ANALISIS PERBANDINGAN PERFORMA CNN KUSTOM DAN MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN DATASET TIDAK SEIMBANG DAN SEIMBANG



Disusun Oleh:

N a m a : Mohammad Farid Anshori
NIM : 22523124

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

ANALISIS PERBANDINGAN PERFORMA CNN KUSTOM DAN MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN DATASET TIDAK SEIMBANG DAN SEIMBANG

TUGAS AKHIR

Disusun Oleh:

N a m a : Mohammad Farid Anshori

NIM : 22523124

Yogyakarta, 31 Agustus 2026

Pembimbing,



(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

ANALISIS PERBANDINGAN PERFORMA CNN KUSTOM DAN MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN DATASET TIDAK SEIMBANG DAN SEIMBANG

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 31 Agustus 2026

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom.

Anggota 1

Dr. HENDRIK, S.T., M.Eng.

Anggota 2

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Mohammad Farid Anshori

NIM : 22523124

Tugas akhir dengan judul:

ANALISIS PERBANDINGAN PERFORMA CNN KUSTOM DAN MOBILENETV2 PADA KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN DATASET TIDAK SEIMBANG DAN SEIMBANG

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Juli 2026



(Mohammad Farid Anshori)

HALAMAN PERSEMBAHAN

Alhamdulillahi rabbil 'alamin, puji dan syukur penulis panjatkan ke hadirat Allah Swt. atas segala nikmat, rahmat, dan kemudahan-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Karya yang sederhana ini penulis persembahkan dengan penuh rasa hormat, bakti, dan terima kasih yang mendalam khusus kepada pihak-pihak tercinta yang namanya tertulis di bawah ini:

1. Ibu Suwarni tercinta yang tanpa lelah mendoakan, memberikan kasih sayang, serta dukungan moril maupun materiil yang tiada henti selama penulis menempuh Pendidikan. Skripsi ini merupakan persembahan kecil dari penulis sebagai wujud bakti atas segala pengorbanan yang telah ibu berikan.
2. Almarhum Bapak Suradi tercinta, yang meskipun telah berpulang, namun garis didikan, teladan, dan impian beliau selalu hidup di hati penulis. Karya ini adalah pemenuhan janji dan harapan yang pernah Beliau titipkan agar penulis menjadi orang yang bermanfaat. Semoga Allah Swt. membrikan tempat terbaik dan termulia di sisi-Nya.
3. Kakak-kakak penulis tersayang Desi Puspitasari, Rudy Prietno, Dini Fitri Nur Utomo dan almarhum Linda Puspitasari, terima kasih atas segala bentuk perhatian, perlindungan, dan motivasi yang diberikan. Keberadaan dan dukungannya selalu menjadi *support system* terbaik yang membuat perjalanan kuliah yang penuh dinamika ini terasa lebih ringan bagi penulis.
4. Sahabat-sahabat penulis di INVACY 22 yang selalu berbagi tawa, mendengarkan keluh kesah, dan selalu memberikan bantuan selama proses penyusunan skripsi ini. Terima kasih telah menjadi bagian dari pendewasaan diri penulis.
5. Sahabat-sahabat SMP penulis Rafael, Tito, Namja, Yongky, Berlin, Bagas, dan Niko, yang selalu menjadi tempat cerita selama proses penyusunan tugas akhir ini.
6. Sahabat-sahabat penulis di lingkungan rumah Ridwan, Dayat, Dava, Cahyo, Pram, Mail, yang tetap hadir sebagai ruang pulang untuk berbagi cerita dan penguat langkah selama proses penyusunan tugas akhir ini.

HALAMAN MOTO

"*maybe its not my weekend, but its gonna be my year*."

(*weightless – All Time Low*)

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirabbil'alamin, segala puji bagi Allah Swt. atas rahmat, karunia, dan kemudahan-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir ini. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad saw., keluarga, sahabat, dan umatnya hingga akhir zaman.

Laporan tugas akhir ini disusun sebagai salah satu syarat menyelesaikan pendidikan Program Sarjana (S-1) Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penelitian ini berjudul "Analisis Perbandingan Performa CNN Kustom dan MobileNetV2 dengan *Fine-tuning* pada Klasifikasi Penyakit Daun Padi Menggunakan Dataset Tidak Seimbang dan Dataset Seimbang". Penelitian ini membahas perbandingan performa kedua model serta pengaruh ketidakseimbangan data terhadap hasil evaluasi. Dalam proses penyusunan tugas akhir ini, penulis menghadapi berbagai kendala. Namun, berkat dukungan berbagai pihak, laporan ini dapat diselesaikan dengan baik. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN Eng selaku Rektor universitas Islam Indonesia.
2. Ibu Dr. Sri Kusumadewi, S.Si., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana.
5. Ibu Arrie Kurniawardhani, S.Si., M.Kom., Selaku dosen pembimbing yang memberi arahan, bimbingan, dan masukan selama penelitian.
6. Ibu Ir. Izzati Muhimmah, S.T., M.Sc., Ph.D. Selaku dosen pembimbing akademik yang memberi masukan selama perkuliahan.

Yogyakarta, Juli 2026



(Mohammad Farid Anshori)

SARI

Padi merupakan salah satu komoditas pangan utama di Indonesia. Dalam proses budidaya, tanaman padi dapat mengalami berbagai penyakit yang salah satunya terlihat melalui gejala pada daun, seperti bercak, perubahan warna, lesi, bagian daun yang mengering, atau perubahan tekstur permukaan daun. Beberapa penyakit daun padi memiliki karakteristik visual yang mirip sehingga proses identifikasi secara manual tidak selalu mudah. Oleh karena itu, penelitian ini memanfaatkan pendekatan *deep learning* untuk melakukan klasifikasi penyakit daun padi berdasarkan citra digital, dengan tujuan membandingkan performa CNN Kustom dan MobileNetV2 dalam mengklasifikasikan tujuh kondisi daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro.

Penelitian ini juga menganalisis pengaruh ketidakseimbangan data pada kelas Tungro dengan membandingkan dua skenario dataset, yaitu dataset original dan dataset *balanced quality-hybrid*. Dataset original berjumlah 5.137 citra, sedangkan dataset *balanced quality-hybrid* berjumlah 4.571 citra setelah dilakukan *quality-based filtering* pada kelas Tungro dan augmentasi pada kelas minoritas. Pendekatan ini digunakan agar distribusi data menjadi lebih seimbang sekaligus mempertimbangkan kualitas visual citra yang digunakan.

Tahapan penelitian meliputi pengumpulan data, *preprocessing*, augmentasi data, pembagian data *training*, *validation*, dan *testing*, pembuatan model, pelatihan model, evaluasi, serta analisis perbandingan. CNN Kustom digunakan sebagai model *baseline* yang dilatih dari awal, sedangkan MobileNetV2 digunakan sebagai model *transfer learning* dengan tahapan *feature extraction* dan *fine-tuning*. Pada MobileNetV2, penentuan jumlah *layer* akhir yang dibuka saat *fine-tuning* dilakukan melalui percobaan ablasi, yaitu 0, 10, 20, 30, dan 50 *layer*. Evaluasi model dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-score*, *classification report*, dan *confusion matrix*, dengan penekanan pada *recall macro* karena penelitian berkaitan dengan klasifikasi penyakit dan ketidakseimbangan data.

Berdasarkan hasil pengujian, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* memperoleh performa terbaik dengan *accuracy* sebesar 92,26%, *recall macro* sebesar 92,21%, dan *F1-score macro* sebesar 92,17%. CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* memperoleh *accuracy* sebesar 76,34%, *recall macro* sebesar 76,07%, dan *F1-score macro* sebesar 75,72%. Hasil penelitian menunjukkan bahwa *balancing* dataset berbasis *quality-based filtering* dan augmentasi dapat meningkatkan performa MobileNetV2, tetapi pengaruhnya tidak selalu sama pada setiap arsitektur model. MobileNetV2 juga memiliki ukuran model dan jumlah parameter yang lebih kecil

dibandingkan CNN Kustom, sehingga memberikan kombinasi performa klasifikasi dan efisiensi model yang lebih baik.

Kata kunci: CNN Kustom, MobileNetV2, *fine-tuning*, klasifikasi penyakit daun padi, *imbalance* dataset, *deep learning*.

GLOSARIUM

Accuracy	Metrik evaluasi yang menunjukkan perbandingan jumlah prediksi benar terhadap seluruh data yang diuji.
Augmentasi data	Teknik untuk menambah variasi citra latih, misalnya dengan membalik atau memutar citra.
CNN	Singkatan dari *Convolutional Neural Network*, yaitu metode *deep learning* yang umum digunakan untuk klasifikasi citra
CNN Kustom	Model CNN yang dirancang sendiri dan dilatih dari awal tanpa menggunakan bobot *pretrained*.
Confusion matrix	Tabel yang menunjukkan perbandingan antara kelas sebenarnya dan kelas hasil prediksi model.
Dataset *balanced*	Dataset yang jumlah data antar kelasnya dibuat lebih seimbang.
Dataset original	Dataset awal sebelum dilakukan penyeimbangan data
Deep learning	Metode pembelajaran mesin yang menggunakan jaringan saraf berlapis untuk mempelajari pola dari data.
Dropout	Teknik regularisasi untuk mengurangi risiko *overfitting* dengan menonaktifkan sebagian neuron secara acak selama pelatihan.
F1-score	Metrik evaluasi yang menggabungkan nilai *precision* dan *recall*.
Feature extraction	Tahap penggunaan model *pretrained* untuk mengekstraksi fitur citra.
Fine-tuning	Tahap pelatihan lanjutan dengan membuka sebagian layer model *pretrained* agar menyesuaikan dengan dataset penelitian.
Imbalance dataset	Kondisi ketika jumlah data pada setiap kelas tidak seimbang.
MobileNetV2	Arsitektur CNN ringan yang digunakan sebagai model *transfer learning* dalam penelitian ini.
Overfitting	Kondisi ketika model terlalu menghafal data *training* sehingga kurang baik saat diuji pada data baru.
Precision	Metrik yang menunjukkan ketepatan prediksi model pada suatu kelas.
Preprocessing	Tahap pengolahan awal citra sebelum dimasukkan ke dalam model.
Recall	Metrik yang menunjukkan kemampuan model dalam menemukan data yang benar pada suatu kelas.
Transfer learning	Pendekatan yang memanfaatkan model yang telah dilatih sebelumnya untuk menyelesaikan tugas baru.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI viii
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 4
1.4 Batasan Masalah 4
1.5 Manfaat Penelitian 5
1.5.1 Manfaat Teoritis 5
1.5.2 Manfaat Praktis 5
1.6 Sistematika Penulisan 5
BAB II LANDASAN TEORI 6
2.1 Dasar Teori 6
2.1.1 Penyakit Daun Padi 6
2.1.2 Citra Digital 6
2.1.3 Convolutional Neural Network 7
2.1.4 Convolutional Neural Network Kustom 8
2.1.5 Transfer learning 8
2.1.6 MobileNetV2 9
2.1.7 Imbalance Dataset 10
2.1.8 Preprocessing 10
2.1.9 Augmentasi Data 11
2.1.10 Metrik Evaluasi 11
2.2 Penelitian Sebelumnya 14
2.3 Pemilihan Metode 18
BAB III METODOLOGI PENELITIAN 21
3.1 Langkah-langkah penelitian 21
3.1.1 Pengumpulan Data 22
3.1.2 *Preprocessing* 26
3.1.3 Pembuatan Model CNN Kustom 26
3.1.4 Pembuatan Model MobileNetV2 31
3.1.5 Pelatihan Model 40
3.1.6 Evaluasi 43
3.1.7 Analisis Perbandingan 44
BAB IV HASIL DAN PEMBAHASAN 46
4.1 Pengumpulan Data 46
4.1.1 Dataset Original 46

4.1.2 Dataset *Balanced* 48
4.2 *Preprocessing* 49
4.3 Pembuatan Model CNN Kustom 50
4.3.1 Implementasi CNN Kustom Awal 50
4.3.2 Implementasi CNN Kustom dengan *Dropout* 51
4.4 Pembuatan Model MobileNetV2 52
4.4.1 Implementasi MobileNetV2 *Feature extraction* 52
4.4.2 Implementasi MobileNetV2 *Fine-tuning* 53
4.5 Pelatihan Model 54
4.5.1 Hasil Pelatihan CNN Kustom 55
4.5.2 Pelatihan MobileNetV2 59
4.5.3 Ringkasan Seluruh Percobaan Pelatihan 62
4.6 Evaluasi Model 63
4.6.1 Evaluasi Seluruh Skenario Model 64
4.6.2 Evaluasi Ablasi MobileNetV2 65
4.6.3 Evaluasi Dataset Original 66
4.6.4 Evaluasi Dataset *Balanced Quality-Hybrid* 68
4.6.5 Analisis Confusion Matrix 69
4.6.6 Penentuan Model Terbaik 72
4.6.7 Pengujian Foto Lapangan 73
4.7 Analisis Perbandingan 76
4.7.1 Performa CNN Kustom dalam Klasifikasi Tujuh Kondisi Daun Padi 76
4.7.2 Performa MobileNetV2 Berbasis Transfer Learning 77
4.7.3 Perbandingan CNN Kustom dan MobileNetV2 pada Dataset Original 77
4.7.4 Perbandingan CNN Kustom dan MobileNetV2 pada Dataset *Balanced Quality-Hybrid* 78
4.7.5 Pengaruh Ketidakseimbangan Data pada Kelas Tungro 78
4.8 Analisis Aspek Komputasional 79
4.8.1 Jumlah Parameter dan Ukuran Model 79
4.8.2 Parameter *Trainable* pada Ablasi MobileNetV2 80
4.8.3 Indokator Waktu Pelatihan 81
4.8.4 Waktu Inferensi 82
BAB V KESIMPULAN DAN SARAN 83
5.1 Kesimpulan 83
5.2 Saran 85
DAFTAR PUSTAKA 87
LAMPIRAN 91

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya....16
Tabel 3.1 Distribusi dataset original....22
Tabel 3.2 Distribusi dataset *balance*....23
Tabel 3.3 Konfigurasi Arsitektur CNN Kustom....28
Tabel 3.4 Hyperparameter Pelatihan CNN Kustom....28
Tabel 3.5 Konfigurasi Arsitektur MobileNetV2....33
Tabel 3.6 Konfigurasi Hyperparameter pelatihan MobileNetV2....34
Tabel 3.7 Skenario eksperimen pelatihan model....40
Tabel 3.8 Kombinasi eksperimen Utama....41
Tabel 3.9 Konfigurasi *hyperparameter* pelatihan model....42
Tabel 4.1 Distribusi dataset original....46
Tabel 4.2 Distribusi dataset *balance*....48
Tabel 4.3 Arsitektur CNN Kustom Awal....50
Tabel 4.4 Arsitektur CNN Kustom dengan *Dropout*....51
Tabel 4.5 Arsitektur MobileNetV2 Featue Extraction....52
Tabel 4.6 Arsitektur MobileNetV2 *Fine-tuning*....53
Tabel 4.7 Ringkasan Trial Pelatihan CNN Kustom....56
Tabel 4.8 Ringkasan Trial Pelatihan MobileNetV2....59
Tabel 4.9 Ringkasan Seluruh Percobaan Pelatihan....62
Tabel 4.10 Ringkasan Hasil Evaluasi Skenario Utama....64
Tabel 4.11 Hasil ablasi MobileNetV2 pada dataset original....65
Tabel 4.12 Hasil ablasi MobileNetV2 pada dataset *balanced quality-hybrid*....65
Tabel 4.13 Hasil evaluasi dataset original....66
Tabel 4.14 *Classification report* CNN Kustom dengan *dropout* dataset original....67
Tabel 4.15 *Classification report* MobileNetV2 *fine-tuning* 50 *layer* dataset original....67
Tabel 4.16 Hasil evaluasi dataset *balanced quality-hybrid*....68
Tabel 4.17 *Classification report* CNN Kustom dengan *dropout* dataset *balanced quality-hybrid*....68
Tabel 4.18 *Classification report* MobileNetV2 *fine-tuning* 50 *layer* dataset *balanced quality-hybrid*....69
Tabel 4.19 Ringkasan penentuan model terbaik....72
Tabel 4.20 Hasil Prediksi Foto Lapangan....74

Tabel 4.21 Perbandingan jumlah parameter dan Ukuran Model 79
Tabel 4.22 Jumlah parameter *trainable* pada ablasi MobileNetV2 80
Tabel 4.23 Indikator pelatihan MobileNetV2 *fine-tuning* 50 *layer* 81
Tabel 4.24 Perbandingan waktu inferensi 82

DAFTAR GAMBAR

Gambar 2.1 Arsitektur *Convolutional Neural Network* 8
Gambar 2.2 Blok arsitektur MobileNetV2 10
Gambar 3.1 Alur penelitian klasifikasi penyakit daun padi 22
Gambar 3.2 Bacterial Leaf Blight 24
Gambar 3.3 Brown Spot 24
Gambar 3.4 Healthy Rice Leaf 25
Gambar 3.5 Leaf Blast 25
Gambar 3.6 Leaf Scald 25
Gambar 3.7 Sheath Blight 25
Gambar 3.8 Tungro 25
Gambar 3.9 Alur arsitektur CNN Kustom 27
Gambar 3.10 Alur arsitektur MobileNetV2 33
Gambar 4.1 Bacterial Leaf Blight 47
Gambar 4.2 Brown Spot 47
Gambar 4.3 Healthy Rice Leaf 47
Gambar 4.4 Leaf Blast 47
Gambar 4.5 Leaf Scald 47
Gambar 4.6 Sheath Blight 47
Gambar 4.7 Tungro 48
Gambar 4.8 Grafik *training* dan *validation* CNN Kustom awal dataset original 57
Gambar 4.9 Grafik *training* dan *validation* CNN Kustom awal dataset *balanced* 57
Gambar 4.10 Grafik *training* dan *validation* CNN Kustom dengan *dropout* dataset original 58
Gambar 4.11 Grafik *training* dan *validation* CNN Kustom dengan *dropout* dataset *balanced quality-hybrid* 58
Gambar 4.12 Grafik *training* dan *validation* MobileNetV2 *feature extraction* dataset original 60
Gambar 4.13 Grafik *training* dan *validation* MobileNetV2 *fine-tuning* 50 layer dataset original 61
Gambar 4.14 Grafik *training* dan *validation* MobileNetV2 *feature extraction* dataset *balanced quality-hybrid* 61
Gambar 4.15 Grafik *training* dan *validation* MobileNetV2 *fine-tuning* 50 *layer* dataset *balanced quality-hybrid* 62

Gambar 4.16 *Confusion matrix* CNN Kustom dengan *dropout* dataset original.....70
Gambar 4.17 *Confusion Matrix MobileNetV2 fine-tuning 50 layer dataset original*.....70
Gambar 4.18 *Confusion Matrix CNN Kustom dengan dropout dataset balanced quality-hybrid* 71
Gambar 4.19 *Confusion Matrix MobileNetV2 fine-tuning 50 layer dataset balanced quality-hybrid* 71
Gambar 4.20 Hasil prediksi foto lapangan menggunakan CNN Kustom dan MobileNetV2 ..76

BAB I
PENDAHULUAN

1.1 Latar Belakang

Padi (*Oryza sativa L.)* merupakan salah satu komoditas pangan utama yang memiliki peran penting dalam memenuhi kebutuhan pangan masyarakat. Produktivitas tanaman padi perlu dijaga karena padi menjadi sumber makanan pokok bagi Sebagian besar masyarakat, khususnya di Indonesia. Dalam proses budidaya, tanaman padi dapat mengalami berbagai gangguan, salah satunya adalah serangan penyakit. Penyakit pada tanaman padi dapat menyerang bagian akar, batang, malai, maupun daun. Daun merupakan salah satu bagian tanaman yang mudah diamati karena gejala penyakit sering terlihat melalui perubahan warna, munculnya bercak, lesi, hingga bagian daun yang mengering.

Penyakit daun padi dapat menurunkan kualitas pertumbuhan tanaman dan berpotensi memengaruhi hasil panen apabila tidak dikenali sejak dini. Beberapa penyakit daun padi memiliki gejala visual yang berbeda, tetapi sebagian lainnya menunjukkan ciri yang cukup mirip. Gejala seperti bercak coklat, perubahan warna kekuningan, lesi memanjang, atau jaringan daun yang mengering dapat muncul pada beberapa jenis penyakit. Kondisi tersebut menyebabkan identifikasi secara manual menjadi tidak selalu mudah, terutama bagi petani atau pengguna yang belum memiliki pengalaman dalam membedakan gejala penyakit secara visual.

Identifikasi penyakit tanaman secara manual umumnya dilakukan dengan pengamatan langsung pada kondisi daun. Namun, metode manual memiliki keterbatasan karena hasilnya sangat bergantung pada pengetahuan dan pengalaman pengamat. Selain itu, faktor pencahayaan, sudut pengambilan gambar, kualitas citra, latar belakang, serta tingkat keparahan penyakit dapat memengaruhi proses gejala. (Deng et al., 2021) menjelaskan bahwa pendekatan berbasis citra dan *deep learning* dapat digunakan untuk membantu diagnosis penyakit padi secara otomatis. Hal ini menunjukkan bahwa teknologi pengolahan citra digital dapat menjadi alternatif untuk membantu proses identifikasi penyakit tanaman.

Perkembangan teknologi kecerdasan buatan, khususnya *deep learning,* memberikan peluang dalam proses klasifikasi citra digital. Salah satu metode yang banyak digunakan dalam klasifikasi citra adalah *Convolutional Neural Network* atau CNN. CNN memiliki kemampuan untuk mengekstraksi fitur visual dari gambar, seperti warna, tekstur, bentuk, tepi objek, serta pola tertentu yang muncul pada citra. Dalam kasus penyakit daun padi, CNN dapat digunakan

untuk mengenali pola visual yang muncul pada permukaan daun, seperti bercak, perubahan warna, serta bentuk lesi penyakit. (Costales et al., 2020) menunjukkan bahwa CNN dapat diterapkan untuk mendeteksi penyakit daun padi berdasarkan citra digital.

Selain CNN yang dibangun dari awal, pendekatan *transfer learning* juga banyak digunakan dalam klasifikasi citra. *Transfer learning* merupakan pendekatan yang memanfaatkan model yang telah dilatih sebelumnya pada dataset berskala besar, kemudian disesuaikan dengan kebutuhan klasifikasi tertentu. Dengan pendekatan ini, model tidak perlu belajar dari nol sepenuhnya karena sudah memiliki kemampuan awal dalam mengenali fitur umum pada citra, seperti garis, bentuk, warna, dan tekstur. Pendekatan *transfer learning* relevan digunakan dalam klasifikasi penyakit daun padi karena model dapat memanfaatkan fitur visual yang telah dipelajari sebelumnya untuk membantu proses klasifikasi pada dataset baru.

Salah satu arsitektur *transfer learning* yang banyak digunakan adalah MobileNetV2. MobileNetV2 merupakan arsitektur CNN yang dirancang agar ringan dan efisien secara komputasi. Arsitektur ini menggunakan konsep *depthwise separable convolution, inverted residual,* dan *linear bottleneck*, sehingga jumlah parameter dan kebutuhan komputasi dapat dibuat lebih kecil dibandingkan arsitektur CNN yang lebih besar (Sandler et al., 2019). Beberapa penelitian sebelumnya menunjukkan bahwa MobileNetV2 dapat digunakan dalam klasifikasi penyakit daun padi. (Costales et al., 2020)menggunakan MobileNetV2 berbasis *transfer learning* untuk deteksi penyakit daun padi, sedangkan (Saddami et al., 2024) membandingkan MobileNetV2 dengan beberapa model *lightweight* CNN lain untuk klasifikasi penyakit daun padi.

Pada penelitian ini digunakan dua model, yaitu CNN Kustom dan MobileNetV2. CNN Kustom digunakan sebagai model dasar atau *baseline* yang dibangun dan dilatih dari awal menggunakan dataset penelitian. Model ini digunakan untuk mengetahui kemampuan CNN sederhana dalam mengenali pola visual penyakit daun padi. Sementara itu, MobileNetV2 digunakan sebagai model pembanding berbasis *transfer learning*. MobileNetV2 dilatih melalui tahap *feature extraction* dan *fine-tuning* agar model dapat memanfaatkan bobot *pretrained* sekaligus menyesuaikan sebagian fiturnya dengan karakteristik citra daun padi.

Dataset yang digunakan dalam penelitian ini terdiri dari tujuh kondisi daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Dataset diperoleh dari dua sumber dataset publik. Dataset pertama digunakan untuk enam kelas, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf

Scald, dan Sheath Blight (M., 2021). Dataset kedua digunakan untuk menambahkan kelas Tungro (Jisan, 2023). Setelah dataset digabungkan, jumlah total citra yang digunakan pada dataset original adalah 5.137 gambar.

Namun, dataset original memiliki distribusi kelas yang tidak sepenuhnya seimbang. Kelas Tungro memiliki jumlah data sebanyak 1.308 gambar, sedangkan kelas lainnya hanya sekitar 600-an gambar. Jumlah data Tungro hampir dua kali lipat dibandingkan kelas lain. Kondisi tersebut disebut *class imbalance*, yaitu kondisi ketika jumlah data antar kelas tidak seimbang. (Johnson & Khoshgoftaar, 2019) menjelaskan bahwa data tidak seimbang dapat menjadi tantangan dalam penerapan *deep learning* karena model dapat cenderung bias terhadap kelas mayoritas.

Ketidakseimbangan data perlu diperhatikan karena dapat memengaruhi proses pelatihan dan interpretasi hasil evaluasi model. Kelas dengan jumlah data lebih banyak cenderung lebih sering dilihat oleh model selama proses pelatihan, sehingga model dapat lebih mudah mengenali kelas tersebut. Selain itu, nilai *accuracy* dan *weighted average* dapat terlihat lebih tinggi karena dipengaruhi oleh kelas yang jumlah datanya dominan. Oleh karena itu, evaluasi pada dataset yang tidak seimbang tidak cukup hanya menggunakan *accuracy*, tetapi perlu memperhatikan metrik lain seperti *precision* macro, *recall* macro, *F1-score macro, F1-score weighted*, *classification report*, dan *confusion matrix*.

Berdasarkan permasalahan tersebut, penelitian ini tidak hanya membandingkan CNN Kustom dan MobileNetV2, tetapi juga membandingkan performa kedua model pada dua skenario dataset, yaitu dataset original yang tidak seimbang dan dataset *balanced*. Pada dataset *balanced*, jumlah data kelas Tungro dikurangi menjadi 636 gambar agar mendekati jumlah kelas lainnya. Dengan adanya dua skenario tersebut, penelitian ini bertujuan untuk mengetahui performa CNN Kustom dan MobileNetV2 dalam klasifikasi penyakit daun padi, serta menganalisis apakah ketidakseimbangan data pada kelas Tungro berpengaruh terhadap hasil evaluasi model.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Bagaimana performa model CNN Kustom dalam mengklasifikasikan tujuh kondisi daun padi berdasarkan citra digital?

b. Bagaimana performa model MobileNetV2 berbasis *transfer learning* dalam mengklasifikasikan tujuh kondisi daun padi berdasarkan citra digital?
c. Bagaimana perbandingan performa CNN Kustom dan MobileNetV2 pada dataset original yang tidak seimbang?
d. Bagaimana perbandingan performa CNN Kustom dan MobileNetV2 pada dataset *balanced?*
e. Apakah ketidakseimbangan data pada kelas Tungro berpengaruh terhadap hasil evaluasi model klasifikasi penyakit daun padi?

1.3 Tujuan Penelitian

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

a. Mengetahui performa model CNN Kustom dalam mengklasifikasikan tujuh kondisi daun padi berdasarkan citra digital.
b. Mengetahui performa model MobileNetV2 berbasis *transfer learning* dalam mengklasifikasikan tujuh kondisi daun padi berdasarkan citra digital.
c. Membandingkan performa CNN Kustom dan MobileNetV2 pada dataset original yang tidak seimbang.
d. Membandingkan performa CNN Kustom dan MobileNetV2 pada dataset *balanced*.
e. Menganalisis pengaruh ketidakseimbangan data pada kelas Tungro terhadap hasil evaluasi model klasifikasi penyakit daun padi.

