: Penelitian menunjukkan bahwa <i>machine learning</i> dan <i>Deep learning</i> banyak digunakan dalam identifikasi penyakit tanaman berbasis citra digital. CNN menjadi salah satu metode yang banyak diterapkan karena mampu mempelajari fitur visual secara otomatis. Keterkaitan/Perbedaan: Relevan karena mendukung penggunaan <i>Deep learning</i>, khususnya CNN, untuk deteksi penyakit daun tanaman. Perbedaannya, penelitian tersebut berupa kajian literatur, sedangkan penelitian ini melakukan pembangunan dan pengujian model CNN menggunakan arsitektur VGG16, ResNet50, dan MobileNetV2.
2	Tian <i>et al.</i> (2022) [10]	Judul: <i>Tomato Leaf Diseases Recognition Based on Deep Convolutional Neural Networks.</i> Metode: <i>Deep Convolutional Neural Network</i> untuk pengenalan penyakit daun tomat berbasis citra digital. Hasil: Penelitian menunjukkan bahwa <i>deep Convolutional Neural Network</i> dapat digunakan untuk mengenali penyakit daun tomat berdasarkan citra daun. Keterkaitan/Perbedaan: Relevan karena sama-sama membahas deteksi penyakit daun

		tanaman menggunakan CNN. Perbedaannya, penelitian ini membandingkan tiga arsitektur CNN, yaitu VGG16, ResNet50, dan MobileNetV2.
3	Pacal <i>et al.</i> (2024) [4]	Judul: <i>A Systematic Review of Deep learning Techniques for Plant Diseases.</i> Metode: <i>Systematic literature review</i> mengenai teknik <i>Deep learning</i> untuk deteksi dan deteksi penyakit tanaman. Hasil: Penelitian menunjukkan bahwa <i>Deep learning</i> memiliki perkembangan yang signifikan dalam deteksi penyakit tanaman, terutama pada deteksi citra daun. Keterkaitan/Perbedaan: Relevan karena memberikan dasar teori mengenai penggunaan <i>Deep learning</i> pada penyakit tanaman. Perbedaannya, penelitian tersebut bersifat kajian literatur, sedangkan penelitian ini melakukan eksperimen langsung terhadap tiga arsitektur CNN.
4	Muslih dan Krismawan (2024) [5]	Judul: <i>Tomato Leaf Diseases Classification using Convolutional Neural Networks with Transfer learning ResNet-50.</i> Metode: CNN dengan <i>transfer learning</i> menggunakan arsitektur ResNet50 untuk deteksi penyakit daun. Hasil: Penelitian menunjukkan bahwa ResNet50 dengan <i>transfer learning</i> dapat digunakan untuk deteksi penyakit daun tomat berbasis citra digital. Keterkaitan/Perbedaan: Relevan karena menggunakan ResNet50, salah satu arsitektur yang juga digunakan dalam penelitian ini. Perbedaannya, penelitian tersebut hanya berfokus pada ResNet50, sedangkan penelitian ini membandingkan ResNet50 dengan VGG16 dan MobileNetV2.
5	Attallah (2023) [6]	Judul: <i>Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer learning and Feature Selection.</i> Metode: <i>Compact CNN, transfer learning,</i>

		dan <i>feature selection</i> untuk deteksi penyakit daun. Hasil: Penelitian menunjukkan bahwa CNN dengan <i>transfer learning</i> dapat membantu meningkatkan performa deteksi penyakit daun tomat. Keterkaitan/Perbedaan: Relevan karena membahas penerapan CNN dan <i>transfer learning</i> pada deteksi penyakit daun tanaman. Perbedaannya, penelitian tersebut menggunakan <i>Compact CNN</i> dan <i>feature selection</i>, sedangkan penelitian ini membandingkan arsitektur VGG16, ResNet50, dan MobileNetV2.
6	Mamatha G. dan Raju (2025) [11]	Judul: <i>Comparative Analysis of Deep learning Models for Tomato Leaf Disease Classification: Insights and Opportunities.</i> Metode: Perbandingan beberapa model <i>Deep learning</i> untuk deteksi penyakit daun. Hasil: Penelitian menunjukkan bahwa perbedaan model <i>Deep learning</i> dapat menghasilkan performa deteksi yang berbeda pada penyakit daun tomat. Keterkaitan/Perbedaan: Sangat relevan karena sama-sama membahas perbandingan model <i>Deep learning</i> untuk deteksi penyakit daun tanaman. Perbedaannya, penelitian ini difokuskan pada pembangunan model deteksi berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2 dalam satu skenario penelitian.
7	Sharma <i>et al.</i> (2025) [12]	Judul: <i>Deep learning Based Ensemble Model for Accurate Tomato Leaf Disease Classification by Leveraging ResNet50 and MobileNetV2 Architectures.</i> Metode: <i>Ensemble Deep learning</i> dengan memanfaatkan arsitektur ResNet50 dan MobileNetV2 untuk deteksi penyakit daun. Hasil: Penelitian menunjukkan bahwa kombinasi ResNet50 dan MobileNetV2 dapat digunakan untuk meningkatkan performa deteksi penyakit daun.

		Keterkaitan/Perbedaan: Relevan karena menggunakan ResNet50 dan MobileNetV2, yang juga digunakan dalam penelitian ini. Perbedaannya, penelitian tersebut menggunakan pendekatan <i>ensemble</i>, sedangkan penelitian ini membandingkan masing-masing arsitektur CNN secara terpisah.
8	Arifin dan Insani (2025) [13]	Judul: <i>Comparative Analysis of CNN, SVM, Decision Tree, Random Forest, and KNN for Maize Leaf Disease Detection Using Color and Texture Feature extraction.</i> Metode: Perbandingan CNN, SVM, <i>Decision Tree</i>, <i>Random Forest</i>, dan KNN untuk deteksi penyakit daun jagung menggunakan fitur warna dan tekstur. Hasil: Penelitian menunjukkan bahwa CNN dan metode <i>machine learning</i> konvensional dapat digunakan untuk deteksi penyakit daun tanaman, tetapi metode konvensional umumnya memerlukan proses ekstraksi fitur terlebih dahulu. Keterkaitan/Perbedaan: Relevan karena menjelaskan adanya metode lain selain CNN, seperti SVM, KNN, <i>Decision Tree</i>, dan <i>Random Forest</i>. Perbedaannya, penelitian tersebut membandingkan CNN dengan metode non-CNN, sedangkan penelitian ini berfokus pada perbandingan tiga arsitektur CNN, yaitu VGG16, ResNet50, dan MobileNetV2.
9	Hukkeri et al. (2024) [14]	Judul: <i>Classification of Various Plant Leaf Disease Using Pretrained CNN Models.</i> Metode: <i>Pretrained CNN models</i> untuk deteksi berbagai penyakit daun tanaman. Hasil: Penelitian menunjukkan bahwa <i>CNN pretrained models</i> dapat digunakan untuk deteksi penyakit daun tanaman dengan hasil yang baik. Keterkaitan/Perbedaan: Relevan karena sama-sama menggunakan pendekatan <i>CNN pretrained</i> untuk deteksi penyakit daun

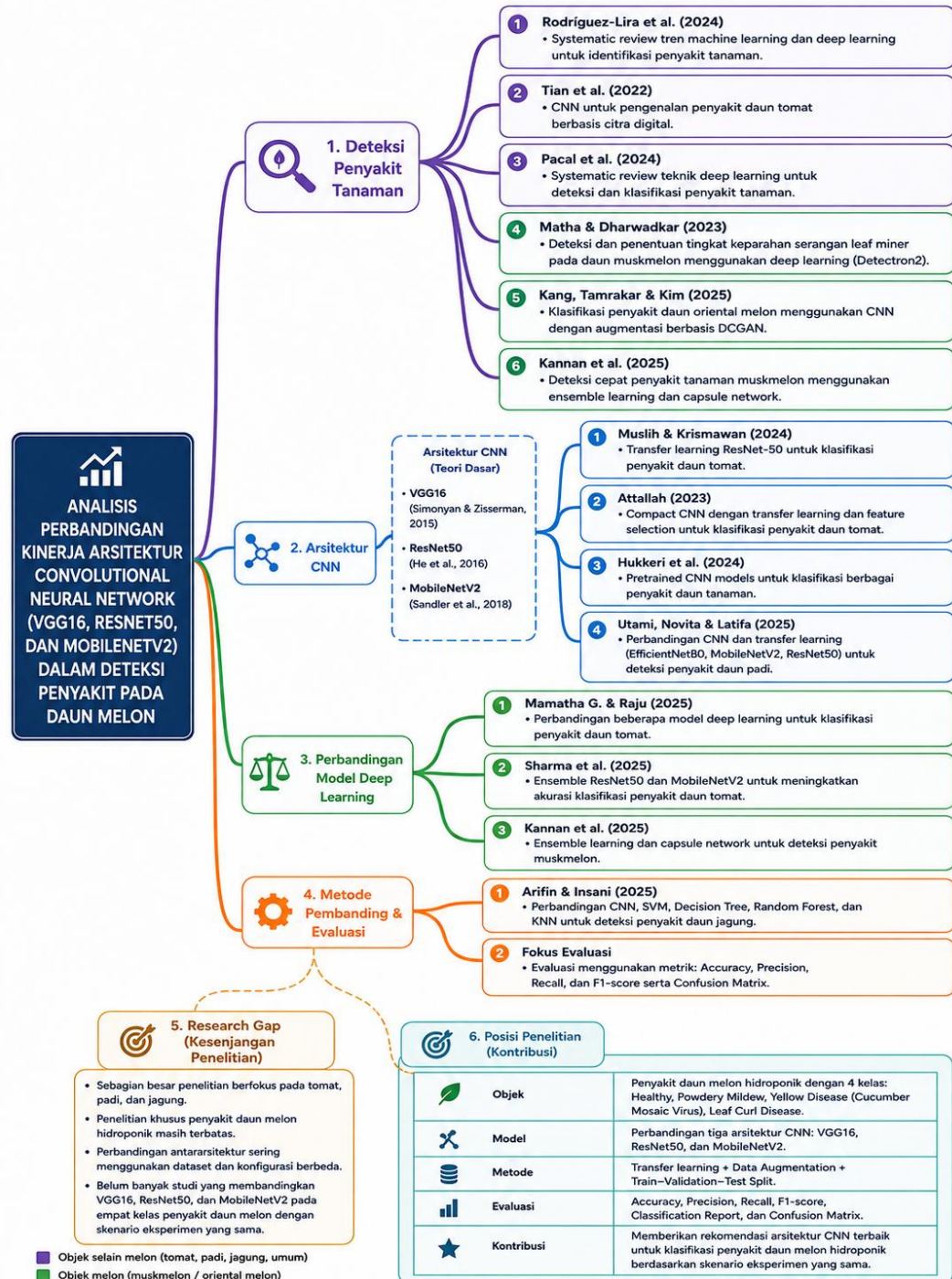
		tanaman. Perbedaannya, penelitian ini secara khusus membandingkan VGG16, ResNet50, dan MobileNetV2 berdasarkan metrik <i>accuracy</i> , <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> .
10	Utami <i>et al.</i> (2025) [15]	Judul: <i>Implementation and Comparative Analysis of CNN and Transfer learning Models (EfficientNetB0, MobileNetV2, and ResNet50) for Rice Leaf Disease Detection Based on Digital Images.</i> Metode: CNN dan <i>transfer learning</i> menggunakan EfficientNetB0, MobileNetV2, dan ResNet50 untuk deteksi penyakit daun padi. Hasil: Penelitian menunjukkan bahwa model CNN dan <i>transfer learning</i> dapat digunakan untuk deteksi penyakit daun padi berbasis citra digital. Keterkaitan/Perbedaan: Relevan karena sama-sama membandingkan model CNN dan <i>transfer learning</i> pada deteksi penyakit daun tanaman. Perbedaannya, penelitian tersebut menggunakan objek daun padi dan arsitektur EfficientNetB0, sedangkan penelitian ini menggunakan VGG16, ResNet50, dan MobileNetV2.
11	Math dan Dharwadkar (2023) [16]	Judul: <i>Deep learning and Computer Vision for Leaf Miner Infestation Severity Detection on Muskmelon (Cucumis melo) Leaves.</i> Metode: Penelitian menggunakan pendekatan <i>Deep learning</i> berbasis Detectron2 untuk mendeteksi, melokalisasi, dan menentukan tingkat keparahan serangan leaf miner pada daun muskmelon. Hasil: Model mampu mendeteksi lokasi serangan leaf miner dan mengelompokkan tingkat keparahan serangan berdasarkan citra daun muskmelon. Keterkaitan dan perbedaan: Penelitian tersebut relevan karena menggunakan objek daun muskmelon dan pengolahan citra berbasis <i>Deep learning</i>. Perbedaannya, penelitian tersebut berfokus pada deteksi

		objek dan tingkat keparahan serangan hama, sedangkan penelitian ini berfokus pada deteksi empat kelas kondisi daun menggunakan VGG16, ResNet50, dan MobileNetV2.
12	Kang, Tamrakar, dan Kim (2025) [17]	Judul: <i>Improving Oriental Melon Leaf Disease Classification via DCGAN-Based Image Augmentation.</i> Metode: Penelitian melakukan deteksi penyakit daun oriental melon menggunakan CNN. Untuk mengatasi keterbatasan jumlah data, penelitian menerapkan augmentasi tradisional, DCGAN, serta modified DCGAN dengan residual connection dan beberapa fungsi aktivasi. Hasil: Augmentasi menggunakan modified DCGAN menghasilkan citra dengan variasi dan karakteristik kelas yang lebih baik serta meningkatkan performa model deteksi penyakit daun oriental melon. Keterkaitan dan perbedaan: Penelitian tersebut sangat relevan karena sama-sama melakukan deteksi penyakit daun melon berbasis citra digital. Perbedaannya, penelitian tersebut berfokus pada peningkatan kualitas dan jumlah data menggunakan DCGAN, sedangkan penelitian ini berfokus pada perbandingan performa tiga arsitektur <i>transfer learning</i> dengan skenario eksperimen yang sama.
13	Kannan et al. (2025) [18]	Judul: <i>Improving Farming by Quickly Detecting Muskmelon Plant Diseases Using Advanced Ensemble Learning and Capsule Networks.</i> Metode: Penelitian menggunakan ensemble learning yang menggabungkan Multi-Class Capsule Network dengan model pretrained menggunakan mekanisme majority voting. Model dibandingkan dengan ResNet101, VGG19, dan GoogleNet. Hasil: Pendekatan ensemble digunakan untuk meningkatkan ketepatan deteksi dini

		penyakit dan hama pada tanaman muskmelon. Keterkaitan dan perbedaan: Penelitian tersebut relevan karena menggunakan objek muskmelon dan membandingkan beberapa model <i>Deep learning</i>. Perbedaannya, penelitian tersebut menggunakan ensemble learning dan capsule network, sedangkan penelitian ini mengevaluasi VGG16, ResNet50, dan MobileNetV2 secara terpisah menggunakan <i>accuracy</i>, <i>precision</i>, <i>recall</i>, <i>F1-score</i>, <i>classification report</i>, dan <i>confusion matrix</i>.
--	--	---

Sumber : Hasil olahan peneliti, 2026.

Diagram *Mind Map* dicontohkan pada gambar berikut ini:



Gambar 1 *Mind Map* Penelitian

B. Landasan Teori

1. Artificial Intelligence

Artificial Intelligence atau kecerdasan buatan merupakan bidang ilmu komputer yang berfokus pada pengembangan sistem yang mampu mengenali pola, mengambil keputusan, dan menyelesaikan permasalahan secara otomatis. Dalam bidang pertanian, *Artificial Intelligence* dapat digunakan untuk membantu proses identifikasi penyakit tanaman melalui analisis data dan citra digital. Pemanfaatan *Artificial Intelligence* dalam identifikasi penyakit tanaman menjadi penting karena proses pengamatan manual masih memiliki keterbatasan, seperti subjektivitas pengamatan, ketergantungan terhadap pengalaman manusia, dan membutuhkan waktu yang relatif lama [3], [4].

Pada penelitian ini, *Artificial Intelligence* digunakan sebagai dasar konseptual dalam pengembangan model deteksi penyakit daun tanaman berbasis citra digital. Pembahasan *Artificial Intelligence* tidak diarahkan pada seluruh cabang kecerdasan buatan, tetapi difokuskan pada penerapannya dalam deteksi citra menggunakan pendekatan *Deep learning* berbasis *Convolutional Neural Network*.

2. Machine learning

Machine learning merupakan bagian dari *Artificial Intelligence* yang memungkinkan sistem belajar dari data untuk melakukan prediksi atau deteksi. Dalam deteksi citra penyakit daun tanaman, *machine learning* dapat digunakan untuk membedakan citra daun sehat dan citra daun yang terkena penyakit berdasarkan fitur tertentu, seperti warna, tekstur, bentuk, dan pola bercak.

Beberapa metode *machine learning* yang dapat digunakan dalam deteksi citra antara lain *Support Vector Machine*, *K-Nearest Neighbor*, *Decision Tree*, *Random Forest*, dan *Naive Bayes*. Metode *machine learning* konvensional umumnya memerlukan proses ekstraksi fitur manual sebelum proses deteksi dilakukan. Berbeda dengan metode tersebut, CNN mampu mempelajari fitur visual secara otomatis dari citra, sehingga lebih sesuai untuk data citra yang memiliki pola visual kompleks [13], [19].

3. Deep learning

Deep learning merupakan pengembangan dari *machine learning* yang menggunakan jaringan saraf tiruan berlapis banyak untuk mempelajari representasi fitur dari data. Pada pengolahan citra, *Deep learning* mampu mempelajari pola visual secara otomatis tanpa memerlukan ekstraksi fitur manual. Kemampuan tersebut membuat *Deep learning* banyak digunakan dalam deteksi citra, termasuk pada deteksi penyakit daun tanaman [3], [4].

Dalam penelitian ini, *Deep learning* digunakan sebagai pendekatan utama untuk membangun model deteksi penyakit daun tanaman berbasis citra digital. Model *Deep learning* yang digunakan adalah *Convolutional Neural Network* dengan tiga arsitektur, yaitu VGG16, ResNet50, dan MobileNetV2. Ketiga arsitektur tersebut dibandingkan untuk mengetahui performa deteksi berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.