1.4 Batasan Masalah

Batasan Masalah dari penelitian ini adalah sebagai berikut:

a. Penelitian hanya menggunakan citra digital daun padi.
b. Kelas yang digunakan terdiri dari tujuh kondisi daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro.
c. Dataset yang digunakan berasal dari dataset publik.
d. Model yang dibandingkan hanya CNN Kustom dan MobileNetV2.
e. Penelitian menggunakan dua skenario dataset, yaitu original tidak seimbang dan dataset *balanced.*
f. Penyeimbangan dataset dilakukan dengan mengurangi jumlah data kelas Tungro agar mendekati jumlah kelas lain.

g. Evaluasi model dilakukan menggunakan *accuracy*, *precision macro*, *recall macro*, *F1-score macro*, *F1-score weighted*, *classification report*, dan *confusion matrix.*
h. Penelitian tidak membahas implementasi model ke dalam aplikasi berbasis *mobile* atau *web.*
i. Penelitian tidak membahas diagnosis penyakit secara biologis secara mendalam, tetapi berfokus pada klasifikasi citra menggunakan metode *deep learning.*

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi dalam bidang pengolahan citra digital dan *deep learning*, khususnya pada klasifikasi penyakit daun padi. Penelitian ini juga dapat menjadi referensi mengenai perbandingan CNN Kustom dan MobileNetV2 pada dataset tidak seimbang dan dataset seimbang. Selain itu, penelitian ini dapat memberikan gambaran mengenai pengaruh imbalance dataset terhadap interpretasi hasil evaluasi model klasifikasi.

1.5.2 Manfaat Praktis

Secara praktis, penelitian ini dapat memberikan gambaran mengenai penggunaan model *deep learning* untuk membantu proses identifikasi penyakit daun padi berdasarkan citra. Hasil penelitian ini juga dapat menjadi pertimbangan dalam memilih model klasifikasi citra yang sesuai, terutama ketika dataset yang digunakan memiliki distribusi kelas yang tidak seimbang.

1.6 Sistematika Penulisan

Sistematika penulisan laporan dalam lima bab, terdiri dari:

a. BAB I Pendahuluan
b. BAB II Landasan Teori
c. BAB III Metodologi Penelitian
d. BAB IV Hasil dan Pembahasan
e. BAB V Kesimpulan dan saran

BAB II
LANDASAN TEORI

2.1 Dasar Teori

2.1.1 Penyakit Daun Padi

Penyakit daun padi merupakan gangguan pada tanaman padi yang gejalanya dapat diamati melalui perubahan visual pada bagian daun. Gejala tersebut dapat berupa bercak, perubahan warna, lesi, bagian daun yang mengering, atau perubahan tekstur permukaan daun. Beberapa penyakit daun padi memiliki gejala yang cukup mirip, sehingga proses identifikasi secara manual dapat menjadi sulit apabila hanya mengandalkan pengamatan visual (Deng et al., 2021; Zhou et al., 2022).

Pada penelitian ini, kondisi daun padi yang diklasifikasikan terdiri dari tujuh kelas, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Kelas Healthy Rice Leaf menunjukkan kondisi daun padi sehat, sedangkan kelas lainnya menunjukkan kondisi daun padi yang mengalami gejala penyakit. (Deng et al., 2021) menjelaskan bahwa penggunaan citra digital dan *deep learning* dapat membantu proses diagnosis penyakit padi secara otomatis karena model dapat mempelajari pola visual penyakit dari citra daun.

2.1.2 Citra Digital

Citra digital merupakan representasi visual suatu objek dalam bentuk piksel. Setiap piksel memiliki nilai tertentu yang menunjukkan intensitas warna atau cahaya. Pada citra berwarna, piksel umumnya memiliki tiga kanal warna, yaitu *red*, *green*, dan *blue* atau RGB. Dalam penelitian klasifikasi penyakit daun padi, citra digital digunakan sebagai data masukan untuk model *deep learning* (Deng et al., 2021).

Citra daun padi mengandung informasi visual yang dapat dipelajari oleh model, seperti warna daun, tekstur, bercak, bentuk lesi, dan pola perubahan warna. Kualitas citra dapat memengaruhi hasil klasifikasi karena model belajar berdasarkan pola yang terdapat pada gambar. Faktor seperti pencahayaan, sudut pengambilan gambar, latar belakang, resolusi, dan variasi posisi daun dapat memengaruhi kemampuan model dalam mengenali kelas penyakit (Zhou et al., 2022).

(Zhou et al., 2022) menjelaskan bahwa konfigurasi dataset citra, termasuk jumlah data dan variasi citra, dapat memengaruhi akurasi model CNN dalam pengenalan penyakit padi. Oleh karena itu, pengelolaan dataset menjadi bagian penting dalam penelitian klasifikasi citra.

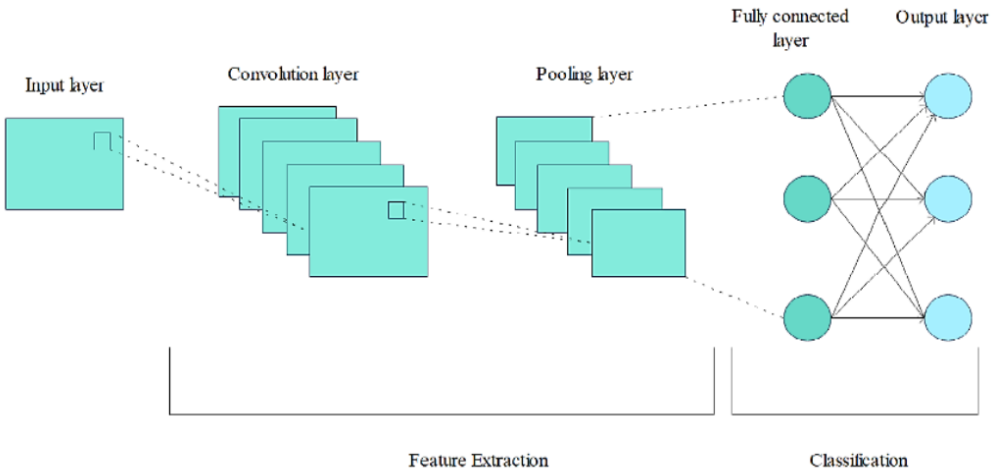
2.1.3 Convolutional Neural Network

Convolutional Neural Network atau CNN merupakan salah satu metode *deep learning* yang banyak digunakan untuk pengolahan citra. CNN dirancang untuk mempelajari fitur visual secara otomatis melalui beberapa lapisan utama, seperti *convolution layer, activation layer, pooling layer, flatten layer, fully connected layer,* dan *output layer* (Costales et al., 2020; Veeramreddy et al., 2024).

Convolution layer berfungsi untuk mengekstraksi fitur dari citra, seperti tepi, garis, warna, tekstur, bercak, dan bentuk tertentu. *Activation layer* digunakan untuk menambahkan fungsi non-linear agar model dapat mempelajari pola yang lebih kompleks. *Pooling layer* berfungsi untuk mengurangi ukuran fitur sehingga proses komputasi menjadi lebih efisien. Setelah fitur diperoleh, *flatten layer* mengubah hasil ekstraksi fitur menjadi bentuk satu dimensi. Selanjutnya, *fully connected layer* digunakan untuk proses klasifikasi sebelum menghasilkan prediksi kelas pada *output layer* (Veeramreddy et al., 2024).

Dalam klasifikasi penyakit daun padi, CNN dapat digunakan untuk mengenali pola visual yang muncul pada daun. (Costales et al., 2020) menunjukkan bahwa CNN dapat diterapkan untuk klasifikasi penyakit daun padi berbasis citra digital. Hal ini mendukung penggunaan CNN sebagai salah satu pendekatan utama dalam penelitian ini.

Arsitektur umum *Convolutional Neural Network* yang terdiri dari proses ekstraksi fitur hingga klasifikasi ditunjukkan pada Gambar 2.1.



Gambar 2.1 Arsitektur *Convolutional Neural Network*
Sumber: (Veeramreddy et al., 2024)

2.1.4 Convolutional Neural Network Kustom

CNN Kustom merupakan model CNN yang dirancang sendiri sesuai kebutuhan penelitian. Model ini dibangun dan dilatih dari awal menggunakan dataset penelitian tanpa menggunakan bobot *pretrained*. Dalam penelitian ini, CNN Kustom digunakan sebagai model dasar atau *baseline* untuk mengetahui kemampuan model yang belajar langsung dari dataset citra daun padi (Utami et al., 2025).

CNN Kustom penting digunakan karena dapat menjadi pembanding terhadap model *transfer learning*. Dengan adanya CNN Kustom, hasil MobileNetV2 tidak berdiri sendiri, tetapi dapat dibandingkan dengan model yang dilatih dari awal. Perbandingan ini membantu menjawab apakah pendekatan *transfer learning* mampu memberikan performa yang lebih baik dibandingkan model CNN dasar (Utami et al., 2025).

(Utami et al., 2025) melakukan perbandingan antara CNN dan beberapa model *transfer learning* untuk deteksi penyakit daun padi berbasis citra digital. Penelitian tersebut menunjukkan bahwa CNN dapat digunakan sebagai baseline dalam eksperimen klasifikasi citra, sedangkan model *transfer learning* digunakan sebagai pembanding untuk melihat performa.

Detail arsitektur CNN Kustom yang digunakan dalam penelitian ini tidak dijelaskan panjang pada BAB II karena bagian tersebut termasuk rancangan metode. Detail *layer* CNN Kustom dijelaskan pada BAB III.

2.1.5 Transfer learning

Transfer learning merupakan pendekatan pembelajaran mesin yang memanfaatkan pengetahuan dari model yang telah dilatih sebelumnya pada dataset besar. Pada klasifikasi citra, model *pretrained* umumnya telah mempelajari fitur dasar seperti warna, garis, tepi, tekstur, dan bentuk objek. Fitur tersebut kemudian dapat digunakan kembali untuk tugas klasifikasi baru dengan menyesuaikan bagian akhir model (Kornblith et al., 2019).

Transfer learning berguna ketika dataset penelitian tidak terlalu besar atau ketika model perlu memperoleh kemampuan ekstraksi fitur yang lebih baik tanpa belajar sepenuhnya dari awal. Pada penelitian ini, *transfer learning* diterapkan melalui model MobileNetV2. Model ini

digunakan untuk membandingkan performa pendekatan *pretrained* dengan CNN Kustom yang dilatih dari awal (Rahman et al., 2024).

(Seelwal & Rohilla, 2023) menunjukkan bahwa pendekatan *transfer learning* dapat digunakan untuk diagnosis penyakit padi berbasis citra. Pendekatan ini relevan karena model dapat memanfaatkan fitur visual yang telah dipelajari sebelumnya untuk membantu proses klasifikasi pada dataset penyakit daun padi.

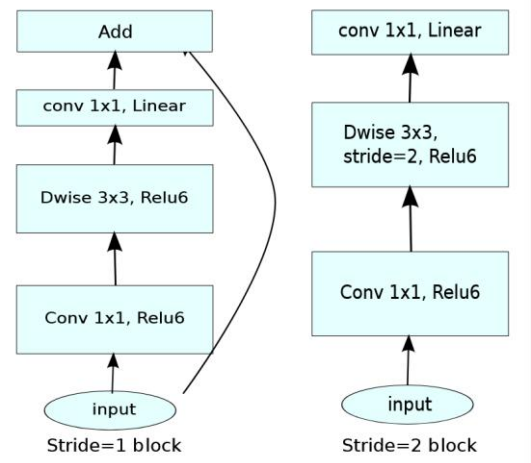
2.1.6 MobileNetV2

MobileNetV2 merupakan salah satu arsitektur CNN yang dirancang agar ringan dan efisien secara komputasi. Arsitektur ini menggunakan konsep *depthwise separable convolution, inverted residual,* dan *linear bottleneck*. Konsep tersebut membuat MobileNetV2 memiliki jumlah parameter yang lebih kecil dibandingkan beberapa arsitektur CNN besar, tetapi tetap mampu mengekstraksi fitur citra dengan baik (Sandler et al., 2019).

Dalam penelitian ini, MobileNetV2 digunakan sebagai model berbasis *transfer learning*. Bagian dasar MobileNetV2 digunakan sebagai *feature extractor*, sedangkan bagian akhir model disesuaikan dengan jumlah kelas penelitian, yaitu tujuh kelas. MobileNetV2 dipilih karena model ini ringan, efisien, dan telah banyak digunakan dalam klasifikasi citra (Liu, 2023; Sandler et al., 2019).

(Liu, 2023) menggunakan MobileNetV2 untuk klasifikasi penyakit padi berbasis model ringan. (Mustofa et al., 2026) juga menggunakan MobileNetV2 untuk klasifikasi multi-kelas penyakit daun padi berbasis *transfer learning*. Hal ini menunjukkan bahwa MobileNetV2 relevan digunakan dalam penelitian klasifikasi penyakit daun padi.

Blok arsitektur MobileNetV2 yang menggunakan konsep *inverted residual* dan *linear bottleneck* ditunjukkan pada Gambar 2.2.



Gambar 2.2 Blok arsitektur MobileNetV2
Sumber: (Sandler et al., 2019)

2.1.7 Imbalance Dataset

Imbalance dataset atau ketidakseimbangan data merupakan kondisi ketika jumlah data antar kelas tidak merata. Dalam klasifikasi, kondisi ini dapat menyebabkan model lebih banyak belajar dari kelas mayoritas dibandingkan kelas minoritas. Akibatnya, model dapat memiliki performa tinggi pada kelas yang jumlah datanya besar, tetapi belum tentu memiliki performa yang sama baik pada kelas lain (Johnson & Khoshgoftaar, 2019).

Pada penelitian ini, dataset original memiliki tujuh kelas. Kelas Tungro memiliki jumlah data sebanyak 1.308 gambar, sedangkan kelas lainnya hanya sekitar 600-an gambar. Jumlah data Tungro hampir dua kali lipat dibandingkan kelas lainnya. Kondisi tersebut menunjukkan adanya ketidakseimbangan data (Johnson & Khoshgoftaar, 2019).

(Johnson & Khoshgoftaar, 2019) menjelaskan bahwa class imbalance menjadi salah satu tantangan dalam *deep learning* karena model dapat cenderung bias terhadap kelas mayoritas. Oleh karena itu, penelitian ini membandingkan hasil model pada dataset original dan dataset *balanced*. Dataset *balanced* dibuat dengan mengurangi jumlah data kelas Tungro menjadi 653 gambar agar lebih sebanding dengan kelas lainnya.

2.1.8 Preprocessing

Preprocessing merupakan tahap persiapan data sebelum digunakan dalam proses pelatihan model. *Preprocessing* dilakukan agar citra memiliki format dan ukuran yang sesuai dengan kebutuhan model. Tahapan *preprocessing* dapat meliputi *resize* citra, normalisasi nilai piksel, pengubahan label ke bentuk numerik, dan pembagian dataset (Zhou et al., 2022).

Resize dilakukan agar seluruh gambar memiliki ukuran *input* yang sama. Normalisasi nilai piksel dilakukan agar nilai piksel yang semula berada pada rentang 0–255 diubah ke rentang tertentu, misalnya 0–1. Proses ini membantu model belajar dengan lebih stabil. Selain itu, label kelas juga perlu diubah ke bentuk numerik agar dapat diproses oleh model (Mao et al., 2023; Zhou et al., 2022).

(Shorten & Khoshgoftaar, 2019) menjelaskan bahwa pengolahan dan augmentasi citra menjadi bagian penting dalam pelatihan model *deep learning* karena dapat membantu model menghadapi variasi data citra.

2.1.9 Augmentasi Data

Augmentasi data merupakan teknik untuk menambah variasi data latih dengan melakukan transformasi pada citra. Teknik augmentasi yang umum digunakan antara lain rotasi, pergeseran, *zoom, flipping*, dan perubahan kecerahan. Tujuan augmentasi adalah agar model tidak hanya menghafal data *training*, tetapi mampu mengenali pola citra pada kondisi yang bervariasi (Shorten & Khoshgoftaar, 2019).

Dalam klasifikasi penyakit daun padi, augmentasi dapat membantu model menghadapi variasi gambar, seperti perbedaan pencahayaan, posisi daun, sudut pengambilan gambar, dan latar belakang. Namun, augmentasi perlu dilakukan secara wajar agar tidak mengubah ciri utama penyakit pada daun (Shorten & Khoshgoftaar, 2019; Zhou et al., 2022).

(Shorten & Khoshgoftaar, 2019) menyatakan bahwa augmentasi citra dapat membantu meningkatkan performa model *deep learning* dengan menyediakan variasi tambahan dari data yang sudah ada. Pada penelitian ini augmentasi diterapkan pada data *training,* sedangkan data *validation* dan *testing* tidak diberi augmentasi agar evalusi tetap objektif.

2.1.10 Metrik Evaluasi

Metrik evaluasi digunakan untuk menilai kemampuan model dalam melakukan klasifikasi citra daun padi. Pada penelitian ini, evaluasi dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-score*, *macro average*, *weighted average*, *classification report*, dan *confusion matrix*. Penggunaan beberapa metrik tersebut diperlukan agar performa model tidak hanya dilihat dari jumlah prediksi benar secara keseluruhan, tetapi juga dari kemampuan model dalam mengenali setiap kelas (Johnson & Khoshgoftaar, 2019).

Dataset original pada penelitian ini memiliki kondisi tidak seimbang karena kelas Tungro memiliki jumlah data lebih banyak dibandingkan kelas lainnya. Pada data tidak seimbang, penggunaan *accuracy* saja belum cukup karena hasil evaluasi dapat dipengaruhi oleh kelas mayoritas. Oleh karena itu, metrik seperti *precision*, *recall*, *F1-score*, *macro average*, dan *weighted average* digunakan untuk memperoleh gambaran performa model yang lebih rinci. (Johnson & Khoshgoftaar, 2019) menjelaskan bahwa *class imbalance* dapat membuat model cenderung bias terhadap kelas mayoritas, sehingga evaluasi model perlu memperhatikan metrik selain *accuracy* (Johnson & Khoshgoftaar, 2019).

Dalam perhitungan metrik evaluasi, terdapat empat istilah dasar yang digunakan, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). TP adalah kondisi ketika data dari suatu kelas diprediksi dengan benar sebagai kelas tersebut. FP

adalah kondisi ketika data dari kelas lain salah diprediksi sebagai kelas yang sedang dihitung. TN adalah kondisi ketika data yang bukan berasal dari suatu kelas juga tidak diprediksi sebagai kelas tersebut. FN adalah kondisi ketika data dari suatu kelas salah diprediksi sebagai kelas lain. Pada klasifikasi multi-kelas, perhitungan TP, FP, TN, dan FN dapat dilakukan untuk setiap kelas dengan pendekatan *one-vs-rest*, yaitu satu kelas dianggap sebagai kelas positif, sedangkan kelas lainnya dianggap sebagai kelas negatif. Penjelasan ini digunakan sebagai dasar dalam menghitung metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Pada dataset tidak seimbang, penggunaan beberapa metrik tersebut diperlukan agar evaluasi tidak hanya bergantung pada *accuracy* yang dapat dipengaruhi oleh kelas mayoritas (Johnson & Khoshgoftaar, 2019).

A. Accuracy

Accuracy merupakan metrik yang menunjukkan perbandingan antara jumlah prediksi benar dengan seluruh data yang diuji. Metrik ini digunakan untuk melihat performa model secara umum. Rumus *accuracy* ditunjukkan pada Persamaan 2.1.

.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} . 100\% \quad (2.1)$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

B. Precision

Precision digunakan untuk mengukur ketepatan model ketika memprediksi suatu kelas. Nilai *precision* menunjukkan seberapa banyak data yang diprediksi sebagai suatu kelas benar-benar berasal dari kelas tersebut. Rumus *precision* ditunjukkan pada Persamaan 2.2.

.

$$Precision = \frac{TP}{TP + FP} . 100\% \quad (2.1)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

C. Recall

Recall digunakan untuk mengukur kemampuan model dalam menemukan data yang sebenarnya berasal dari suatu kelas. Nilai *recall* menunjukkan seberapa banyak data dari suatu kelas yang berhasil dikenali dengan benar oleh model. Rumus *recall* ditunjukkan pada Persamaan 2.3.

$$Recall = \frac{TP}{TP + FN} . 100\% \quad (2.3)$$

Keterangan:

TP = *True Positive*

FN = *False Negative*

D. F1-*score*

F1-score merupakan metrik yang menggabungkan *precision* dan *recall* ke dalam satu nilai. Metrik ini digunakan untuk melihat keseimbangan antara ketepatan prediksi dan kemampuan model dalam mengenali data dari suatu kelas. Rumus *F1-score* ditunjukkan pada Persamaan 2.4.

.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} . 100\% \quad (2.4)$$

E. Macro average

Macro average merupakan rata-rata nilai metrik dari seluruh kelas dengan bobot yang sama. Dengan menggunakan *macro average*, setiap kelas memiliki pengaruh yang setara terhadap nilai akhir, baik kelas tersebut memiliki jumlah data besar maupun kecil. Metrik ini penting digunakan pada dataset tidak seimbang karena dapat menunjukkan performa rata-rata model pada setiap kelas secara lebih adil. Rumus *macro average* ditunjukkan pada Persamaan 2.5.

.

$$Macro\ average = \frac{1}{N} \sum_{i=1}^{N} M_i . 100\% \quad (2.5)$$

Keterangan:

N = jumlah kelas

M_i = nilai metrik pada kelas ke-i

F. Weighted average

Weighted average merupakan rata-rata nilai metrik yang dihitung dengan mempertimbangkan jumlah data pada setiap kelas. Kelas dengan jumlah data lebih banyak akan memberikan pengaruh lebih besar terhadap nilai akhir. Pada penelitian ini, *weighted average* digunakan untuk melihat performa model dengan tetap memperhatikan distribusi jumlah data pada setiap kelas. Rumus *weighted average* ditunjukkan pada Persamaan 2.6.

.

$$Weighted\ average = \frac{\sum_{i=1}^{N} w_{iM_i}}{\sum_{i=1}^{N} w_i}.100\% \quad (2.6)$$

Keterangan:

N = jumlah kelas

w_i = jumlah data pada kelas ke-i

M_i = nilai metrik pada kelas ke-i

2.2 Penelitian Sebelumnya

Penelitian mengenai klasifikasi penyakit daun padi berbasis citra digital telah banyak dilakukan dengan memanfaatkan metode *deep learning*, khususnya *Convolutional Neural Network* (CNN) dan *transfer learning*. Metode tersebut digunakan karena mampu mempelajari pola visual pada citra daun, seperti warna, tekstur, bercak, tepi daun, dan bentuk lesi secara otomatis. Beberapa penelitian sebelumnya menunjukkan bahwa CNN dapat digunakan untuk mendeteksi penyakit tanaman secara efektif, sedangkan *transfer learning* dapat membantu meningkatkan performa model ketika dataset yang digunakan tidak terlalu besar.

(Deng et al., 2021) melakukan penelitian mengenai diagnosis otomatis penyakit padi menggunakan *deep learning*. Penelitian tersebut menunjukkan bahwa *deep learning* dapat digunakan untuk mengenali penyakit padi berdasarkan citra visual. Penelitian ini menjadi salah satu dasar bahwa citra daun padi dapat diproses menggunakan model *deep learning* untuk membantu proses identifikasi penyakit secara otomatis.

(Zhou et al., 2022) meneliti pengaruh konfigurasi dataset terhadap akurasi pengenalan penyakit padi berbasis CNN. Penelitian tersebut menunjukkan bahwa konfigurasi dataset, termasuk jumlah data, variasi citra, dan pembagian data, dapat memengaruhi performa model. Hal ini relevan dengan penelitian ini karena dataset yang digunakan memiliki kondisi tidak seimbang, terutama pada kelas Tungro yang jumlah datanya lebih banyak dibandingkan kelas lain.

(Costales et al., 2020) mengembangkan *prototipe* aplikasi deteksi penyakit padi menggunakan CNN dan *transfer learning*. Penelitian tersebut menggunakan MobileNetV2 sebagai base model dengan bobot *pretrained* ImageNet. Hasil penelitiannya menunjukkan bahwa *transfer learning*, augmentasi data, dan *fine-tuning* dapat meningkatkan performa model pada klasifikasi penyakit daun padi. Penelitian tersebut menjadi salah satu acuan penggunaan MobileNetV2 dalam penelitian ini.

(Saddami et al., 2024) membandingkan beberapa model *lightweight* CNN, yaitu EfficientNet-B0, MobileNetV2, dan ShuffleNet untuk klasifikasi penyakit daun padi. Penelitian tersebut menekankan pentingnya model yang ringan dan efisien, terutama jika model akan diterapkan pada perangkat dengan keterbatasan komputasi. Hasil penelitian tersebut menunjukkan bahwa model *pretrained* dapat digunakan untuk klasifikasi penyakit daun padi, tetapi performanya dapat berbeda tergantung arsitektur, konfigurasi pelatihan, dan karakteristik dataset.

(Miftahuddin et al., 2025) membandingkan arsitektur MobileNetV2 dan MobileNetV3 dalam klasifikasi penyakit daun padi menggunakan pendekatan *transfer learning*. Penelitian tersebut menggunakan dataset citra daun padi yang terdiri dari 4.684 gambar dan membagi data dengan rasio 80:20 untuk *training* dan *validation*. Tahap *preprocessing* dilakukan dengan mengubah ukuran citra menjadi 224 x 224 piksel, normalisasi, dan augmentasi data. Hasil penelitian menunjukkan bahwa MobileNetV2 memperoleh *accuracy* sebesar 96%, sedangkan MobileNetV3 memperoleh *accuracy* sebesar 99%. Penelitian tersebut relevan dengan penelitian ini karena sama-sama menggunakan MobileNetV2 pada klasifikasi penyakit daun padi, meskipun penelitian ini membandingkan MobileNetV2 dengan CNN Kustom serta menambahkan analisis dataset original dan dataset *balanced*.

(Paemo et al., 2025) membandingkan MobileNetV2 dan EfficientNetB0 untuk klasifikasi penyakit daun padi berbasis *transfer learning*. Penelitian tersebut menunjukkan bahwa *transfer learning* dapat digunakan untuk membangun model klasifikasi citra daun padi dengan performa yang baik. Namun, penelitian tersebut lebih menekankan perbandingan antar model *pretrained*, sedangkan penelitian ini membandingkan model CNN Kustom yang dilatih dari awal dengan MobileNetV2 berbasis *transfer learning*.

Selain penelitian yang berfokus pada model, beberapa referensi juga membahas aspek teknis yang digunakan dalam penelitian ini. (Johnson & Khoshgoftaar, 2019) menjelaskan bahwa *imbalance* dataset dapat menjadi tantangan dalam *deep learning* karena model cenderung lebih mudah mengenali kelas mayoritas. (Shorten & Khoshgoftaar, 2019)

menjelaskan bahwa augmentasi citra dapat membantu meningkatkan variasi data *training*. (Srivastava et al., 2014) memperkenalkan *dropout* sebagai teknik regularisasi untuk mengurangi *overfitting*. Referensi tersebut relevan dengan penelitian ini karena eksperimen yang dilakukan tidak hanya membandingkan model, tetapi juga membahas pengaruh dataset original dan dataset *balanced* serta penggunaan *Dropout* pada CNN Kustom.

Ringkasan penelitian sebelumnya yang relevan dengan penelitian ini ditunjukkan pada Tabel 2.1.

Tabel 2.1 Penelitian Sebelumnya

No	**Peneliti**	**Topik Penelitian**	**Metode**	**Hasil/Temuan Utama**	**Relevansi dengan Penelitian ini**
1	(Deng et al., 2021)	Diagnosis otomatis penyakit padi	*Deep learning*	Konfigurasi dataset memengaruhi akurasi model	Menjadi dasar bahwa penyakit padi dapat diklasifikasikan menggunakan citra digital dan *deep learning*
2	(Zhou et al., 2022)	Pengaruh konfigurasi dataset pada pengenalan penyakit padi	CNN	Konfigurasi dataset memengaruhi akurasi model	Mendukung pembahasan dataset original dan dataset *balanced*
3	(Costales et al., 2020)	Aplikasi deteksi penyakit padi	CNN dan MobileNetV2	*Transfer learning* dan augmentasi membantu meningkatkan performa model	Mendukung penggunaan MobileNetV2 dan *fine-tuning*
4	(Saddami et al., 2024)	*Lightweight* CNN untuk penyakit daun padi	EfficientNet-B0, MobileNetV2, ShuffleNet	Model *lightweight* dapat digunakan untuk klasifikasi penyakit daun padi dengan	Mendukung pemilihan MobileNetV2 sebagai model ringan berbasis *transfer learning*

				efisiensi komputasi	
5	(Miftahuddin et al., 2025)	Perbandingan MobileNetV2 dan MobileNetV3	*Transfer learning*	*Transfer learning* | MobileNetV3 memperoleh *accuracy* 99%, sedangkan MobileNetV2 memperoleh *accuracy* 96%	Mendukung penggunaan MobileNetV2 sebagai model pembanding berbasis *transfer learning*
6	(Paemo et al., 2025)	Perbandingan MobileNetV2 dan EfficientNetB0	*Transfer learning*	Model *pretrained* dapat digunakan untuk klasifikasi penyakit daun padi	Menjadi pembanding penelitian yang sama-sama menggunakan MobileNetV2
7	(Johnson & Khoshgoftaar, 2019)	*Imbalance* dataset pada *deep learning*	*Literature review*	Data tidak seimbang dapat membuat model bias terhadap kelas mayoritas	Mendukung analisis pengaruh *imbalance* dataset
8	(Shorten & Khoshgoftaar, 2019)	Augmentasi citra untuk *deep learning*	*Literature review*	Augmentasi dapat meningkatkan variasi data *training*	Mendukung penggunaan RandomFlip dan RandomRotation
9	(Srivastava et al., 2014)	*Dropout* sebagai regularisasi	*Dropout*	*Dropout* dapat mengurangi *overfitting* pada neural network	Mendukung penambahan *dropout* pada CNN Kustom
10	(Sandler et al., 2019)	Arsitektur MobileNetV2	MobileNetV2	MobileNetV2 menggunakan *inverted residual* dan	Menjadi dasar teknis penggunaan MobileNetV2

				linear bottleneck untuk efisiensi komputasi	

Berdasarkan Tabel 2.1, dapat diketahui bahwa CNN dan *transfer learning* telah banyak digunakan dalam klasifikasi penyakit daun padi. Namun, masih terdapat ruang penelitian untuk membandingkan CNN Kustom yang dilatih dari awal dengan MobileNetV2 berbasis *transfer learning* pada dataset yang sama. Selain itu, penelitian ini juga membahas pengaruh *imbalance* dataset dengan membandingkan dataset original dan dataset *balanced*. Dengan demikian, penelitian ini tidak hanya melihat model mana yang memiliki akurasi lebih tinggi, tetapi juga menganalisis bagaimana distribusi data memengaruhi performa model pada setiap kelas.

Gap penelitian yang menjadi dasar penelitian ini adalah sebagai berikut:

a. Sebagian penelitian sebelumnya lebih banyak berfokus pada peningkatan akurasi model, tetapi belum selalu membahas pengaruh distribusi data antar kelas.
b. Beberapa penelitian menggunakan model *transfer learning*, tetapi tidak selalu membandingkannya dengan CNN Kustom yang dilatih dari awal pada dataset yang sama.
c. Penelitian terkait penyakit daun padi sering menggunakan jumlah kelas yang berbeda, sehingga diperlukan eksperimen yang disesuaikan dengan dataset dan kelas yang digunakan dalam penelitian ini.
d. Analisis performa perlu dilakukan tidak hanya berdasarkan *accuracy*, tetapi juga berdasarkan *precision*, *recall*, *F1-score*, *classification report*, dan *confusion matrix* agar terlihat kelas mana yang sulit diprediksi.

2.3 Pemilihan Metode

Metode yang digunakan dalam penelitian ini dipilih berdasarkan tujuan utama penelitian, yaitu membandingkan performa CNN Kustom dan MobileNetV2 dalam klasifikasi penyakit daun padi serta menganalisis pengaruh *imbalance* dataset. Oleh karena itu, penelitian ini menggunakan dua pendekatan model, yaitu CNN Kustom sebagai model baseline dan MobileNetV2 sebagai model berbasis *transfer learning*.

CNN Kustom dipilih karena model ini dibangun dan dilatih dari awal menggunakan dataset penelitian. Dengan menggunakan CNN Kustom, penelitian dapat mengetahui kemampuan model yang belajar langsung dari citra daun padi tanpa menggunakan bobot

pretrained. Model ini digunakan sebagai *baseline* untuk melihat sejauh mana arsitektur CNN sederhana mampu mengenali tujuh kelas penyakit daun padi.

MobileNetV2 dipilih sebagai model pembanding karena memiliki arsitektur yang ringan dan efisien. MobileNetV2 menggunakan konsep *depthwise separable convolution, inverted residual,* dan *linear bottleneck* sehingga dapat mengekstraksi fitur citra dengan jumlah parameter yang relatif lebih kecil dibandingkan beberapa arsitektur CNN besar. Selain itu, MobileNetV2 dapat menggunakan bobot *pretrained* ImageNet, sehingga model tidak belajar sepenuhnya dari awal. Pada penelitian ini, MobileNetV2 dilatih melalui dua tahap, yaitu *feature extraction* dan *fine-tuning*.

Feature extraction digunakan pada tahap awal pelatihan MobileNetV2 dengan membekukan base model. Pada tahap ini, *layer* tambahan di bagian akhir model dilatih untuk menyesuaikan fitur yang diekstraksi oleh MobileNetV2 dengan jumlah kelas penyakit daun padi. Setelah itu, *fine-tuning* dilakukan dengan membuka sebagian *layer* akhir MobileNetV2 agar bobot *pretrained* dapat menyesuaikan diri dengan karakteristik citra daun padi. Strategi ini dipilih karena citra daun padi memiliki karakteristik visual yang berbeda dari dataset ImageNet.

Selain membandingkan dua model, penelitian ini juga menggunakan dua skenario dataset, yaitu dataset original dan dataset *balanced*. Dataset original digunakan untuk melihat performa model pada kondisi data awal, yaitu ketika kelas Tungro memiliki jumlah data lebih banyak dibandingkan kelas lainnya. Dataset *balanced* digunakan untuk melihat apakah penyetaraan jumlah data kelas Tungro dapat membuat performa model menjadi lebih stabil. Perbandingan dua skenario dataset ini penting karena *imbalance* dataset dapat memengaruhi hasil evaluasi model, terutama pada nilai *recall* dan *F1-score* setiap kelas.

Pada CNN Kustom, penelitian juga melakukan percobaan dengan menambahkan *dropout. Dropout* dipilih karena model CNN yang dilatih dari awal berisiko mengalami *overfitting*, terutama ketika model terlalu menyesuaikan diri terhadap data *training*. Dengan menambahkan *Dropout*, beberapa neuron dinonaktifkan secara acak selama proses pelatihan sehingga model diharapkan dapat belajar pola yang lebih umum dan tidak hanya menghafal data *training*.

Preprocessing dilakukan dengan mengubah ukuran citra menjadi 256 x 256 piksel dan menjaga tiga kanal warna RGB, sehingga *input* model berukuran 256 x 256 x 3. Augmentasi data dilakukan menggunakan RandomFlip dan RandomRotation pada data *training*. Augmentasi digunakan untuk menambah variasi citra latih agar model lebih tahan terhadap

perubahan posisi daun dan orientasi citra. Data *validation* dan *testing* tidak diberi augmentasi agar evaluasi tetap objektif.

Evaluasi model dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-score*, macro average, weighted average, *classification report*, dan *confusion matrix*. Penggunaan metrik tersebut dipilih karena penelitian ini melibatkan dataset dengan kondisi tidak seimbang. *Accuracy* digunakan untuk melihat performa umum model, sedangkan *precision*, *recall*, *F1-score,* dan *macro average* digunakan untuk melihat performa model pada setiap kelas secara lebih adil.

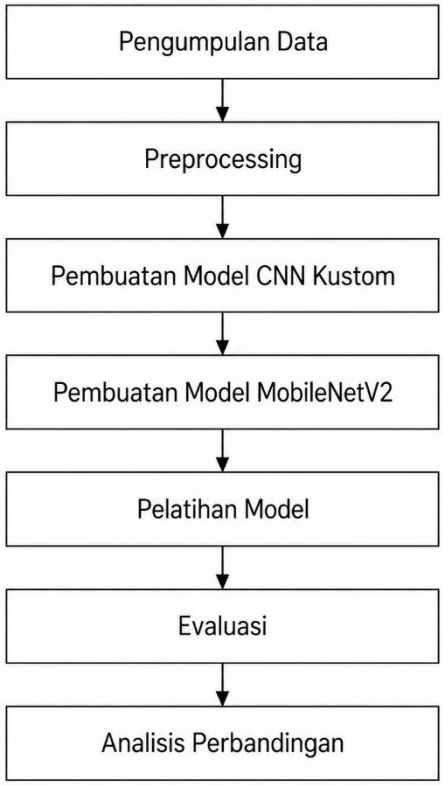
BAB III
METODOLOGI PENELITIAN

3.1 Langkah-langkah penelitian

Gambar 3.1 menunjukkan alur yang dilakukan dalam penelitian ini. Langkah-langkah penelitian terdiri dari pengumpulan data, *preprocessing*, pembuatan model CNN Kustom, pembuatan model MobileNetV2, pelatihan model, evaluasi, dan analisis perbandingan. Tahapan tersebut disusun untuk membandingkan performa CNN Kustom dan MobileNetV2 dalam klasifikasi penyakit daun padi berdasarkan citra digital.

Penelitian ini menggunakan dua skenario dataset, yaitu dataset original dan dataset *balanced*. Dataset original digunakan untuk melihat performa model pada kondisi data yang tidak seimbang, sedangkan dataset *balanced* digunakan untuk melihat perubahan performa model setelah jumlah data kelas Tungro diseimbangkan. Dengan adanya dua skenario tersebut, penelitian ini tidak hanya membandingkan performa dua model, tetapi juga menganalisis pengaruh ketidakseimbangan data terhadap hasil evaluasi model.

Alur penelitian klasifikasi penyakit daun padi menggunakan CNN Kustom dan MobilrNetV2 ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur penelitian klasifikasi penyakit daun padi

3.1.1 Pengumpulan Data

Pada penelitian ini, sumber data utama yang digunakan merujuk pada dua dataset citra daun padi yang tersedia pada platform Kaggle. Dataset pertama adalah *Rice Disease Dataset* yang dapat diakses melalui https://www.kaggle.com/datasets/anshulm257/rice-disease-dataset. Dataset tersebut digunakan untuk memperoleh enam kelas citra daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, dan Sheath Blight. Dataset kedua adalah *Rice Leaf Dataset from Mendeley Data* yang dapat diakses melalui https://www.kaggle.com/datasets/maimunulkjisan/rice-leaf-dataset-from-mendeley-data/data. Pada penelitian ini, dataset kedua digunakan khusus untuk menambahkan kelas Tungro.

Dataset tersebut dipilih karena memuat citra daun padi yang sesuai dengan konteks penelitian, yaitu klasifikasi penyakit daun padi berdasarkan citra digital. Citra pada dataset memperlihatkan variasi visual daun padi, seperti bercak, perubahan warna, pola lesi, dan kondisi daun sehat. Variasi tersebut diperlukan agar model dapat mempelajari perbedaan pola visual pada setiap kelas penyakit.

Setelah kedua dataset digabungkan, jumlah data pada dataset original adalah 5.137 gambar. Dataset tersebut terdiri dari tujuh kelas, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Distribusi jumlah citra pada dataset original ditunjukkan pada Tabel 3.1.

Tabel 3.1 Distribusi dataset original

Kelas	Jumlah Gambar
Bacterial Leaf Blight	636
Brown Spot	646
Healthy Rice Leaf	653
Leaf Blast	634
Leaf Scald	628
Sheath Blight	631
Tungro	1308
Total	5137

Berdasarkan Tabel 3.1, kelas Tungro memiliki jumlah citra paling banyak, yaitu 1.308 gambar. Jumlah tersebut hampir dua kali lipat dibandingkan kelas lainnya yang berada pada kisaran 628 sampai 653 gambar. Kondisi ini menunjukkan adanya ketidakseimbangan data atau *imbalance* dataset. Ketidakseimbangan data perlu diperhatikan karena model berpotensi lebih dominan mempelajari kelas dengan jumlah data lebih banyak.

Berdasarkan kondisi tersebut, penelitian ini menggunakan dua skenario dataset, yaitu dataset original dan dataset *balanced*. Dataset original digunakan untuk melihat performa model pada kondisi data awal yang tidak seimbang. Sementara itu, dataset *balanced* dibuat dari dataset original untuk melihat pengaruh distribusi data yang lebih seimbang terhadap hasil evaluasi model.

Pembuatan dataset *balanced* dilakukan menggunakan pendekatan *hybrid* pada level data, yaitu *quality-based filtering* dan augmentasi kelas minoritas. *Quality-based filtering* digunakan pada kelas mayoritas Tungro dengan mempertahankan 653 citra yang memiliki kualitas visual lebih baik. Kualitas citra dinilai berdasarkan ketajaman, kontras, kecerahan, dan ukuran citra. Dengan demikian, citra Tungro yang kurang jelas, blur, atau memiliki kualitas visual lebih rendah tidak digunakan pada dataset *balanced*.

Untuk kelas yang jumlah datanya lebih sedikit dari target, dilakukan augmentasi citra. Augmentasi digunakan agar jumlah data setiap kelas menjadi sama tanpa harus menghapus variasi citra asli pada kelas minoritas. Target jumlah citra pada dataset *balanced* adalah 653 citra untuk setiap kelas.

Distribusi jumlah citra pada dataset *balanced* ditunjukkan pada Tabel 3.2.

Tabel 3.2 Distribusi dataset *balance*

Kelas	Data Awal	Citra Asli Dipakai	Citra dikeluarkan	Citra Augmentasi	Total Akhir
Bacterial Leaf Blight	636	636	0	17	653
Brown Spot	646	646	0	7	653
Healthy Rice Leaf	653	653	0	0	653
Leaf Blast	634	634	0	19	653
Leaf Scald	628	628	0	25	653
Sheath Blight	632	632	0	21	653
Tungro	636	653	655	0	653
Total	4465	4482	655	89	4571

Berdasarkan Tabel 3.2 kelas Tungro dikurangi dari 1.308 citra menjadi 653 citra dengan mempertahankan citra yang memiliki kualitas visual lebih baik. Sementara itu, kelas minoritas ditambah menggunakan augmentasi hingga masing-masing kelas memiliki 653 citra. Dengan demikian, dataset *balanced* tidak hanya dibuat untuk menyamakan jumlah data antar kelas, tetapi juga mempertimbangkan kualitas citra yang digunakan.

Sementara itu, dataset *balanced* dibuat dengan mengurangi jumlah data kelas Tungro menjadi 636 gambar agar distribusinya lebih mendekati kelas lainnya. Tahap penyusunan dataset dilakukan dengan cara:

a. Mengunduh dataset citra daun padi dari dua sumber dataset publik pada platform Kaggle.
b. Mengambil enam kelas dari dataset pertama, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, dan Sheath Blight.
c. Mengambil kelas Tungro dari dataset kedua sebagai tambahan kelas penelitian.
d. Menggabungkan seluruh citra ke dalam struktur folder berdasarkan nama kelas.
e. Menghitung jumlah citra pada setiap kelas untuk mengetahui distribusi dataset original.
f. Membuat dataset *balanced* dengan mengurangi jumlah citra kelas Tungro menjadi 636 gambar.
g. Memastikan setiap folder kelas memiliki label yang sesuai agar dapat digunakan dalam proses pelatihan dan pengujian model.

Secara konseptual, penggunaan dataset citra daun padi yang sesuai dengan domain penelitian membantu model mempelajari variasi visual penyakit yang muncul pada daun. Variasi seperti warna daun, bentuk bercak, ukuran lesi, dan kondisi pencahayaan dapat memengaruhi proses pembelajaran model. Selain itu, distribusi jumlah data pada setiap kelas juga perlu diperhatikan karena ketidakseimbangan data dapat membuat model cenderung lebih mudah mengenali kelas yang jumlah datanya lebih banyak (Johnson & Khoshgoftaar, 2019).

Contoh citra pada masing-masing kelas daun padi dalam dataset penelitian ditunjukkan pada Gambar 3.2 sampai Gambar 3.8. Gambar 3.2 menunjukkan contoh citra kelas Bacterial Leaf Blight, Gambar 3.3 menunjukkan kelas Brown Spot, Gambar 3.4 menunjukkan kelas Healthy Rice Leaf, Gambar 3.5 menunjukkan kelas Leaf Blast, Gambar 3.6 menunjukkan kelas Leaf Scald, Gambar 3.7 menunjukkan kelas Sheath Blight, dan Gambar 3.8 menunjukkan kelas Tungro.



Gambar 3.2 Bacterial Leaf Blight



Gambar 3.3 Brown Spot



Gambar 3.4 Healthy Rice Leaf



Gambar 3.5 Leaf Blast



Gambar 3.6 Leaf Scald



Gambar 3.7 Sheath Blight



Gambar 3.8 Tungro

3.1.2 *Preprocessing*

Tahap *preprocessing* dilakukan untuk menyiapkan citra agar dapat digunakan sebagai *input* model. Proses ini diperlukan karena model *deep learning* membutuhkan data dengan format dan ukuran yang konsisten. *Preprocessing* dilakukan pada dataset original dan dataset *balanced* agar kedua skenario memiliki tahapan pengolahan data yang sama.

Tahapan *preprocessing* yang dilakukan meliputi penyamaan ukuran gambar, normalisasi nilai piksel, dan penyesuaian label kelas. Penyamaan ukuran gambar dilakukan agar seluruh citra memiliki ukuran *input* yang sama sesuai kebutuhan model. Normalisasi dilakukan dengan mengubah nilai piksel ke rentang tertentu, misalnya dari 0–255 menjadi 0–1, agar proses pelatihan model menjadi lebih stabil.

Selain itu, label kelas diubah ke dalam bentuk numerik agar dapat diproses oleh model. Setelah *preprocessing* selesai, dataset dibagi menjadi data *training*, *validation*, dan *testing*. Data *training* digunakan untuk melatih model, data *validation* digunakan untuk memantau performa model selama pelatihan, sedangkan data *testing* digunakan untuk mengukur performa akhir model.

3.1.3 Pembuatan Model CNN Kustom

Tahap pembuatan model CNN Kustom dilakukan untuk membangun model dasar atau *baseline* pada penelitian ini. Model ini disusun dari awal tanpa menggunakan bobot *pretrained*, sehingga seluruh proses pembelajaran dilakukan langsung dari dataset citra daun padi yang digunakan. Penggunaan CNN Kustom bertujuan untuk mengetahui kemampuan model CNN yang dibangun secara mandiri dalam mengenali pola visual penyakit daun padi, seperti warna daun, bercak, tekstur, tepi daun, dan bentuk lesi. CNN dipilih karena arsitektur ini telah banyak digunakan pada tugas klasifikasi citra dan terbukti mampu mengekstraksi fitur visual secara bertahap dari lapisan awal hingga lapisan akhir.

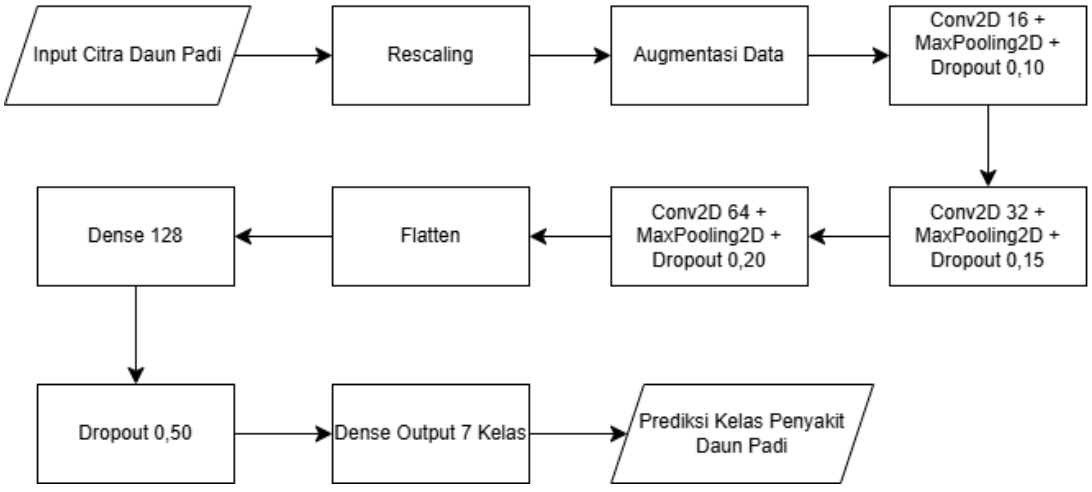
Pada penelitian ini, arsitektur CNN Kustom terdiri atas beberapa tahapan utama, yaitu *input layer*, *rescaling*, augmentasi data, blok konvolusi, *max pooling*, *dropout*, *flatten*, *dense layer*, dan *output layer*. Citra daun padi terlebih dahulu masuk ke *input layer* dengan ukuran 256 × 256 × 3. Selanjutnya dilakukan *rescaling* untuk mengubah rentang nilai piksel dari 0–255 menjadi 0–1 agar proses pelatihan lebih stabil. Setelah itu, augmentasi data diterapkan pada data *training* menggunakan *RandomFlip* dan *RandomRotation* untuk menambah variasi citra latih serta membantu model mengenali perubahan orientasi dan posisi objek. Penggunaan

augmentasi citra dinilai penting untuk meningkatkan keragaman data dan mengurangi risiko *overfitting* selama pelatihan model.

Setelah tahap awal tersebut, citra diproses melalui tiga blok konvolusi. Blok pertama terdiri atas *Conv2D* dengan 16 filter berukuran 3 × 3, *MaxPooling2D*, dan *Dropout* 0,10. Blok kedua terdiri atas *Conv2D* dengan 32 filter berukuran 3 × 3, *MaxPooling2D*, dan *Dropout* 0,15. Blok ketiga terdiri atas *Conv2D* dengan 64 filter berukuran 3 × 3, *MaxPooling2D*, dan *Dropout* 0,20. Susunan bertahap ini digunakan agar model dapat mempelajari fitur visual dari tingkat sederhana hingga lebih kompleks. Penambahan *dropout* pada tiap blok dilakukan untuk mengurangi kemungkinan model terlalu menyesuaikan diri terhadap data latih. Teknik *dropout* sendiri dikenal efektif untuk membantu mengurangi *overfitting* pada jaringan saraf.

Fitur hasil ekstraksi kemudian diubah menjadi vektor satu dimensi melalui *flatten layer*. Selanjutnya, fitur diproses oleh *dense layer* dengan 128 neuron untuk mempelajari hubungan antarfitur sebelum dilakukan klasifikasi akhir. Sebelum menuju *output layer*, diterapkan kembali *dropout* sebesar 0,50 untuk memperkuat regularisasi pada tahap klasifikasi. Pada bagian akhir, *output layer* terdiri atas 7 neuron yang disesuaikan dengan jumlah kelas pada penelitian ini, yaitu *Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight*, dan *Tungro*.

Alur arsitektur CNN Kustom yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3.9.



Gambar 3.9 Alur arsitektur CNN Kustom

Setelah alur arsitektur ditentukan, konfigurasi *layer* CNN Kustom disusun sesuai kebutuhan klasifikasi tujuh kelas penyakit daun padi. Konfigurasi tersebut ditunjukkan pada Tabel 3.3.

Tabel 3.3 Konfigurasi Arsitektur CNN Kustom

Tahap	**Konfigurasi**
Input	256 × 256 × 3
Rescaling	1/255
Augmentasi	RandomFlip(horizontal_and_vertical), RandomRotation(0.2)
Convo2D	16 filter, karnel 3 × 3, padding same, aktivasi ReLU
MaxPooling2D	Pooling 2 × 2
Dropout	0,10
Convo2D	32 filter, karnel 3 × 3, padding same, aktivasi ReLU
MaxPooling2D	Pooling 2 × 2
Dropout	0,15
Convo2D	64 filter, karnel 3 × 3, padding same, aktivasi ReLU
MaxPooling2D	Pooling 2 × 2
Dropout	0,20
Flatten	Mengubah *feature map* menjadi vector satu dimensi
Dense	128 neuron, aktivasi ReLU
Dropout	0,50
Output	7 neuron

Berdasarkan Tabel 3.3, setelah konfigurasi *layer* ditetapkan, proses pelatihan model CNN Kustom dilakukan menggunakan beberapa hyperparameter yang ditentukan berdasarkan rujukan penelitian terdahulu dan hasil percobaan (*trial and error*) penulis. Pada tahap awal, penulis melakukan percobaan pipeline menggunakan 20 epoch untuk memastikan proses mulai dari pemuatan dataset, *preprocessing*, pelatihan, hingga evaluasi dapat berjalan dengan baik. Setelah itu, epoch maksimum dinaikkan menjadi 100 sesuai arahan dosen, kemudian diterapkan *early stopping* agar model tetap dapat dihentikan ketika performa validasi tidak lagi mengalami perbaikan. Selain itu, *dropout* pada blok konvolusi dan sebelum lapisan *output* juga dipertahankan karena pada percobaan penulis strategi tersebut membantu mengurangi kecenderungan *overfitting*.

Konfigurasi hyperparameter pelatihan CNN Kustom ditunjukkan pada Tabel 3.4.