4. Pengolahan Citra Digital

Pengolahan citra digital merupakan proses pemrosesan gambar menggunakan komputer untuk memperoleh informasi tertentu dari citra. Dalam deteksi penyakit daun tanaman, pengolahan citra digital digunakan untuk menyiapkan citra agar dapat diproses oleh model deteksi. Tahapan ini dapat meliputi penyesuaian ukuran citra, normalisasi nilai piksel, augmentasi data, dan pembagian dataset [6], [10].

Pada penelitian ini, citra digital yang digunakan berupa gambar daun tanaman yang memiliki label kelas penyakit dan kelas daun sehat. Pengolahan citra dilakukan agar data memiliki ukuran dan format yang seragam sebelum dimasukkan ke dalam model CNN. Dengan *preprocessing* yang tepat, tahap *preprocessing* meliputi:

- a. Akuisisi citra
- b. *Resize* gambar
- c. Normalisasi data
- d. Augmentasi data
- e. Input ke model CNN

5. *Convolutional Neural Network* (CNN)

Convolutional Neural Network merupakan salah satu metode *Deep learning* yang dirancang untuk memproses data berbentuk citra. CNN bekerja dengan mengekstraksi fitur visual melalui *convolution layer*, menyederhanakan fitur melalui *pooling layer*, dan melakukan deteksi melalui *fully connected layer*. Keunggulan utama CNN adalah kemampuannya dalam mempelajari fitur secara otomatis dari citra tanpa memerlukan proses ekstraksi fitur manual [4], [6].

Dalam identifikasi penyakit tanaman, CNN banyak digunakan karena mampu mengenali pola visual pada daun, seperti warna, bercak, tekstur, dan bentuk kerusakan. Pada penelitian ini, CNN menjadi metode utama yang digunakan untuk membangun model deteksi penyakit daun tanaman. Arsitektur CNN yang dibandingkan adalah VGG16, ResNet50, dan MobileNetV2.

a. Arsitektur VGG16

VGG16 merupakan arsitektur CNN yang dikembangkan oleh Simonyan dan Zisserman. Arsitektur ini menggunakan susunan lapisan konvolusi dengan ukuran kernel kecil, yaitu 3×3 , dan memiliki 16 lapisan berbobot. VGG16 dikenal sebagai arsitektur yang sederhana, terstruktur, dan banyak digunakan dalam berbagai tugas deteksi citra [7].

Pada penelitian ini, VGG16 digunakan sebagai salah satu arsitektur CNN yang dibandingkan. Penggunaan VGG16 bertujuan untuk mengetahui performa arsitektur tersebut dalam deteksi penyakit daun tanaman berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, penelitian terkait deteksi penyakit daun tanaman juga menunjukkan bahwa VGG16 dapat digunakan sebagai salah satu model *Deep learning* yang dibandingkan dalam deteksi citra penyakit daun [11].

b. Arsitektur ResNet50

ResNet50 merupakan arsitektur CNN yang menggunakan konsep residual learning. Arsitektur ini dikembangkan untuk membantu proses pelatihan jaringan yang lebih dalam. ResNet menggunakan residual connection atau skip connection, yaitu mekanisme yang memungkinkan informasi dari lapisan sebelumnya diteruskan secara langsung ke lapisan berikutnya [8].

Pada penelitian ini, ResNet50 digunakan sebagai salah satu arsitektur CNN yang dibandingkan. Penggunaan ResNet50 bertujuan untuk mengetahui kemampuan arsitektur jaringan yang lebih dalam dalam melakukan deteksi penyakit daun tanaman berbasis citra digital. Beberapa penelitian deteksi penyakit daun tanaman juga menggunakan ResNet50 sebagai model utama maupun sebagai model pembanding karena kemampuannya dalam mempelajari fitur citra secara mendalam [5], [12], [15].

c. Arsitektur MobileNetV2

MobileNetV2 merupakan arsitektur CNN yang dirancang agar lebih ringan dibandingkan beberapa arsitektur CNN besar. Arsitektur ini menggunakan *inverted residual*, linear bottleneck, dan *depthwise separable convolution*. Konsep tersebut membuat MobileNetV2 dapat mengurangi jumlah operasi komputasi tanpa menghilangkan kemampuan model dalam mempelajari fitur citra [9].

Pada penelitian ini, MobileNetV2 digunakan sebagai salah satu arsitektur CNN yang dibandingkan dengan VGG16 dan ResNet50. Penggunaan MobileNetV2 bertujuan untuk mengetahui performa deteksi model dalam mengenali penyakit daun tanaman berdasarkan citra digital. Penelitian sebelumnya juga menunjukkan bahwa

MobileNetV2 banyak digunakan dalam deteksi penyakit daun tanaman, baik sebagai model tunggal maupun sebagai bagian dari pendekatan ensemble atau *transfer learning* [12], [15].

6. *Transfer learning*

Transfer learning merupakan teknik pemanfaatan model yang telah dilatih sebelumnya pada dataset besar untuk menyelesaikan permasalahan baru. Dalam deteksi citra, model *pre-trained* telah mempelajari fitur dasar seperti garis, tepi, bentuk, tekstur, dan pola visual. Fitur tersebut dapat dimanfaatkan kembali untuk deteksi penyakit daun tanaman dengan menyesuaikan lapisan akhir model sesuai jumlah kelas yang digunakan [5], [6].

Pada penelitian ini, *transfer learning* digunakan pada arsitektur VGG16, ResNet50, dan MobileNetV2. Pendekatan ini dipilih karena dapat membantu proses pelatihan model dan mendukung penerapan CNN pada dataset citra penyakit daun tanaman. Beberapa penelitian sebelumnya juga menggunakan *transfer learning* untuk deteksi penyakit daun tanaman dengan model seperti ResNet50, MobileNetV2, dan EfficientNetB0 [5], [12], [15].

7. Dataset Citra Penyakit Daun Tanaman

Dataset merupakan kumpulan data yang digunakan dalam proses pelatihan, validasi, dan pengujian model. Dalam penelitian deteksi penyakit daun tanaman, dataset berisi citra daun yang telah diberi label sesuai jenis tanaman dan kelas penyakitnya. Label pada dataset digunakan sebagai acuan agar model dapat mempelajari perbedaan antar kelas penyakit [10], [14].

Pada penelitian ini, dataset yang digunakan berupa citra daun tanaman yang terdiri dari beberapa kelas penyakit dan kelas daun sehat. Dataset tersebut perlu melalui proses seleksi dan *preprocessing* agar citra memiliki kualitas dan format yang sesuai untuk proses pelatihan model CNN. Dataset citra penyakit daun tanaman banyak digunakan dalam penelitian *Deep learning* karena memungkinkan model mempelajari pola visual dari data berlabel [4].

Dalam pengumpulan dataset penyakit tanaman berbasis citra, tidak terdapat standar universal yang menetapkan jumlah megapiksel minimum kamera. Literatur lebih menekankan kualitas dan keterwakilan citra, seperti objek daun dan gejala penyakit terlihat jelas, citra berada dalam kondisi fokus, pencahayaan memadai, serta terdapat variasi sudut, latar belakang, orientasi, dan kondisi pengambilan. Penelitian terdahulu menunjukkan bahwa citra penyakit daun yang dikumpulkan pada kondisi lapangan dapat memiliki variasi pencahayaan, sudut pengambilan, permukaan, dan noise agar dataset lebih merepresentasikan kondisi nyata [20], [21].

Citra yang diperoleh menggunakan perangkat bergerak juga dapat dipengaruhi oleh pencahayaan, sudut pengambilan, latar belakang, dan karakteristik kamera. Oleh karena itu, kualitas dataset tidak hanya ditentukan oleh resolusi kamera, tetapi juga oleh prosedur pengambilan citra, ketajaman objek, keterbacaan gejala, dan variasi kondisi lapangan [22].

8. *Preprocessing Data*

Preprocessing data merupakan tahap awal untuk menyiapkan citra sebelum digunakan dalam proses pelatihan model. *Preprocessing* diperlukan agar citra memiliki ukuran, format, dan nilai piksel yang seragam sehingga dapat diproses oleh model *Deep learning*. Tahapan *preprocessing* pada deteksi citra umumnya meliputi *resize citra*, normalisasi nilai piksel, augmentasi data, serta pembagian dataset menjadi data latih, validasi, dan uji [6], [10].

Pada penelitian ini, *preprocessing* dilakukan dengan mengubah ukuran citra menjadi 224×224 piksel, melakukan normalisasi nilai piksel, serta menerapkan augmentasi data seperti rotasi, flip, dan zoom. Tahapan ini bertujuan untuk meningkatkan variasi data dan mengurangi risiko *overfitting* pada proses pelatihan model [15].

9. Pembagian Dataset dan Fungsi Data *Validation*

Dalam pengembangan model *machine learning*, dataset perlu dipisahkan agar proses pembelajaran, pemilihan model, dan evaluasi akhir tidak menggunakan data yang sama. Secara umum, dataset dapat dikelompokkan menjadi data pengembangan model dan data pengujian. Data pengembangan selanjutnya dapat dibagi menjadi data *Training* dan data *validation*. Data *Training* digunakan untuk memperbarui bobot model, sedangkan data *validation* digunakan untuk mengevaluasi performa sementara selama proses pelatihan dan membantu pemilihan konfigurasi atau *checkpoint* model. Data *testing* dipisahkan dari proses pengembangan dan digunakan untuk memberikan estimasi performa akhir pada data yang belum digunakan dalam pelatihan maupun pemilihan model [23], [24], [25].

Penggunaan data *validation* diperlukan agar pemantauan performa model tidak dilakukan menggunakan data *testing*. Apabila data *testing* ikut digunakan untuk menentukan model, epoch, atau parameter yang dipilih, hasil evaluasi akhir dapat mengalami selection bias dan tidak lagi mencerminkan pengujian yang independen [24].

Pada implementasi Keras, data yang diberikan melalui parameter *validation_data* digunakan untuk menghitung nilai *Loss* dan metrik pada akhir setiap epoch, tetapi data tersebut tidak digunakan untuk melatih atau memperbarui bobot model. Dengan demikian, *validation* berperan sebagai bagian dari proses pengembangan model, sedangkan *testing* tetap menjadi

dasar evaluasi akhir. Dokumentasi Keras juga menjelaskan bahwa *validation_data* dievaluasi setiap akhir epoch tanpa digunakan untuk *Training*.

10. Kriteria Visual Penyakit Daun Tanaman

Kriteria visual penyakit daun tanaman merupakan gejala yang tampak pada permukaan daun dan dapat digunakan sebagai dasar pengenalan penyakit. Gejala tersebut dapat berupa perubahan warna, munculnya bercak, perubahan tekstur permukaan, daun mengering, daun berlubang, daun menggulung, atau perubahan bentuk daun. Informasi visual tersebut menjadi dasar dalam proses pengenalan penyakit daun berbasis citra digital [4], [10].

Dalam pendekatan berbasis CNN, kriteria visual tersebut tidak ditentukan secara manual oleh sistem. Model CNN akan mempelajari pola visual dari citra selama proses pelatihan. Namun, penjelasan kriteria visual tetap diperlukan agar ruang lingkup deteksi penyakit daun tanaman dapat dipahami secara jelas.[16]

a. *Healthy*.

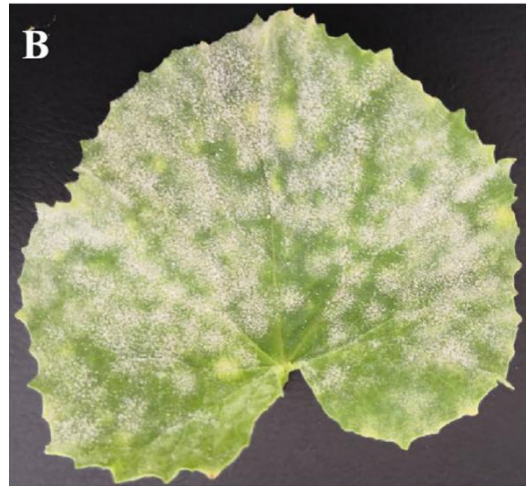
Daun sehat digunakan sebagai kelas kontrol dan secara operasional ditandai oleh warna hijau yang relatif merata, bentuk daun normal, serta tidak menunjukkan lapisan putih seperti tepung, pola mosaik atau bercak klorotik, maupun gejala daun menggulung. Karakteristik sehat daun melon ditunjukkan pada Gambar 2 dibawah ini [26].



Gambar 2 Daun melon sehat berdasarkan literatur

b. *Powdery Mildew*

Powdery Mildew pada melon berkaitan dengan infeksi jamur yang menimbulkan bercak atau lapisan putih seperti tepung pada permukaan daun. Infeksi dapat berkembang pada jaringan daun dan memengaruhi struktur daun, kandungan klorofil, proses fotosintesis, kualitas, serta hasil tanaman melon. Karakteristik sehat daun melon ditunjukkan pada Gambar 3 dibawah ini [27], [28].



Gambar 3 Gejala *Powdery Mildew* pada daun melon berdasarkan literatur

c. *Cucumber Mosaic Virus*

Tanaman melon yang terinfeksi *Cucumber Mosaic Virus* dapat menunjukkan bercak klorotik dan pola mosaik sistemik pada daun. Gejala tersebut dapat disertai perubahan bentuk daun dan gangguan pertumbuhan, sedangkan kepastian penyebab virus memerlukan pengujian serologis atau molekuler. Karakteristik sehat daun melon ditunjukkan pada Gambar 4 dibawah ini [29], [30].



Gambar 4 Gejala *Cucumber Mosaic Virus* pada daun melon

d. *Leaf Curl Disease*

Penyakit daun keriting pada melon dapat menunjukkan gejala awal berupa bercak kuning, kemudian berkembang menjadi penebalan tulang daun, daun menggulung atau keriting, serta pertumbuhan tanaman yang terhambat. Penelitian pada melon di Yogyakarta mengidentifikasi Begomovirus sebagai penyebab penyakit daun keriting, sedangkan penelitian lain melaporkan klorosis tidak merata dan leaf curling pada infeksi Melon chlorotic leaf curl virus. Karakteristik sehat daun melon ditunjukkan pada Gambar 5 dibawah ini [31].



Gambar 5 Tanaman melon tanpa gejala dan dengan gejala Leaf Curl Disease

Tabel 2 Tabel kriteria visual penyakit daun tanaman

No	penyakit	Ciri visual
1	<i>Healthy</i>	Hijau normal
2	<i>Powdery Mildew</i>	Lapisan putih seperti tepung
3	<i>Cucumber Mosaic Virus</i>	Daun menguning dan mosaik
4	<i>Leaf Curl Disease</i>	Daun menggulung

11. *Confusion matrix*

Confusion matrix merupakan tabel evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya. Dalam deteksi, *confusion matrix* digunakan sebagai dasar untuk menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. *Confusion matrix* banyak digunakan dalam evaluasi performa deteksi karena mampu menunjukkan jumlah prediksi benar dan salah pada masing-masing kelas [32].

Fungsi masing-masing metrik:

- a. *Accuracy*
 - 1) Mengukur persentase prediksi yang benar dari seluruh data.
 - 2) Cocok jika jumlah data setiap kelas seimbang.
- b. *Precision*
 - 1) Mengukur ketepatan prediksi suatu kelas.
 - 2) Semakin tinggi *precision*, semakin sedikit prediksi positif yang salah (*False Positive*).
- c. *Recall*
 - 1) Mengukur kemampuan model menemukan seluruh data yang benar pada suatu kelas.
 - 2) Semakin tinggi *recall*, semakin sedikit data positif yang terlewat (*False Negative*).
- d. *F1-score*
 - 1) Merupakan rata-rata harmonik antara *precision* dan *recall*.
 - 2) Sangat berguna jika ingin melihat keseimbangan antara *precision* dan *recall*.

Confusion matrix diperoleh dari hasil pengujian model terhadap data *testing*. Pada tahap pengujian, model menghasilkan prediksi kelas untuk setiap citra daun tanaman. Hasil prediksi tersebut kemudian dibandingkan dengan label sebenarnya pada data *testing*. Dari perbandingan tersebut diperoleh jumlah prediksi yang benar dan salah, yang selanjutnya dikelompokkan ke dalam komponen *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Keterangan komponen *confusion matrix* adalah sebagai berikut:

- a. TP atau *True Positive* adalah jumlah data positif yang berhasil diprediksi benar sebagai positif.
- b. TN atau *True Negative* adalah jumlah data negatif yang berhasil diprediksi benar sebagai negatif.
- c. FP atau *False Positive* adalah jumlah data negatif yang salah diprediksi sebagai positif.
- d. FN atau *False Negative* adalah jumlah data positif yang salah diprediksi sebagai negatif.

12. Evaluasi Performa Model

Evaluasi performa model dilakukan untuk mengetahui kemampuan model dalam melakukan deteksi penyakit daun tanaman. Dalam penelitian ini, evaluasi performa model dibatasi pada metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Keempat metrik tersebut digunakan untuk membandingkan performa model deteksi berbasis CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2 [32].

a. *Accuracy*

Accuracy merupakan metrik evaluasi yang digunakan untuk mengukur persentase jumlah prediksi benar terhadap seluruh data pengujian. Metrik ini memberikan gambaran umum mengenai kemampuan model dalam melakukan deteksi. Rumus *Accuracy* adalah sebagai berikut:

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

Accuracy cocok digunakan ketika jumlah data pada setiap kelas relatif seimbang. Namun, apabila dataset memiliki ketidakseimbangan jumlah data antar kelas, *Accuracy* perlu didukung oleh metrik lain seperti *precision*, *recall*, dan *F1-score*.

b. *Precision*

Precision merupakan metrik yang digunakan untuk mengukur ketepatan model ketika memberikan prediksi pada suatu kelas. *Precision* menunjukkan perbandingan antara jumlah prediksi positif yang benar dengan seluruh data yang diprediksi positif oleh model. Rumus *precision* adalah sebagai berikut:

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

Precision yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan yang rendah dalam memberikan prediksi positif.

c. *Recall*

Recall merupakan metrik yang digunakan untuk mengukur kemampuan model dalam menemukan seluruh data positif yang sebenarnya. *Recall* menunjukkan perbandingan antara jumlah data positif yang berhasil diprediksi benar dengan seluruh data positif yang sebenarnya. Rumus *recall* adalah sebagai berikut:

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

Recall yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar data positif dan memiliki tingkat kegagalan deteksi yang rendah.

d. *F1-score*

F1-score merupakan metrik evaluasi yang menggabungkan *precision* dan *recall* dalam satu nilai. Metrik ini digunakan untuk mengetahui keseimbangan antara ketepatan prediksi dan kemampuan model dalam menemukan data positif. Rumus *F1-score* adalah sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F1-score sangat berguna ketika dataset memiliki distribusi kelas yang tidak seimbang, karena metrik ini mempertimbangkan *precision* dan *recall* secara bersamaan.