Tabel 3.4 Hyperparameter Pelatihan CNN Kustom

Hyperparameter	**Nilai**	**Dasar Penentuan**
Optimizer	Adam	Adam digunakan karena mampu melakukan pembaruan bobot secara adaptif dan umum diterapkan pada pelatihan model *deep learning* (Kingma & Ba, 2017). Pada percobaan awal pipeline

		20 epoch, optimizer Adam dapat menjalankan proses pelatihan CNN Kustom dari tahap *training* sampai evaluasi dengan stabil.
Learning rate	0.0003	Nilai *learning rate* 0.0003 digunakan berdasarkan hasil trial and error penulis agar proses pelatihan CNN Kustom berjalan stabil. *Learning rate* kecil dipilih untuk mengurangi perubahan bobot yang terlalu besar selama *training*, sehingga proses pembelajaran tidak mudah berfluktuasi. Penentuan *learning rate* dalam pelatihan jaringan saraf perlu dilakukan secara hati-hati karena berpengaruh terhadap kestabilan proses optimasi (Kingma & Ba, 2017).
Batch size	32	Batch size 32 digunakan karena masih sesuai dengan kemampuan komputasi yang digunakan dan dapat menjalankan proses *training* tanpa kendala memori pada percobaan pipeline awal. Nilai ini dipertahankan karena menghasilkan proses pelatihan yang stabil pada dataset penelitian. Penggunaan ukuran batch juga berkaitan dengan stabilitas gradien dan generalisasi model selama pelatihan (Keskar et al., 2017).
Epoch maksimum	100	Pada percobaan awal, model diuji menggunakan 20 epoch untuk memastikan pipeline dapat berjalan. Setelah itu, epoch maksimum dinaikkan menjadi 100 berdasarkan trial and error penulis agar model memiliki kesempatan lebih besar dalam mempelajari pola visual citra daun padi. Penggunaan epoch maksimum yang lebih besar tetap dikombinasikan dengan *early stopping* agar pelatihan dapat dihentikan ketika *validation* loss tidak lagi membaik (Prechelt, 1998).
Loss function	SparseCategorical Crossentropy(from_logits=True)	Loss function ini digunakan karena penelitian merupakan klasifikasi multi-kategori dengan label numerik. Pada CNN Kustom, *output layer* tidak menggunakan aktivasi softmax secara eksplisit, sehingga konfigurasi loss menggunakan from_logits=True. Penggunaan cross-entropy umum diterapkan pada tugas klasifikasi karena dapat mengukur

		perbedaan antara label sebenarnya dan hasil prediksi model (Mao et al., 2023).
Input size	256 × 256 × 3	Ukuran *input* digunakan untuk menyeragamkan citra sebelum masuk ke model. Angka 3 menunjukkan bahwa citra menggunakan kanal warna RGB. Ukuran 256 × 256 × 3 dipilih berdasarkan hasil *preprocessing* dan dipertahankan pada seluruh percobaan agar format *input* konsisten pada dataset original maupun *balanced*. Penyeragaman ukuran citra merupakan tahapan penting dalam pemrosesan citra sebelum pelatihan model CNN (Zhou et al., 2022).
Jumlah kelas	7 kelas	Jumlah kelas ditentukan berdasarkan kategori citra daun padi yang digunakan dalam penelitian, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Penentuan jumlah kelas mengikuti struktur folder dan label pada dataset penelitian (M., 2021).
Output layer	7 neuron	Jumlah neuron pada *output layer* disesuaikan dengan jumlah kelas yang diklasifikasikan. Karena penelitian menggunakan tujuh kelas, *output layer* pada CNN Kustom menggunakan 7 neuron, sehingga setiap neuron merepresentasikan satu kategori daun padi. Penyesuaian jumlah neuron *output* dengan jumlah kelas merupakan bagian dari rancangan model klasifikasi multi-kelas (Mao et al., 2023).
Augmentasi data	RandomFlip horizontal_and_vertical dan RandomRotation 0.2	Augmentasi data digunakan pada data *training* untuk menambah variasi citra latih, seperti perubahan orientasi dan posisi daun. Penggunaan augmentasi dipertahankan berdasarkan percobaan penulis karena dapat membantu model tidak hanya menghafal data *training*. Augmentasi citra juga digunakan untuk meningkatkan keragaman data dan mengurangi risiko *overfitting* pada pelatihan model *deep learning* (Shorten & Khoshgoftaar, 2019).
Dropout	0.10, 0.15, 0.20, dan 0.50	*Dropout* ditambahkan berdasarkan trial and error penulis pada percobaan CNN Kustom untuk mengurangi *overfitting*. *Dropout* ditempatkan setelah *MaxPooling*

		pada setiap blok konvolusi dan sebelum *output* akhir. Nilai *Dropout* dibuat bertahap, yaitu 0.10, 0.15, dan 0.20 pada blok konvolusi, serta 0.50 sebelum *output* akhir, agar regularisasi lebih kuat pada tahap klasifikasi. Teknik *dropout* dikenal dapat mengurangi *overfitting* dengan menonaktifkan sebagian neuron secara acak selama pelatihan (Srivastava et al., 2014).
ModelCheckpoint	*Validation* loss terendah	Digunakan untuk menyimpan model dengan performa validasi terbaik berdasarkan nilai *validation loss* terendah, yang merupakan praktik umum dalam pelatihan model *deep learning* untuk memilih model terbaik (Veeramreddy et al., 2024), (Paemo et al., 2025).
EarlyStopping	*Validation* loss	*Early Stopping* digunakan pada percobaan 100 epoch untuk menghentikan pelatihan ketika *validation loss* tidak menunjukkan perbaikan. Penggunaan *early stopping* bertujuan agar model tidak terus dilatih ketika performa validasi mulai stagnan atau menurun, sehingga risiko *overfitting* dapat dikurangi. *Early stopping* merupakan salah satu strategi yang umum digunakan untuk menghentikan pelatihan berdasarkan performa validasi (Prechelt, 1998).

Berdasarkan Tabel 3.4, model CNN Kustom tersebut diterapkan pada dua skenario dataset, yaitu dataset original dan dataset *balanced*. Dengan demikian, performa CNN Kustom dapat dibandingkan pada kondisi data tidak seimbang dan kondisi data yang lebih seimbang.

3.1.4 Pembuatan Model MobileNetV2

Tahap pembuatan model MobileNetV2 dilakukan untuk membangun model pembanding berbasis transfer learning. Berbeda dengan CNN Kustom yang dilatih dari awal, MobileNetV2 memanfaatkan bobot *pretrained* ImageNet sehingga model telah memiliki kemampuan awal dalam mengenali fitur visual umum, seperti tepi, bentuk, tekstur, dan pola warna. Pendekatan ini digunakan agar model tidak mempelajari seluruh fitur citra dari awal, tetapi menyesuaikan pengetahuan awal tersebut dengan karakteristik citra daun padi.

Pada penelitian ini, MobileNetV2 digunakan sebagai base model untuk proses ekstraksi fitur. Bagian klasifikasi bawaan MobileNetV2 tidak digunakan karena jumlah kelas pada ImageNet berbeda dengan jumlah kelas penelitian. Oleh karena itu, bagian akhir model diganti dengan layer klasifikasi

baru yang disesuaikan dengan tujuh kelas daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro.

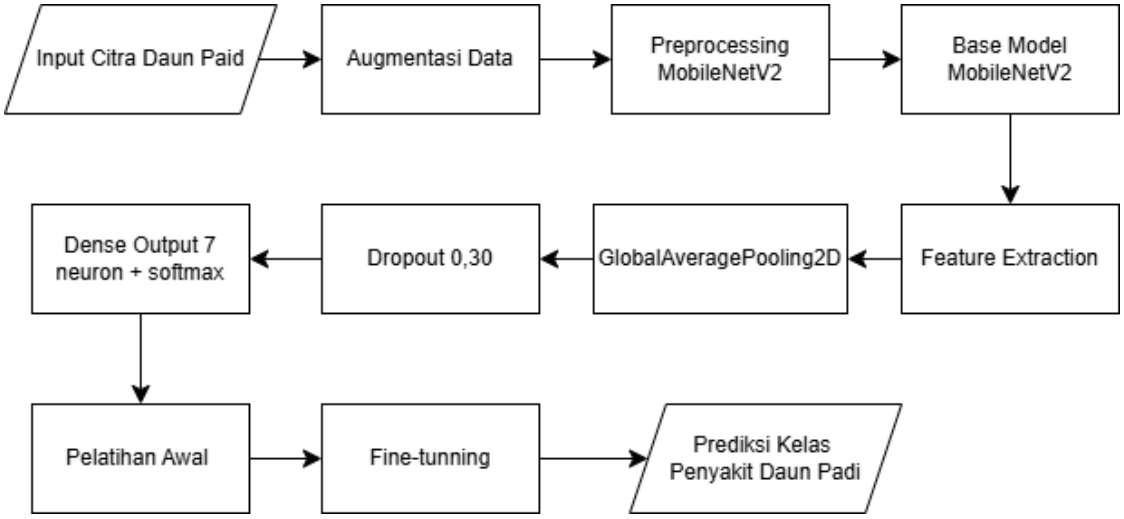
Alur model MobileNetV2 dimulai dari *input* citra berukuran 256 x 256 x 3. Citra pada data *training* kemudian melalui augmentasi berupa RandomFlip dan RandomRotation. Setelah itu, citra diproses menggunakan fungsi *preprocessing* khusus MobileNetV2 agar skala dan format *input* sesuai dengan format yang digunakan oleh model *pretrained*. Selanjutnya, citra masuk ke base model MobileNetV2 untuk proses ekstraksi fitur. Pada tahap ini, MobileNetV2 menghasilkan *feature map*, yaitu keluaran fitur yang memuat informasi visual penting dari citra, seperti pola, tekstur, tepi, dan bentuk objek. *Feature map* tersebut kemudian diringkas menggunakan GlobalAveragePooling2D agar menjadi representasi fitur yang lebih ringkas. Representasi fitur tersebut dilanjutkan ke *Dropout* 0,30 untuk mengurangi risiko overfitting, kemudian masuk ke *Dense output* dengan tujuh neuron dan aktivasi softmax untuk menghasilkan prediksi kelas.

Pelatihan MobileNetV2 dilakukan melalui dua tahap. Tahap pertama adalah *feature extraction*. Pada tahap ini, seluruh layer pada base model MobileNetV2 dibekukan atau dibuat tidak *trainable*, sehingga bobot *pretrained* tidak diperbarui selama pelatihan awal. Bobot yang diperbarui hanya berada pada layer klasifikasi tambahan, yaitu GlobalAveragePooling2D, *Dropout*, dan *Dense output*. Tahap ini bertujuan untuk melatih *classifier* baru berdasarkan fitur yang dihasilkan oleh MobileNetV2.

Tahap kedua adalah *fine-tuning*. Pada tahap ini, sebagian layer akhir MobileNetV2 dibuka agar bobot *pretrained* dapat menyesuaikan diri dengan karakteristik citra daun padi. *Fine-tuning* dilakukan menggunakan learning rate yang lebih kecil, yaitu 0,00001, agar perubahan bobot berlangsung lebih hati-hati dan tidak merusak fitur umum yang telah dipelajari dari ImageNet. Layer BatchNormalization tetap dibekukan untuk menjaga stabilitas proses pelatihan.

Jumlah layer akhir yang dibuka pada tahap *fine-tuning* tidak ditentukan secara langsung, tetapi melalui percobaan ablasi. Percobaan ablasi dilakukan dengan membandingkan beberapa skenario jumlah layer akhir yang dibuka, yaitu 0, 10, 20, 30, dan 50 layer. Skenario 0 layer merepresentasikan *feature extraction*, sedangkan skenario 10, 20, 30, dan 50 layer digunakan untuk melihat pengaruh tingkat *fine-tuning* terhadap performa model. Hasil percobaan ablasi digunakan sebagai dasar pemilihan konfigurasi final MobileNetV2..

Alur arsitektur MobileNetV2 yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3.10.



Gambar 3.10 Alur arsitektur MobileNetV2

Berdasarkan Gambar 3.10, konfigurasi MobileNetV2 disusun sesuai kebutuhan klasifikasi tujuh kelas penyakit daun padi. Konfigurasi tersebut ditunjukkan pada Tabel 3.5.

Tabel 3.5 Konfigurasi Arsitektur MobileNetV2

Tahap	**Konfigurasi**
Input	256 × 256 × 3
Rescaling	1/255
Augmentasi	RandomFlip dan RandomRotation
Preprocessing MobileNetV2	mobilenet_v2.preprocess_*input*
Base Model	MobileNetV2
Bobot Awal	*Pretrained* ImageNet
Include Top	False
Model Awal	*Trainable*=False
Feature extraction	Ekstraksi fitur menggunakan MobileNetV2
GlobalAveragePooling2D	Mengubah *feature map* menjadi representasi fitur ringkas
Dropout	0,30
Dense Output	7 neuron, aktivasi softmax
Pelatihan Awal	Melatih *layer* klasifikasi tambahan
Fine-tuning	0, 10, 20, 30, 50 layer
BatchNormalization	7 neuron
Output	Prediksi 7 kelas penyakit daun padi

Berdasarkan Tabel 3.5, setelah konfigurasi arsitektur ditetapkan, proses pelatihan MobileNetV2 dilakukan menggunakan beberapa hyperparameter. Hyperparameter ditentukan berdasarkan rujukan penelitian terdahulu dan hasil percobaan atau trial and error penulis. Pada percobaan awal, MobileNetV2 dijalankan dengan epoch 20 untuk memastikan pipeline dapat berjalan dari tahap

pemuatan dataset, *preprocessing*, pelatihan, hingga evaluasi. Pada tahap awal tersebut, MobileNetV2 hanya digunakan sebagai feature extractor.

Setelah percobaan awal berjalan, epoch maksimum dinaikkan menjadi 100 dengan *early stopping* agar model memiliki kesempatan mencapai kondisi konvergen tanpa harus selalu menyelesaikan seluruh epoch. Selain itu, dilakukan percobaan lanjutan berupa *fine-tuning* karena hasil MobileNetV2 dengan *feature extraction* saja belum sepenuhnya optimal. *Fine-tuning* dilakukan dengan membuka 30 *layer* akhir MobileNetV2 dan menggunakan *learning rate* yang lebih kecil, yaitu 0,00001. Strategi ini digunakan agar bobot *pretrained* dapat menyesuaikan dengan dataset daun padi secara lebih hati-hati.

Konfigurasi hyperparameter pelatihan MobileNetV2 ditunjukkan pada Tabel 3.6.

Tabel 3.6 Konfigurasi Hyperparameter pelatihan MobileNetV2

Hyperparameter	**Nilai**	**Dasar penentuan**
Optimizer	Adam	Adam digunakan karena mampu melakukan pembaruan bobot secara adaptif dan umum diterapkan pada pelatihan model *deep learning* (Kingma & Ba, 2017). Pada percobaan MobileNetV2, optimizer ini digunakan pada tahap *feature extraction* maupun *fine-tuning* dan mampu menjalankan proses pelatihan secara stabil.
Learning rate feature extraction	*0,001*	*Learning rate* 0,001 digunakan pada tahap *feature extraction* karena base model MobileNetV2 masih dibekukan, sehingga bobot yang diperbarui hanya berada pada *layer* klasifikasi tambahan. Nilai ini digunakan pada percobaan awal sebagai *learning rate* standar Adam dan dapat menjalankan pelatihan secara stabil (Kingma & Ba, 2017; Liu, 2023).

Learning rate fine-tuning	*0,00001*	*Learning rate fine-tuning* dibuat lebih kecil agar perubahan bobot *pretrained* tidak terlalu besar ketika 30 *layer* akhir MobileNetV2 dibuka. Nilai kecil digunakan berdasarkan trial and error penulis agar proses *fine-tuning* lebih stabil dan tidak merusak fitur yang telah dipelajari sebelumnya (Kingma & Ba, 2017; Kornblith et al., 2019).
Batch size	32	Batch size 32 dipilih karena masih sesuai dengan kemampuan komputasi yang digunakan dan pada percobaan awal dapat menjalankan proses *training* tanpa kendala memori. Nilai ini juga umum digunakan pada eksperimen *deep learning* berbasis citra (Keskar et al., 2017; Saddami et al., 2024).

Epoch maksimum feature extraction	100	Pada percobaan awal, model diuji menggunakan 20 epoch untuk memastikan pipeline berjalan. Setelah itu, epoch maksimum dinaikkan menjadi 100 agar classifier tambahan memiliki kesempatan lebih besar untuk mempelajari pola fitur dari MobileNetV2. *Early stopping* digunakan untuk menghentikan *training* ketika *validation* loss tidak lagi membaik (Kornblith et al., 2019; Prechelt, 1998).
Epoch maksimum fine-tuning	*30*	Epoch *fine-tuning* dibuat lebih kecil karena model sudah memiliki bobot hasil *feature extraction*. *Fine-tuning* hanya bertujuan menyesuaikan sebagian *layer* akhir MobileNetV2 terhadap karakteristik citra daun padi, bukan melatih model dari awal (Utami et al., 2025).
Loss function	Sparse categorical crossentropy	Digunakan karena penelitian merupakan klasifikasi multi-kelas dengan label numerik. Label tidak diubah ke bentuk one-hot encoding, sehingga sparse categorical crossentropy sesuai dengan format label yang digunakan pada pipeline penelitian (Costales et al., 2020; Utami et al., 2025).
Input size	256 × 256 × 3	Digunakan untuk menyeragamkan ukuran

		seluruh citra sebelum masuk ke model. Ukuran 256 × 256 dipilih agar detail visual seperti bercak, tekstur, dan perubahan warna masih dapat terlihat, sedangkan angka 3 menunjukkan kanal warna RGB (Gunawan & Darmansyah, 2024; Zhou et al., 2022).
Jumlah Kelas	7 Kelas	Ditentukan berdasarkan jumlah kategori citra daun padi yang digunakan dalam penelitian, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro (Jisan, 2023; M., 2021).
OutputLayer	7 neuron, softmax	Jumlah neuron pada lapisan *output* disesuaikan dengan jumlah kelas, sehingga setiap neuron merepresentasikan satu kategori klasifikasi. Aktivasi softmax digunakan untuk menghasilkan probabilitas prediksi masing-masing kelas (Costales et al., 2020; Utami et al., 2025).
Augmentasi Data	RandomFlip dan RandomRotation	Digunakan untuk menambah variasi data *training* dan membantu model mempelajari perubahan orientasi serta posisi citra. Augmentasi umum digunakan pada klasifikasi citra untuk meningkatkan keragaman data latih dan mengurangi risiko *overfitting*

		(Saddami et al., 2024; Shorten & Khoshgoftaar, 2019).
Dropout	0,30	*Dropout* ditambahkan setelah GlobalAveragePooling2D untuk mengurangi risiko *overfitting* pada *layer* klasifikasi tambahan. Nilai 0,30 dipertahankan berdasarkan rancangan model dan trial and error penulis selama pelatihan MobileNetV2 (Saddami et al., 2024; Srivastava et al., 2014).
Pretrained weights	ImageNet	Bobot ImageNet digunakan agar model memiliki pengetahuan awal terhadap fitur visual umum. Penggunaan *pretrained* weights membantu model *transfer learning* agar tidak belajar dari awal sepenuhnya (Deng et al., 2021; Rahman et al., 2024).
Fine-tuning layer	0, 10, 20, 30, 50 *layer akhir MobileNetV2*	Jumlah *layer* akhir ditentukan berdasarkan percobaan ablasi penulis. *Layer* akhir dibuka karena bagian akhir model cenderung mempelajari fitur yang lebih spesifik terhadap tugas klasifikasi, sehingga dapat disesuaikan dengan citra daun padi (Costales et al., 2020; Miftahuddin et al., 2025; Utami et al., 2025).
BatchNormalization	Tetap *frozen*	BatchNormalization tetap dibekukan saat *fine-tuning* agar distribusi aktivasi tetap stabil

		dan proses pelatihan tidak mengalami perubahan bobot yang terlalu besar (Ioffe & Szegedy, 2015).
ModelCheckpoint	*Validation loss* terendah	Digunakan untuk menyimpan model terbaik berdasarkan performa validasi selama proses pelatihan. Pada percobaan penulis, model terbaik tidak selalu berada pada epoch terakhir, sehingga *checkpoint* diperlukan untuk menyimpan model dengan *validation* loss terendah (Prechelt, 1998; Saddami et al., 2024).
EarlyStopping	*Vlidation loss*	Digunakan untuk menghentikan pelatihan ketika *validation* loss tidak lagi menunjukkan perbaikan. *Early stopping* digunakan karena epoch maksimum dinaikkan menjadi 100, sehingga diperlukan mekanisme untuk mengurangi risiko *overfitting* dan menghentikan *training* ketika model sudah tidak membaik (Prechelt, 1998; Saddami et al., 2024).

Berdasarkan Tabel 3.6 model MobileNetV2 tersebut diterapkan pada dataset original dan dataset *balanced*. Hasil dari kedua skenario tersebut kemudian dibandingkan dengan hasil CNN Kustom untuk mengetahui model yang memiliki performa lebih baik dalam klasifikasi penyakit daun padi.

3.1.5 Pelatihan Model

Tahap pelatihan model dilakukan setelah proses pengumpulan data, *preprocessing*, pembagian data, dan perancangan arsitektur model selesai. Pelatihan bertujuan agar model dapat mempelajari pola visual pada citra daun padi, seperti warna daun, bercak, tekstur, tepi daun, bentuk lesi, dan perubahan warna yang menjadi karakteristik dari masing-masing kelas.

Pada penelitian ini, pelatihan dilakukan menggunakan dua skenario dataset, yaitu dataset original dan dataset *balanced*. Dataset original digunakan untuk mengetahui performa model pada kondisi data awal yang masih tidak seimbang, terutama karena jumlah data pada kelas Tungro lebih banyak dibandingkan kelas lainnya. Sementara itu, dataset *balanced* digunakan untuk mengetahui performa model setelah distribusi data dibuat lebih seimbang melalui proses *quality-based filtering* pada kelas Tungro dan augmentasi pada kelas minoritas.

Secara umum, proses pelatihan model dilakukan melalui beberapa tahap, yaitu pemanggilan dataset hasil *preprocessing*, pemanggilan arsitektur model, kompilasi model, proses *training*, validasi model, penyimpanan model terbaik, dan pengujian menggunakan data *testing*. Data *training* digunakan untuk memperbarui bobot model selama proses pembelajaran. Data *validation* digunakan untuk memantau performa model selama proses pelatihan, sedangkan data *testing* digunakan untuk mengevaluasi performa akhir model setelah proses pelatihan selesai.

Pelatihan CNN Kustom dilakukan dalam dua bentuk, yaitu CNN Kustom awal dan CNN Kustom dengan *Dropout*. CNN Kustom awal digunakan sebagai *baseline* untuk mengetahui kemampuan dasar model yang dilatih dari awal tanpa menggunakan bobot *pretrained*. Sementara itu, CNN Kustom dengan *Dropout* digunakan untuk mengetahui pengaruh regularisasi terhadap kemampuan generalisasi model. *Dropout* ditambahkan setelah proses *pooling* dan sebelum lapisan *output* agar model tidak terlalu bergantung pada fitur tertentu selama proses *training*.

Pelatihan MobileNetV2 dilakukan melalui dua tahap, yaitu *feature extraction* dan *fine-tuning*. Pada tahap *feature extraction*, *base model* MobileNetV2 dibekukan sehingga bobot *pretrained* tidak diperbarui selama proses pelatihan. Model hanya melatih lapisan klasifikasi tambahan yang disesuaikan dengan tujuh kelas daun padi. Setelah tahap tersebut, dilakukan *fine-tuning* dengan membuka sebagian lapisan akhir MobileNetV2 menggunakan *learning rate* yang lebih kecil. Jumlah lapisan akhir yang dibuka tidak ditentukan secara langsung, tetapi diuji melalui percobaan ablasi pada 0, 10, 20, 30, dan 50 layer. Hasil percobaan ablasi digunakan untuk menentukan konfigurasi *fine-tuning* yang memberikan performa terbaik.

Skenario percobaan yang dilakukan dalam penelitian ini ditunjukkan pada Tabel 3.7.

Tabel 3.7 Skenario eksperimen pelatihan model

Percobaan	Dataset	Model	Konfigurasi yang Diuji	Tujuan

Percobaan awal pipeline	Original dan *balanced*	CNN Kustom dan MobileNetV2	Epoch 20	Memastikan pipeline penelitian berjalan dari pemuatan data sampai evaluasi
Percobaan epoch maksimum	Original dan *balanced*	CNN Kustom dan MobileNetV2	Epoch maksimum 100 + *early stopping*	Memberikan kesempatan model mencapai kondisi konvergen dan mengurangi *overfitting*
Percobaan MobileNetV2 awal	Original dan *balanced*	MobileNetV2	*Feature extraction*	Melatih *layer* klasifikasi tambahan dengan base model MobileNetV2 dalam kondisi frozen
Percobaan MobileNetV2 lanjutan	Original dan *balanced*	MobileNetV2	*Feature extraction* + *fine-tuning*	Menyesuaikan *layer* akhir MobileNetV2 terhadap karakteristik citra daun padi
Percobaan CNN Kustom awal	Original dan *balanced*	CNN Kustom	Tanpa *dropout* tambahan	Mengetahui performa CNN sebagai baseline awal
Percobaan CNN Kustom lanjutan	Original dan *balanced*	CNN Kustom	Penambahan *dropout* setelah MaxPooling2D dan *Dense*	Mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model

Berdasarkan skenario pada Tabel 3.7, eksperimen utama dalam penelitian ini terdiri atas kombinasi dataset dan model. Kombinasi tersebut digunakan untuk membandingkan performa model pada data original dan data *balanced*, serta untuk melihat pengaruh perubahan konfigurasi model terhadap hasil klasifikasi. Kombinasi eksperimen utama ditunjukkan pada Tabel 3.8.

Tabel 3.8 Kombinasi eksperimen Utama

No	**Dataset**	**Model**	**Konfigurasi yang Diuji**
1	Dataset original	CNN Kustom	Menguji CNN Kustom pada data tidak seimbang
2	Dataset original	MobileNetV2	Menguji MobileNetV2 pada data tidak seimbang
3	Dataset *balanced*	MobileNetV2	Menguji CNN Kustomsetelah kelas Tungro diseimbangkan
4	Dataset *balanced*	MobileNetV2	Menguji MobileNetV2 setelah kelas Tungro diseimbangkan

Berdasarkan Tabel 3.8 pada pelatihan CNN Kustom, model dibangun dari awal tanpa menggunakan bobot *pretrained*. Optimizer yang digunakan adalah Adam dengan *learning rate* 0,001. Fungsi loss yang digunakan adalah SparseCategoricalCrossentropy(from_logits=True) karena *output* model CNN Kustom tidak menggunakan aktivasi softmax secara eksplisit pada *layer output*.

Pada pelatihan MobileNetV2, proses *training* dilakukan dalam dua tahap. Tahap pertama adalah *feature extraction* dengan *learning rate* 0,001. Pada tahap ini, base model MobileNetV2 dibekukan sehingga hanya *layer* klasifikasi tambahan yang dilatih. Tahap kedua adalah *fine-tuning* dengan *learning rate* 0,00001. Pada tahap *fine-tuning*, dilakukan dengan percobaan ablasi untuk menentukan layer akhir dengan membuka 0, 10, 20, 30, dan 50 *layer* akhir MobileNetV2, sedangkan *layer* BatchNormalization tetap dibekukan agar proses pelatihan tetap stabil.

Konfigurasi hyperparameter pelatihan model ditunjukkan pada Tabel 3.9.

Tabel 3.9 Konfigurasi *hyperparameter* pelatihan model

Hyperparameter	**CNN Kustom**	**MobileNetV2 *Feature extraction***	**MobileNetV2 *Fine-tuning***
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0,00001
Batch size	32	32	32
Epoch	100	100	30
Loss function	*SparseCategorical Crossentropy(from_logits=True)*	*Sparse_categorial_crossentropy*	*Sparse_categorial_crossentropy*
Input size	256×256×3	256×256×3	256×256×3
Jumlah kelas	7 kelas	7 kelas	7 kelas
Output	7 neuron	7 neuron + softmax	7 neuron + softmax
Callback	EarlyStopping, ModelCheckpoint	EarlyStopping, ModelCheckpoint	EarlyStopping, ModelCheckpoint
Dasar model terbaik	*Validation loss* terendah	*Validation loss* terendah	*Validation loss* terendah

Berdasarkan Tabel 3.9, *Early stopping* digunakan untuk menghentikan pelatihan ketika nilai *validation* loss tidak lagi mengalami perbaikan dalam beberapa epoch berturut-turut. Penggunaan *early stopping* dilakukan karena epoch maksimum yang digunakan cukup besar, yaitu 100 epoch. Dengan *early stopping*, proses pelatihan dapat berhenti lebih awal apabila model sudah tidak menunjukkan peningkatan pada data *validation*. Hal ini bertujuan untuk mengurangi risiko *overfitting* dan menghemat waktu pelatihan.

Selain *early stopping*, digunakan juga ModelCheckpoint. ModelCheckpoint berfungsi untuk menyimpan model dengan nilai *validation* loss terbaik selama proses pelatihan. Dengan demikian, model yang digunakan pada tahap evaluasi bukan selalu model pada epoch terakhir, melainkan model terbaik berdasarkan performa validasi. Hal ini penting karena pada proses pelatihan, nilai *validation* loss dapat mengalami naik turun, sehingga model terbaik dapat terjadi sebelum epoch terakhir.

Hasil dari setiap skenario percobaan kemudian disimpan dalam bentuk model terlatih, grafik *accuracy* dan loss, *classification report*, *confusion matrix*, serta tabel perbandingan hasil evaluasi. Seluruh hasil percobaan tersebut dianalisis pada Bab IV untuk mengetahui model terbaik dan pengaruh *imbalance* dataset terhadap performa klasifikasi penyakit daun padi.

3.1.6 Evaluasi

Evaluasi dilakukan pada data testing dari hasil pembagian dataset. Tahap ini bertujuan untuk menilai kemampuan model dalam mengklasifikasikan citra daun padi ke dalam tujuh kelas, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Evaluasi dilakukan setelah proses pelatihan selesai dan model terbaik diperoleh berdasarkan nilai *validation* loss terendah melalui ModelCheckpoint.

Evaluasi dilakukan terhadap seluruh skenario percobaan yang telah dijalankan. Pada CNN Kustom, evaluasi dilakukan terhadap CNN Kustom awal dan CNN Kustom dengan penambahan *Dropout*. Pada MobileNetV2, evaluasi dilakukan terhadap tahap *feature extraction* dan *fine-tuning* berdasarkan skenario ablasi jumlah layer. Evaluasi juga dilakukan pada dua skenario dataset, yaitu dataset original dan dataset *balanced*.

Pengujian dilakukan dengan membandingkan hasil prediksi model terhadap label sebenarnya pada data *testing*. Metrik evaluasi yang digunakan dalam penelitian ini meliputi:

a. *Accuracy*, yaitu proporsi prediksi yang benar terhadap seluruh data *testing*.
b. *Precison*, yaitu tingkat ketepatan model dalam memprediksi suatu kelas
c. *Recall,* yaitu kemampuan model dalam mengenali data yang sebenarnya berasal dari suatu kelas.
d. *F1-score,* yaitu metrik yang menggabungkan *precision* dan *recall* untuk melihat keseimbangan keduanya.
e. *Classification report,* yaitu ringkasan evaluasi yang menampilkan *precision*, *recall*, F1-score, dan support pada setiap kelas.
f. *Confusion matrix,* yaitu tabel yang menunjukkan jumlah prediksi benar dan salah pada setiap kelas.

Dalam penelitian ini, metrik yang lebih ditekankan untuk menentukan performa model adalah recall macro. Recall macro digunakan karena penelitian berkaitan dengan klasifikasi penyakit daun padi dan ketidakseimbangan dataset. Pada kasus klasifikasi penyakit, kemampuan model dalam mengenali setiap kelas penyakit menjadi penting, sehingga model tidak hanya dinilai dari jumlah prediksi benar secara keseluruhan, tetapi juga dari kemampuannya menemukan data pada masing-masing kelas. Recall macro menghitung rata-rata recall seluruh kelas dengan bobot yang sama, sehingga setiap kelas tetap diperhatikan meskipun jumlah datanya berbeda.

F1-score macro digunakan sebagai metrik pendamping untuk melihat keseimbangan antara precision dan recall pada setiap kelas. Metrik ini membantu menilai apakah model tidak hanya mampu mengenali kelas tertentu, tetapi juga memiliki keseimbangan antara ketepatan prediksi dan kemampuan mengenali data tiap kelas. *Accuracy* tetap digunakan sebagai metrik pendukung untuk memberikan gambaran performa model secara umum, tetapi tidak dijadikan satu-satunya dasar pemilihan model terbaik karena *accuracy* dapat dipengaruhi oleh kelas dengan jumlah data lebih banyak.

Berdasarkan pertimbangan tersebut, penentuan model terbaik dilakukan dengan memprioritaskan recall macro, kemudian didukung oleh F1-score macro dan *accuracy*. Model yang dipilih sebagai model

terbaik adalah model yang mampu memberikan nilai recall macro tinggi dan tetap memiliki F1-score macro serta *accuracy* yang baik. Dengan demikian, evaluasi tidak hanya melihat seberapa banyak prediksi yang benar secara keseluruhan, tetapi juga melihat kemampuan model dalam mengenali seluruh kelas penyakit secara lebih merata.

Classification report digunakan untuk melihat performa model pada masing-masing kelas secara lebih rinci. Melalui classification report, dapat diketahui nilai precision, recall, F1-score, dan support untuk setiap kelas penyakit daun padi. Support menunjukkan jumlah data testing pada masing-masing kelas. Informasi ini penting karena dataset original memiliki distribusi jumlah data yang tidak seimbang, terutama pada kelas Tungro.

Selain classification report, confusion matrix digunakan untuk mengetahui pola kesalahan prediksi model. Confusion matrix dapat menunjukkan kelas mana yang sering diklasifikasikan dengan benar dan kelas mana yang sering tertukar dengan kelas lain. Analisis ini diperlukan karena beberapa penyakit daun padi memiliki kemiripan visual, seperti bercak daun, perubahan warna, dan pola kerusakan yang mirip.

Hasil evaluasi dari setiap skenario kemudian digunakan pada tahap analisis perbandingan untuk mengetahui model terbaik, pengaruh perubahan konfigurasi model, serta pengaruh balancing dataset terhadap performa klasifikasi penyakit daun padi.

3.1.7 Analisis Perbandingan

Analisis perbandingan dilakukan setelah seluruh hasil evaluasi diperoleh. Tahap ini bertujuan untuk membandingkan performa model pada setiap skenario percobaan yang telah dilakukan. Analisis tidak hanya dilakukan untuk membandingkan CNN Kustom dan MobileNetV2, tetapi juga untuk melihat pengaruh perubahan konfigurasi model dan pengaruh balancing dataset terhadap hasil klasifikasi.

Pada penelitian ini, analisis perbandingan dilakukan dalam beberapa bentuk, yaitu:

a. Perbandingan antar model, yaitu membandingkan hasil CNN Kustom dan MobileNetV2 pada dataset yang sama.
b. Perbandingan antar skenario dataset, yaitu membandingkan hasil model pada dataset original dan dataset *balanced*.
c. Perbandingan konfigurasi CNN Kustom, yaitu membandingkan CNN Kustom awal dengan CNN Kustom yang ditambahkan *Dropout*.
d. Perbandingan konfigurasi MobileNetV2, yaitu membandingkan MobileNetV2 pada tahap *feature extraction* dengan MobileNetV2 setelah *fine-tuning*.

Perbandingan antar model digunakan untuk mengetahui model yang memiliki performa lebih baik dalam klasifikasi penyakit daun padi. Perbandingan antar dataset digunakan untuk mengetahui apakah ketidakseimbangan jumlah data pada kelas Tungro berpengaruh terhadap hasil evaluasi model. Sementara itu, perbandingan konfigurasi model digunakan untuk mengetahui apakah perubahan yang

dilakukan pada model, seperti penambahan *Dropout* pada CNN Kustom dan *fine-tuning* pada MobileNetV2, memberikan peningkatan performa.

Metrik yang lebih ditekankan dalam analisis perbandingan adalah recall macro. Metrik ini dipilih karena recall macro menunjukkan rata-rata kemampuan model dalam mengenali data dari setiap kelas dengan bobot yang sama. Hal ini penting pada penelitian klasifikasi penyakit daun padi karena model diharapkan mampu mengenali seluruh kelas penyakit, bukan hanya menghasilkan prediksi benar secara umum atau dominan pada kelas tertentu. Selain itu, recall macro juga relevan digunakan karena penelitian membahas ketidakseimbangan dataset.

F1-score macro digunakan sebagai metrik pendukung utama untuk melihat keseimbangan antara precision dan recall pada setiap kelas. *Accuracy* digunakan sebagai metrik pendukung untuk memberikan gambaran performa model secara keseluruhan. Precision macro, F1-score weighted, classification report, dan confusion matrix juga digunakan untuk memperkuat analisis, tetapi tidak menjadi satu-satunya dasar dalam menentukan model terbaik.

Analisis perbandingan juga memperhatikan nilai F1-score weighted. F1-score weighted digunakan untuk melihat performa model dengan mempertimbangkan jumlah data pada setiap kelas. Dengan membandingkan F1-score macro dan F1-score weighted, penelitian dapat melihat apakah performa model merata pada seluruh kelas atau lebih dipengaruhi oleh kelas yang memiliki jumlah data lebih banyak. Namun, karena fokus penelitian adalah melihat kemampuan model mengenali setiap kelas secara lebih seimbang, recall macro dan F1-score macro tetap lebih ditekankan dalam pembahasan.

Hasil analisis perbandingan digunakan untuk menentukan model terbaik pada penelitian ini. Model terbaik ditentukan dengan memprioritaskan nilai recall macro, kemudian didukung oleh F1-score macro dan *accuracy*. Selain itu, classification report dan confusion matrix digunakan untuk melihat pola kesalahan prediksi antar kelas, sehingga pemilihan model terbaik tidak hanya berdasarkan angka evaluasi secara umum, tetapi juga berdasarkan kemampuan model mengenali setiap kelas penyakit daun padi secara lebih merata.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Tahap pengumpulan data menghasilkan dua skenario dataset yang digunakan dalam penelitian, yaitu dataset original dan dataset *balanced*. Dataset original digunakan untuk merepresentasikan kondisi awal data setelah penggabungan dua sumber dataset. Sementara itu, dataset *balanced* digunakan untuk melihat pengaruh ketidakseimbangan jumlah data, terutama pada kelas Tungro yang memiliki jumlah citra lebih banyak dibandingkan kelas lainnya.

4.1.1 Dataset Original

Dataset original terdiri dari tujuh kelas citra daun padi, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf scald, Sheath Blight, dan Tungro. Jumlah keseluruhan citra pada dataset original adalah 5.137 gambar. Distribusi data pada dataset original ditunjukkan pada Tabel 4.1.

Tabel 4.1 Distribusi dataset original

Kelas	Jumlah Gambar
Bacterial Leaf Blight	636
Brown Spot	646
Healthy Rice Leaf	653
Leaf Blast	634
Leaf Scald	628
Sheath Blight	631
Tungro	1308
Total	5137

Berdasarkan Tabel 4.1, kelas Tungro memiliki jumlah data sebanyak 1.308 gambar. Jumlah tersebut hampir dua kali lipat dibandingkan kelas lain yang berada pada kisaran 600 gambar. Kondisi ini menunjukkan adanya ketidakseimbangan data pada dataset original.

Contoh citra pada masing-masing kelas daun padi dalam dataset penelitian ditunjukkan pada Gambar 4.1 sampai Gambar 4.7. Gambar 4.1 menunjukkan contoh citra kelas Bacterial Leaf Blight, Gambar 4.2 menunjukkan kelas Brown Spot, Gambar 4.3 menunjukkan kelas Healthy Rice Leaf, Gambar 4.4 menunjukkan kelas Leaf Blast, Gambar 4.5 menunjukkan kelas Leaf Scald, Gambar 4.6 menunjukkan kelas Sheath Blight, dan Gambar 4.7 menunjukkan kelas Tungro.



Gambar 4.1 Bacterial Leaf Blight



Gambar 4.2 Brown Spot



Gambar 4.3 Healthy Rice Leaf



Gambar 4.4 Leaf Blast



Gambar 4.5 Leaf Scald



Gambar 4.6 Sheath Blight



Gambar 4.7 Tungro

4.1.2 Dataset *Balanced*

Dataset *balanced* dibuat dari dataset original untuk mengurangi ketimpangan jumlah data antar kelas. Pada dataset original, kelas Tungro memiliki jumlah citra paling banyak, yaitu 1.308 citra, sedangkan kelas lainnya berada pada kisaran 628 sampai 653 citra. Oleh karena itu, penyeimbangan dilakukan dengan target 653 citra pada setiap kelas.

Penyeimbangan dataset dilakukan melalui dua tahap, yaitu *quality-based filtering* pada kelas mayoritas dan augmentasi pada kelas minoritas. Pada kelas Tungro, citra dikurangi dari 1.308 menjadi 653 citra dengan mempertahankan citra yang memiliki kualitas visual lebih baik. Sementara itu, kelas dengan jumlah citra kurang dari 653 ditambah menggunakan augmentasi hingga mencapai jumlah target. Distribusi data pada dataset *balanced* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Distribusi dataset *balance*

Kelas	Data Awal	Citra Asli Dipakai	Citra dikeluarkan	Citra Augmentasi	Total Akhir
Bacterial Leaf Blight	636	636	0	17	653
Brown Spot	646	646	0	7	653
Healthy Rice Leaf	653	653	0	0	653

Leaf Blast	634	634	0	19	653
Leaf Scald	628	628	0	25	653
Sheath Blight	632	632	0	21	653
Tungro	636	653	655	0	653
Total	4465	4482	655	89	4571

Berdasarkan Tabel 4.2, dataset *balanced* memiliki total akhir 4.571 citra dengan jumlah 653 citra pada setiap kelas. Jumlah tersebut diperoleh dari 4.482 citra asli yang dipertahankan dan 89 citra hasil augmentasi. Pada kelas Tungro, sebanyak 655 citra dikeluarkan berdasarkan proses *quality-based filtering*, sedangkan kelas minoritas ditambah melalui augmentasi agar distribusi data menjadi seimbang.

Strategi ini berbeda dari *undersampling* sederhana karena citra yang dikeluarkan tidak dipilih secara acak, melainkan berdasarkan kualitas visual. Selain itu, kelas minoritas tidak dibiarkan berada di bawah jumlah target, tetapi ditambah melalui augmentasi. Dengan skenario ini, model tidak hanya diuji pada dataset original, tetapi juga pada dataset *balanced* yang memiliki distribusi kelas lebih merata.

4.2 *Preprocessing*

Tahap *preprocessing* dimulai dengan memuat citra daun padi dari struktur folder dataset yang telah disusun berdasarkan nama kelas. Setiap folder merepresentasikan satu label kelas penyakit daun padi. Dataset kemudian dibagi menjadi data *training*, *validation*, dan testing. Data *training* digunakan untuk melatih model, data *validation* digunakan untuk memantau performa selama pelatihan, sedangkan data testing digunakan untuk mengevaluasi performa akhir model.

Pada skenario dataset original, data testing yang digunakan pada tahap evaluasi berjumlah 768 citra. Sementara itu, pada skenario dataset *balanced*, data testing yang digunakan berjumlah 672 citra. Perbedaan jumlah data testing ini terjadi karena jumlah keseluruhan citra pada dataset original dan dataset *balanced* berbeda. Dataset original memiliki 5.137 citra, sedangkan dataset *balanced* memiliki 4.571 citra. Selain itu, pembagian *validation* dan testing pada pipeline dilakukan berdasarkan jumlah batch, sehingga jumlah citra pada *validation* dan testing tidak selalu sama persis.

Setelah pembagian data, seluruh citra diubah ke ukuran yang sama, yaitu 256 x 256 piksel. Proses resize dilakukan karena citra pada dataset memiliki ukuran awal yang bervariasi. Karena citra yang digunakan merupakan citra berwarna RGB, dimensi *input* model menjadi 256 x 256 x 3. Angka 3 menunjukkan tiga kanal warna, yaitu red, green, dan blue.

Tahap *preprocessing* berikutnya disesuaikan dengan model yang digunakan. Pada CNN Kustom, nilai piksel citra dinormalisasi menggunakan rescaling 1/255 sehingga nilai piksel yang semula berada pada rentang 0-255 diubah menjadi rentang 0-1. Pada MobileNetV2, *preprocessing* dilakukan menggunakan fungsi khusus MobileNetV2 karena model tersebut menggunakan bobot awal ImageNet, sehingga format *input* citra perlu disesuaikan dengan format yang digunakan oleh model *pretrained*. Selain itu, augmentasi data diterapkan pada data *training* menggunakan RandomFlip dan RandomRotation untuk menambah variasi citra selama proses pelatihan.

4.3 Pembuatan Model CNN Kustom

Hasil pembuatan model CNN Kustom menghasilkan dua rancangan arsitektur yang digunakan dalam percobaan, yaitu CNN Kustom awal dan CNN Kustom dengan *Dropout*. Kedua rancangan tersebut sama-sama dibangun dari awal tanpa menggunakan bobot *pretrained*. Perbedaan utama terdapat pada penambahan lapisan *Dropout* pada rancangan kedua untuk mengurangi risiko *overfitting* selama proses pelatihan.

4.3.1 Implementasi CNN Kustom Awal

Pada percobaan awal, CNN Kustom disusun menggunakan tiga blok konvolusi. Setiap blok terdiri dari Conv2D dan MaxPooling2D. Lapisan Conv2D berfungsi mengekstraksi pola visual dari citra daun padi, sedangkan MaxPooling2D digunakan untuk mengurangi ukuran *feature map* agar proses komputasi lebih ringan. Setelah fitur diekstraksi, *feature map* diubah menjadi vektor menggunakan *Flatten*, kemudian diproses oleh *Dense layer* untuk menghasilkan skor prediksi kelas. Arsitektur CNN Kustom awal ditunjukkan pada Tabel 4.3.

Tabel 4.3 Arsitektur CNN Kustom Awal

Tahap	Konfigurasi	Keterangan
Input	256 × 256 × 3	Menerima citra RGB daun padi sebagai *input* model
Rescaling	1/255	Mengubah nilai piksel dari rentang 0-255 menjadi 0-1
Augmentasi	RandomFlip(horizontal_and_vertical), RandomRotation(0.2)	Menambah variasi data *training*
Convo2D	16 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mengekstraksi fitur visual awal
MaxPooling2D	Pooling 2 × 2	Mengurangi ukuran *feature map*
Convo2D	32 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mengektraksi fitur visual lanjutan

MaxPooling2D	Pooling 2 × 2	Mengurangi ukuran *feature map* tahap kedua
Convo2D	64 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mempelajari pola visual yang lebih kompleks
MaxPooling2D	Pooling 2 × 2	Merangkum fitur penting sebelum klasifikasi
Flatten	Mengubah *feature map* menjadi vector satu dimensi	Mengubah *feature map* menjadi vektor satu dimensi
Dense	128 neuron, aktivasi ReLU	Mengolah fitur untuk proses klasifikasi
Output	7 neuron	Menghasilkan skor prediksi untuk tujuh kelas

Berdasarkan Tabel 4.3, model pada percobaan awal ini belum menggunakan *Dropout*. Oleh karena itu, rancangan ini digunakan sebagai baseline untuk melihat kemampuan awal CNN Kustom sebelum dilakukan penambahan regularisasi.

4.3.2 Implementasi CNN Kustom dengan *Dropout*

Pada percobaan awal, CNN Kustom disusun menggunakan tiga blok konvolusi. Setiap blok terdiri dari Conv2D dan MaxPooling2D. Lapisan Conv2D berfungsi mengekstraksi pola visual dari citra daun padi, sedangkan MaxPooling2D digunakan untuk mengurangi ukuran *feature map* agar proses komputasi lebih ringan. Setelah fitur diekstraksi, *feature map* diubah menjadi vektor menggunakan *Flatten*, kemudian diproses oleh *Dense layer* untuk menghasilkan skor prediksi kelas. Arsitektur CNN Kustom awal ditunjukkan pada Tabel 4.4.

Tabel 4.4 Arsitektur CNN Kustom dengan *Dropout*

Tahap	**Konfigurasi**	**Keterangan**
Input	256 × 256 × 3	Menerima citra RGB daun padi sebagai *input* model
Rescaling	1/255	Mengubah nilai piksel dari rentang 0-255 menjadi 0-1
Augmentasi	RandomFlip(horizontal_and_vertical), RandomRotation(0.2)	Menambah variasi data *training*
Convo2D	16 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mengekstraksi fitur visual awal
MaxPooling2D	Pooling 2 × 2	Mengurangi ukuran *feature map*
Convo2D	32 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mengektraksi fitur visual lanjutan
MaxPooling2D	Pooling 2 × 2	Mengurangi ukuran *feature map* tahap kedua
Convo2D	64 filter, karnel 3 × 3, padding same, aktivasi ReLU	Mempelajari pola visual yang lebih kompleks

MaxPooling2D	Pooling 2 × 2	Merangkum fitur penting sebelum klasifikasi
Flatten	Mengubah *feature map* menjadi vector satu dimensi	Mengubah *feature map* menjadi vektor satu dimensi
Dense	128 neuron, aktivasi ReLU	Mengolah fitur untuk proses klasifikasi
Output	7 neuron	Menghasilkan skor prediksi untuk tujuh kelas

Berdasarkan Tabel 4.4, dengan adanya penambahan *Dropout*, arsitektur CNN Kustom menjadi lebih kuat terhadap risiko *overfitting* dibandingkan rancangan awal. Percobaan ini digunakan untuk melihat apakah perubahan kecil pada arsitektur CNN Kustom dapat memberikan dampak terhadap performa klasifikasi pada dataset original dan dataset *balanced*.

4.4 Pembuatan Model MobileNetV2

Hasil pembuatan model MobileNetV2 menghasilkan rancangan model *transfer learning* yang digunakan sebagai pembanding terhadap CNN Kustom. Model ini menggunakan MobileNetV2 dengan bobot awal ImageNet dan *layer* klasifikasi tambahan yang disesuaikan dengan tujuh kelas penyakit daun padi. Pada penelitian ini, MobileNetV2 diuji dalam dua tahap, yaitu *feature extraction* dan *fine-tuning*. Kedua tahap tersebut menggunakan susunan *layer* yang sama, tetapi berbeda pada status *layer* base model yang dilatih.

4.4.1 Implementasi MobileNetV2 *Feature extraction*

Pada tahap *feature extraction*, base model MobileNetV2 digunakan sebagai ekstraktor fitur. Seluruh bobot pada base model dibekukan, sehingga proses pelatihan hanya dilakukan pada *layer* tambahan di bagian akhir model. Tahap ini digunakan untuk mengetahui kemampuan fitur *pretrained* ImageNet ketika diterapkan pada citra penyakit daun padi tanpa mengubah bobot utama MobileNetV2. Implementasi MobileNetV2 pada tahap *feature extraction* ditunjukkan pada Tabel 4.5.

Tabel 4.5 Arsitektur MobileNetV2 Featue Extraction

Tahap	**Konfigurasi**	**Keterangan**
Input	256 × 256 × 3	Menerima Citra RGB daun padi sebagai *input* model
Augmentasi	RandomFlip dan RandomRotation	Menambahkan variasi data *training*
Preprocessing MobileNetV2	mobilenet_v2.preprocess_*input*	Menyesuaikan format *input* dengan kebutuhan MobileNetV2 *pretrained*
Base Model	MobileNetV2	Model dasar *transfer learning*

Bobot Awal	*Pretrained* ImageNet	Menggunakan bobot *pretrained*
Include Top	False	*Layer* klasifikasi bawaan ImageNet tidak digunakan
Model Awal	*Trainable*=False	Bobot MobileNetV2 tidak diperbarui pada *feature extraction*
Feature extraction	Base model MobileNetV2	Mengekstraksi fitur visual dari citra daun padi
GlobalAveragePooling2D	-	Mengubah *feature map* menjadi representasi fitur ringkas
Dropout	0,30	Mengurangi risiko *overfitting*
Dense Output	7 neuron, aktivasi softmax	Menghasilkan prediksi tujuh kelas

Berdasarkan Tabel 4.5, MobileNetV2 belum menyesuaikan bobot utamanya terhadap dataset penelitian. Model hanya memanfaatkan fitur umum yang telah dipelajari dari ImageNet, kemudian *layer* klasifikasi tambahan dilatih untuk membedakan tujuh kelas penyakit daun padi.

4.4.2 Implementasi MobileNetV2 *Fine-tuning*

Setelah tahap *feature extraction*, percobaan MobileNetV2 dilanjutkan dengan *fine-tuning*. Pada tahap ini, sebagian layer akhir MobileNetV2 dibuka agar sebagian bobot *pretrained* dapat menyesuaikan diri dengan karakteristik citra daun padi. Jumlah layer akhir yang dibuka tidak langsung ditentukan pada satu nilai tertentu, tetapi diuji melalui percobaan ablasi. Skenario ablasi yang digunakan adalah 0, 10, 20, 30, dan 50 layer akhir. Skenario 0 layer merepresentasikan tahap *feature extraction*, sedangkan skenario 10 sampai 50 layer digunakan untuk melihat pengaruh jumlah layer yang dibuka terhadap performa model.

Pada proses *fine-tuning*, layer BatchNormalization tetap dibekukan untuk menjaga stabilitas pelatihan. *Fine-tuning* dilakukan menggunakan learning rate yang lebih kecil, yaitu 0,00001, agar perubahan bobot *pretrained* tidak terlalu besar. Implementasi MobileNetV2 pada tahap *fine-tuning* ditunjukkan pada Tabel 4.6.

Tabel 4.6 Arsitektur MobileNetV2 *Fine-tuning*

Tahap	**Konfigurasi**	**Keterangan**
Input	256 × 256 × 3	Menerima Citra RGB daun padi sebagai *input* model
Augmentasi	RandomFlip dan RandomRotation	Menambahkan variasi data *training*

Preprocessing MobileNetV2	mobilenet_v2.preprocess_*input*	Menyesuaikan format *input* dengan kebutuhan MobileNetV2 *pretrained*
Base Model	MobileNetV2	Model dasar *transfer learning*
Bobot Awal	*Pretrained* ImageNet	Menggunakan bobot *pretrained*
Include Top	False	*Layer* klasifikasi bawaan ImageNet tidak digunakan
Tahap *Feature extraction*	Base model frozen / *Trainable*=False	Bobot MobileNetV2 tidak diperbarui pada *feature extraction*
GlobalAveragePooling2D	-	Mengubah *feature map* menjadi representasi fitur ringkas
Dropout	0,30	Mengurangi risiko *overfitting*
Dense Output	7 neuron, aktivasi softmax	Menghasilkan prediksi tujuh kelas
Tahap *fine-tuning*	Membuka 30 *layer* akhir MobileNetV2	Menyesuaikan sebagian bobot *pretrained* dengan karakteristik citra daun padi
BatchNormalization	Tetap frozen saat *fine-tuning*	Menjaga kestabilan proses *fine-tuning*agar statistic normalisasi tidak berubah drastic
Learning rate Fine-tuning	0,00001	Mengatur pembaruan bobot agar tidak terlalu besar saat *fine-tuning*
Output	Prediksi 7 kelas penyakit daun padi	Hasil akhir odel berupa kelas penyakit dengan probabilitas tertinggi

Berdasarkan Tabel 4.6, *fine-tuning* dilakukan sebagai tahap lanjutan setelah *feature extraction*. Perbedaan utama keduanya terletak pada status base model. Pada *feature extraction*, seluruh layer MobileNetV2 dibekukan sehingga hanya layer klasifikasi tambahan yang dilatih. Pada *fine-tuning*, sebagian layer akhir MobileNetV2 dibuka berdasarkan skenario ablasi agar model dapat mempelajari pola visual yang lebih sesuai dengan dataset penyakit daun padi. Hasil dari skenario ablasi tersebut kemudian digunakan untuk menentukan jumlah layer *fine-tuning* yang paling sesuai.

4.5 Pelatihan Model

Pelatihan model dilakukan melalui beberapa skenario percobaan. Percobaan awal dilakukan menggunakan 20 epoch untuk memastikan pipeline pelatihan, evaluasi, dan penyimpanan hasil dapat berjalan dengan baik. Setelah itu, percobaan dilanjutkan menggunakan epoch maksimum 100 dengan *early stopping* agar model memiliki kesempatan belajar lebih lama, tetapi tetap dapat berhenti ketika performa validasi tidak lagi membaik.

Pada penelitian ini, trial and error dilakukan pada dua sisi. Pertama, pada CNN Kustom dilakukan percobaan antara arsitektur awal dan arsitektur dengan penambahan *Dropout*. Kedua, pada MobileNetV2 dilakukan percobaan antara *feature extraction* dan *fine-tuning*. Seluruh percobaan diterapkan pada dua skenario dataset, yaitu dataset original dan dataset *balanced*.

4.5.1 Hasil Pelatihan CNN Kustom

Pada CNN Kustom, percobaan pertama dilakukan menggunakan arsitektur awal dengan epoch 20. Percobaan ini digunakan sebagai tahap awal untuk melihat apakah model dapat belajar dari dataset dan menghasilkan evaluasi akhir. Hasil percobaan awal menunjukkan bahwa model sudah mampu mempelajari pola citra daun padi, tetapi performanya masih dapat ditingkatkan.

Hasil pelatihan CNN Kustom pada percobaan epoch 20 tercatat pada log pelatihan. Log tersebut memuat nilai loss, *accuracy*, *validation* loss, dan *validation accuracy* dari epoch pertama sampai epoch ke-20. Namun, pada penelitian ini grafik pelatihan epoch 20 tidak disajikan sebagai gambar terpisah karena file grafik yang tersimpan merupakan grafik dari percobaan lanjutan. Oleh karena itu, hasil percobaan awal epoch 20 dibahas melalui tabel evaluasi pada subbab 4.6.1, sedangkan grafik pelatihan yang ditampilkan pada subbab ini difokuskan pada percobaan lanjutan dan model final.