BAB III METODOLOGI

A. Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan selama dua bulan, yaitu mulai Maret hingga Juli 2026. Tahapan penelitian dimulai dari *studi literatur*, pengumpulan referensi, penentuan dan pengambilan dataset citra daun melon hidroponik, *preprocessing* data, perancangan model *Convolutional Neural Network* (CNN), pelatihan model, pengujian model, evaluasi performa, analisis hasil penelitian, hingga penyusunan laporan akhir skripsi.

Kegiatan penyusunan penelitian, pengolahan data, implementasi model, pelatihan, pengujian, dan analisis hasil dilakukan di lingkungan Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali Cilacap. Proses komputasi dilakukan menggunakan perangkat komputer atau laptop milik peneliti dengan dukungan perangkat lunak Python, TensorFlow, Keras, Scikit-learn, dan Google Colab.

Pengumpulan dataset dilakukan secara langsung di *Green House Labiq Farm*, Kecamatan Purwojati, Kabupaten Banyumas. Dataset yang diperoleh berupa citra daun melon hidroponik yang dikelompokkan ke dalam empat kelas, yaitu *Healthy*, *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*.

B. Alat dan Bahan

Alat dan bahan dalam penelitian ini digunakan untuk mendukung proses pengumpulan data, *preprocessing* citra, perancangan model, pelatihan model, pengujian model, dan evaluasi performa. Alat yang digunakan berupa perangkat keras dan perangkat lunak, sedangkan bahan yang digunakan berupa dataset citra penyakit daun tanaman melon hidroponik.

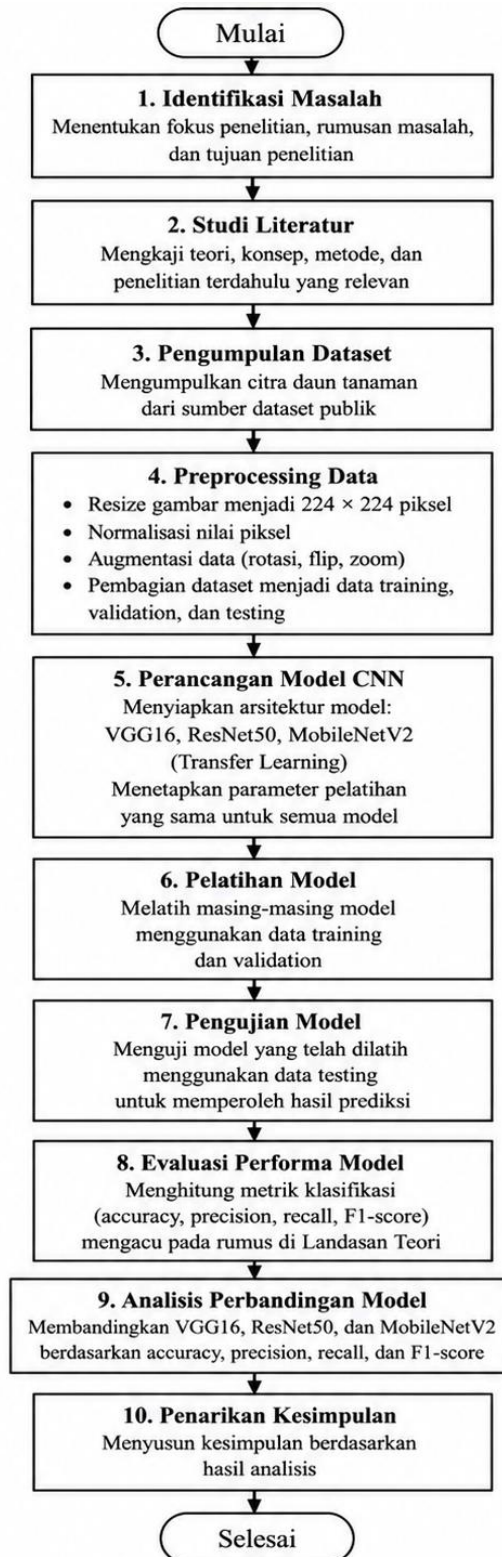
Tabel 3 Alat dan Bahan Penelitian

No	Alat/Bahan	Keterangan
1	Laptop/komputer (Lenovo IdeaPad Flex 5)	Digunakan untuk penyusunan penelitian, pengolahan data, implementasi model, dan analisis hasil penelitian. Spesifikasi perangkat yang digunakan adalah Lenovo IdeaPad Flex 5 (Model 82R9), AMD Ryzen 5 5500U with Radeon Graphics, RAM 16 GB, Windows 11 Home Single Language 64-bit, dan DirectX 12.
2	Google Colab	Digunakan sebagai lingkungan komputasi untuk menjalankan proses pelatihan dan pengujian model CNN.
3	Python	Digunakan sebagai bahasa pemrograman utama dalam implementasi model deteksi.

4	<i>Scikit-learn</i>	Digunakan untuk menghitung metrik evaluasi seperti <i>confusion matrix</i> , <i>accuracy</i> , <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> .
5	Dataset daun melon	Digunakan sebagai bahan penelitian berupa citra daun melon yang terdiri dari 4 kelas: <i>Healthy</i> , <i>Powdery Mildew</i> , <i>Cucumber Mosaic Virus</i> , <i>Leaf Curl Disease</i> .
6	TensorFlow dan Keras	Digunakan untuk membangun, melatih, dan menguji model CNN dengan arsitektur VGG16, ResNet50, dan MobileNetV2.
7	Smartphone (Samsung Galaxy S24 FE)	Digunakan sebagai perangkat pengambilan citra daun melon di lokasi penelitian. Perangkat memiliki kamera utama 50 MP dengan <i>autofocus</i> dan <i>Optical Image Stabilization</i> , serta kamera <i>ultra-wide</i> 12 MP dan <i>telephoto</i> 8 MP. Spesifikasi tersebut memadai untuk menghasilkan citra beresolusi tinggi dengan objek dan gejala visual daun yang dapat diamati dengan jelas.

C. Prosedur Penelitian

Prosedur penelitian merupakan tahapan operasional yang disusun secara sistematis mulai dari tahap awal hingga tahap akhir penelitian. Pada penelitian ini, prosedur penelitian difokuskan pada pembangunan dan perbandingan performa model deteksi penyakit daun tanaman berbasis *Convolutional Neural Network* (CNN) menggunakan tiga arsitektur, yaitu VGG16, ResNet50, dan MobileNetV2. Penelitian ini menggunakan metode eksperimen komparatif dengan pendekatan kuantitatif. Eksperimen dilakukan untuk memperoleh hasil pengukuran performa model secara objektif berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.



Gambar 6 *Flowchart* Penelitian

Flowchart penelitian menunjukkan urutan tahapan penelitian mulai dari identifikasi masalah hingga diperoleh kesimpulan mengenai performa model CNN dalam deteksi penyakit daun tanaman. *Flowchart* tersebut digunakan sebagai pedoman pelaksanaan penelitian agar setiap tahapan, mulai dari studi

literatur, pengumpulan dataset, *preprocessing* data, perancangan model, pelatihan model, pengujian model, evaluasi performa, hingga analisis perbandingan hasil dapat dilakukan secara terstruktur.

Pengujian dilakukan terhadap tiga arsitektur CNN menggunakan skenario yang sama. Setiap model menggunakan dataset, *preprocessing*, pembagian data, parameter pelatihan, dan metrik evaluasi yang sama agar hasil perbandingan dapat dilakukan secara objektif

1. Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk menentukan fokus utama penelitian. Permasalahan yang dikaji adalah perlunya mengetahui performa model deteksi penyakit daun tanaman berbasis CNN dengan menggunakan arsitektur VGG16, ResNet50, dan MobileNetV2.

Hasil dari tahap ini menjadi dasar dalam penyusunan rumusan masalah, tujuan penelitian, batasan masalah, dan rancangan eksperimen. Pada tahap ini juga ditentukan bahwa penelitian tidak merancang arsitektur CNN baru, tetapi membangun model deteksi menggunakan arsitektur CNN yang sudah tersedia

2. Studi Literatur

Tahap studi literatur dilakukan untuk memperoleh dasar teori, konsep, dan referensi yang mendukung pelaksanaan penelitian. Literatur yang dikaji berkaitan dengan *Artificial Intelligence*, *Machine learning*, *Deep learning*, *Convolutional Neural Network*, deteksi penyakit daun tanaman, *transfer learning*, arsitektur VGG16, ResNet50, MobileNetV2, metode deteksi selain CNN, *confusion matrix*, serta metrik evaluasi *accuracy*, *precision*, *recall*, dan *F1-score*.

Studi literatur juga dilakukan terhadap penelitian terdahulu yang relevan, baik yang membahas deteksi penyakit daun tanaman menggunakan CNN maupun metode *machine learning* konvensional seperti *Support Vector Machine*, *K-Nearest Neighbor*, *Decision Tree*, *Random Forest*, dan *Naive Bayes*. Hasil dari tahap ini digunakan sebagai dasar dalam menyusun landasan teori, memperkuat *research gap*, serta menentukan metode dan skenario pengujian yang digunakan dalam penelitian.

3. Pengumpulan Dataset

Tahap pengumpulan dataset dilakukan dengan memperoleh citra daun tanaman melon hidroponik dari sumber dataset pribadi yang saya dapatkan dari data lapangan di Green House Labiq Farm Kecamatan Purwojati Kabupaten Banyumas. Dataset yang digunakan berupa citra daun tanaman yang telah memiliki label kelas penyakit dan kelas daun sehat.

Dataset yang dikumpulkan kemudian diperiksa untuk memastikan kesesuaian label, kualitas citra, dan keterbacaan gambar. Citra yang tidak sesuai, rusak, buram, atau tidak relevan dapat dieliminasi agar tidak

memengaruhi proses pelatihan dan pengujian model. Hasil dari tahap ini adalah dataset citra daun tanaman yang siap diproses pada tahap *preprocessing*.

4. *Preprocessing* Data

Tahap *preprocessing* dilakukan untuk menyeragamkan data citra sebelum digunakan dalam proses pelatihan model. Tahapan ini bertujuan agar seluruh citra memiliki format dan skala nilai piksel yang sesuai dengan kebutuhan input model CNN.

Tahapan *preprocessing* yang dilakukan meliputi:

- a. *Resize* gambar menjadi 224×224 piksel.
- b. Normalisasi nilai piksel.
- c. Augmentasi data menggunakan rotasi, flip, dan zoom.
- d. Pembagian dataset menjadi data *Training*, *validation*, dan *testing*.

Normalisasi dilakukan agar nilai piksel citra sesuai dengan kebutuhan masukan model *Deep learning*. Augmentasi data diterapkan untuk menambah variasi data latih dan mengurangi risiko *overfitting*. Selanjutnya, pembagian dataset dilakukan agar proses pengembangan model dan pengujian akhir menggunakan data yang terpisah.

Dataset pada penelitian ini dibagi menjadi dua kelompok utama, yaitu data pengembangan model sebesar 85% dan data pengujian akhir sebesar 15%. Data pengembangan model terdiri atas data *Training* sebesar 70% dan data *validation* sebesar 15% dari keseluruhan dataset. Dengan demikian, secara operasional terdapat tiga subset data, yaitu *Training*, *validation*, dan *testing*, tetapi data *validation* ditempatkan sebagai bagian dari proses pengembangan model, bukan sebagai data pengujian akhir.

Data *Training* digunakan dalam proses optimasi untuk memperbarui bobot model. Data *validation* tidak digunakan untuk memperbarui bobot, tetapi digunakan untuk menghitung *validation accuracy* dan *validation Loss* pada akhir setiap epoch, memantau kecenderungan *overfitting*, serta membantu menentukan *checkpoint* model apabila *callback* berbasis *validation* diterapkan. Sementara itu, data *testing* dipisahkan sejak awal dan hanya digunakan setelah proses pelatihan dan pemilihan model selesai untuk menghitung performa akhir.

Pembagian dataset menggunakan rasio 70:15:15 dan menghasilkan 305 citra sebagai data pengembangan model, yang terdiri atas 251 citra *Training* dan 54 citra *validation*, serta 56 citra sebagai data *testing*. Pemisahan data *testing* dari proses pengembangan diperlukan agar nilai *accuracy*, *precision*, *recall*, *F1-score*, *classification report*, dan *confusion matrix* mencerminkan performa model pada data yang tidak digunakan dalam proses pelatihan maupun pemilihan model.

5. Perancangan Model CNN

Tahap perancangan model dilakukan dengan menyiapkan tiga arsitektur CNN yang akan dibandingkan, yaitu VGG16, ResNet50, dan MobileNetV2. Ketiga model dirancang menggunakan pendekatan *transfer learning* dengan menyesuaikan lapisan akhir sesuai jumlah kelas pada dataset.

Agar hasil perbandingan bersifat objektif, setiap model menggunakan skenario pelatihan yang sama, meliputi ukuran input citra, pembagian dataset, *batch size*, jumlah *epoch*, *optimizer*, *learning rate*, dan metrik evaluasi. Parameter pelatihan yang digunakan dalam penelitian adalah sebagai berikut:

Tabel 4 Parameter Pelatihan Model

No	Parameter	Nilai
1	<i>Input citra</i>	224 × 224 piksel
2	<i>Batch size</i>	16
3	<i>Epoch</i>	20 epoch
4	<i>Optimizer</i>	Adam
5	<i>Learning rate</i>	0.0001
6	<i>Loss function</i>	<i>Sparse Categorical Crossentropy</i>
7	<i>Pre-trained weights</i>	<i>ImageNet</i>
8	<i>Metrik Training</i>	<i>Accuracy</i>

Ketiga model menggunakan parameter pelatihan yang sama agar proses perbandingan performa deteksi dapat dilakukan secara adil. Dengan demikian, perbedaan hasil evaluasi yang diperoleh lebih dipengaruhi oleh karakteristik masing-masing arsitektur model, bukan oleh perbedaan skenario pelatihan.

6. Pelatihan Model

Tahap pelatihan model dilakukan menggunakan data *Training* dan data *validation*. Data *Training* digunakan untuk melatih model dalam mengenali pola visual pada citra daun, sedangkan data *validation* digunakan untuk memantau performa model selama proses pelatihan.

Proses pelatihan dilakukan pada masing-masing arsitektur, yaitu VGG16, ResNet50, dan MobileNetV2. Ketiga model dilatih menggunakan dataset, *preprocessing*, dan parameter pelatihan yang sama. Hasil dari tahap ini adalah model CNN yang telah dilatih dan siap diuji menggunakan data *testing*.

7. Pengujian Model

Tahap pengujian dilakukan setelah proses pelatihan dan pemilihan model selesai. Pengujian menggunakan data *testing* yang telah dipisahkan sejak awal dan tidak digunakan dalam proses pelatihan maupun validasi. Data *testing* terdiri atas 56 citra daun melon, yaitu 13 citra *Cucumber*

Mosaic Virus, 18 citra *Healthy*, 7 citra *Leaf Curl Disease*, dan 18 citra *Powdery Mildew*. Tujuan pengujian adalah mengukur kemampuan setiap model dalam mendeteksi kondisi daun melon ke dalam kelas yang sesuai.

Model MobileNetV2, VGG16, dan ResNet50 diuji menggunakan data *testing* dan skenario pengujian yang sama agar hasil perbandingan dapat dilakukan secara objektif. Setiap model menghasilkan nilai probabilitas untuk empat kelas, kemudian kelas dengan nilai probabilitas tertinggi ditetapkan sebagai hasil prediksi. Hasil prediksi tersebut dibandingkan dengan label sebenarnya untuk mengetahui jumlah prediksi yang benar dan salah pada setiap kelas.

Hasil pengujian selanjutnya digunakan untuk menghitung *accuracy*, *precision*, *recall*, dan *F1-score*, serta menghasilkan *classification report* dan *confusion matrix*. Nilai *precision*, *recall*, dan *F1-score* keseluruhan dihitung menggunakan metode *weighted average* untuk mempertimbangkan perbedaan jumlah citra pada setiap kelas. Ketiga arsitektur diuji menggunakan data *testing* yang sama sehingga perbedaan hasil lebih mencerminkan karakteristik masing-masing model.

Pengujian setiap arsitektur dilakukan satu kali menggunakan satu pembagian dataset dan konfigurasi parameter yang sama. Oleh karena itu, hasil penelitian diinterpretasikan berdasarkan dataset dan skenario eksperimen yang digunakan dalam penelitian ini. Skenario untuk pengujian sebagai berikut:

- a. Memuat model hasil pelatihan.
- b. Memasukkan data testing.
- c. Menghasilkan probabilitas kelas.
- d. Menentukan kelas prediksi.
- e. Membandingkan prediksi dengan label sebenarnya.
- f. Menghitung metrik evaluasi.

8. Evaluasi Performa Deteksi

Tahap evaluasi performa model dilakukan untuk mengukur kemampuan model dalam melakukan deteksi penyakit daun tanaman. Evaluasi dilakukan menggunakan *confusion matrix* dan metrik deteksi berupa *accuracy*, *precision*, *recall*, dan *F1-score*.

Accuracy digunakan untuk mengetahui persentase prediksi benar terhadap seluruh data uji. *Precision* digunakan untuk mengetahui ketepatan model dalam memberikan prediksi pada suatu kelas. *Recall* digunakan untuk mengetahui kemampuan model dalam mengenali data pada kelas yang sebenarnya. *F1-score* digunakan untuk mengetahui keseimbangan antara *precision* dan *recall*.

Perhitungan metrik evaluasi mengacu pada rumus yang telah dijelaskan pada bagian landasan teori. Hasil evaluasi digunakan untuk membandingkan performa VGG16, ResNet50, dan MobileNetV2

9. Analisis Perbandingan Model

Tahap analisis perbandingan model dilakukan dengan membandingkan hasil evaluasi dari tiga arsitektur CNN, yaitu VGG16, ResNet50, dan MobileNetV2. Perbandingan dilakukan berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

Hasil analisis disajikan dalam bentuk tabel agar perbedaan performa setiap model dapat terlihat secara jelas. Model dengan nilai evaluasi terbaik akan dianalisis sebagai model yang memiliki performa paling baik dalam deteksi penyakit daun tanaman berdasarkan metrik yang digunakan.

Tabel 5 Rancangan Hasil Perbandingan Model

No	Model	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
1	VGG16	-	-	-	-
2	ResNet50	-	-	-	-
3	MobileNetV2	-	-	-	-

Tabel tersebut digunakan sebagai rancangan penyajian hasil pengujian. Nilai pada tabel akan diisi berdasarkan hasil eksperimen setelah proses pelatihan, pengujian, dan evaluasi model selesai dilakukan.

10. Penarikan Kesimpulan

Tahap akhir penelitian adalah penarikan kesimpulan berdasarkan hasil pengujian, evaluasi, dan analisis perbandingan model. Kesimpulan disusun untuk menjawab rumusan masalah penelitian.

Kesimpulan pertama berkaitan dengan skenario pembangunan model deteksi penyakit daun tanaman berbasis CNN menggunakan arsitektur VGG16, ResNet50, dan MobileNetV2. Kesimpulan kedua berkaitan dengan performa ketiga arsitektur CNN berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil akhir penelitian diharapkan dapat memberikan gambaran mengenai arsitektur CNN yang memiliki performa terbaik dalam deteksi penyakit daun tanaman berbasis citra digital.

D. Analisis Data

Analisis data dilakukan setelah proses pelatihan dan pengujian model selesai. Data yang dianalisis dalam penelitian ini berupa hasil prediksi dari tiga model *Convolutional Neural Network* (CNN), yaitu VGG16, ResNet50, dan MobileNetV2 terhadap data *testing*. Hasil prediksi tersebut dibandingkan dengan label sebenarnya untuk memperoleh *confusion matrix* dan nilai metrik evaluasi.

Metrik evaluasi yang digunakan dalam penelitian ini meliputi *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* digunakan untuk mengetahui tingkat ketepatan model secara keseluruhan. *Precision* digunakan untuk mengetahui ketepatan model dalam menghasilkan prediksi pada suatu kelas. *Recall* digunakan untuk mengetahui kemampuan model dalam mengenali data pada kelas yang sebenarnya. *F1-score* digunakan untuk mengetahui keseimbangan antara *precision* dan *recall*.

Hasil evaluasi dari masing-masing model kemudian dibandingkan untuk mengetahui perbedaan performa antara VGG16, ResNet50, dan MobileNetV2. Perbandingan dilakukan berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score* yang diperoleh dari hasil pengujian. Model dengan nilai evaluasi terbaik dianalisis sebagai model yang memiliki performa paling baik dalam deteksi penyakit daun tanaman melon berdasarkan metrik yang digunakan.

Hasil analisis disajikan dalam bentuk tabel perbandingan agar perbedaan performa setiap model dapat terlihat secara jelas. Selain itu, hasil pelatihan seperti nilai *Accuracy* dan *Loss* pada data *Training* dan *validation* dapat digunakan sebagai informasi pendukung untuk melihat kestabilan proses pelatihan model.

E. Jadwal Penelitian

Jadwal penelitian disusun sebagai pedoman pelaksanaan penelitian agar setiap tahapan dapat berjalan secara sistematis dan terarah. Penelitian ini direncanakan berlangsung selama 3 bulan, mulai dari tahap persiapan hingga tahap akhir penyusunan laporan dan sidang skripsi. Bentuk jadwal penelitian yang digunakan pada penelitian ini adalah tabel jadwal penelitian.

Tabel 6 Jadwal Penelitian

No	Nama Kegiatan	Bulan ke-					
		1	2	3	4	5	6
1	Tahap Persiapan						
	a. Studi Literatur						
	b. Rumusan masalah						
	c. Penetapan Metode						
	d. Seminar Proposal						
2	Tahap Penelitian						
	a. Pengumpulan Dataset						
	b. <i>Preprocessing</i> Data						
	c. Pelatihan Model CNN						
	d. Pengujian dan Evaluasi Model						
	e. Analisis Perbandingan Hasil						
	f. Penulisan Skripsi						
3	Tahap Akhir						
	a. Sidang Skripsi						
	b. Publikasi ilmiah						

BAB IV HASIL DAN PEMBAHASAN

A. Hasil

1. Hasil Pengumpulan Dataset

Proses pengumpulan dataset dilakukan dengan mengambil citra daun melon hidroponik yang telah dikelompokkan ke dalam empat folder kelas pada direktori penyimpanan Google Drive. Berdasarkan hasil pemeriksaan direktori dataset, diperoleh sebanyak 361 citra daun melon yang terbagi ke dalam empat kelas, yaitu *Cucumber Mosaic Virus*, *Healthy*, *Leaf Curl Disease*, dan *Powdery Mildew*. Rincian jumlah citra pada masing-masing kelas sebagai berikut:

Tabel 7 Jumlah Citra Dataset Setiap Kelas

Kelas	Jumlah
<i>Healthy</i>	118
<i>Powdery Mildew</i>	116
<i>Cucumber Mosaic Virus</i>	83
<i>Leaf Curl Disease</i>	44
Total	361

Sumber: Hasil pengumpulan dataset, 2026.

distribusi jumlah citra antar kelas tidak seimbang atau *imbalanced*. Kelas *Healthy* memiliki jumlah citra terbanyak, yaitu 118 citra, sedangkan kelas *Leaf Curl Disease* memiliki jumlah citra paling sedikit, yaitu 44 citra. Kondisi tersebut menjadi pertimbangan pada tahap pelatihan model sehingga diterapkan strategi pembobotan kelas atau *class weighting* untuk mengurangi kecenderungan model terhadap kelas dengan jumlah citra lebih banyak.

Selain berdasarkan jumlah dan distribusi kelas, hasil pengumpulan dataset juga ditinjau dari karakteristik pengambilan citra. Pengambilan citra dilakukan menggunakan smartphone Samsung Galaxy S24 FE dengan menempatkan daun sebagai objek utama. Jarak kamera disesuaikan dengan ukuran daun dan posisi tanaman agar sebagian besar permukaan daun berada di dalam bidang citra serta karakteristik visualnya dapat diamati dengan jelas. Karakteristik pengambilan citra dataset disajikan sebagai berikut:

Tabel 8 Karakteristik Pengambilan Citra Dataset Daun Melon

No	Aspek	Hasil penerapan dataset
1	Objek citra	Daun melon menjadi objek utama dan mengisi sekitar 75–90% bidang citra.
2	Jarak pemotretan	Kamera diposisikan sekitar 20–30 cm dari daun, disesuaikan dengan ukuran daun agar pangkal, tepi, tulang utama, tekstur, dan bagian yang menunjukkan gejala berada dalam bidang citra.
3	Sudut pemotretan	Kamera diarahkan mendekati tegak lurus terhadap permukaan daun untuk mengurangi perubahan bentuk akibat sudut pengambilan.
4	Fokus dan detail	Fokus diarahkan pada permukaan daun sehingga tulang daun, tekstur, perubahan warna, bercak, lapisan putih, pola mosaik, atau gejala keriting dapat diamati.
5	Seleksi kualitas	Citra yang terpotong berlebihan, buram, mengalami pantulan cahaya, terlalu gelap, atau tidak memperlihatkan bagian gejala tidak digunakan.

Citra yang digunakan dalam penelitian memiliki variasi ukuran daun karena pengambilan dataset dilakukan langsung pada tanaman melon dalam kondisi lapangan. Variasi tersebut mencakup daun yang berukuran relatif kecil hingga daun yang berukuran lebih besar. Ukuran daun tidak ditentukan menggunakan satuan sentimeter, tetapi seluruh citra dipilih dari daun yang telah berkembang dan memiliki permukaan yang dapat diamati.

- a. Pada kelas *Healthy*, citra dipilih dari daun yang telah berkembang, berwarna hijau relatif merata, berbentuk normal, dan tidak menunjukkan gejala penyakit. Daun yang telah mengering serta daun muda atau daun pucuk yang belum berkembang secara penuh tidak digunakan karena perubahan warna, ukuran, bentuk, dan teksturnya.
- b. Pada kelas *Powdery Mildew*, citra memperlihatkan daun dengan bercak atau lapisan putih menyerupai tepung pada bagian permukaan daun. Luas lapisan putih pada setiap citra dapat berbeda, tetapi citra

dipilih ketika karakteristik tersebut masih dapat diamati pada bagian daun yang berada dalam bidang gambar.

- c. Pada kelas *Cucumber Mosaic Virus*, citra menunjukkan perubahan warna pada permukaan daun berupa pola mosaik, bercak klorotik, atau warna hijau dan kuning yang tidak merata. Pada beberapa citra, perubahan tersebut juga disertai perubahan bentuk permukaan daun. Citra digunakan apabila pola visual yang menjadi dasar pengelompokan kelas dapat diamati.
- d. Pada kelas *Leaf Curl Disease*, citra memperlihatkan perubahan bentuk daun berupa bagian daun yang menggulung, mengeriting, atau mengalami perubahan bentuk dibandingkan daun normal. Beberapa citra juga menunjukkan perubahan warna dan bentuk tulang daun. Pemilihan citra dilakukan dengan memastikan bahwa bagian daun yang mengalami perubahan bentuk berada dalam bidang citra.

Daun pucuk yang masih sangat muda tidak digunakan dalam dataset karena bentuk, ukuran, warna, dan permukaan daunnya belum berkembang secara penuh. Kondisi daun pucuk tersebut dapat menghasilkan variasi visual yang berasal dari tahap pertumbuhan, bukan dari kondisi penyakit. Oleh karena itu, objek penelitian dibatasi pada daun yang telah berkembang agar karakteristik visual antar kelas dapat dibandingkan secara lebih konsisten. Contoh visual daun melon sebagai berikut:



Gambar 7 Daun *Healthy*



Gambar 8 Daun *Leaf Curl Disease*



Gambar 9 Daun *Powdery Mildew*



Gambar 10 Daun *Cucumber Mosaic Virus*

2. Hasil *Preprocessing*

Pada tahap *preprocessing*, seluruh citra daun melon dipersiapkan sebelum digunakan dalam proses pelatihan model CNN. Tahapan *preprocessing* bertujuan agar seluruh citra memiliki ukuran dan format yang seragam sehingga dapat diproses oleh model secara optimal. Tahapan *preprocessing* yang dilakukan meliputi:

a. Pembagian Dataset (*Data Splitting*)

Dataset yang telah terkumpul dibagi menjadi dua kelompok utama, yaitu data pengembangan model sebesar 85% dan data pengujian akhir sebesar 15%. Data pengembangan model selanjutnya terdiri atas data latih (*Training*) sebesar 70% dan data validasi (*validation*) sebesar 15% dari keseluruhan dataset. Dengan demikian, secara operasional terdapat tiga subset data, yaitu *Training*, *validation*, dan *testing*.

Pembagian dataset dilakukan secara acak menggunakan teknik *random shuffle* dengan *seed* tetap agar hasil pembagian dapat direproduksi. Hasil pembagian dataset ditampilkan pada Tabel 9 di bawah ini.

Tabel 9 Hasil pembagian dataset

Kelas	Train	Validation	Test	Total
<i>Cucumber Mosaic Virus</i>	58	12	13	83
<i>Healthy</i>	82	18	18	118
<i>Leaf Curl Disease</i>	30	7	7	44
<i>Powdery Mildew</i>	81	17	18	116
Total	251	54	56	361

Berdasarkan hasil pembagian tersebut, diperoleh 305 citra sebagai data pengembangan model, yang terdiri atas 251 citra *Training* dan 54 citra *validation*, serta 56 citra sebagai data *testing*. Seluruh subset data kemudian dimuat menggunakan fungsi `image_dataset_from_directory` dengan ukuran *batch* sebesar 16 citra.

Data *Training* digunakan untuk memperbarui bobot model selama proses optimasi. Data *validation* digunakan untuk memantau nilai *validation accuracy* dan *validation Loss* pada setiap epoch serta membantu mengevaluasi perkembangan model selama pelatihan. Sementara itu, data *testing* dipisahkan sejak awal dan hanya digunakan setelah proses pelatihan selesai untuk mengukur performa akhir model melalui *accuracy*, *precision*, *recall*, *F1-score*, *classification report*, dan *confusion matrix*.

b. *Resize citra*

Seluruh citra diubah ukurannya menjadi 224×224 piksel agar sesuai dengan ukuran input yang digunakan oleh arsitektur VGG16, ResNet50, dan MobileNetV2.



Gambar 11 Sebelum *resize citra*



Gambar 12 Setelah *resize citra*

seluruh citra daun melon telah diubah ukurannya menjadi 224×224 piksel. Proses *resize* dilakukan untuk menyeragamkan dimensi seluruh citra sehingga sesuai dengan ukuran masukan (input) yang digunakan oleh model VGG16, ResNet50, dan MobileNetV2. Dengan ukuran citra yang seragam, proses pelatihan model dapat dilakukan secara lebih optimal dan konsisten.

c. Labeling Dataset

Setelah proses *resize* selesai, setiap citra diberi label berdasarkan kelas penyakitnya. Proses pelabelan dilakukan secara otomatis menggunakan fungsi `label_mode="int"` berdasarkan nama folder dataset sehingga setiap citra memiliki label kelas yang sesuai.

d. Normalisasi

dilakukan menggunakan fungsi *preprocessing* bawaan masing-masing arsitektur CNN. Nilai piksel citra dinormalisasi menggunakan fungsi `preprocess_input()` sehingga distribusi data sesuai dengan kebutuhan model *pretrained ImageNet*.

e. Data Augmentation

Untuk mengurangi risiko *overfitting* akibat jumlah data yang relatif terbatas, dilakukan augmentasi data secara *on-the-fly* menggunakan lapisan *RandomFlip (horizontal)*, *RandomRotation(0,2)*, dan *RandomZoom(0,2)*. Augmentasi diterapkan hanya pada data latih dan dijalankan sebagai bagian dari arsitektur model sehingga tidak menambah ukuran dataset secara fisik, namun memberikan variasi citra pada setiap epoch pelatihan.



Gambar 13 Hasil augmentasi

f. Pembobotan Kelas (Class Weighting)

Mengingat distribusi data yang tidak seimbang, dihitung bobot kelas menggunakan metode *balanced* dari *scikit-learn*. Hasil perhitungan menghasilkan bobot sebesar 1,0819 untuk *Cucumber Mosaic Virus*, 0,7652 untuk *Healthy*, 2,0917 untuk *Leaf Curl Disease*, dan 0,7747 untuk *Powdery Mildew*. Bobot yang lebih besar diberikan pada kelas dengan jumlah data lebih sedikit (*Leaf Curl Disease*) agar kontribusi kelas tersebut terhadap fungsi *Loss* menjadi lebih signifikan selama pelatihan.

3. Hasil Pelatihan

a. Model MobileNetV2

Model MobileNetV2 dirancang menggunakan pendekatan *transfer learning* dengan bobot awal dari *ImageNet*. Pada penelitian ini digunakan parameter `include_top=False` sehingga model hanya dimanfaatkan sebagai *feature extractor*.

Berbeda dengan VGG16 dan ResNet50, MobileNetV2 menggunakan arsitektur *Depthwise separable convolution* dan *Inverted residual Block* yang membuat jumlah parameter menjadi lebih sedikit sehingga proses komputasi lebih cepat dan ringan.

Setelah proses ekstraksi fitur selesai, keluaran model diproses menggunakan *Global Average Pooling 2D*, kemudian diteruskan ke lapisan *Dense* sebanyak 256 *neuron* dengan fungsi aktivasi ReLU. Untuk mengurangi *overfitting*, ditambahkan *Dropout* sebesar 0,3. Lapisan keluaran menggunakan fungsi aktivasi *Softmax* dengan empat *neuron* yang menghasilkan prediksi kelas penyakit daun melon hidroponik.

b. Model VGG16

Model VGG16 dirancang menggunakan pendekatan *transfer learning* dengan memanfaatkan bobot awal (*pre-trained weights*) dari *ImageNet*. Pada penelitian ini, lapisan deteksi bawaan VGG16

dihilangkan dengan mengatur parameter `include_top=False`, sehingga model hanya digunakan sebagai *feature extractor* untuk mengekstraksi ciri visual dari citra daun melon.

Selanjutnya, keluaran dari model VGG16 dihubungkan dengan lapisan *Global Average Pooling 2D* untuk mereduksi dimensi fitur tanpa menghilangkan informasi penting. Setelah itu ditambahkan lapisan *Dense* sebanyak 256 neuron dengan fungsi aktivasi ReLU sebagai proses deteksi lanjutan. Untuk mengurangi risiko *overfitting*, digunakan lapisan *Dropout* sebesar 0,3. Pada lapisan keluaran digunakan fungsi aktivasi *Softmax* dengan empat neuron yang merepresentasikan empat kelas penyakit daun melon, yaitu *Healthy*, *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*.

c. Model ResNet50

Model ResNet50 juga dibangun menggunakan teknik *transfer learning* dengan bobot awal dari *ImageNet*. Sama seperti VGG16, lapisan deteksi bawaan dihilangkan (`include_top=False`) sehingga model hanya berfungsi sebagai *feature extractor*.

Keunggulan utama ResNet50 adalah penggunaan *Residual Block* (*Skip Connection*) yang memungkinkan informasi dari lapisan sebelumnya diteruskan ke lapisan berikutnya. Mekanisme ini membantu mengurangi masalah *vanishing gradient* sehingga jaringan yang lebih dalam tetap dapat dilatih secara efektif.

Output dari ResNet50 kemudian diproses menggunakan *Global Average Pooling 2D*, dilanjutkan dengan lapisan *Dense* sebanyak 256 neuron menggunakan fungsi aktivasi ReLU, kemudian Dropout sebesar 0,3. Lapisan terakhir menggunakan fungsi aktivasi *Softmax* dengan empat neuron sebagai keluaran deteksi penyakit daun melon.