Setelah percobaan awal, CNN Kustom kemudian dilatih kembali dengan epoch maksimum 100 dan *early stopping*. Penggunaan epoch maksimum yang lebih besar bertujuan agar model memiliki kesempatan belajar lebih lama sampai proses pelatihan mulai konvergen. *Early stopping* digunakan agar pelatihan dapat berhenti ketika *validation* loss tidak lagi menunjukkan perbaikan.

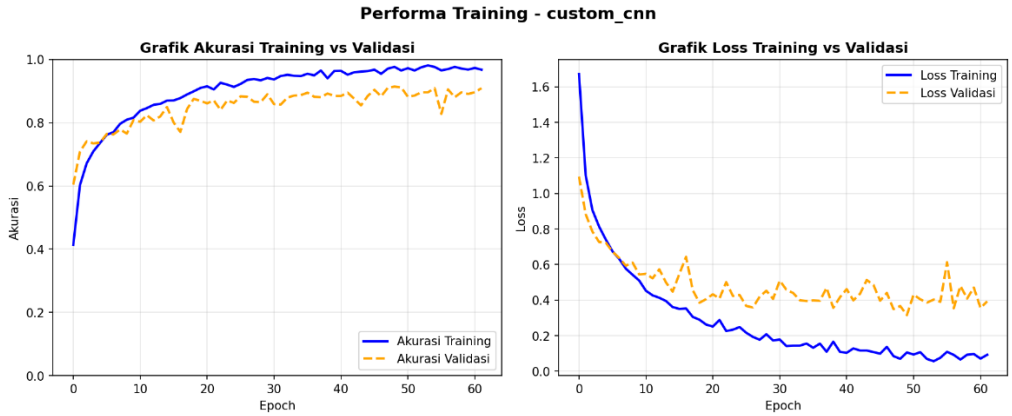
Berdasarkan hasil pelatihan lanjutan, dilakukan percobaan tambahan dengan menambahkan *Dropout* pada CNN Kustom. *Dropout* ditempatkan setelah setiap MaxPooling2D dan sebelum *output layer*. Percobaan ini dilakukan karena model CNN Kustom awal masih berpotensi mengalami *overfitting*. Ringkasan trial and error pelatihan CNN Kustom ditunjukkan pada Tabel 4.7.

Tabel 4.7 Ringkasan Trial Pelatihan CNN Kustom

Percobaan	Dataset	Konfigurasi Pelatihan	Tujuan Percobaan	Tujuan
CNN Kustom awal epoch 20	Original	Arsitektur awal, epoch 20	Menguji pipeline awal dan kemampuan dasar CNN Kustom	Model sudah dapat belajar, tetapi performanya masih lebih rendah dibandingkan skenario lanjutan
CNN Kustom awal epoch maksimum 100	Original dan *balanced*	Arsitektur awal, *early stopping*, *checkpoint* model terbaik	performa CNN Kustom jika diberi kesempatan *training* lebih lama	Digunakan sebagai baseline pembanding terhadap CNN Kustom dengan *Dropout*
CNN Kustom dengan *Dropout*	Original	*Dropout* setelah MaxPooling2D dan sebelum *output layer*, *early stopping*, *checkpoint* model terbaik	Menguji pengaruh regularisasi pada dataset *original*	Memberikan hasil lebih baik dibandingkan CNN Kustom awal pada dataset original
CNN Kustom dengan *Dropout*	*Balanced quality-hybrid*	*Dropout* setelah MaxPoolong2D dan Sebelum *output* layer, EarlyStopping, dan ModelCheckpoint	Menguji pengaruh regularisasi pada dataset *balanced* terbaru	Performa belum lebih baik dibandingkan MobileNetV2 dan digunakan sebagai pembanding akhir

Berdasarkan Tabel 4.7, percobaan CNN Kustom dilakukan secara bertahap. Percobaan dimulai dari arsitektur awal, kemudian dilanjutkan dengan epoch maksimum 100 yang dikombinasikan dengan *EarlyStopping* dan *ModelCheckpoint*. Setelah itu, model dikembangkan dengan penambahan *Dropout*. *EarlyStopping* digunakan agar proses pelatihan dapat berhenti ketika *validation loss* tidak lagi mengalami perbaikan, sedangkan *ModelCheckpoint* digunakan untuk menyimpan model terbaik berdasarkan nilai *validation loss* terendah.

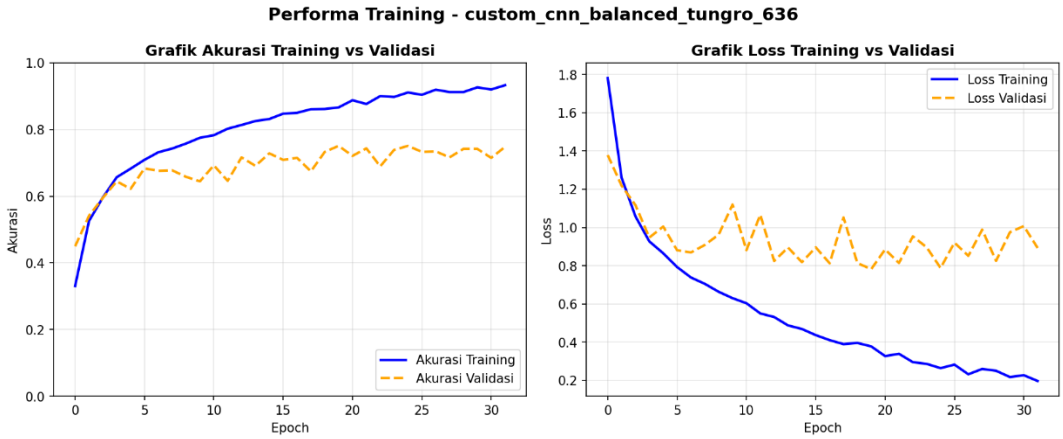
Grafik pelatihan CNN Kustom awal pada dataset *original* ditunjukkan pada Gambar 4.8. Grafik ini digunakan untuk melihat perkembangan *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss* ketika model dilatih sebagai *baseline* pada dataset *original*.



Gambar 4.8 Grafik *training* dan *validation* CNN Kustom awal dataset original

Berdasarkan Gambar 4.8 Grafik *training* dan *validation* CNN Kustom awal dataset original, proses pelatihan CNN Kustom awal pada dataset original dapat diamati dari perubahan nilai *accuracy* dan loss pada data *training* maupun *validation*. Grafik tersebut membantu menunjukkan apakah model mengalami peningkatan performa dan kapan *validation* loss mulai tidak membaik.

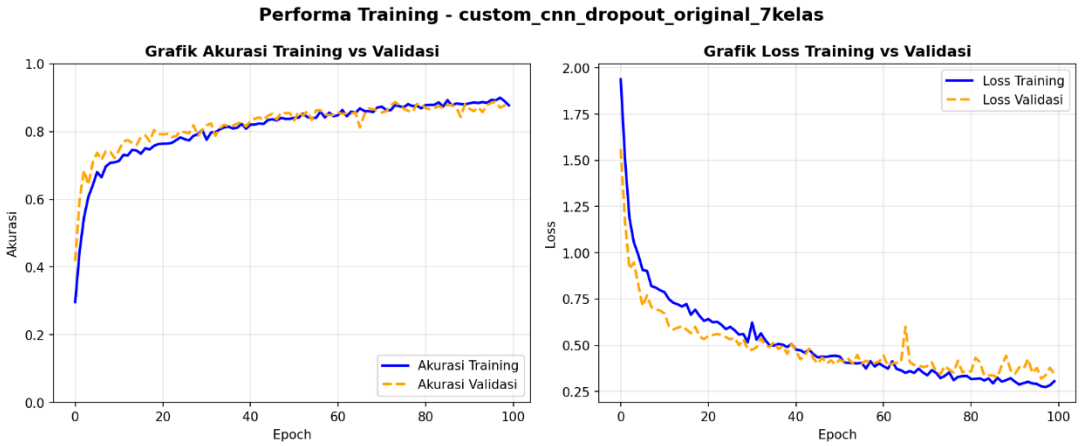
Grafik pelatihan CNN Kustom awal pada dataset *balanced* ditunjukkan pada Gambar 4.9.



Gambar 4.9 Grafik *training* dan *validation* CNN Kustom awal dataset *balanced*

Berdasarkan Gambar 4.9, grafik pelatihan CNN Kustom awal pada dataset *balanced* digunakan untuk melihat pengaruh penyeimbangan jumlah data Tungro terhadap proses pembelajaran model. Grafik ini juga menjadi pembanding terhadap hasil CNN Kustom awal pada dataset original.

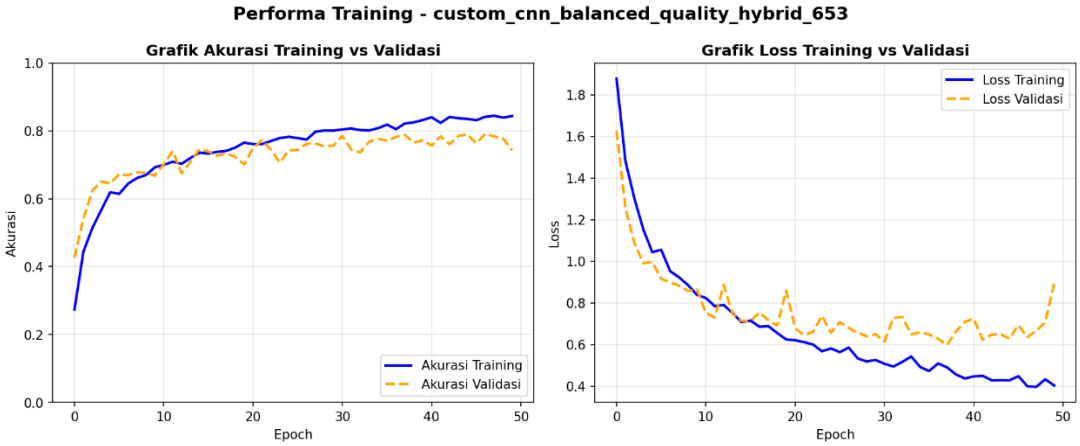
Setelah percobaan CNN Kustom awal, grafik pelatihan CNN Kustom dengan *Dropout* pada dataset original ditunjukkan pada Gambar 4.10.



Gambar 4.10 Grafik *training* dan *validation* CNN Kustom dengan *dropout* dataset original

BerdasarkanGambar 4.10, grafik pelatihan CNN Kustom dengan *Dropout* pada dataset original digunakan untuk melihat apakah penambahan *Dropout* membantu mengurangi jarak antara performa *training* dan *validation*. Hal ini penting karena *Dropout* diterapkan untuk mengurangi risiko *overfitting*.

Grafik pelatihan CNN Kustom dengan *Dropout* pada dataset *balanced* ditunjukkan pada Gambar 4.11.



Gambar 4.11 Grafik *training* dan *validation* CNN Kustom dengan *dropout* dataset *balanced quality-hybrid*

Berdasarkan Gambar 4.11, grafik pelatihan CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* digunakan untuk melihat perkembangan *training accuracy*, *validation accuracy*, *training* loss, dan *validation* loss setelah data diseimbangkan menggunakan quality-based filtering dan augmentasi. Grafik ini juga menjadi pembanding terhadap hasil CNN Kustom dengan *Dropout* pada dataset original.

4.5.2 Pelatihan MobileNetV2

Pada MobileNetV2, pelatihan dilakukan menggunakan pendekatan *transfer learning*. Proses pelatihan dilakukan dalam dua tahap, yaitu *feature extraction* dan *fine-tuning*. Pada tahap *feature extraction*, *base model* MobileNetV2 berada dalam kondisi *frozen*, sehingga bobot *pretrained* tidak diperbarui. Pelatihan pada tahap ini hanya dilakukan pada *layer* klasifikasi tambahan yang disesuaikan dengan tujuh kelas daun padi.

Setelah tahap *feature extraction* selesai, pelatihan dilanjutkan ke tahap *fine-tuning*. Pada tahap ini, sebagian *layer* akhir MobileNetV2 dibuka agar sebagian bobot *pretrained* dapat menyesuaikan diri dengan karakteristik citra penyakit daun padi. Jumlah *layer* akhir yang dibuka diuji melalui beberapa skenario, yaitu 10, 20, 30, dan 50 *layer* akhir. Selain itu, skenario 0 *layer* digunakan sebagai pembanding karena merepresentasikan kondisi *feature extraction* tanpa membuka *layer* akhir MobileNetV2.

Fine-tuning dilakukan menggunakan *learning rate* yang lebih kecil, yaitu **0,00001**, agar perubahan bobot *pretrained* tidak terlalu besar selama proses pelatihan. *Layer* BatchNormalization tetap dibekukan untuk menjaga kestabilan proses *fine-tuning*. Seluruh skenario pelatihan menggunakan *EarlyStopping* dan *ModelCheckpoint*, sehingga proses *training* dapat berhenti ketika *validation loss* tidak lagi membaik dan model terbaik tetap disimpan berdasarkan nilai *validation loss* terendah.

Ringkasan skenario pelatihan MobileNetV2 ditunjukkan pada Tabel 4.8.

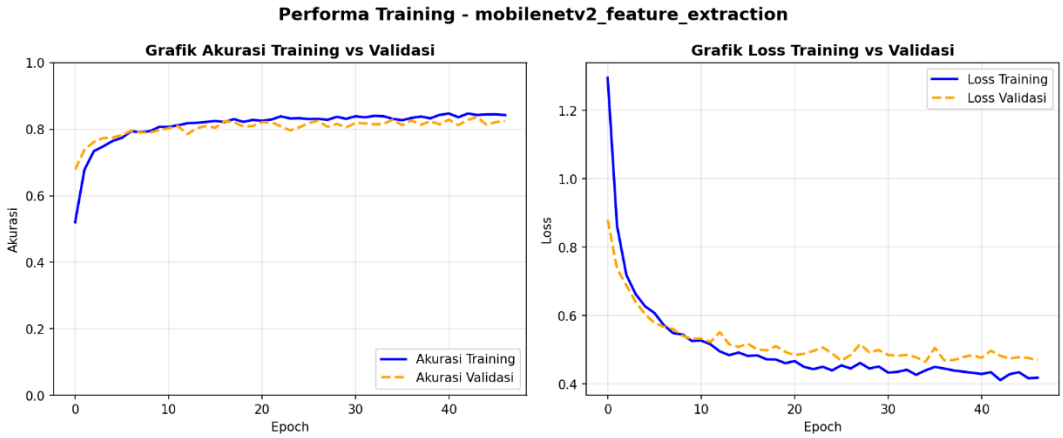
Tabel 4.8 Ringkasan Trial Pelatihan MobileNetV2

Percobaan	Dataset	Konfigurasi Pelatihan	Keterangan
Feature extraction	Original dan Balande	Base model frozen epoch 20	Melatih layer klasifikasi tambahan
Fine-tuning 10 *layer*	Original dan *balanced*	Membuka 10 *layer* akhir, BatchNormalization tetap *frozen*, *learning rate* 0,00001	Melatih sebagian kecil *layer* akhir MobileNetV2
Fine-tuning 20 *layer*	Original dan *balanced*	Membuka 20 *layer* akhir, BatchNormalization tetap *frozen*, *learning rate* 0,00001	Melatih sebagian kecil *layer* akhir MobileNetV2
Fine-tuning 30 *layer*	Original dan *Balanced*	Membuka 30 *layer* akhir, BatchNormalization tetap *frozen*,	Melatih sebagian kecil *layer* akhir MobileNetV2

		learning rate 0,00001	
Fine-tuning 50 *layer*	Original dan *Balanced*	Membuka 50 *layer* akhir, BatchNormalization tetap *frozen*, *learning rate* 0,00001	Melatih sebagian kecil *layer* akhir MobileNetV2

Berdasarkan Tabel 4.8, pelatihan MobileNetV2 pada dataset *original* dan dataset *balanced* dilakukan dengan alur yang sama. Perbedaannya terletak pada skenario dataset yang digunakan, yaitu dataset *original* dan dataset *balanced quality-hybrid*. Dengan alur tersebut, proses pelatihan MobileNetV2 dapat diamati mulai dari tahap *feature extraction* hingga beberapa variasi *fine-tuning*.

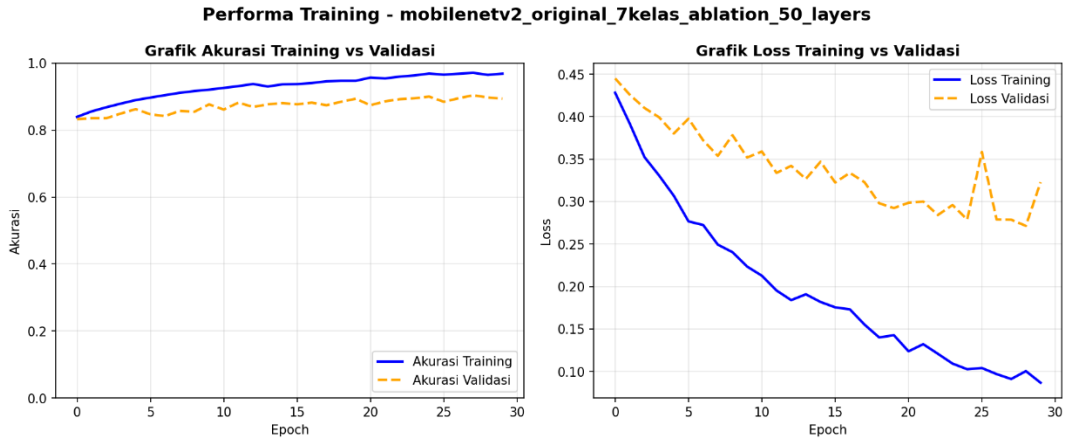
Grafik pelatihan MobileNetV2 *feature extraction* pada dataset original ditunjukkan pada Gambar 4.12.



Gambar 4.12 Grafik *training* dan *validation* MobileNetV2 *feature extraction* dataset original

Berdasarkan Gambar 4.12, grafik pelatihan MobileNetV2 pada tahap *feature extraction* menggunakan dataset *original* digunakan untuk melihat perkembangan *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss* ketika *base model* MobileNetV2 masih dalam kondisi *frozen*. Grafik ini menunjukkan proses pelatihan *layer* klasifikasi tambahan sebelum dilakukan *fine-tuning*.

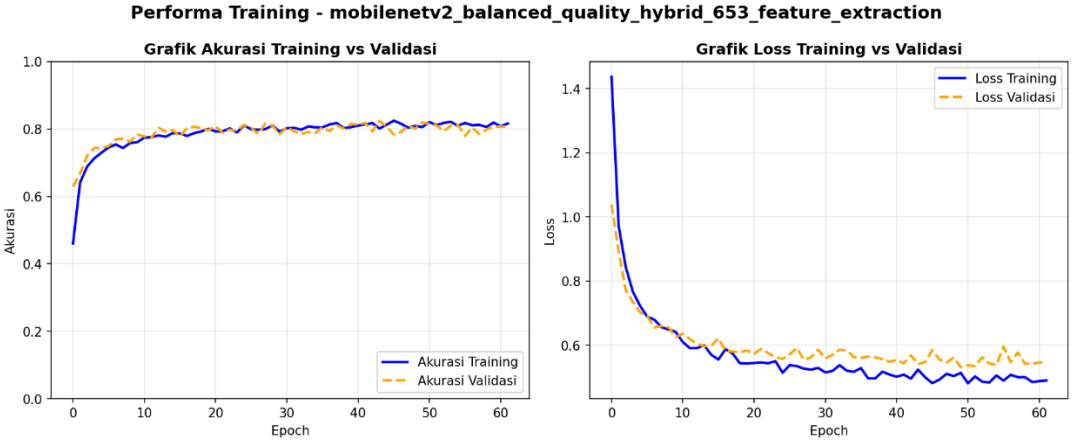
Grafik pelatihan MobileNetV2 *fine-tuning* pada dataset original ditunjukkan pada Gambar 4.13.



Gambar 4.13 Grafik *training* dan *validation* MobileNetV2 *fine-tuning* 50 layer dataset original

Berdasarkan Gambar 4.13, grafik pelatihan MobileNetV2 *fine-tuning* 50 *layer* pada dataset *original* digunakan untuk melihat perkembangan *training* dan *validation* setelah sebagian *layer* akhir MobileNetV2 dibuka. Grafik ini menjadi pembanding terhadap tahap *feature extraction* pada dataset *original*.

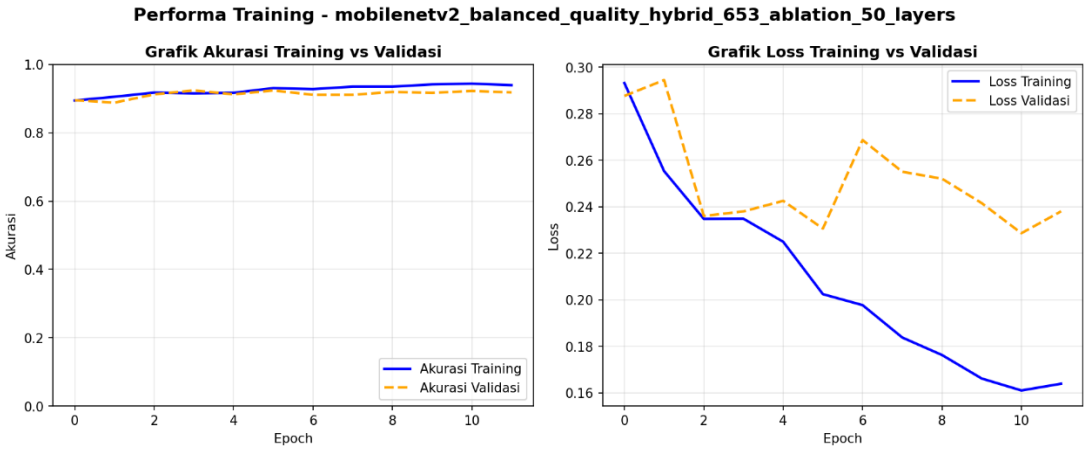
Grafik pelatihan MobileNetV2 *feature extraction* pada dataset *balanced* ditunjukkan pada Gambar 4.14.



Gambar 4.14 Grafik *training* dan *validation* MobileNetV2 *feature extraction* dataset *balanced quality-hybrid*

Berdasarkan Gambar 4.14, grafik *feature extraction* pada dataset *balanced* digunakan untuk melihat performa MobileNetV2 ketika distribusi kelas Tungro sudah diseimbangkan. Grafik ini menjadi pembanding terhadap *feature extraction* pada dataset original.

Grafik pelatihan MobileNetV2 *fine-tuning* pada dataset *balanced* ditunjukkan padaGambar 4.15.



Gambar 4.15 Grafik *training* dan *validation* MobileNetV2 *fine-tuning* 50 *layer* dataset *balanced quality-hybrid*

Berdasarkan Gambar 4.15, grafik pelatihan MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced* digunakan untuk melihat perkembangan *training* dan *validation* setelah sebagian bobot *pretrained* disesuaikan dengan dataset penyakit daun padi yang lebih seimbang. Grafik ini menjadi pembanding terhadap grafik *fine-tuning* MobileNetV2 pada dataset *original*.

4.5.3 Ringkasan Seluruh Percobaan Pelatihan

Secara keseluruhan, percobaan pelatihan dalam penelitian ini dilakukan secara bertahap. Tahap pertama dilakukan untuk memastikan alur pelatihan berjalan dengan baik dan mengetahui kemampuan awal model. Setelah itu, percobaan dilanjutkan dengan penggunaan epoch maksimum, *EarlyStopping*, *ModelCheckpoint*, penambahan *Dropout* pada CNN Kustom, serta *feature extraction* dan *fine-tuning* pada MobileNetV2. Ringkasan seluruh percobaan pelatihan ditunjukkan pada Tabel 4.9.

Tabel 4.9 Ringkasan Seluruh Percobaan Pelatihan

No	Percobaan	Dataset	Model	Tujuan Percobaan	Tujuan
1	Percobaan awal	Original	CNN Kustom Awal	Epoch 20 tanpa *dropout* tambahan	Menguji pipeline awal dan melihat kemampuan dasar CNN Kustom
2	Percobaan awal	*Balanced*	CNN Kustom Awal	Epoch 20 tanpa *dropout* tambahan	Melihat performa CNN Kustom pada dataset *balanced*
3	Percobaan epoch maksimum	Original	CNN Kustom Awal	Epoch maksimum 100, *early stopping*	Melihat performa CNN Kustom awal dengan waktu pelatihan lebih Panjang

4	Percobaan epoch maksimum	*Balanced*	CNN Kustom Awal	Epoch maksimum 100, *early stopping*	Melihat pengaruh dataset *balanced* terhadap CNN Kustom awal
5	Percobaan MobileNetV2 *feature extraction*	Original	MobileNetV2	Base model frozen, epoch maksimum 100, *early stopping*	Melihat *layer* klasifikasi tambahan pada dataset original
6	Percobaan MobileNetV2 *feature extraction*	*Balanced quality-hybrid*	MobileNetV2	Base model frozen, epoch maksimum 100, *early stopping*	Melihat *layer* klasifikasi tambahan pada dataset *balanced*
7	Percobaan MobileNetV2 *fine-tuning* ablasi	Original	MobileNetV2	Membuka 10, 20, 30, dan 50 *layer* akhir, *learning rate* 0,00001	Mengamati proses *fine-tuning* pada beberapa jumlah *layer* akhir
8	Percobaan MobileNetV2 *fine-tuning* ablasi	*Balanced quality-hybrid*	MobileNetV2	Membuka 10, 20, 30, dan 50 *layer* akhir, *learning rate* 0,00001	Mengamati proses *fine-tuning* pada dataset *balanced*
9	Percobaan CNN Kustom dengan *dropout*	Original	CNN Kustom + *Dropout*	*Dropout* 0,10; 0,15; 0,20; 0,50, *early stopping*	Mengurangi *overfitting* pada dataset original
10	Percobaan CNN Kustom dengan *dropout*	*Balanced quality hybrid*	CNN Kustom + *Dropout*	*Dropout* 0,10; 0,15; 0,20; 0,50, *early stopping*	Mengurangi *overfitting* pada dataset *balanced*

Berdasarkan Tabel 4.9, percobaan pelatihan mencakup dua model utama, yaitu CNN Kustom dan MobileNetV2. Pada CNN Kustom, percobaan dilakukan mulai dari arsitektur awal hingga penambahan *Dropout*. Sementara itu, pada MobileNetV2, percobaan dilakukan mulai dari tahap *feature extraction* hingga *fine-tuning* dengan beberapa skenario jumlah *layer* akhir yang dibuka.

4.6 Evaluasi Model

Evaluasi model dilakukan menggunakan data *testing* dari masing-masing skenario dataset. Data *testing* digunakan untuk mengetahui kemampuan model dalam mengklasifikasikan citra daun padi yang tidak digunakan selama proses pelatihan. Pada dataset *original*, jumlah data *testing* yang digunakan adalah 768 citra. Pada dataset *balanced quality-hybrid*, jumlah data *testing* yang digunakan adalah 672 citra.

Metrik evaluasi yang digunakan meliputi *accuracy*, *precision*, *recall*, *F1-score*, *classification report*, dan *confusion matrix*. Pada penelitian ini, *recall macro* lebih ditekankan karena penelitian berkaitan dengan klasifikasi penyakit daun padi. *Recall macro* digunakan untuk melihat rata-rata kemampuan model dalam mengenali data dari setiap kelas dengan

bobot yang sama. Dengan demikian, evaluasi tidak hanya melihat jumlah prediksi benar secara keseluruhan, tetapi juga memperhatikan kemampuan model dalam mengenali setiap kelas penyakit.

F1-score macro digunakan sebagai metrik pendukung utama karena metrik ini menunjukkan keseimbangan antara *precision* dan *recall* pada setiap kelas. *Accuracy* tetap digunakan untuk memberikan gambaran performa model secara umum, tetapi tidak dijadikan satu-satunya dasar dalam menentukan model terbaik karena *accuracy* dapat dipengaruhi oleh distribusi jumlah data pada setiap kelas.