4. Hasil Pengujian Model

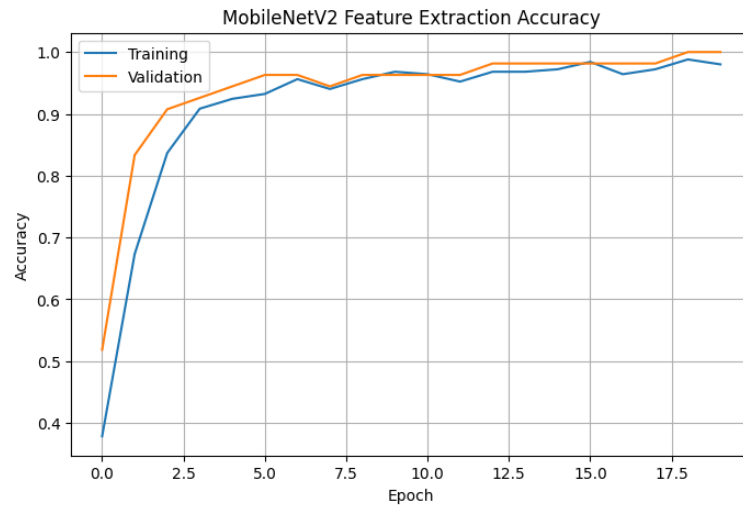
a. Hasil pengujian MobileNetV2

Model MobileNetV2 juga dibangun menggunakan metode *transfer learning* dengan bobot awal *ImageNet*. Pada tahap *feature extraction*, proses pelatihan dilakukan selama 20 epoch. Berdasarkan grafik *accuracy*, model mengalami peningkatan performa yang cukup konsisten. Akurasi validasi meningkat dari 0,5185 pada epoch pertama menjadi 100% pada epoch ke-19 hingga epoch ke-20.

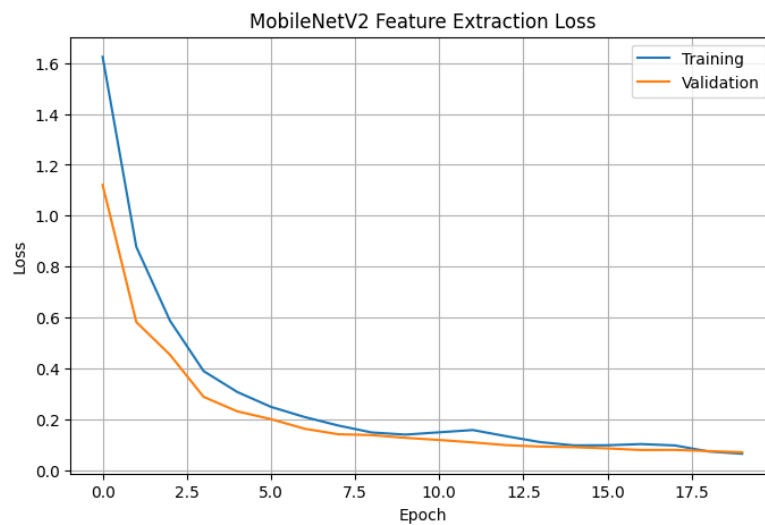
Seiring meningkatnya *accuracy*, nilai *validation Loss* terus mengalami penurunan dari 1,1209 pada epoch pertama menjadi 0,0699 pada epoch terakhir. Hal ini menunjukkan bahwa model semakin baik dalam mengenali pola penyakit daun melon dan menghasilkan prediksi dengan tingkat keyakinan yang lebih tinggi.

Berdasarkan hasil pelatihan selama 20 epoch, MobileNetV2 mampu mencapai akurasi validasi yang tinggi dengan nilai *validation*

Loss yang terus menurun. Hasil tersebut menunjukkan bahwa MobileNetV2 mampu mempelajari karakteristik penyakit daun melon secara efektif menggunakan pendekatan *transfer learning* berbasis *feature extraction*. Selain itu, jumlah parameter yang lebih sedikit membuat MobileNetV2 memiliki waktu pelatihan yang relatif lebih singkat dibandingkan model CNN yang lebih besar sehingga menjadi salah satu model yang efisien untuk deteksi penyakit daun melon.



Gambar 14 Grafik *Accuracy* MobileNetV2



Gambar 15 Grafik *Loss* MobileNetV2

```

Epoch 1/20
16/16 ----- 0s 1s/step - accuracy: 0.3679 - loss: 2.0099
Epoch 1: val_accuracy improved from None to 0.51852, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 1: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 57s 3s/step - accuracy: 0.3785 - loss: 1.6248 - val_accuracy: 0.5185 - val_loss: 1.1209
Epoch 2/20
16/16 ----- 0s 2s/step - accuracy: 0.6263 - loss: 0.9661
Epoch 2: val_accuracy improved from 0.51852 to 0.83333, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 2: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 30s 2s/step - accuracy: 0.6733 - loss: 0.8777 - val_accuracy: 0.8333 - val_loss: 0.5820
Epoch 3/20
16/16 ----- 0s 2s/step - accuracy: 0.8568 - loss: 0.6663
Epoch 3: val_accuracy improved from 0.83333 to 0.90741, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 3: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 31s 2s/step - accuracy: 0.8367 - loss: 0.5874 - val_accuracy: 0.9074 - val_loss: 0.4533
Epoch 4/20
16/16 ----- 0s 1s/step - accuracy: 0.8979 - loss: 0.4282
Epoch 4: val_accuracy improved from 0.90741 to 0.92593, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 4: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 32s 2s/step - accuracy: 0.9084 - loss: 0.3890 - val_accuracy: 0.9259 - val_loss: 0.2876
Epoch 5/20
16/16 ----- 0s 2s/step - accuracy: 0.9108 - loss: 0.3304
Epoch 5: val_accuracy improved from 0.92593 to 0.94444, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 5: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 41s 2s/step - accuracy: 0.9243 - loss: 0.3068 - val_accuracy: 0.9444 - val_loss: 0.2308
Epoch 6/20
16/16 ----- 0s 1s/step - accuracy: 0.9486 - loss: 0.2501
Epoch 6: val_accuracy improved from 0.94444 to 0.96296, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 6: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 31s 2s/step - accuracy: 0.9323 - loss: 0.2483 - val_accuracy: 0.9630 - val_loss: 0.2000
Epoch 7/20
16/16 ----- 0s 1s/step - accuracy: 0.9667 - loss: 0.1823
Epoch 7: val_accuracy did not improve from 0.96296
16/16 ----- 31s 2s/step - accuracy: 0.9562 - loss: 0.2084 - val_accuracy: 0.9630 - val_loss: 0.1624
Epoch 8/20
16/16 ----- 0s 1s/step - accuracy: 0.9230 - loss: 0.1867
Epoch 8: val_accuracy did not improve from 0.96296
16/16 ----- 34s 2s/step - accuracy: 0.9402 - loss: 0.1750 - val_accuracy: 0.9444 - val_loss: 0.1407
Epoch 9/20
16/16 ----- 0s 2s/step - accuracy: 0.9662 - loss: 0.1506
Epoch 9: val_accuracy did not improve from 0.96296
16/16 ----- 30s 2s/step - accuracy: 0.9562 - loss: 0.1475 - val_accuracy: 0.9630 - val_loss: 0.1373
Epoch 10/20
16/16 ----- 0s 2s/step - accuracy: 0.9741 - loss: 0.1568
Epoch 10: val_accuracy did not improve from 0.96296
Epoch 9: val_accuracy did not improve from 0.96296
16/16 ----- 30s 2s/step - accuracy: 0.9562 - loss: 0.1475 - val_accuracy: 0.9630 - val_loss: 0.1373
Epoch 10/20
16/16 ----- 0s 2s/step - accuracy: 0.9741 - loss: 0.1568
Epoch 10: val_accuracy did not improve from 0.96296
16/16 ----- 31s 2s/step - accuracy: 0.9681 - loss: 0.1393 - val_accuracy: 0.9630 - val_loss: 0.1265
Epoch 11/20
16/16 ----- 0s 2s/step - accuracy: 0.9782 - loss: 0.1291
Epoch 11: val_accuracy did not improve from 0.96296
16/16 ----- 41s 2s/step - accuracy: 0.9641 - loss: 0.1483 - val_accuracy: 0.9630 - val_loss: 0.1182
Epoch 12/20
16/16 ----- 0s 2s/step - accuracy: 0.9601 - loss: 0.1714
Epoch 12: val_accuracy did not improve from 0.96296
16/16 ----- 40s 2s/step - accuracy: 0.9522 - loss: 0.1572 - val_accuracy: 0.9630 - val_loss: 0.1085
Epoch 13/20
16/16 ----- 0s 1s/step - accuracy: 0.9684 - loss: 0.1165
Epoch 13: val_accuracy improved from 0.96296 to 0.98148, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 13: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 45s 2s/step - accuracy: 0.9681 - loss: 0.1327 - val_accuracy: 0.9815 - val_loss: 0.0977
Epoch 14/20
16/16 ----- 0s 2s/step - accuracy: 0.9731 - loss: 0.1057
Epoch 14: val_accuracy did not improve from 0.98148
16/16 ----- 30s 2s/step - accuracy: 0.9681 - loss: 0.1097 - val_accuracy: 0.9815 - val_loss: 0.0922
Epoch 15/20
16/16 ----- 0s 2s/step - accuracy: 0.9890 - loss: 0.0621
Epoch 15: val_accuracy did not improve from 0.98148
16/16 ----- 42s 2s/step - accuracy: 0.9721 - loss: 0.0969 - val_accuracy: 0.9815 - val_loss: 0.0899
Epoch 16/20
16/16 ----- 0s 2s/step - accuracy: 0.9968 - loss: 0.0777
Epoch 16: val_accuracy did not improve from 0.98148
16/16 ----- 30s 2s/step - accuracy: 0.9841 - loss: 0.0974 - val_accuracy: 0.9815 - val_loss: 0.0850
Epoch 17/20
16/16 ----- 0s 2s/step - accuracy: 0.9538 - loss: 0.0976
Epoch 17: val_accuracy did not improve from 0.98148
16/16 ----- 41s 2s/step - accuracy: 0.9641 - loss: 0.1020 - val_accuracy: 0.9815 - val_loss: 0.0787
Epoch 18/20
16/16 ----- 0s 2s/step - accuracy: 0.9767 - loss: 0.1147
Epoch 18: val_accuracy did not improve from 0.98148
16/16 ----- 30s 2s/step - accuracy: 0.9721 - loss: 0.0967 - val_accuracy: 0.9815 - val_loss: 0.0791
Epoch 19/20
16/16 ----- 0s 2s/step - accuracy: 0.9945 - loss: 0.0670
Epoch 19: val_accuracy improved from 0.98148 to 1.00000, saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras

Epoch 19: finished saving model to /content/drive/MyDrive/Model_Melon/mobilenet_feature_best.keras
16/16 ----- 31s 2s/step - accuracy: 0.9880 - loss: 0.0729 - val_accuracy: 1.0000 - val_loss: 0.0746
Epoch 20/20
16/16 ----- 0s 2s/step - accuracy: 0.9708 - loss: 0.0699
Epoch 20: val_accuracy did not improve from 1.00000
16/16 ----- 40s 2s/step - accuracy: 0.9801 - loss: 0.0640 - val_accuracy: 1.0000 - val_loss: 0.0699
Restoring model weights from the end of the best epoch: 20.

```

Gambar 16 Hasil *Training* MobileNetV2

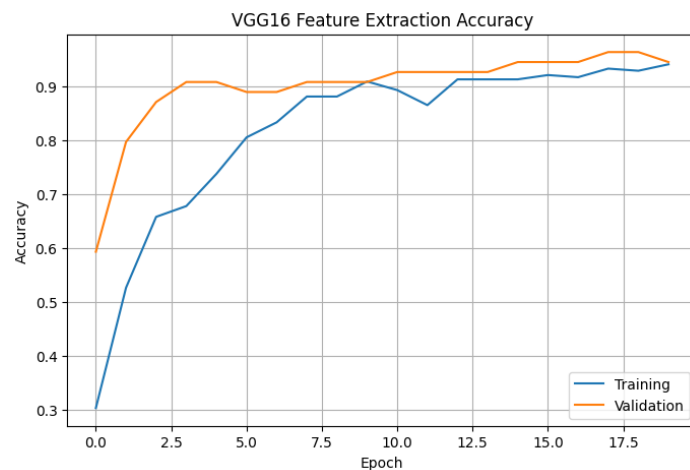
b. Hasil pengujian VGG16

Proses pelatihan model VGG16 dilakukan menggunakan metode *transfer learning* dengan bobot awal *ImageNet*. Pada tahap *feature extraction*, proses pelatihan berlangsung selama 20 epoch. Berdasarkan grafik *accuracy*, model menunjukkan peningkatan performa secara bertahap. Akurasi validasi meningkat dari 0,5926

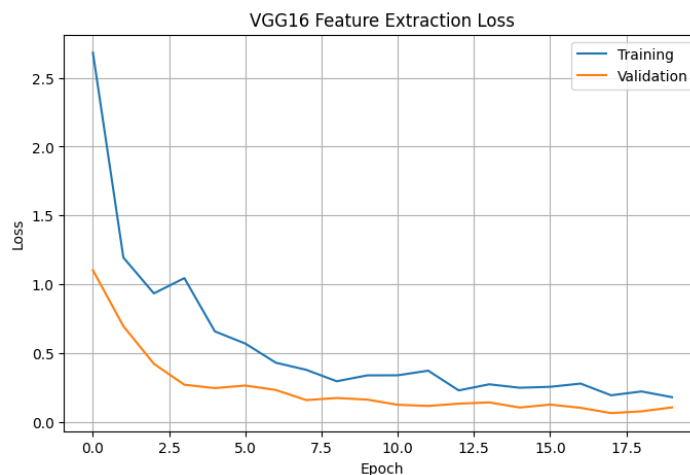
pada epoch pertama menjadi 0,9630 pada epoch ke-18, sedangkan akurasi pelatihan mencapai sekitar 0,9402 pada epoch terakhir.

Selain peningkatan *accuracy*, nilai *validation Loss* juga mengalami penurunan yang signifikan dari 1,1021 pada epoch pertama menjadi 0,0636 pada epoch ke-18. Penurunan *Loss* tersebut menunjukkan bahwa model semakin mampu mempelajari karakteristik citra daun melon dan menghasilkan prediksi yang lebih baik.

Berdasarkan hasil pelatihan selama 20 epoch, model VGG16 menunjukkan peningkatan performa secara bertahap dengan nilai *Accuracy* yang semakin tinggi dan *validation Loss* yang semakin rendah. Hasil tersebut menunjukkan bahwa VGG16 mampu mempelajari karakteristik citra daun melon dengan baik menggunakan pendekatan *transfer learning* berbasis *feature extraction*.



Gambar 17 Grafik *Accuracy* Model VGG16



Gambar 18 Grafik *Loss* Model VGG16

```

Epoch 1/20
16/16 ----- 0s 2s/step - accuracy: 0.2520 - loss: 3.5837
Epoch 1: val_accuracy improved from None to 0.59259, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 1: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 44s 2s/step - accuracy: 0.3028 - loss: 2.6828 - val_accuracy: 0.5926 - val_loss: 1.1021
Epoch 2/20
16/16 ----- 0s 2s/step - accuracy: 0.4742 - loss: 1.3892
Epoch 2: val_accuracy improved from 0.59259 to 0.79630, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 2: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 38s 2s/step - accuracy: 0.5259 - loss: 1.1940 - val_accuracy: 0.7963 - val_loss: 0.6954
Epoch 3/20
16/16 ----- 0s 2s/step - accuracy: 0.6162 - loss: 0.9471
Epoch 3: val_accuracy improved from 0.79630 to 0.87037, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 3: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 33s 2s/step - accuracy: 0.6574 - loss: 0.9336 - val_accuracy: 0.8704 - val_loss: 0.4221
Epoch 4/20
16/16 ----- 0s 2s/step - accuracy: 0.7070 - loss: 1.0040
Epoch 4: val_accuracy improved from 0.87037 to 0.90741, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 4: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 32s 2s/step - accuracy: 0.6773 - loss: 1.0447 - val_accuracy: 0.9074 - val_loss: 0.2698
Epoch 5/20
16/16 ----- 0s 2s/step - accuracy: 0.7552 - loss: 0.6444
Epoch 5: val_accuracy did not improve from 0.90741
16/16 ----- 40s 2s/step - accuracy: 0.7371 - loss: 0.6576 - val_accuracy: 0.9074 - val_loss: 0.2453
Epoch 6/20
16/16 ----- 0s 2s/step - accuracy: 0.7691 - loss: 0.6706
Epoch 6: val_accuracy did not improve from 0.90741
16/16 ----- 31s 2s/step - accuracy: 0.8048 - loss: 0.5690 - val_accuracy: 0.8889 - val_loss: 0.2640
Epoch 7/20
16/16 ----- 0s 2s/step - accuracy: 0.7957 - loss: 0.5157
Epoch 7: val_accuracy did not improve from 0.90741
16/16 ----- 38s 2s/step - accuracy: 0.8327 - loss: 0.4300 - val_accuracy: 0.8889 - val_loss: 0.2316
Epoch 8/20
16/16 ----- 0s 2s/step - accuracy: 0.8726 - loss: 0.4460
Epoch 8: val_accuracy did not improve from 0.90741
16/16 ----- 32s 2s/step - accuracy: 0.8805 - loss: 0.3782 - val_accuracy: 0.9074 - val_loss: 0.1582
Epoch 9/20
16/16 ----- 0s 2s/step - accuracy: 0.8611 - loss: 0.3302
Epoch 9: val_accuracy did not improve from 0.90741
16/16 ----- 36s 2s/step - accuracy: 0.8805 - loss: 0.2951 - val_accuracy: 0.9074 - val_loss: 0.1734
Epoch 10/20
16/16 ----- 0s 2s/step - accuracy: 0.8748 - loss: 0.4097
Epoch 10: val_accuracy did not improve from 0.90741
16/16 ----- 35s 2s/step - accuracy: 0.9084 - loss: 0.3379 - val_accuracy: 0.9074 - val_loss: 0.1617
Epoch 11/20
16/16 ----- 0s 2s/step - accuracy: 0.9085 - loss: 0.2801
Epoch 11: val_accuracy improved from 0.90741 to 0.92593, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 11: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 31s 2s/step - accuracy: 0.8924 - loss: 0.3384 - val_accuracy: 0.9259 - val_loss: 0.1242
Epoch 12/20
16/16 ----- 0s 2s/step - accuracy: 0.8572 - loss: 0.3647
Epoch 12: val_accuracy did not improve from 0.92593
16/16 ----- 32s 2s/step - accuracy: 0.8645 - loss: 0.3721 - val_accuracy: 0.9259 - val_loss: 0.1159
Epoch 13/20
16/16 ----- 0s 1s/step - accuracy: 0.9240 - loss: 0.1845
Epoch 13: val_accuracy did not improve from 0.92593
16/16 ----- 31s 2s/step - accuracy: 0.9124 - loss: 0.2287 - val_accuracy: 0.9259 - val_loss: 0.1323
Epoch 14/20
16/16 ----- 0s 2s/step - accuracy: 0.9027 - loss: 0.2834
Epoch 14: val_accuracy did not improve from 0.92593
16/16 ----- 39s 2s/step - accuracy: 0.9124 - loss: 0.2729 - val_accuracy: 0.9259 - val_loss: 0.1413
Epoch 15/20
16/16 ----- 0s 2s/step - accuracy: 0.8874 - loss: 0.3338
Epoch 15: val_accuracy improved from 0.92593 to 0.94444, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 15: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 31s 2s/step - accuracy: 0.9124 - loss: 0.2480 - val_accuracy: 0.9444 - val_loss: 0.1044
Epoch 16/20
16/16 ----- 0s 2s/step - accuracy: 0.9429 - loss: 0.1930
Epoch 16: val_accuracy did not improve from 0.94444
16/16 ----- 41s 2s/step - accuracy: 0.9203 - loss: 0.2547 - val_accuracy: 0.9444 - val_loss: 0.1257
Epoch 17/20
16/16 ----- 0s 2s/step - accuracy: 0.9368 - loss: 0.2121
Epoch 17: val_accuracy did not improve from 0.94444
16/16 ----- 31s 2s/step - accuracy: 0.9163 - loss: 0.2782 - val_accuracy: 0.9444 - val_loss: 0.1021
Epoch 18/20
16/16 ----- 0s 2s/step - accuracy: 0.9340 - loss: 0.1721
Epoch 18: val_accuracy improved from 0.94444 to 0.96296, saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras

Epoch 18: finished saving model to /content/drive/MyDrive/Model_Melon/vgg16_feature_best.keras
16/16 ----- 37s 2s/step - accuracy: 0.9323 - loss: 0.1931 - val_accuracy: 0.9630 - val_loss: 0.0636
Epoch 19/20
16/16 ----- 0s 2s/step - accuracy: 0.9138 - loss: 0.2683
Epoch 19: val_accuracy did not improve from 0.96296
16/16 ----- 31s 2s/step - accuracy: 0.9283 - loss: 0.2210 - val_accuracy: 0.9630 - val_loss: 0.0765
Epoch 20/20
16/16 ----- 0s 2s/step - accuracy: 0.9402 - loss: 0.1543
Epoch 20: val_accuracy did not improve from 0.96296
16/16 ----- 30s 2s/step - accuracy: 0.9402 - loss: 0.1794 - val_accuracy: 0.9444 - val_loss: 0.1052
Restoring model weights from the end of the best epoch: 18.

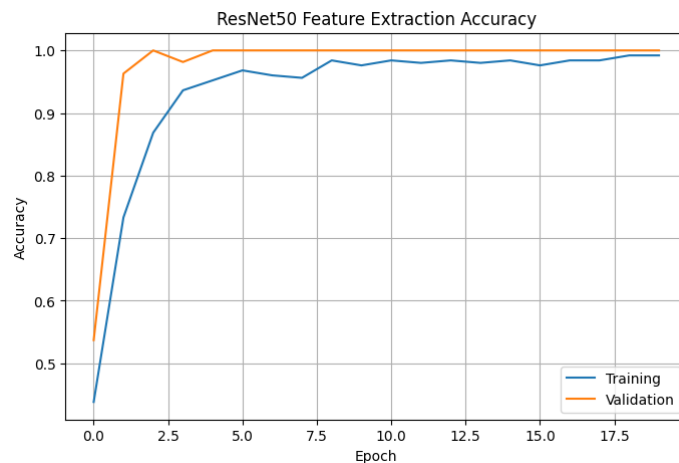
```

Gambar 19 Hasil *Training* VGG16

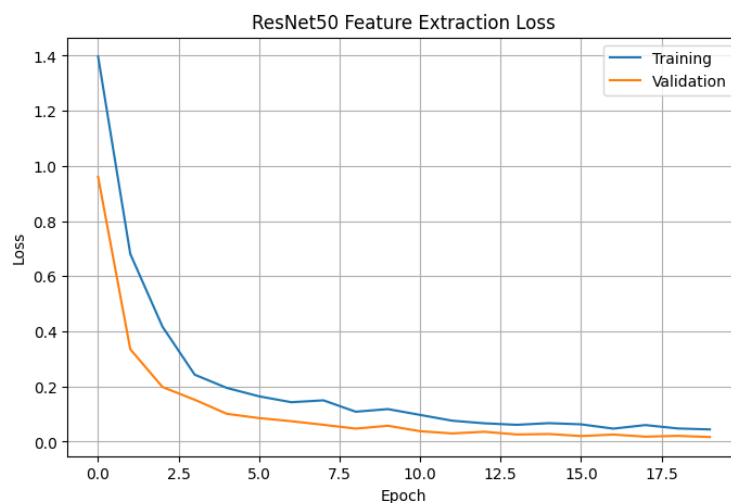
c. Hasil pengujian ResNet50

Proses pelatihan model ResNet50 dilakukan menggunakan metode *transfer learning* dengan bobot awal (*pre-trained weights*) *ImageNet*. Pada tahap *feature extraction*, proses pelatihan berlangsung selama 20 epoch. Berdasarkan grafik *accuracy*, model menunjukkan peningkatan performa yang sangat cepat. Akurasi validasi meningkat dari 0,5370 pada epoch pertama menjadi 100% mulai epoch ke-3 hingga akhir proses pelatihan, sedangkan akurasi pelatihan meningkat dari 0,4382 menjadi 0,9920 pada epoch ke-20.

Seiring dengan meningkatnya nilai *accuracy*, nilai *validation Loss* terus mengalami penurunan dari 0,9605 pada epoch pertama menjadi 0,0168 pada epoch ke-20. Penurunan nilai *Loss* tersebut menunjukkan bahwa model semakin mampu mempelajari karakteristik citra daun melon dan menghasilkan prediksi dengan tingkat kesalahan yang semakin kecil. Hasil ini menunjukkan bahwa proses *feature extraction* pada ResNet50 berjalan dengan baik dan mampu menghasilkan model yang stabil.



Gambar 20 Grafik Accuracy Model ResNet50



Gambar 21 Grafik Loss Model ResNet50

```

Epoch 1/20
16/16 ----- 0s 1s/step - accuracy: 0.3308 - loss: 1.6542
Epoch 1: val_accuracy improved from None to 0.53704, saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras

Epoch 1: finished saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras
16/16 ----- 48s 2s/step - accuracy: 0.4382 - loss: 1.3971 - val_accuracy: 0.5370 - val_loss: 0.9605
Epoch 2/20
16/16 ----- 0s 2s/step - accuracy: 0.6077 - loss: 0.8304
Epoch 2: val_accuracy improved from 0.53704 to 0.96296, saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras

Epoch 2: finished saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras
16/16 ----- 34s 2s/step - accuracy: 0.7331 - loss: 0.6810 - val_accuracy: 0.9630 - val_loss: 0.3353
Epoch 3/20
16/16 ----- 0s 2s/step - accuracy: 0.8912 - loss: 0.4273
Epoch 3: val_accuracy improved from 0.96296 to 1.00000, saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras

Epoch 3: finished saving model to /content/drive/MyDrive/Model_Melon/resnet50_feature_best.keras
16/16 ----- 32s 2s/step - accuracy: 0.8685 - loss: 0.4170 - val_accuracy: 1.0000 - val_loss: 0.1982
Epoch 4/20
16/16 ----- 0s 2s/step - accuracy: 0.9234 - loss: 0.2814
Epoch 4: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9363 - loss: 0.2431 - val_accuracy: 0.9815 - val_loss: 0.1521
Epoch 5/20
16/16 ----- 0s 2s/step - accuracy: 0.9507 - loss: 0.2175
Epoch 5: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9522 - loss: 0.1948 - val_accuracy: 1.0000 - val_loss: 0.1010
Epoch 6/20
16/16 ----- 0s 2s/step - accuracy: 0.9752 - loss: 0.1596
Epoch 6: val_accuracy did not improve from 1.00000
16/16 ----- 38s 2s/step - accuracy: 0.9681 - loss: 0.1645 - val_accuracy: 1.0000 - val_loss: 0.0856
Epoch 7/20
16/16 ----- 0s 2s/step - accuracy: 0.9636 - loss: 0.1482
Epoch 7: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9602 - loss: 0.1432 - val_accuracy: 1.0000 - val_loss: 0.0742
Epoch 8/20
16/16 ----- 0s 2s/step - accuracy: 0.9490 - loss: 0.1608
Epoch 8: val_accuracy did not improve from 1.00000
16/16 ----- 32s 2s/step - accuracy: 0.9562 - loss: 0.1499 - val_accuracy: 1.0000 - val_loss: 0.0611
Epoch 9/20
16/16 ----- 0s 1s/step - accuracy: 0.9817 - loss: 0.1078
Epoch 9: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9841 - loss: 0.1087 - val_accuracy: 1.0000 - val_loss: 0.0474
Epoch 10/20
16/16 ----- 0s 1s/step - accuracy: 0.9792 - loss: 0.1027
Epoch 10: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9761 - loss: 0.1182 - val_accuracy: 1.0000 - val_loss: 0.0578
Epoch 11/20
16/16 ----- 0s 2s/step - accuracy: 0.9831 - loss: 0.0886
Epoch 11: val_accuracy did not improve from 1.00000
16/16 ----- 35s 2s/step - accuracy: 0.9841 - loss: 0.0971 - val_accuracy: 1.0000 - val_loss: 0.0380
Epoch 12/20
Epoch 8: val_accuracy did not improve from 1.00000
16/16 ----- 32s 2s/step - accuracy: 0.9562 - loss: 0.1499 - val_accuracy: 1.0000 - val_loss: 0.0611
Epoch 9/20
16/16 ----- 0s 1s/step - accuracy: 0.9817 - loss: 0.1078
Epoch 9: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9841 - loss: 0.1087 - val_accuracy: 1.0000 - val_loss: 0.0474
Epoch 10/20
16/16 ----- 0s 1s/step - accuracy: 0.9792 - loss: 0.1027
Epoch 10: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9761 - loss: 0.1182 - val_accuracy: 1.0000 - val_loss: 0.0578
Epoch 11/20
16/16 ----- 0s 2s/step - accuracy: 0.9831 - loss: 0.0886
Epoch 11: val_accuracy did not improve from 1.00000
16/16 ----- 35s 2s/step - accuracy: 0.9841 - loss: 0.0971 - val_accuracy: 1.0000 - val_loss: 0.0380
Epoch 12/20
16/16 ----- 0s 2s/step - accuracy: 0.9750 - loss: 0.0821
Epoch 12: val_accuracy did not improve from 1.00000
16/16 ----- 37s 2s/step - accuracy: 0.9801 - loss: 0.0761 - val_accuracy: 1.0000 - val_loss: 0.0299
Epoch 13/20
16/16 ----- 0s 2s/step - accuracy: 0.9812 - loss: 0.0780
Epoch 13: val_accuracy did not improve from 1.00000
16/16 ----- 36s 2s/step - accuracy: 0.9841 - loss: 0.0666 - val_accuracy: 1.0000 - val_loss: 0.0360
Epoch 14/20
16/16 ----- 0s 2s/step - accuracy: 0.9876 - loss: 0.0461
Epoch 14: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9801 - loss: 0.0611 - val_accuracy: 1.0000 - val_loss: 0.0262
Epoch 15/20
16/16 ----- 0s 2s/step - accuracy: 0.9842 - loss: 0.0570
Epoch 15: val_accuracy did not improve from 1.00000
16/16 ----- 41s 2s/step - accuracy: 0.9841 - loss: 0.0672 - val_accuracy: 1.0000 - val_loss: 0.0277
Epoch 16/20
16/16 ----- 0s 2s/step - accuracy: 0.9835 - loss: 0.0613
Epoch 16: val_accuracy did not improve from 1.00000
16/16 ----- 37s 2s/step - accuracy: 0.9761 - loss: 0.0629 - val_accuracy: 1.0000 - val_loss: 0.0204
Epoch 17/20
16/16 ----- 0s 2s/step - accuracy: 0.9927 - loss: 0.0422
Epoch 17: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9841 - loss: 0.0473 - val_accuracy: 1.0000 - val_loss: 0.0258
Epoch 18/20
16/16 ----- 0s 2s/step - accuracy: 0.9866 - loss: 0.0581
Epoch 18: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9841 - loss: 0.0602 - val_accuracy: 1.0000 - val_loss: 0.0182
Epoch 19/20
16/16 ----- 0s 2s/step - accuracy: 0.9897 - loss: 0.0554
Epoch 19: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9920 - loss: 0.0479 - val_accuracy: 1.0000 - val_loss: 0.0212
Epoch 20/20
16/16 ----- 0s 2s/step - accuracy: 0.9963 - loss: 0.0334
Epoch 20: val_accuracy did not improve from 1.00000
16/16 ----- 31s 2s/step - accuracy: 0.9920 - loss: 0.0445 - val_accuracy: 1.0000 - val_loss: 0.0168
Restoring model weights from the end of the best epoch: 20.

```

Gambar 22 Hasil Training ResNet50

5. Hasil Perbandingan Model

Setelah proses pelatihan selesai, ketiga model final (mobilenetv2_final.keras, vgg16_final.keras, dan resnet50_final.keras) diuji menggunakan 56 citra data uji yang belum pernah dilihat oleh model selama pelatihan. Evaluasi dilakukan dengan menghitung metrik *accuracy*, *precision*, *recall*, dan *F1-score* menggunakan pustaka *scikit-learn*. Ringkasan hasil pengujian ditampilkan pada Tabel Berikut:

Tabel 10 Hasil Pengujian dan Perbandingan Model

No	Model	Accuracy	Precision	Recall	F1-score
1	MobileNetV2	0.9643	0.9722	0.9643	0.9655
2	VGG16	0.9643	0.9679	0.9643	0.9642
3	ResNet50	0.9821	0.9844	0.9821	0.9825

Berdasarkan Tabel, model ResNet50 memperoleh nilai *accuracy*, *precision*, *recall*, dan *F1-score* tertinggi di antara ketiga arsitektur, yaitu sebesar 0,9821 untuk *accuracy*, 0,9844 untuk *precision*, 0,9821 untuk *recall*, dan 0,9825 untuk *F1-score*. Model MobileNetV2 dan VGG16 memperoleh *Accuracy* yang sama (0,9643), namun MobileNetV2 memiliki *precision* dan *F1-score* yang sedikit lebih tinggi dibandingkan VGG16.

6. Classification report

a. Classification report MobileNetV2

Tabel 11 Classification report MobileNetV2

Kelas	Precision	Recall	F1-score	Support
<i>Cucumber Mosaic Virus</i>	1.00	1.00	1.00	13
<i>Healthy</i>	1.00	1.00	1.00	18
<i>Leaf Curl Disease</i>	0.78	1.00	0.88	7
<i>Powdery Mildew</i>	1.00	0.89	0.94	18

Berdasarkan Tabel 10, kelas *Cucumber Mosaic Virus* dan *Healthy* memperoleh nilai *precision*, *recall*, dan *F1-score* sebesar 1,00, yang menunjukkan bahwa seluruh data pada kedua kelas tersebut berhasil dideteksi dengan benar. Sementara itu, kelas *Leaf Curl Disease* memiliki nilai *precision* terendah sebesar 0,78 karena masih terdapat

beberapa prediksi yang keliru terhadap kelas *Powdery Mildew*. Meskipun demikian, nilai *recall* sebesar 1,00 menunjukkan bahwa seluruh data *Leaf Curl Disease* berhasil dikenali oleh model.

b. *Classification report VGG16*

Tabel 12 *Classification report VGG16*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	Support
<i>Cucumber Mosaic Virus</i>	1.00	0.92	0.96	13
<i>Healthy</i>	1.00	1.00	1.00	18
<i>Leaf Curl Disease</i>	1.00	0.86	0.92	7
<i>Powdery Mildew</i>	0.90	1.00	0.95	18

Berdasarkan Tabel 11, model VGG16 mampu mendeteksi kelas *Healthy* dengan sangat baik, ditunjukkan oleh nilai *precision*, *recall*, dan *F1-score* sebesar 1,00. Namun, nilai *recall* pada kelas *Cucumber Mosaic Virus* sebesar 0,92 dan *Leaf Curl Disease* sebesar 0,86 menunjukkan bahwa masih terdapat beberapa citra yang salah dideteksi, sehingga memengaruhi nilai *F1-score* pada kedua kelas tersebut.

c. *Classification report ResNet50*

Tabel 13 *Classification report ResNet50*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	Support
<i>Cucumber Mosaic Virus</i>	1.00	1.00	1.00	13
<i>Healthy</i>	1.00	0.94	0.97	18
<i>Leaf Curl Disease</i>	0.88	1.00	0.93	7
<i>Powdery Mildew</i>	1.00	1.00	1.00	18

Berdasarkan Tabel 12, model ResNet50 menunjukkan performa yang paling baik dibandingkan model lainnya. Kelas *Cucumber Mosaic Virus* dan *Powdery Mildew* memperoleh nilai *precision*, *recall*, dan *F1-score* sebesar 1,00. Pada kelas *Healthy*, nilai *recall* sebesar 0,94 menunjukkan masih terdapat satu citra yang salah dideteksi, sedangkan kelas *Leaf Curl Disease* memperoleh nilai *precision* sebesar 0,88 dengan *recall* sebesar 1,00. Secara

keseluruhan, nilai metrik pada seluruh kelas menunjukkan bahwa ResNet50 memiliki kemampuan deteksi yang sangat baik.