4.6.1 Evaluasi Seluruh Skenario Model

Evaluasi dilakukan terhadap model CNN Kustom dan MobileNetV2 pada dataset *original* dan dataset *balanced quality-hybrid*. Pada CNN Kustom, evaluasi dilakukan terhadap CNN Kustom awal dan CNN Kustom dengan *Dropout*. Pada MobileNetV2, evaluasi dilakukan terhadap hasil *fine-tuning* berdasarkan percobaan ablasi jumlah *layer* akhir yang dibuka. Ringkasan hasil evaluasi seluruh skenario utama ditunjukkan pada Tabel 4.10.

Tabel 4.10 Ringkasan Hasil Evaluasi Skenario Utama

Model	**Dataset**	**Konfigurasi**	***Accuracy***	***Precision* Macro**	***Recall* Macro**	**F1-score Macro**
CNN Kustom Awal	Original	Baseline	83,20%	81,11%	80,79%	80,74%
CNN Kustom + *Dropout*	Original	*Dropout*	86,98%	85,41%	85,22%	85,10%
MobileNetV2	Original	*Fine-tuning* 50 *layer*	87,89%	86,08%	85,96%	85,87%
CNN Kustom + *Dropout*	*Balanced quality-hybrid*	*Dropout*	76,34%	75,97%	76,07%	75,72%
MobileNetV2	*Balanced quality-hybrid*	*Fine-tuning* 50 *layer*	92,26%	92,14%	92,21%	92,17%

Berdasarkan Tabel 4.10, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* memperoleh nilai tertinggi pada seluruh metrik utama, yaitu *accuracy* 92,26%, *precision macro* 92,14%, *recall macro* 92,21%, dan *F1-score macro* 92,17%. Nilai *recall macro* yang tinggi menunjukkan bahwa model mampu mengenali data pada setiap kelas penyakit secara lebih merata.

Pada dataset *original*, MobileNetV2 *fine-tuning* 50 *layer* juga memperoleh nilai tertinggi dibandingkan CNN Kustom awal dan CNN Kustom dengan *Dropout*. Namun, peningkatannya

tidak sebesar pada dataset *balanced quality-hybrid*. Hal ini menunjukkan bahwa performa MobileNetV2 dapat meningkat ketika data yang digunakan lebih seimbang dan memiliki kualitas visual yang lebih terkontrol.

Pada CNN Kustom, penambahan *Dropout* meningkatkan performa pada dataset *original* dibandingkan CNN Kustom awal. Namun, pada dataset *balanced quality-hybrid*, performa CNN Kustom dengan *Dropout* belum setinggi MobileNetV2. Hal ini menunjukkan bahwa CNN Kustom yang dilatih dari awal lebih sensitif terhadap variasi data, sedangkan MobileNetV2 memiliki keuntungan dari bobot *pretrained* ImageNet.

4.6.2 Evaluasi Ablasi MobileNetV2

Percobaan ablasi MobileNetV2 dilakukan untuk menentukan jumlah *layer* akhir yang paling sesuai untuk dibuka pada tahap *fine-tuning*. Skenario yang diuji adalah 0, 10, 20, 30, dan 50 *layer*. Skenario 0 *layer* merepresentasikan *feature extraction* karena tidak ada *layer* akhir MobileNetV2 yang dibuka. Hasil ablasi MobileNetV2 pada dataset *original* ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil ablasi MobileNetV2 pada dataset original

Jumlah *Layer* yang dibuka	***Acuracy***	***Precision Macro***	***Recall Macro***	***F1-score Macro***	***Best Epoch***
0 *Layer*	80,08%	76,94%	76,96%	76,79%	-
10 *Layer*	86,85%	84,85%	84,83%	84,77%	30
20 *Layer*	84,24%	82,06%	81,85%	81,83%	23
30 *Layer*	85,29%	82,89%	82,93%	82,88%	24
50 *Layer*	87,89%	86,08%	85,96%	85,87%	29

Berdasarkan Tabel 4.11, konfigurasi 50 *layer* memperoleh hasil terbaik pada dataset *original*. Nilai *recall macro* sebesar 85,96% menunjukkan bahwa konfigurasi ini memiliki kemampuan paling baik dalam mengenali setiap kelas dibandingkan skenario ablasi lainnya pada dataset *original*.

Hasil ablasi MobileNetV2 pada dataset *balanced quality-hybrid* ditunjukkan pada Tabel 4.12.

Tabel 4.12 Hasil ablasi MobileNetV2 pada dataset *balanced quality-hybrid*

Jumlah *Layer* yang dibuka	***Acuracy***	***Precision Macro***	***Recall Macro***	***F1-score Macro***	***Best Epoch***
0 *Layer*	79,91%	79,66%	79,75%	79,57%	-
10 *Layer*	87,20%	87,08%	86,98%	87,01%	30

20 *Layer*	90,18%	90,17%	90,16%	90,05%	29
30 *Layer*	91,67%	91,77%	91,56%	91,58%	30
50 *Layer*	92,26%	92,14%	92,21%	92,17%	29

Berdasarkan Tabel 4.12, konfigurasi 50 *layer* juga memperoleh hasil terbaik pada dataset *balanced quality-hybrid*. Peningkatan nilai *recall macro* dari 79,75% pada 0 *layer* menjadi 92,21% pada 50 *layer* menunjukkan bahwa *fine-tuning* membantu MobileNetV2 menyesuaikan fitur *pretrained* dengan karakteristik citra penyakit daun padi.

4.6.3 Evaluasi Dataset Original

Evaluasi pada dataset *original* dilakukan untuk melihat performa model pada kondisi data awal yang masih tidak seimbang, terutama karena jumlah data kelas Tungro lebih banyak dibandingkan kelas lainnya. Hasil evaluasi percobaan lanjutan dan model final ditunjukkan pada Tabel 4.13.

Tabel 4.13 Hasil evaluasi dataset original

Model	***Accuracy***	***Precision* Macro**	***Recall* Macro**	**F1-score Macro**
CNN Kustom awal	83,20%	81,11%	80,79%	80,74%
CNN Kustom + *Dropout*	86,98%	85,41%	85,22%	85,10%
MobileNetV2 *fine-tuning* 50 *layer*	87,89%	86,08%	85,96%	85,87%

Berdasarkan Tabel 4.13, MobileNetV2 *fine-tuning* 50 *layer* memperoleh hasil tertinggi pada dataset *original*. Jika dilihat dari *recall macro*, MobileNetV2 memperoleh 85,96%, CNN Kustom dengan *Dropout* memperoleh 85,22%, dan CNN Kustom awal memperoleh 80,79%. Hal ini menunjukkan bahwa *fine-tuning* MobileNetV2 mampu meningkatkan kemampuan model dalam mengenali setiap kelas pada dataset *original*.

Classification report CNN Kustom dengan *Dropout* pada dataset *original* ditunjukkan pada Tabel 4.14.

Tabel 4.14 *Classification report* CNN Kustom dengan *dropout* dataset original

Kategori	*Precision*	*Recall*	F1-score	Support
Bacterial Leaf Blight	0,74	0,79	0,76	95
Brown Spot	0,85	0,71	0,77	101
Healthy Rice Leaf	0,96	0,94	0,95	104
Leaf Blast	0,78	0,77	0,77	94
Leaf Scald	0,75	0,90	0,81	88
Sheath Blight	0,91	0,86	0,88	98
Tungro	0,99	1,00	1,00	188
Macro Average	**0,85**	**0,85**	**0,85**	**768**
Weighted Average	**0,87**	**0,87**	**0,87**	**768**

Berdasarkan Tabel 4.14, CNN Kustom dengan *Dropout* pada dataset *original* masih memiliki nilai *recall* yang lebih rendah pada kelas Brown Spot dan Leaf Blast dibandingkan kelas lainnya. Sementara itu, kelas Tungro dan Healthy Rice Leaf memiliki nilai *F1-score* yang tinggi, sehingga kedua kelas tersebut relatif lebih mudah dikenali oleh model.

Classification report CNN Kustom dengan *Dropout* pada dataset *original* ditunjukkan pada Tabel 4.15.

Tabel 4.15 *Classification report* MobileNetV2 *fine-tuning* 50 *layer* dataset original

Kategori	*Precision*	*Recall*	F1-score	Support
Bacterial Leaf Blight	0,73	0,85	0,79	95
Brown Spot	0,83	0,84	0,84	101
Healthy Rice Leaf	0,98	1,00	0,99	104
Leaf Blast	0,76	0,64	0,69	94
Leaf Scald	0,81	0,81	0,81	88
Sheath Blight	0,91	0,88	0,90	98
Tungro	1,00	1,00	1,00	188
Macro Average	**0,86**	**0,86**	**0,86**	**768**
Weighted Average	**0,88**	**0,88**	**0,88**	**768**

Berdasarkan Tabel 4.15, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *original* memiliki performa tinggi pada kelas Tungro, Healthy Rice Leaf, dan Sheath Blight. Namun, kelas Leaf Blast masih menjadi kelas yang lebih sulit dikenali karena memiliki *recall* 0,64 dan *F1-score* 0,69. Hal ini menunjukkan bahwa kemiripan visual antarpenyakit tetap memengaruhi kesalahan prediksi meskipun model telah menggunakan *transfer learning*.

4.6.4 Evaluasi Dataset *Balanced Quality-Hybrid*

Evaluasi pada dataset *balanced quality-hybrid* dilakukan untuk melihat performa model setelah distribusi data dibuat lebih seimbang menggunakan *quality-based filtering* pada kelas Tungro dan augmentasi pada kelas minoritas. Hasil evaluasi pada dataset *balanced quality-hybrid* ditunjukkan pada Tabel 4.16.

Tabel 4.16 Hasil evaluasi dataset *balanced quality-hybrid*

Model	***Accuracy***	***Precision* Macro**	***Recall* Macro**	**F1-score Macro**
MobileNetV2 + *fine-tuning*	76,34%	75,97%	76,07%	75,72%
CNN Kustom + *Dropout*	92,26%	92,14%	92,21%	92,17%

Berdasarkan Tabel 4.16, MobileNetV2 *fine-tuning* 50 *layer* memperoleh performa yang lebih tinggi dibandingkan CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid*. Perbedaan yang cukup besar pada *recall macro* menunjukkan bahwa MobileNetV2 lebih mampu mengenali data dari setiap kelas secara merata setelah dataset diseimbangkan.

Classification report CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* ditunjukkan pada Tabel 4.17.

Tabel 4.17 *Classification report* CNN Kustom dengan *dropout* dataset *balanced quality-hybrid*

Kategori	***Precision***	***Recall***	**F1-score**	**Support**
Bacterial Leaf Blight	0,80	0,62	0,70	105
Brown Spot	0,64	0,68	0,66	90
Healthy Rice Leaf	0,88	0,89	0,88	102
Leaf Blast	0,68	0,77	0,72	95
Leaf Scald	0,63	0,56	0,59	91
Sheath Blight	0,70	0,81	0,75	88
Tungro	0,99	1,00	1,00	101
Macro Average	**0,76**	**0,76**	**0,76**	**672**
Weighted Average	**0,77**	**0,76**	**0,76**	**672**

Berdasarkan Tabel 4.17, CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* masih mengalami kesulitan pada kelas Leaf Scald, Bacterial Leaf Blight, dan Brown Spot. Nilai *recall* pada Leaf Scald sebesar 0,56 menunjukkan bahwa sebagian citra Leaf Scald

belum dikenali dengan baik oleh model. Sebaliknya, kelas Tungro dan Healthy Rice Leaf memiliki performa yang lebih tinggi.

Classification report MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* ditunjukkan pada Tabel 4.18.

Tabel 4.18 *Classification report* MobileNetV2 *fine-tuning* 50 *layer* dataset *balanced quality-hybrid*

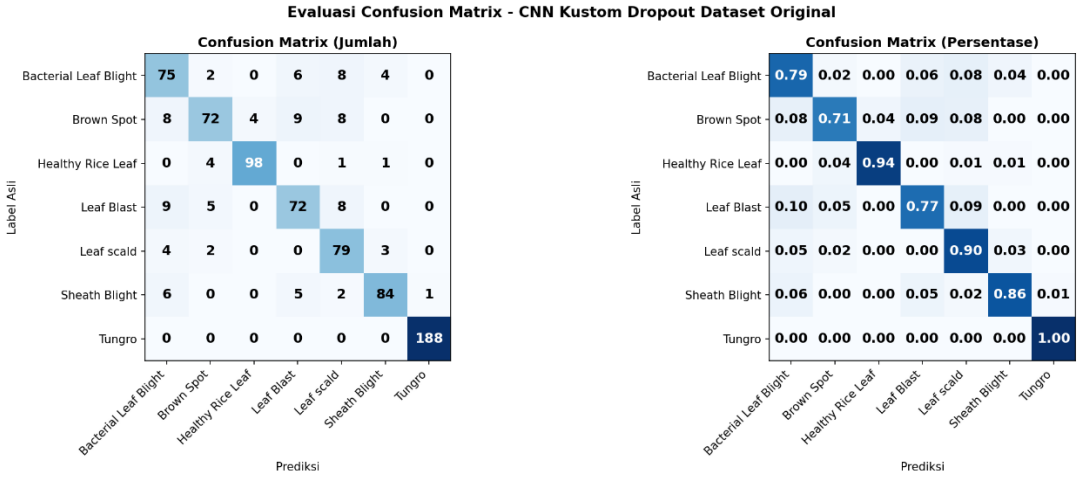
Kategori	***Precision***	***Recall***	F1-score	Support
Bacterial Leaf Blight	0,91	0,88	0,89	104
Brown Spot	0,90	0,90	0,90	94
Healthy Rice Leaf	0,98	0,98	0,98	107
Leaf Blast	0,82	0,83	0,82	93
Leaf Scald	0,91	0,93	0,92	89
Sheath Blight	0,92	0,93	0,93	90
Tungro	1,00	1,00	1,00	95
Macro Average	**0,92**	**0,92**	**0,92**	**672**
Weighted Average	**0,92**	**0,92**	**0,92**	**672**

Berdasarkan Tabel 4.18, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* menunjukkan performa yang stabil pada hampir seluruh kelas. Nilai *recall* setiap kelas berada pada kisaran 0,83 sampai 1,00. Hal ini menunjukkan bahwa model mampu mengenali setiap kelas penyakit dengan lebih merata dibandingkan CNN Kustom dengan *Dropout*.

4.6.5 Analisis Confusion Matrix

Confusion matrix digunakan untuk melihat pola prediksi benar dan salah pada setiap kelas. Pada *confusion matrix*, nilai pada diagonal utama menunjukkan jumlah data yang diprediksi dengan benar, sedangkan nilai di luar diagonal menunjukkan kesalahan prediksi antar kelas.

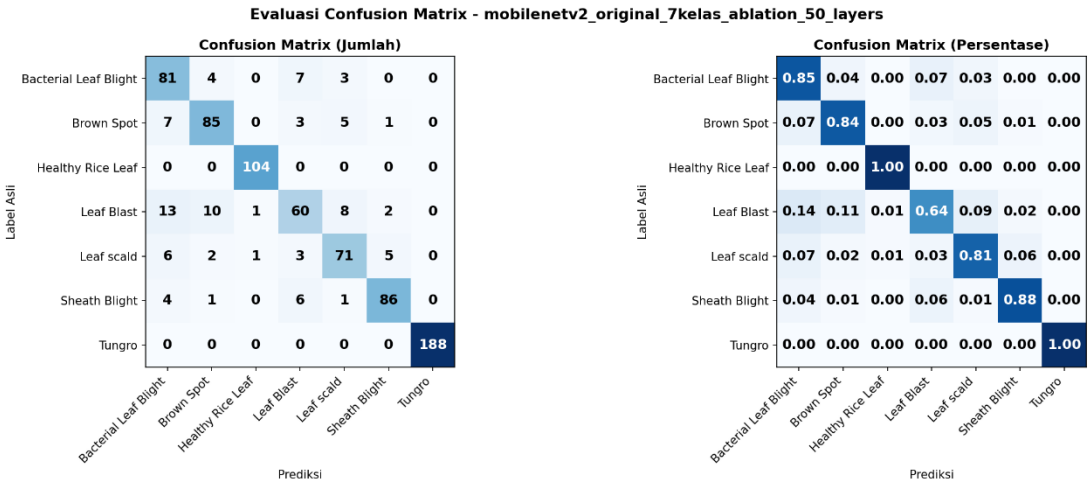
Confusion matrix CNN Kustom dengan *Dropout* pada dataset *original* ditunjukkan pada Gambar 4.16.



Gambar 4.16 *Confusion matrix* CNN Kustom dengan *dropout* dataset original

Berdasarkan Gambar 4.16, *confusion matrix* CNN Kustom dengan *Dropout* pada dataset *original* digunakan untuk melihat kelas yang masih sering tertukar. Kesalahan prediksi perlu diperhatikan terutama pada kelas yang memiliki ciri visual mirip, seperti Brown Spot, Leaf Blast, dan Leaf Scald.

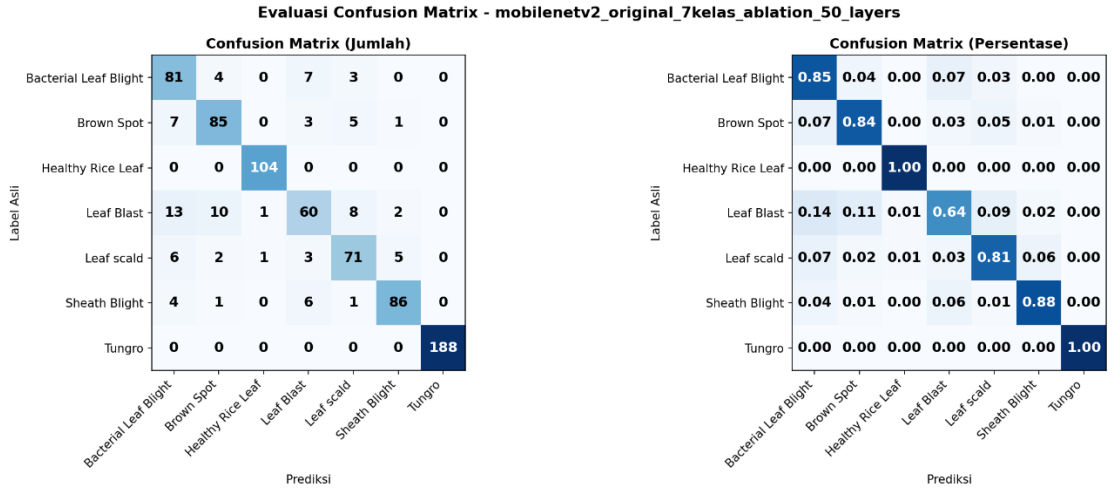
Confusion matrix MobileNetV2 *fine-tuning* 50 *layer* pada dataset *original* ditunjukkan pada Gambar 4.17.



Gambar 4.17 *Confusion Matrix MobileNetV2 fine-tuning 50 layer dataset original*

Berdasarkan Gambar 4.17, MobileNetV2 pada dataset *original* menunjukkan prediksi yang lebih stabil dibandingkan CNN Kustom awal. Namun, kelas Leaf Blast masih menjadi salah satu kelas yang relatif sulit dikenali karena memiliki kemiripan visual dengan penyakit daun lain.

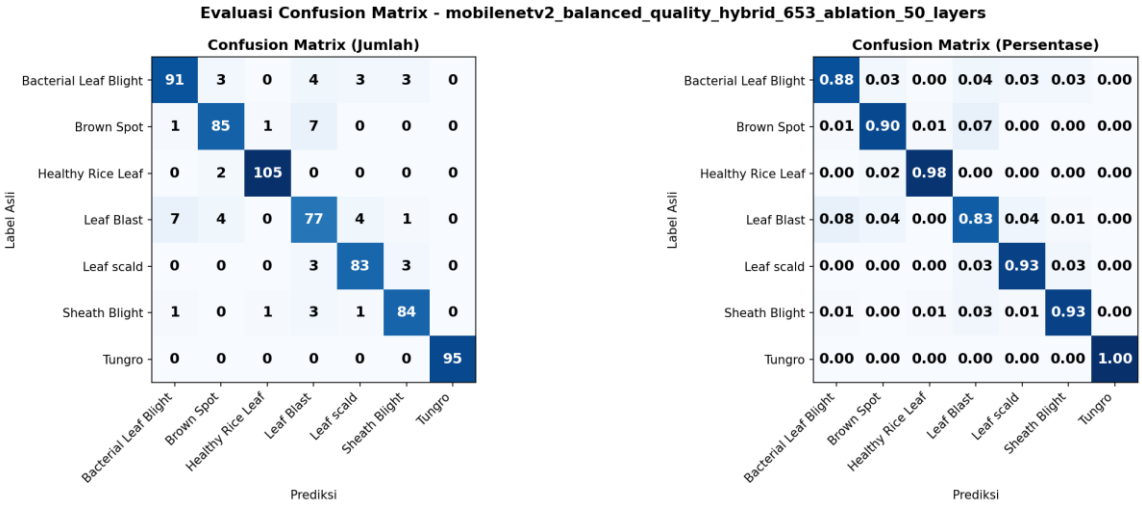
Confusion matrix CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* ditunjukkan pada Gambar 4.18.



Gambar 4.18 *Confusion Matrix CNN Kustom dengan dropout dataset balanced quality-hybrid*

Berdasarkan Gambar 4.18, CNN Kustom dengan *Dropout* pada dataset *balanced quality-hybrid* masih menunjukkan kesalahan prediksi pada beberapa kelas, terutama Leaf Scald, Bacterial Leaf Blight, dan Brown Spot. Hal ini sejalan dengan *classification report* yang menunjukkan nilai *recall* lebih rendah pada kelas-kelas tersebut.

Confusion matrix MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* ditunjukkan pada Gambar 4.19.



Gambar 4.19 *Confusion Matrix MobileNetV2 fine-tuning 50 layer dataset balanced quality-hybrid*

Berdasarkan Gambar 4.19, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* menghasilkan pola prediksi yang lebih stabil. Sebagian besar prediksi berada pada diagonal utama, yang menunjukkan bahwa model lebih mampu mengenali kelas dengan

benar. Kesalahan prediksi masih dapat terjadi pada kelas yang memiliki kemiripan visual, tetapi jumlahnya lebih rendah dibandingkan CNN Kustom.

4.6.6 Penentuan Model Terbaik

Penentuan model terbaik dilakukan berdasarkan hasil evaluasi pada data *testing*. Metrik yang lebih diprioritaskan dalam penelitian ini adalah *recall macro*, kemudian didukung oleh *F1-score macro* dan *accuracy*. *Recall macro* diprioritaskan karena penelitian ini berkaitan dengan klasifikasi penyakit daun padi, sehingga kemampuan model dalam mengenali setiap kelas penyakit menjadi penting.

Accuracy tetap digunakan untuk melihat performa model secara umum, tetapi tidak dijadikan satu-satunya dasar pemilihan model terbaik. Hal ini disebabkan oleh adanya perbedaan jumlah data pada setiap kelas, terutama pada dataset *original* yang memiliki kelas Tungro sebagai kelas mayoritas. Oleh karena itu, *recall macro* digunakan untuk melihat apakah model mampu mengenali setiap kelas secara lebih merata.

Ringkasan hasil evaluasi model utama ditunjukkan pada Tabel 4.19.

Tabel 4.19 Ringkasan penentuan model terbaik

Dataset	**Model**	***Accuracy***	***Recal Macro***	***F1-Score Macro***	**Keterangan**
Original	CNN Kustom awal	83,20%	80,79%	80,74%	*Baseline* CNN
Original	CNN Kustom + *Dropout*	86,98%	85,22%	85,10%	CNN dengan regularisasi *Dropout*
Original	MobileNet V2 *fine-tuning* 50 layer	87,89%	85,96%	85,87%	Hasil terbaik pada dataset *original*
Balanced quality-hybrid	CNN Kustom + *Dropout*	76,34%	76,07%	75,72%	CNN pada dataset *balanced quality-hybrid*
Balanced quality-hybrid	MobileNet V2 *fine-tuning* 50 layer	92,26%	92.21%	92,17%	Hasil terbaik keseluruhan

Berdasarkan Tabel 4.19, MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid* menjadi model terbaik pada penelitian ini. Model tersebut memperoleh *accuracy* sebesar **92,26%**, *recall macro* sebesar **92,21%**, dan *F1-score macro* sebesar **92,17%**.

Nilai *recall macro* yang tinggi menunjukkan bahwa model mampu mengenali setiap kelas penyakit secara lebih merata.

Pada dataset *original*, MobileNetV2 *fine-tuning* 50 *layer* juga memperoleh hasil tertinggi dengan *recall macro* sebesar **85,96%**. Nilai tersebut sedikit lebih tinggi dibandingkan CNN Kustom dengan *Dropout* yang memperoleh *recall macro* sebesar **85,22%**. Hal ini menunjukkan bahwa MobileNetV2 tetap memiliki performa yang lebih baik pada dataset *original*, meskipun selisihnya tidak terlalu besar.

Pada dataset *balanced quality-hybrid*, perbedaan performa antara MobileNetV2 dan CNN Kustom terlihat lebih besar. MobileNetV2 memperoleh *recall macro* sebesar **92,21%**, sedangkan CNN Kustom dengan *Dropout* memperoleh *recall macro* sebesar **76,07%**. Hal ini menunjukkan bahwa MobileNetV2 lebih mampu memanfaatkan dataset yang telah diseimbangkan dengan *quality-based filtering* dan augmentasi.

Hasil tersebut menunjukkan bahwa strategi *balancing* dataset berbasis *quality-based filtering* dan augmentasi lebih efektif ketika dikombinasikan dengan MobileNetV2. MobileNetV2 memiliki bobot *pretrained* ImageNet yang membantu proses ekstraksi fitur visual, sedangkan *fine-tuning* memungkinkan model menyesuaikan sebagian bobot *pretrained* dengan karakteristik citra penyakit daun padi.

Dengan demikian, model terbaik pada penelitian ini adalah **MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid***. Model tersebut dipilih karena memiliki nilai *recall macro*, *F1-score macro*, dan *accuracy* tertinggi dibandingkan skenario lainnya.

4.6.7 Pengujian Foto Lapangan

Selain evaluasi menggunakan data *testing* dari dataset, dilakukan juga pengujian tambahan menggunakan foto daun padi yang diambil secara langsung di area persawahan. Pengujian ini bertujuan untuk melihat respons model terhadap citra baru yang berasal dari kondisi lapangan. Foto lapangan tidak digunakan pada proses pelatihan, validasi, maupun *testing* utama.

Pengujian foto lapangan dilakukan menggunakan 11 citra daun padi. Foto tersebut tidak memiliki label *ground truth* yang terverifikasi seperti dataset utama, sehingga pengujian ini tidak digunakan untuk menghitung metrik *accuracy*, *precision*, *recall*, maupun *F1-score*. Hasil pengujian digunakan sebagai analisis tambahan secara kualitatif.

Model yang digunakan pada pengujian foto lapangan adalah CNN Kustom dengan *Dropout* dan MobileNetV2 *fine-tuning* 50 *layer*. Kedua model yang digunakan merupakan

model hasil pelatihan pada dataset *balanced quality-hybrid*. Hasil prediksi foto lapangan ditunjukkan pada Tabel 4.20.