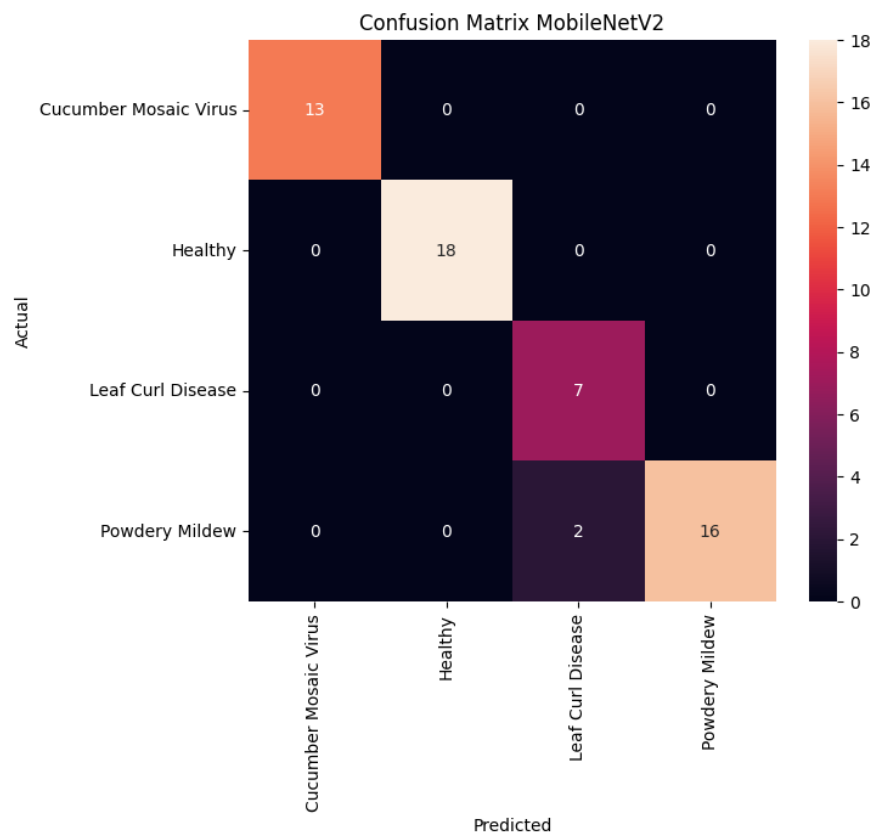
Secara umum, nilai *precision*, *recall*, dan *F1-score* pada kelas *Leaf Curl Disease* cenderung lebih bervariasi dibandingkan kelas lainnya pada ketiga model. Hal ini sejalan dengan jumlah data pada kelas tersebut yang paling sedikit, yaitu sebanyak 44 citra, sehingga model memiliki lebih sedikit variasi pola yang dapat dipelajari. Kondisi tersebut berpengaruh terhadap kemampuan model dalam membedakan kelas *Leaf Curl Disease* dengan kelas lain yang memiliki karakteristik visual yang serupa.

7. Hasil *Confusion matrix*

Confusion matrix digunakan untuk menganalisis lebih lanjut pola kesalahan deteksi tiap model pada 56 data uji, dengan urutan kelas *Cucumber Mosaic Virus*, *Healthy*, *Leaf Curl Disease*, dan *Powdery Mildew*.

a. *Confusion matrix* Model MobileNetV2

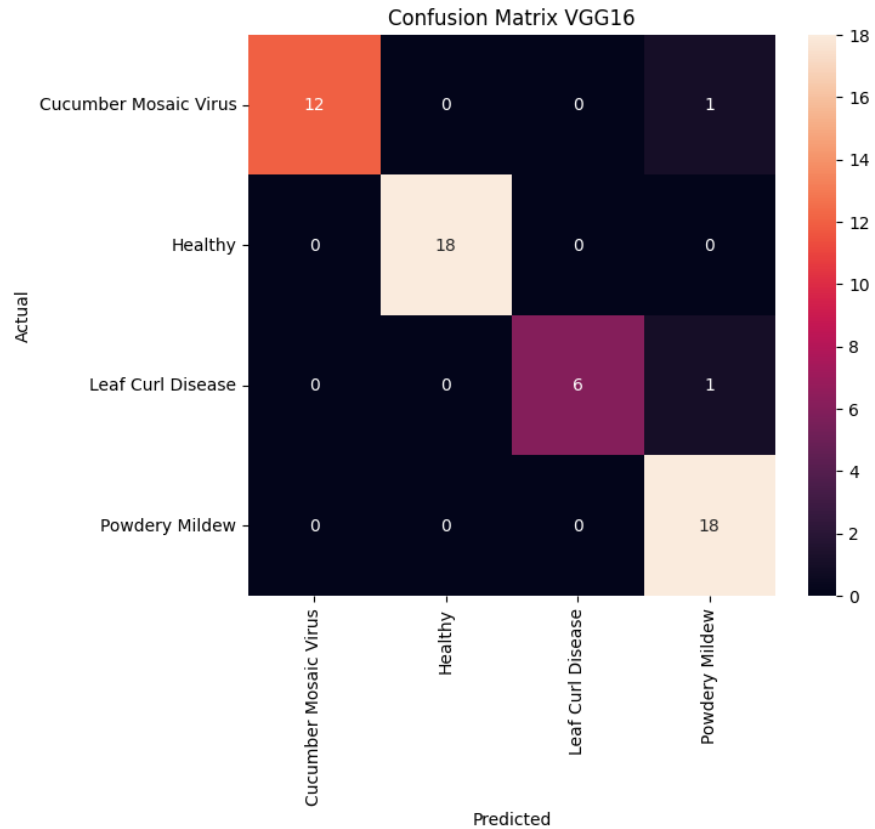
Pada model MobileNetV2, seluruh citra kelas *Cucumber Mosaic Virus*, *Healthy*, dan *Leaf Curl Disease* dideteksi dengan benar. Kesalahan deteksi hanya terjadi pada 2 citra kelas *Powdery Mildew* yang diprediksi sebagai *Leaf Curl Disease*.



Gambar 23 *Confusion matrix* Model MobileNetV2

b. *Confusion matrix* Model VGG16

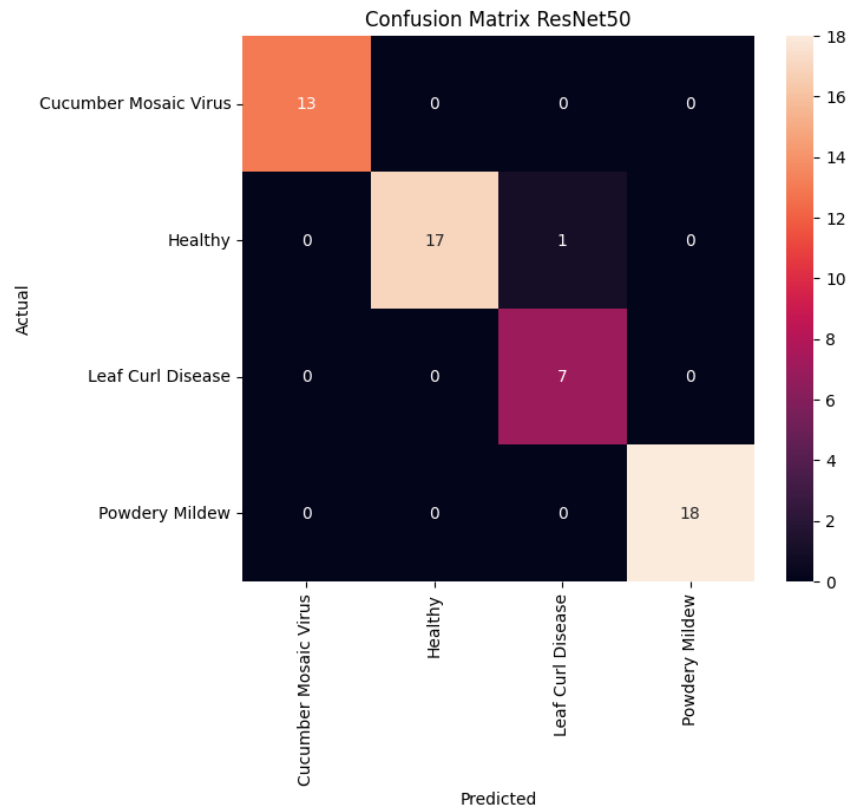
Pada model VGG16, kesalahan deteksi terjadi pada 1 citra *Cucumber Mosaic Virus* dan 1 citra *Leaf Curl Disease* yang keduanya diprediksi sebagai *Powdery Mildew*, sedangkan kelas *Healthy* dan *Powdery Mildew* dideteksi dengan sempurna.



Gambar 24 *Confusion matrix* Model VGG16

c. *Confusion matrix* Model ResNet50

Pada model ResNet50, kesalahan deteksi hanya terjadi pada 1 citra kelas *Healthy* yang diprediksi sebagai *Leaf Curl Disease*, sedangkan seluruh citra pada tiga kelas lainnya dideteksi dengan tepat. Jumlah kesalahan deteksi yang lebih sedikit ini konsisten dengan nilai *accuracy*, *precision*, *recall*, dan *F1-score* ResNet50 yang lebih tinggi dibandingkan kedua model lainnya.



Gambar 25 *Confusion matrix* Model ResNet50

B. Pembahasan

1. Implementasi Arsitektur CNN

Penelitian ini berhasil mengimplementasikan tiga arsitektur *Convolutional Neural Network* (CNN), yaitu MobileNetV2, VGG16, dan ResNet50, untuk mendeteksi penyakit daun melon hidroponik ke dalam empat kelas, yaitu *Healthy*, *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*. Implementasi dilakukan menggunakan pendekatan *transfer learning* dengan memanfaatkan bobot awal (*pre-trained weights*) dari *ImageNet*. Pendekatan ini dipilih karena mampu meningkatkan efektivitas proses pembelajaran model, terutama ketika jumlah dataset yang digunakan relatif terbatas.

Tahap *preprocessing* memberikan kontribusi penting terhadap keberhasilan implementasi model. Proses perubahan ukuran citra menjadi 224×224 piksel menghasilkan ukuran masukan yang sesuai dengan spesifikasi ketiga arsitektur CNN. Normalisasi nilai piksel membantu menghasilkan distribusi data yang lebih seragam sehingga proses optimasi selama pelatihan menjadi lebih stabil. Selain itu, augmentasi data pada data latih mampu meningkatkan variasi citra yang dipelajari model sehingga mengurangi kemungkinan terjadinya *overfitting*. Penelitian ini juga menerapkan *class weight* untuk

mengurangi pengaruh ketidakseimbangan jumlah data pada setiap kelas sehingga proses pembelajaran menjadi lebih proporsional.

Ketiga model dibangun dengan menghilangkan lapisan deteksi bawaan menggunakan parameter *include_top=False* sehingga hanya berfungsi sebagai feature extractor. Selanjutnya ditambahkan lapisan *Global Average Pooling 2D*, Dense sebanyak 256 neuron dengan fungsi aktivasi ReLU, Dropout sebesar 0,3, serta lapisan keluaran Softmax yang terdiri atas empat neuron sesuai dengan jumlah kelas penyakit daun melon. Seluruh model kemudian dilatih menggunakan optimizer Adam selama 20 epoch.

Berdasarkan hasil pelatihan dan pengujian, ketiga model mampu mempelajari karakteristik visual penyakit daun melon dengan baik. Hal tersebut ditunjukkan oleh peningkatan nilai *Accuracy* dan penurunan nilai *Loss* selama proses pelatihan serta tingginya nilai metrik evaluasi pada data uji. Dengan demikian, implementasi arsitektur CNN menggunakan pendekatan *transfer learning* terbukti mampu diterapkan secara efektif untuk deteksi penyakit daun melon hidroponik dan menjawab rumusan masalah pertama pada penelitian ini.

2. Perbandingan Berdasarkan Metrik Evaluasi

Perbandingan performa MobileNetV2, VGG16, dan ResNet50 dilakukan menggunakan empat metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Keempat metrik tersebut digunakan untuk memberikan gambaran menyeluruh mengenai kemampuan model dalam mendeteksi penyakit daun melon hidroponik.

Berdasarkan hasil pengujian, ResNet50 memperoleh performa terbaik dengan *Accuracy* sebesar 0,9821, *precision* sebesar 0,9844, *recall* sebesar 0,9821, dan *F1-score* sebesar 0,9825. Sementara itu, MobileNetV2 dan VGG16 masing-masing memperoleh *Accuracy* sebesar 0,9643 dengan perbedaan nilai *precision*, *recall*, dan *F1-score* yang relatif kecil. Hasil tersebut menunjukkan bahwa ketiga model memiliki kemampuan deteksi yang baik, namun ResNet50 mampu memberikan prediksi yang lebih akurat dan konsisten.

Keunggulan ResNet50 dipengaruhi oleh penggunaan residual *block* (*skip connection*) yang memungkinkan proses pembelajaran pada jaringan yang lebih dalam berlangsung secara optimal sehingga fitur-fitur penting dari citra daun dapat dipelajari dengan lebih baik. Sebaliknya, MobileNetV2 tetap menunjukkan performa yang kompetitif meskipun menggunakan jumlah parameter yang lebih sedikit, sedangkan VGG16 menghasilkan performa yang stabil namun belum mampu melampaui ResNet50. Berdasarkan keempat metrik evaluasi tersebut, ResNet50 menjadi model dengan performa terbaik pada penelitian ini.

3. Analisis *Classification report* dan *Confusion matrix*

Analisis *classification report* dan *confusion matrix* dilakukan untuk mengetahui performa model pada setiap kelas serta mengidentifikasi pola kesalahan deteksi yang terjadi. Berdasarkan hasil evaluasi, sebagian besar kelas memperoleh nilai *precision*, *recall*, dan *F1-score* yang tinggi, terutama pada kelas *Healthy* dan *Cucumber Mosaic Virus* yang hampir seluruh citranya berhasil dideteksi dengan benar oleh ketiga model.

Sebaliknya, kelas *Leaf Curl Disease* menunjukkan performa yang lebih rendah dibandingkan kelas lainnya, terutama pada nilai *precision* dan *recall*. Hasil *confusion matrix* menunjukkan bahwa sebagian besar kesalahan deteksi terjadi pada kelas tersebut. Kondisi ini dipengaruhi oleh jumlah data *Leaf Curl Disease* yang lebih sedikit dibandingkan kelas lainnya serta kemiripan karakteristik visual dengan beberapa penyakit daun lainnya, sehingga model masih mengalami kesulitan dalam membedakan kelas tersebut secara konsisten.

Secara keseluruhan, ResNet50 menghasilkan jumlah kesalahan deteksi paling sedikit dibandingkan MobileNetV2 dan VGG16. Hasil tersebut sejalan dengan nilai *accuracy*, *precision*, *recall*, dan *F1-score* yang lebih tinggi sehingga menunjukkan bahwa ResNet50 memiliki kemampuan generalisasi yang lebih baik dalam mendeteksi penyakit daun melon hidroponik.

4. Analisis Karakteristik Arsitektur

Perbedaan performa ketiga model dipengaruhi oleh karakteristik arsitektur masing-masing. MobileNetV2 menggunakan *depthwise separable convolution* dan *inverted residual block* sehingga memiliki jumlah parameter yang lebih sedikit dan proses komputasi yang lebih efisien. Meskipun demikian, model ini tetap mampu menghasilkan performa deteksi yang kompetitif.

VGG16 menggunakan arsitektur CNN konvensional dengan jumlah parameter yang lebih besar sehingga mampu mempelajari representasi fitur secara stabil, namun membutuhkan sumber daya komputasi yang lebih tinggi. Pada penelitian ini, peningkatan kompleksitas tersebut belum mampu menghasilkan performa yang lebih baik dibandingkan MobileNetV2 maupun ResNet50.

Sementara itu, ResNet50 memanfaatkan residual block (*skip connection*) yang memungkinkan jaringan mempelajari fitur yang lebih kompleks tanpa mengalami penurunan performa akibat vanishing gradient. Karakteristik tersebut berkontribusi terhadap kemampuan ResNet50 dalam menghasilkan nilai *accuracy*, *precision*, *recall*, dan *F1-score* tertinggi. Dengan demikian, pemilihan arsitektur CNN tidak hanya dipengaruhi oleh tingkat akurasi, tetapi juga perlu mempertimbangkan efisiensi komputasi sesuai kebutuhan implementasi.

Tabel 14 Karakteristik Arsitektur CNN

Aspek	MobileNetV2	VGG16	ResNet50
Jenis arsitektur	<i>Depthwise separable convolution</i>	CNN konvensional	Residual Network
Karakteristik utama	Ringan dan efisien	Sederhana dan stabil	Residual Block (<i>Skip Connection</i>)
Kompleksitas komputasi	Rendah	Tinggi	Tinggi
Jumlah parameter	Paling sedikit	Sangat banyak	Lebih banyak dari MobileNetV2, lebih sedikit dari VGG16
<i>Accuracy</i> penelitian	0,9643	0,9643	0,9821
Kelebihan	Efisien untuk perangkat terbatas	Struktur sederhana dan stabil	Ekstraksi fitur dan generalisasi terbaik
Kekurangan	Sedikit di bawah ResNet50 pada akurasi	Parameter besar, komputasi lebih berat	Komputasi lebih kompleks dibanding MobileNetV2

5. Perbandingan dengan Penelitian Terdahulu

Hasil penelitian ini sejalan dengan penelitian Muslih dan Krismawan yang menunjukkan bahwa penerapan *transfer learning* menggunakan ResNet50 mampu menghasilkan performa deteksi penyakit daun yang tinggi. Selain itu, penelitian Sharma *et al.* juga melaporkan bahwa ResNet50 dan MobileNetV2 merupakan arsitektur yang efektif dalam deteksi penyakit tanaman.

Perbedaan objek penelitian, jumlah kelas, serta karakteristik dataset menyebabkan adanya variasi nilai akurasi antar penelitian. Meskipun demikian, kecenderungan bahwa ResNet50 memberikan performa terbaik tetap terlihat konsisten. Temuan tersebut menunjukkan bahwa penggunaan residual block mampu meningkatkan kemampuan ekstraksi fitur pada berbagai jenis citra penyakit tanaman.

Dengan demikian, penelitian ini memperkuat hasil penelitian sebelumnya sekaligus menunjukkan bahwa ResNet50 merupakan arsitektur yang efektif untuk deteksi penyakit daun melon hidroponik, sedangkan MobileNetV2 tetap menjadi alternatif yang baik apabila efisiensi komputasi menjadi pertimbangan utama.

BAB V KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, implementasi arsitektur *Convolutional Neural Network* (CNN) menggunakan MobileNetV2, VGG16, dan ResNet50 dengan pendekatan *transfer learning* berhasil diterapkan untuk mendeteksi penyakit daun melon hidroponik ke dalam empat kelas, yaitu *Healthy*, *Powdery Mildew*, *Cucumber Mosaic Virus*, dan *Leaf Curl Disease*. Tahapan *preprocessing* yang meliputi perubahan ukuran citra menjadi 224×224 piksel, normalisasi nilai piksel, augmentasi data, serta penerapan *class weight* mampu mendukung proses pelatihan sehingga ketiga model dapat mempelajari karakteristik visual penyakit daun dengan baik. Hal tersebut ditunjukkan oleh peningkatan nilai *Accuracy* dan penurunan nilai *Loss* selama proses pelatihan pada seluruh model.

Berdasarkan hasil perbandingan kinerja menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, ResNet50 merupakan arsitektur dengan performa terbaik. ResNet50 memperoleh *Accuracy* sebesar 0,9821, *precision* sebesar 0,9844, *recall* sebesar 0,9821, dan *F1-score* sebesar 0,9825. Selain itu, hasil *classification report* dan *confusion matrix* menunjukkan bahwa ResNet50 memiliki jumlah kesalahan deteksi paling sedikit, sehingga menunjukkan kemampuan generalisasi yang lebih baik dibandingkan MobileNetV2 dan VGG16. Keunggulan tersebut dipengaruhi oleh penggunaan residual block (*skip connection*) yang memungkinkan model mengekstraksi fitur citra secara lebih optimal.

MobileNetV2 memperoleh *Accuracy* sebesar 0,9643 dengan nilai *precision*, *recall*, dan *F1-score* yang juga tinggi, sehingga menunjukkan kemampuan deteksi yang kompetitif meskipun menggunakan jumlah parameter yang paling sedikit. Karakteristik tersebut menjadikan MobileNetV2 lebih efisien dari sisi komputasi dan sesuai apabila diterapkan pada perangkat dengan sumber daya terbatas. Sementara itu, VGG16 juga memperoleh *Accuracy* sebesar 0,9643 dan menghasilkan performa deteksi yang stabil. Namun, kompleksitas model dan jumlah parameter yang lebih besar menyebabkan kebutuhan komputasi lebih tinggi tanpa memberikan peningkatan performa yang signifikan dibandingkan MobileNetV2 maupun ResNet50 pada penelitian ini.

Dengan demikian, ResNet50 direkomendasikan sebagai arsitektur yang paling optimal untuk deteksi penyakit daun melon hidroponik apabila prioritas utama adalah tingkat akurasi dan kemampuan generalisasi. Sebaliknya, MobileNetV2 dapat menjadi alternatif yang baik apabila efisiensi komputasi menjadi pertimbangan utama, sedangkan VGG16 tetap mampu menghasilkan performa yang baik, tetapi memerlukan sumber daya komputasi yang lebih besar.

B. Saran/Rekomendasi

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat menjadi bahan pertimbangan untuk penelitian selanjutnya.

1. Penelitian selanjutnya disarankan menggunakan jumlah dataset yang lebih banyak dengan distribusi data yang lebih seimbang pada setiap kelas agar kemampuan model dalam mempelajari karakteristik penyakit daun menjadi lebih optimal.
2. Penelitian selanjutnya dapat menambahkan variasi kondisi pengambilan citra, seperti perbedaan pencahayaan, latar belakang, dan sudut pengambilan gambar, sehingga model memiliki kemampuan generalisasi yang lebih baik terhadap kondisi nyata di lapangan.
3. Penelitian selanjutnya dapat membandingkan arsitektur CNN lainnya, seperti *EfficientNet*, *DenseNet*, atau *ConvNeXt*, maupun mengoptimalkan model menggunakan teknik *hyperparameter* tuning untuk memperoleh performa deteksi yang lebih baik.
4. Penelitian selanjutnya dapat mengimplementasikan model terbaik ke dalam aplikasi berbasis web atau perangkat mobile sebagai sistem pendukung identifikasi penyakit daun melon hidroponik secara otomatis sehingga hasil penelitian dapat dimanfaatkan secara lebih praktis oleh petani maupun masyarakat.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, “Statistik Hortikultura 2024,” 2024. Accessed: Jul. 04, 2026. [Online]. Available: <https://www.bps.go.id/>
- [2] Direktorat Jenderal Hortikultura, “Statistik Hortikultura Tahun 2024,” Jakarta, 2024. Accessed: Jul. 04, 2026. [Online]. Available: <https://hortikultura.pertanian.go.id/>
- [3] D. C. Rodríguez-Lira, D. M. Córdova-Esparza, J. M. Álvarez-Alvarado, J. Terven, J. A. Romero-González, and J. Rodríguez-Reséndiz, “Trends in Machine and Deep Learning Techniques for Plant Disease Identification: A Systematic Review,” Dec. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/agriculture14122188.
- [4] I. Pacal *et al.*, “A systematic review of deep learning techniques for plant diseases,” *Artif. Intell. Rev.*, vol. 57, no. 11, Nov. 2024, doi: 10.1007/s10462-024-10944-7.
- [5] A. Danang Krismawan, “Tomato Leaf Diseases Classification using Convolutional Neural Networks with Transfer Learning Resnet-50,” *Computer Network, Computing, Electronics, and Control Journal*, vol. 9, no. 2, pp. 149–158, 2024.
- [6] O. Attallah, “Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection,” *Horticulturae*, vol. 9, no. 2, p. 149, Jan. 2023, doi: 10.3390/horticulturae9020149.
- [7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Apr. 2015, [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015, [Online]. Available: <http://arxiv.org/abs/1512.03385>
- [9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Mar. 2019, [Online]. Available: <http://arxiv.org/abs/1801.04381>
- [10] K. Tian, J. Zeng, T. Song, Z. Li, A. Evans, and J. Li, “Tomato leaf diseases recognition based on deep convolutional neural networks,” *Journal of Agricultural Engineering*, Aug. 2022, doi: 10.4081/jae.2022.1432.
- [11] Mamatha. G and G T Raju, “Comparative Analysis of Deep Learning Models for Tomato Leaf Disease Classification: Insights and Opportunities,” *International Journal of Computational and Experimental Science and Engineering*, vol. 11, no. 3, Jul. 2025, doi: 10.22399/ijcesen.3483.

- [12] J. Sharma *et al.*, “Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures,” *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-98015-x.
- [13] N. Arifin and C. N. Insani, “Comparative Analysis of CNN, SVM, Decision Tree, Random Forest, and KNN for Maize Leaf Disease Detection Using Color and Texture Feature Extraction,” *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 5, pp. 3572–3586, Oct. 2025, doi: 10.52436/1.jutif.2025.6.5.5128.
- [14] G. S. Hukkeri, B. C. Soundarya, H. L. Gururaj, and V. Ravi, “Classification of Various Plant Leaf Disease Using Pretrained Convolutional Neural Network On Imagenet,” *Open Agric. J.*, vol. 18, no. 1, May 2024, doi: 10.2174/0118743315305194240408034912.
- [15] T. W. Utami, M. Novita, and K. Latifa, “Implementation and Comparative Analysis of CNN and Transfer Learning Models (EfficientNetB0, MobileNetV2, and ResNet50) for Rice Leaf Disease Detection Based on Digital Images,” 2025. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [16] R. M. Math and N. V. Dharwadkar, “Deep learning and computer vision for leaf miner infestation severity detection on muskmelon (*Cucumis melo*) leaves,” *Computers and Electrical Engineering*, vol. 110, p. 108843, Sep. 2023, doi: 10.1016/j.compeleceng.2023.108843.
- [17] M. Kang, N. Tamrakar, and H. T. Kim, “Improving Oriental Melon Leaf Disease Classification via DCGAN-Based Image Augmentation,” *Agriculture (Switzerland)*, vol. 15, no. 22, Nov. 2025, doi: 10.3390/agriculture15222324.
- [18] D. Kannan *et al.*, “Improving farming by quickly detecting muskmelon plant diseases using advanced ensemble learning and capsule networks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 38, no. 3, p. 2090, Jun. 2025, doi: 10.11591/ijeecs.v38.i3.pp2090-2100.
- [19] B. K. Hatuwal, A. Shakya, and B. Joshi, “Plant Leaf Disease Recognition Using Random Forest, KNN, SVM and CNN,” 2020, doi: 10.17562/PB-62-2.
- [20] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan, “The Plant Pathology Challenge 2020 data set to classify foliar disease of apples,” *Appl. Plant Sci.*, vol. 8, no. 9, Sep. 2020, doi: 10.1002/aps3.11390.
- [21] A. Ramcharan *et al.*, “Assessing a mobile-based deep learning model for plant disease surveillance.”
- [22] A. Ahmad, D. Saraswat, A. El Gamal, and G. Johal, “CD&S Dataset: Handheld Imagery Dataset Acquired Under Field Conditions for Corn

- Disease Identification and Severity Estimation,” Oct. 2021, [Online]. Available: <http://arxiv.org/abs/2110.12084>
- [23] Ian Goodfellow; Yoshua Bengio; Aaron Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016.
 - [24] G. C. Cawley and N. L. C. Talbot, “On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation,” 2010. doi: 10.5555/1756006.1859921.
 - [25] L. Prechelt, “Automatic early stopping using cross validation: quantifying the criteria,” *Neural Networks*, vol. 11, no. 4, pp. 761–767, Jun. 1998, doi: 10.1016/S0893-6080(98)00010-0.
 - [26] C. Sáez *et al.*, “RNA-Seq Transcriptome Analysis Provides Candidate Genes for Resistance to Tomato Leaf Curl New Delhi Virus in Melon,” *Front. Plant Sci.*, vol. 12, Jan. 2022, doi: 10.3389/fpls.2021.798858.
 - [27] Y. Cao, Q. Diao, Y. Chen, H. Jin, Y. Zhang, and H. Zhang, “Development of KASP Markers and Identification of a QTL Underlying Powdery Mildew Resistance in Melon (*Cucumis melo* L.) by Bulk Segregant Analysis and RNA-Seq,” *Front. Plant Sci.*, vol. 11, Feb. 2021, doi: 10.3389/fpls.2020.593207.
 - [28] M. Tian *et al.*, “Effect of Powdery Mildew on the Photosynthetic Parameters and Leaf Microstructure of Melon,” *Agriculture (Switzerland)*, vol. 14, no. 6, Jun. 2024, doi: 10.3390/agriculture14060886.
 - [29] R. Rasoulpour, A. Afsharifar, and K. Izadpanah, “Partial biological and molecular characterization of a Cucumber mosaic virus isolate naturally infecting *Cucumis melo* in Iran,” *Virusdisease*, vol. 27, no. 2, pp. 193–197, Jun. 2016, doi: 10.1007/s13337-016-0310-3.
 - [30] C. De Moya-Ruiz, P. Gómez, and M. Juárez, “Occurrence, Distribution, and Management of Aphid-Transmitted Viruses in Cucurbits in Spain,” Mar. 01, 2023, *MDPI*. doi: 10.3390/pathogens12030422.
 - [31] F. Wilisiani, S. Somowiyarjo, and S. Hartono, “IDENTIFIKASI MOLEKULER VIRUS PENYEBAB PENYAKIT DAUN KERITING ISOLAT BANTUL PADA MELON MOLECULAR IDENTIFICATION OF VIRUS CAUSING LEAF CURL DISEASE BANTUL ISOLATE ON MELON.”
 - [32] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.

LAMPIRAN

A. Surat Permohonan Observasi


UNUGHA CILACAP
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER (FMIKOM)
Keputusan Kemendikbud RI Nomor : 264/E/O/2014 Tanggal 23 Juli 2014

Nomor	: B/230/Ybk.041.10/PT/2026	Cilacap, 01 Juli 2026
Lampiran	: -	
Hal	: Permohonan Observasi	

Yth.
Pimpinan Labiq Farm Purwojati Banyumas
di –
Tempat

Assalamu 'alaikum Wr.Wb.

Salam silaturahmi dan sejahtera kami sampaikan semoga kita senantiasa mendapatkan ridlo dan pertolongan dari Allah SWT dalam menjalankan aktivitas sehari-hari. Amin.

Sehubungan dengan tuntutan kebutuhan mahasiswa untuk mendapatkan pengalaman nyata di lapangan, maka Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali Cilacap menugaskan kepada mahasiswa kami:

Nama	: Duffa Imanulloh Gunawan
NIM	: 22EO10016
Prodi	: Informatika

Untuk mengadakan riset terkait dengan skripsi yang sedang di kerjakan dengan Judul "Perbandingan Kinerja Arsitektur *Convolutional Neural Network* (Vgg16, Resnet50, dan Mobilenetv2) dalam Klasifikasi Penyakit Daun Tanaman". Berkenaan dengan hal tersebut, maka kami mengajukan permohonan kepada Bapak Pimpinan Labiq Farm Purwojati Banyumas untuk mengizinkan mahasiswa kami melaksanakan kegiatan tersebut.

Demikian surat ini kami sampaikan, atas bimbingan dan kerjasamanya disampaikan terimakasih.
Wassalamu 'alaikum Wr.Wb.

Dekan,


Mochamad T.A. Aziz Zein, M.Kom
NIK. 41 230714 020

UNIVERSITAS NAHDLATUL 'ULAMA AL GHAZALI CILACAP
Jln. Kemerdekaan Barat No.17 Kesugihan Cilacap Jawa Tengah K.Pos 53274, [http:// unugha.ac.id](http://unugha.ac.id),
Email : fmikom@unugha.ac.id Telp. : (0282) 695415, 695407, Fax : (0282) 695407

B. Pengambilan Dataset



C. Kode MobileNetV2

```
[ ]  inputs = Input(  
    shape=(224,224,3)  
)  
  
x = data_augmentation(  
    inputs  
)  
  
x = base_model_mobile(  
    x,  
    training=False  
)  
  
x = GlobalAveragePooling2D()(x)  
  
x = Dropout(  
    0.3  
)(x)  
  
x = Dense(  
    256,  
    activation="relu"
```

```
)(x)
```

```
outputs = Dense(  
    num_classes,  
    activation="softmax"
```

```
)(x)
```

```
model_mobile = Model(  
    inputs,  
    outputs  
)
```

```
print("Model MobileNetV2 dibuat")
```

```
... Model MobileNetV2 dibuat
```

D. *Kode Training*

```
[ ]  history_mobile = model_mobile.fit(  
    train_mobile,  
  
    validation_data=val_mobile,  
  
    epochs=20,  
  
    class_weight=class_weights,  
  
    callbacks=[  
        checkpoint_mobile,  
        earllystop_mobile  
    ]  
)
```

E. Kode Grafik *Accuracy Training* dan *Validation*

```
[ ] plt.figure(  
    figsize=(8,5)  
)  
  
plt.plot(  
    history_mobile.history["accuracy"],  
    label="Training"  
)  
  
plt.plot(  
    history_mobile.history["val_accuracy"],  
    label="Validation"  
)  
  
plt.title(  
    "MobileNetV2 Feature Extraction Accuracy"  
)  
  
plt.xlabel(  
    "Epoch"  
)  
  
plt.ylabel(  
    "Accuracy"  
)  
  
plt.legend()  
  
plt.grid(True)  
  
plt.show()
```

F. Kode Grafik *Loss Training* dan *Validation*

```
[ ] plt.figure(  
    figsize=(8,5)  
)  
  
plt.plot(  
    history_mobile.history["loss"],  
    label="Training"  
)  
  
plt.plot(  
    history_mobile.history["val_loss"],  
    label="Validation"  
)  
  
plt.title(  
    "MobileNetV2 Feature Extraction Loss"  
)  
  
plt.xlabel(  
    "Epoch"  
)  
  
plt.ylabel(  
    "Loss"  
)  
  
plt.legend()  
  
plt.grid(True)  
  
plt.show()
```

G. Kode Perbandingan MobileNetV2, VGG16, dan ResNet50

```
results = pd.DataFrame(  
    [  
        result_mobile,  
        result_vgg,  
        result_resnet  
    ],  
    columns=[  
        "Model",  
        "Accuracy",  
        "Precision",  
        "Recall",  
        "F1-Score"  
    ]  
)  
  
results
```

	Model	Accuracy	Precision	Recall	F1-Score
0	MobileNetV2	0.964286	0.972222	0.964286	0.965467
1	VGG16	0.964286	0.967857	0.964286	0.964182
2	ResNet50	0.982143	0.984375	0.982143	0.982483

H. *Code Confusion matrix*

```
[ ] ▶ cm_mobile = confusion_matrix(  
    y_true,  
    y_pred_mobile  
)  
  
plt.figure(  
    figsize=(7,6)  
)  
  
sns.heatmap(  
    cm_mobile,  
    annot=True,  
    fmt="d",  
    xticklabels=class_names,  
    yticklabels=class_names  
)  
  
plt.title(  
    "Confusion Matrix MobileNetV2"  
)  
  
plt.xlabel(  
    "Predicted"  
)  
  
plt.ylabel(  
    "Actual"  
)  
  
plt.show()
```

I. Kode grafik perbandingan Accuracy

```
[ ] ▶ plt.figure(  
    figsize=(8,5)  
)  
  
plt.bar(  
    results["Model"],  
    results["Accuracy"]  
)  
  
plt.title(  
    "Perbandingan Accuracy Model"  
)  
  
plt.xlabel(  
    "Model"  
)  
  
plt.ylabel(  
    "Accuracy"  
)  
  
plt.ylim(  
    0,  
    1  
)  
  
plt.grid(  
    axis="y"  
)  
  
plt.show()
```