Tabel 4.20 Hasil Prediksi Foto Lapangan

File	**Prediksi CNN Kustom + *Dropout***	***Confidence* CNN**	**Prediksi MobileNetV2**	***Confidence MobileNetV2***	**Prediksi Sama**
Foto_001.jpeg	Healthy Rice Leaf	100,00%	Healthy Rice Leaf	84,25%	Ya
Foto_002.jpeg	Healthy Rice Leaf	99,41%	Healthy Rice Leaf	92,57%	Ya
Foto_003.jpeg	Bacterial Leaf Blight	45,65%	Healthy Rice Leaf	91,62%	Tidak
Foto_004.jpeg	Healthy Rice Leaf	98,90%	Healthy Rice Leaf	99,45%	Ya
Foto_005.jpeg	Healthy Rice Leaf	97,47%	Healthy Rice Leaf	99,71%	Ya
Foto_006.jpeg	Healthy Rice Leaf	97,42	Healthy Rice Leaf	99,33%	Ya
Foto_007.jpeg	Healthy Rice Leaf	100,00%	Healthy Rice Leaf	99,96%	Ya
Foto_008.jpeg	Healthy Rice Leaf	97,66%	Tungro	55,15%	Tidak
Foto_009.jpeg	Healthy Rice Leaf	99,83%	Sheath Blight	84,37%	Tidak
Foto_010.jpeg	Healthy Rice Leaf	99,97%	Healthy Rice Leaf	56,56%	Ya
Foto_011.jpeg	Healthy Rice Leaf	99,68%	Healthy Rice Leaf	78,65%	Ya

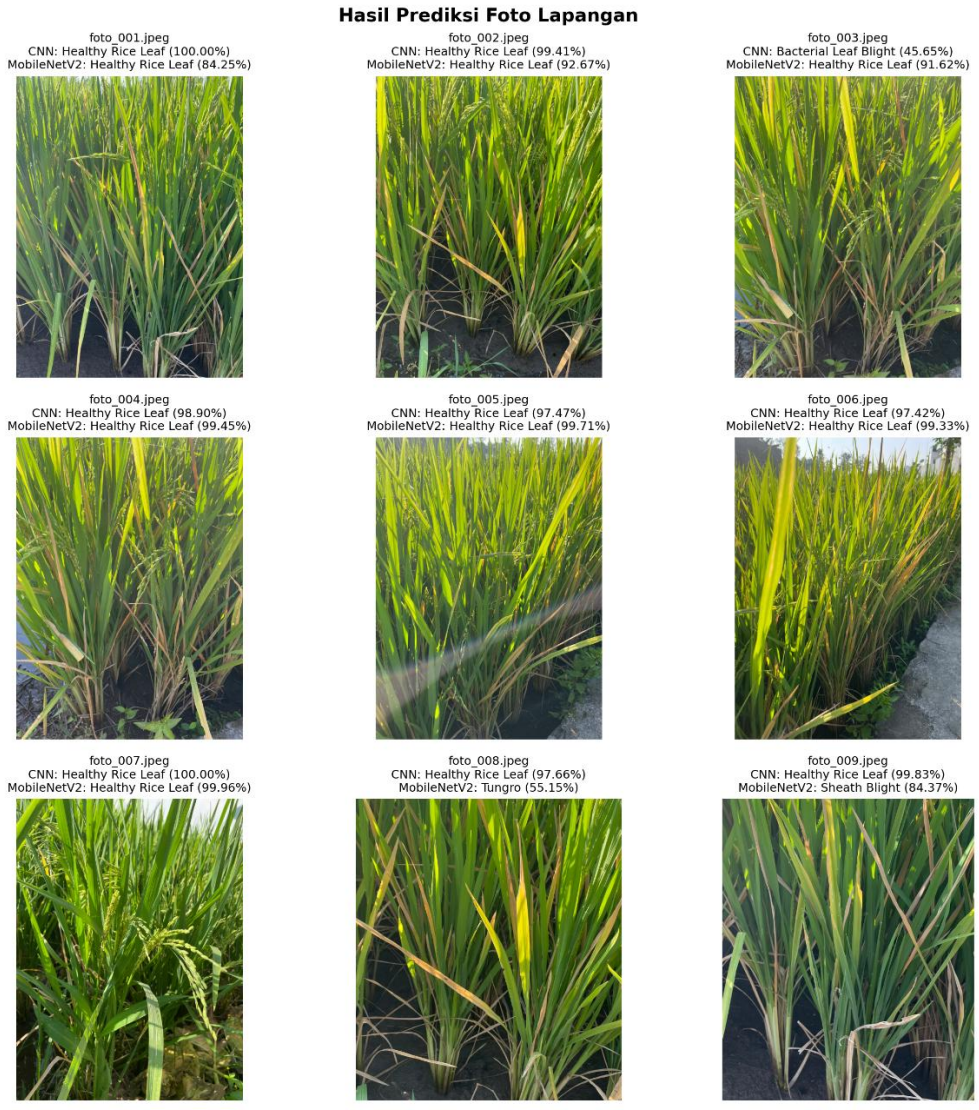
Berdasarkan Tabel 4.20, terdapat 8 dari 11 foto yang memperoleh hasil prediksi sama dari kedua model. Pada foto tersebut, CNN Kustom dengan *Dropout* dan MobileNetV2 sama-sama memprediksi kelas Healthy Rice Leaf. Hal ini menunjukkan bahwa kedua model cenderung memberikan hasil yang sama pada sebagian besar foto lapangan yang diuji.

Terdapat 3 foto yang memperoleh hasil prediksi berbeda, yaitu foto_003.jpeg, foto_008.jpeg, dan foto_009.jpeg. Pada foto_003.jpeg, CNN Kustom dengan *Dropout*

memprediksi Bacterial Leaf Blight dengan *confidence* 45,65%, sedangkan MobileNetV2 memprediksi Healthy Rice Leaf dengan *confidence* 91,62%. Pada foto_008.jpeg, CNN Kustom dengan *Dropout* memprediksi Healthy Rice Leaf, sedangkan MobileNetV2 memprediksi Tungro. Pada foto_009.jpeg, CNN Kustom dengan *Dropout* memprediksi Healthy Rice Leaf, sedangkan MobileNetV2 memprediksi Sheath Blight.

Perbedaan prediksi tersebut menunjukkan bahwa kedua model dapat memberikan respons yang berbeda terhadap citra lapangan. Hal ini dapat disebabkan oleh perbedaan arsitektur, proses ekstraksi fitur, kondisi pencahayaan, posisi pengambilan gambar, latar belakang, dan kualitas visual foto. Karena foto lapangan tidak memiliki label *ground truth* yang terverifikasi, hasil ini tidak digunakan untuk menentukan model mana yang benar atau salah, tetapi digunakan untuk melihat kecenderungan prediksi model terhadap citra baru.

Visualisasi hasil prediksi foto lapangan ditunjukkan pada Gambar 4.20.



Gambar 4.20 Hasil prediksi foto lapangan menggunakan CNN Kustom dan MobileNetV2

Berdasarkan Gambar 4.20, hasil prediksi foto lapangan memperlihatkan bahwa sebagian besar foto diprediksi sebagai Healthy Rice Leaf oleh kedua model. Namun, MobileNetV2 memberikan prediksi berbeda pada beberapa foto, yaitu Tungro dan Sheath Blight. Hasil ini menunjukkan bahwa pengujian lapangan dapat memberikan gambaran tambahan mengenai respons model terhadap citra nyata di luar dataset utama.

Dengan demikian, pengujian foto lapangan digunakan sebagai pengujian tambahan secara kualitatif. Pengujian ini tidak menggantikan evaluasi utama menggunakan data *testing*, tetapi memberikan gambaran awal mengenai bagaimana model merespons citra daun padi yang diambil langsung dari lingkungan nyata.

4.7 Analisis Perbandingan

Analisis perbandingan dilakukan untuk menjawab rumusan masalah penelitian berdasarkan hasil evaluasi yang telah diperoleh. Analisis ini membahas performa CNN Kustom, performa MobileNetV2, perbandingan kedua model pada dataset *original*, perbandingan kedua model pada dataset *balanced quality-hybrid*, serta pengaruh ketidakseimbangan data pada kelas Tungro terhadap hasil evaluasi model.

4.7.1 Performa CNN Kustom dalam Klasifikasi Tujuh Kondisi Daun Padi

Berdasarkan hasil evaluasi, CNN Kustom mampu melakukan klasifikasi tujuh kondisi daun padi, tetapi performanya berbeda pada setiap konfigurasi dan skenario dataset. Pada dataset *original*, CNN Kustom awal memperoleh *accuracy* 83,20%, *recall macro* 80,79%, dan *F1-score macro* 80,74%. Setelah ditambahkan *Dropout*, performa CNN Kustom pada dataset *original* meningkat menjadi *accuracy* 86,98%, *recall macro* 85,22%, dan *F1-score macro* 85,10%.

Hasil tersebut menunjukkan bahwa penambahan *Dropout* membantu meningkatkan kemampuan generalisasi CNN Kustom pada dataset *original*. Namun, pada dataset *balanced quality-hybrid*, CNN Kustom dengan *Dropout* memperoleh *accuracy* 76,34%, *recall macro* 76,07%, dan *F1-score macro* 75,72%. Nilai tersebut menunjukkan bahwa CNN Kustom masih cukup sensitif terhadap variasi citra dan perubahan distribusi data karena model dilatih dari awal tanpa bobot *pretrained*.

Dengan demikian, rumusan masalah pertama dapat dijawab bahwa CNN Kustom dapat digunakan untuk mengklasifikasikan tujuh kondisi daun padi, terutama setelah ditambahkan

Dropout. Namun, performanya belum selalu stabil pada setiap skenario dataset, khususnya pada dataset *balanced quality-hybrid.*

4.7.2 Performa MobileNetV2 Berbasis Transfer Learning

MobileNetV2 menunjukkan performa yang lebih baik dibandingkan CNN Kustom pada skenario akhir. Pada dataset *original*, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *accuracy* 87,89%, *recall macro* 85,96%, dan *F1-score macro* 85,87%. Pada dataset *balanced quality-hybrid*, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *accuracy* 92,26%, *recall macro* 92,21%, dan *F1-score macro* 92,17%.

Hasil tersebut menunjukkan bahwa MobileNetV2 berbasis *transfer learning* mampu mengenali tujuh kondisi daun padi dengan baik. Bobot *pretrained* ImageNet membantu model dalam mengekstraksi fitur visual umum, sedangkan *fine-tuning* membantu menyesuaikan sebagian bobot *pretrained* dengan karakteristik citra penyakit daun padi. Berdasarkan percobaan ablasi, konfigurasi 50 *layer* akhir yang dibuka memberikan hasil terbaik pada dataset *original* maupun dataset *balanced quality-hybrid*.

Dengan demikian, rumusan masalah kedua dapat dijawab bahwa MobileNetV2 berbasis *transfer learning* memiliki performa yang baik dalam klasifikasi tujuh kondisi daun padi, terutama ketika dilakukan *fine-tuning* pada sebagian *layer* akhir.

4.7.3 Perbandingan CNN Kustom dan MobileNetV2 pada Dataset Original

Pada dataset *original*, performa MobileNetV2 *fine-tuning* 50 *layer* lebih tinggi dibandingkan CNN Kustom awal dan CNN Kustom dengan *Dropout*. CNN Kustom awal memperoleh *recall macro* 80,79%, CNN Kustom dengan *Dropout* memperoleh *recall macro* 85,22%, sedangkan MobileNetV2 *fine-tuning* 50 *layer* memperoleh *recall macro* 85,96%.

Perbedaan antara MobileNetV2 dan CNN Kustom dengan *Dropout* pada dataset *original* tidak terlalu besar, tetapi MobileNetV2 tetap memiliki nilai *recall macro* dan *F1-score macro* yang lebih tinggi. Hal ini menunjukkan bahwa *transfer learning* memberikan keuntungan dalam proses klasifikasi karena model tidak belajar seluruh fitur visual dari awal.

Hasil tersebut menunjukkan bahwa *fine-tuning* membuat MobileNetV2 lebih sesuai dengan domain citra daun padi dibandingkan hanya menggunakan *feature extraction*. Namun, pada hasil akhir penelitian, performa MobileNetV2 masih sedikit di bawah CNN Kustom dengan *Dropout* pada dataset *balanced*.

Dengan demikian, rumusan masalah ketiga dapat dijawab bahwa pada dataset *original* yang tidak seimbang, MobileNetV2 *fine-tuning* 50 *layer* memberikan performa terbaik dibandingkan CNN Kustom. Namun, CNN Kustom dengan *Dropout* juga menunjukkan hasil yang kompetitif pada dataset *original*.

4.7.4 Perbandingan CNN Kustom dan MobileNetV2 pada Dataset *Balanced Quality-Hybrid*

Pada dataset *balanced quality-hybrid*, perbedaan performa antara CNN Kustom dengan *Dropout* dan MobileNetV2 *fine-tuning* 50 *layer* terlihat lebih besar. CNN Kustom dengan *Dropout* memperoleh *accuracy* 76,34%, *recall macro* 76,07%, dan *F1-score macro* 75,72%. Sementara itu, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *accuracy* 92,26%, *recall macro* 92,21%, dan *F1-score macro* 92,17%.

Hasil tersebut menunjukkan bahwa MobileNetV2 lebih mampu memanfaatkan dataset *balanced quality-hybrid* dibandingkan CNN Kustom. Dataset *balanced quality-hybrid* memiliki distribusi kelas yang lebih merata dan kualitas citra yang lebih terkontrol. Kondisi tersebut membantu MobileNetV2 menyesuaikan fitur *pretrained* melalui *fine-tuning*, sedangkan CNN Kustom yang dilatih dari awal masih belum mampu menghasilkan performa yang sama stabilnya.

Dengan demikian, rumusan masalah keempat dapat dijawab bahwa pada dataset *balanced quality-hybrid*, MobileNetV2 *fine-tuning* 50 *layer* memiliki performa yang lebih baik dibandingkan CNN Kustom dengan *Dropout*.

4.7.5 Pengaruh Ketidakseimbangan Data pada Kelas Tungro

Ketidakseimbangan data pada kelas Tungro berpengaruh terhadap hasil evaluasi model. Pada dataset *original*, kelas Tungro memiliki jumlah data paling banyak, yaitu 1.308 citra, sedangkan kelas lain berada pada kisaran 628 sampai 653 citra. Kondisi ini dapat membuat evaluasi model terlihat baik secara umum karena kelas mayoritas lebih dominan dalam data.

Setelah dataset diseimbangkan menggunakan *quality-based filtering* dan augmentasi, MobileNetV2 mengalami peningkatan performa. Pada dataset *original*, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *recall macro* 85,96%, sedangkan pada dataset *balanced quality-hybrid* memperoleh *recall macro* 92,21%. Peningkatan ini menunjukkan bahwa distribusi data yang lebih seimbang dapat membantu model mengenali setiap kelas dengan lebih merata.

Namun, pengaruh *balancing* dataset tidak selalu sama pada setiap arsitektur model. Pada CNN Kustom, performa dataset *balanced quality-hybrid* lebih rendah dibandingkan dataset *original*. Hal ini menunjukkan bahwa selain distribusi data, arsitektur model, kemampuan ekstraksi fitur, bobot *pretrained*, variasi citra, dan kemiripan visual antar kelas juga memengaruhi hasil evaluasi.

Dengan demikian, rumusan masalah kelima dapat dijawab bahwa ketidakseimbangan data pada kelas Tungro berpengaruh terhadap hasil evaluasi model klasifikasi penyakit daun padi. Penyeimbangan dataset dapat membantu meningkatkan performa, terutama pada MobileNetV2, tetapi dampaknya tetap bergantung pada karakteristik model yang digunakan.

4.8 Analisis Aspek Komputasional

Selain evaluasi berdasarkan metrik klasifikasi, penelitian ini juga menganalisis aspek komputasional dari model CNN Kustom dan MobileNetV2. Analisis ini dilakukan karena MobileNetV2 dipilih sebagai model *transfer learning* yang dirancang ringan dan efisien. Oleh karena itu, perbandingan model tidak hanya dilihat dari *accuracy*, *recall macro*, dan *F1-score macro*, tetapi juga dari jumlah parameter, ukuran model, serta waktu inferensi.

Aspek komputasional yang dianalisis meliputi jumlah parameter model, jumlah parameter yang dilatih, ukuran file model, jumlah epoch berjalan, dan waktu rata-rata prediksi satu citra. Jumlah parameter digunakan untuk melihat kompleksitas model. Ukuran file model digunakan untuk melihat kebutuhan penyimpanan. Waktu inferensi digunakan untuk melihat kecepatan model ketika melakukan prediksi citra baru.

4.8.1 Jumlah Parameter dan Ukuran Model

Jumlah parameter dan ukuran model diperoleh dari model yang telah disimpan pada folder *output*s. Perbandingan jumlah parameter dan ukuran file model ditunjukkan pada Tabel 4.21.

Tabel 4.21 Perbandingan jumlah parameter dan Ukuran Model

Model	**Dataset**	**Total Parameter**	***Trainable* Parameter**	***Non-trainable* Parameter**	**Ukutan File Model**
CNN Kustom + *Dropout*	Original	8.413.223	8.413.223	0	96,34 MB
CNN Kustom + *Dropout*	*Balanced quality-hybrid*	8.413.223	8.413.223	0	96,34 MB

MobileNetV2 *fine-tuning* 50 layer	Original	2.266.951	1.843.399	423.552	23,06 MB
MobileNetV2 *fine-tuning* 50 layer	*Balanced quality-hybrid*	2.266.951	1.843.399	423.552	23,06 MB

Berdasarkan Tabel 4.21, CNN Kustom dengan *Dropout* memiliki jumlah parameter yang lebih besar dibandingkan MobileNetV2 *fine-tuning* 50 *layer*. CNN Kus tom memiliki 8.413.223 parameter dan seluruhnya bersifat *trainable*. Sementara itu, MobileNetV2 memiliki 2.266.951 parameter, dengan 1.843.399 parameter *trainable* dan 423.552 parameter *non-trainable* pada skenario *fine-tuning* 50 *layer*.

Ukuran file model CNN Kustom dengan *Dropout* juga lebih besar, yaitu sekitar 96,34 MB. Sementara itu, MobileNetV2 *fine-tuning* 50 *layer* memiliki ukuran file sekitar 23,06 MB. Perbedaan ini menunjukkan bahwa MobileNetV2 memiliki kebutuhan penyimpanan yang lebih kecil dibandingkan CNN Kustom pada penelitian ini.

4.8.2 Parameter *Trainable* pada Ablasi MobileNetV2

Pada MobileNetV2, jumlah parameter *trainable* berubah sesuai dengan jumlah *layer* akhir yang dibuka pada tahap *fine-tuning*. Semakin banyak *layer* yang dibuka, semakin banyak bobot *pretrained* yang ikut dilatih. Perbandingan parameter *trainable* pada skenario ablasi MobileNetV2 ditunjukkan pada Tabel 4.22.

Tabel 4.22 Jumlah parameter *trainable* pada ablasi MobileNetV2

Jumlah Layer Akhir Dibuka	***Trainable* Parameter**	***Non-trainable* Parameter**	**Keterangan**
0 layer	8.967	2.257.984	*Feature extraction*, hanya layer klasifikasi tambahan yang dilatih
10 layer	734.407	1.532.544	*Fine-tuning* ringan
20 layer	1.203.847	1.063.104	*Fine-tuning* dengan layer akhir lebih banyak
30 layer	1.519.687	747.264	*Fine-tuning* menengah
50 layer	1.843.399	423.552	*Fine-tuning dengan cakupan layer akhir lebih besar*

Berdasarkan Tabel 4.22, skenario 0 *layer* memiliki parameter *trainable* paling sedikit karena seluruh *base model* MobileNetV2 masih dibekukan. Pada skenario 50 *layer*, jumlah parameter *trainable* meningkat menjadi 1.843.399 karena lebih banyak *layer* akhir MobileNetV2 yang dibuka. Peningkatan jumlah parameter *trainable* ini membuat model memiliki ruang adaptasi yang lebih besar terhadap karakteristik citra daun padi, tetapi juga meningkatkan kebutuhan komputasi selama pelatihan.

4.8.3 Indokator Waktu Pelatihan

Waktu pelatihan aktual dalam satuan menit tidak dicatat secara otomatis pada seluruh log percobaan. Oleh karena itu, analisis waktu pelatihan pada penelitian ini dilihat melalui indikator jumlah epoch berjalan dan *best epoch*. Jumlah epoch berjalan menunjukkan banyaknya proses pelatihan yang benar-benar dilakukan sebelum berhenti, sedangkan *best epoch* menunjukkan epoch dengan *validation loss* terbaik yang digunakan sebagai dasar penyimpanan model melalui *ModelCheckpoint*.

Ringkasan indikator pelatihan pada skenario MobileNetV2 *fine-tuning* 50 *layer* ditunjukkan pada Tabel 4.23.

Tabel 4.23 Indikator pelatihan MobileNetV2 *fine-tuning* 50 *layer*

Dataset	**Epoch Maksimum**	**Epoch Berjalan**	**Epoch Terbaik**	***Trainable* Parameter**
Original	30	30	29	1.843.399
Balanced quality-hybrid	30	30	29	1.843.399

Berdasarkan Tabel 4.23, MobileNetV2 *fine-tuning* 50 *layer* berjalan sampai 30 epoch pada dataset *original* maupun dataset *balanced quality-hybrid*. Model terbaik diperoleh pada epoch ke-29 untuk kedua skenario dataset. Hal ini menunjukkan bahwa proses *fine-tuning* masih membutuhkan hampir seluruh batas epoch maksimum untuk memperoleh *validation loss* terbaik.

Pada CNN Kustom, seluruh parameter model bersifat *trainable* sejak awal karena model tidak menggunakan bobot *pretrained*. Hal ini berbeda dengan MobileNetV2 yang dapat mengatur jumlah parameter *trainable* melalui *feature extraction* dan *fine-tuning*. Dengan demikian, MobileNetV2 memberikan fleksibilitas komputasi karena jumlah *layer* yang dilatih dapat disesuaikan melalui skenario ablasi.

4.8.4 Waktu Inferensi

Waktu inferensi diukur untuk mengetahui rata-rata waktu yang dibutuhkan model dalam memprediksi satu citra. Pengukuran dilakukan dengan memuat model tersimpan, melakukan satu kali prediksi awal sebagai *warm-up*, kemudian menghitung rata-rata waktu prediksi dari 20 kali pengujian menggunakan *input* berukuran 256 × 256 × 3. Hasil pengukuran waktu inferensi ditunjukkan pada Tabel 4.24.

Tabel 4.24 Perbandingan waktu inferensi

Model	**Dataset**	**Rata-rata Waktu Inferansi per CItra**
CNN Kustom + *Dropout*	Original	54,75 ms
CNN Kustom + *Dropout*	*Balanced quality-hybrid*	56,91 ms
MobileNetV2 *fine-tuning* 50 layer	Original	78,49 ms
MobileNetV2 *fine-tuning* 50 layer	*Balanced quality-hybrid*	75,67 ms

Berdasarkan Tabel 4.24, CNN Kustom dengan *Dropout* memiliki waktu inferensi per citra yang lebih cepat pada lingkungan pengujian ini. Namun, MobileNetV2 memiliki jumlah parameter dan ukuran file model yang jauh lebih kecil. Hal ini menunjukkan bahwa efisiensi model tidak hanya dapat dinilai dari satu aspek, tetapi perlu dilihat dari beberapa indikator, seperti jumlah parameter, ukuran model, dan waktu prediksi.

Perbedaan waktu inferensi dapat dipengaruhi oleh struktur operasi pada masing-masing arsitektur, lingkungan perangkat keras, serta cara *framework* menjalankan *layer* model. MobileNetV2 dirancang ringan melalui penggunaan *depthwise separable convolution*, *inverted residual*, dan *linear bottleneck*, tetapi pada pengujian satu citra di lingkungan ini waktu prediksinya sedikit lebih tinggi dibandingkan CNN Kustom. Oleh karena itu, pada penelitian ini MobileNetV2 lebih efisien dari sisi jumlah parameter dan ukuran model, sedangkan CNN Kustom lebih cepat pada pengujian inferensi satu citra.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa model CNN Kustom dan MobileNetV2 dapat digunakan untuk melakukan klasifikasi tujuh kondisi daun padi berdasarkan citra digital, yaitu Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight, dan Tungro. Setiap model menunjukkan performa yang berbeda pada dataset *original* dan dataset *balanced quality-hybrid*.

CNN Kustom mampu mengklasifikasikan tujuh kondisi daun padi, tetapi performanya masih dipengaruhi oleh konfigurasi model dan skenario dataset yang digunakan. Pada dataset *original*, CNN Kustom awal memperoleh *accuracy* sebesar 83,20%, *recall macro* sebesar 80,79%, dan *F1-score macro* sebesar 80,74%. Setelah ditambahkan *Dropout*, performa CNN Kustom pada dataset *original* meningkat menjadi *accuracy* sebesar 86,98%, *recall macro* sebesar 85,22%, dan *F1-score macro* sebesar 85,10%. Hal ini menunjukkan bahwa penambahan *Dropout* dapat membantu meningkatkan kemampuan generalisasi CNN Kustom pada dataset *original*.

MobileNetV2 berbasis *transfer learning* menunjukkan performa yang lebih baik dibandingkan CNN Kustom pada skenario akhir penelitian. Pada dataset *original*, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *accuracy* sebesar 87,89%, *recall macro* sebesar 85,96%, dan *F1-score macro* sebesar 85,87%. Pada dataset *balanced quality-hybrid*, MobileNetV2 *fine-tuning* 50 *layer* memperoleh hasil terbaik dengan *accuracy* sebesar 92,26%, *recall macro* sebesar 92,21%, dan *F1-score macro* sebesar 92,17%. Nilai *recall macro* yang tinggi menunjukkan bahwa model mampu mengenali setiap kelas daun padi secara lebih merata.

Perbandingan pada dataset *original* menunjukkan bahwa MobileNetV2 *fine-tuning* 50 *layer* memiliki performa lebih tinggi dibandingkan CNN Kustom awal dan CNN Kustom dengan *Dropout*. Namun, selisih performa antara MobileNetV2 dan CNN Kustom dengan *Dropout* pada dataset *original* tidak terlalu besar. Hal ini menunjukkan bahwa CNN Kustom dengan *Dropout* masih memiliki performa yang cukup kompetitif pada dataset *original*, meskipun MobileNetV2 tetap lebih unggul karena memanfaatkan bobot *pretrained* ImageNet.

Perbandingan pada dataset *balanced quality-hybrid* menunjukkan perbedaan performa yang lebih besar. CNN Kustom dengan *Dropout* memperoleh *accuracy* sebesar 76,34%, *recall*

macro sebesar 76,07%, dan *F1-score macro* sebesar 75,72%. Sementara itu, MobileNetV2 *fine-tuning* 50 *layer* memperoleh *accuracy* sebesar 92,26%, *recall macro* sebesar 92,21%, dan *F1-score macro* sebesar 92,17%. Hasil ini menunjukkan bahwa MobileNetV2 lebih mampu memanfaatkan dataset *balanced quality-hybrid* dibandingkan CNN Kustom.

Ketidakseimbangan data pada kelas Tungro terbukti berpengaruh terhadap hasil evaluasi model. Pada dataset *original*, kelas Tungro memiliki jumlah data paling banyak dibandingkan kelas lainnya, sehingga evaluasi model dapat dipengaruhi oleh dominasi kelas mayoritas. Setelah dataset diseimbangkan menggunakan *quality-based filtering* pada kelas Tungro dan augmentasi pada kelas minoritas, MobileNetV2 menunjukkan peningkatan performa yang lebih baik dan lebih stabil. Namun, pengaruh *balancing* dataset tidak selalu sama pada setiap arsitektur model, karena CNN Kustom yang dilatih dari awal masih lebih sensitif terhadap variasi citra dan kemiripan visual antar kelas.

Berdasarkan percobaan ablasi MobileNetV2, konfigurasi *fine-tuning* dengan membuka 50 *layer* akhir memberikan hasil terbaik dibandingkan skenario 0, 10, 20, dan 30 *layer*. Oleh karena itu, pemilihan 50 *layer* akhir sebagai konfigurasi final MobileNetV2 didasarkan pada hasil percobaan, bukan ditentukan secara langsung tanpa pembanding.

Dari aspek komputasional, MobileNetV2 memiliki jumlah parameter dan ukuran model yang lebih kecil dibandingkan CNN Kustom dengan *Dropout*. MobileNetV2 *fine-tuning* 50 *layer* memiliki 2.266.951 parameter dengan ukuran model sekitar 23,06 MB, sedangkan CNN Kustom dengan *Dropout* memiliki 8.413.223 parameter dengan ukuran model sekitar 96,34 MB. Meskipun CNN Kustom memiliki waktu inferensi satu citra yang lebih cepat pada lingkungan pengujian ini, MobileNetV2 tetap lebih unggul secara keseluruhan karena memberikan kombinasi antara performa klasifikasi yang tinggi dan ukuran model yang lebih kecil.

Pengujian tambahan menggunakan foto lapangan menunjukkan bahwa model dapat memberikan prediksi terhadap citra daun padi yang diambil langsung dari lingkungan nyata. Dari 11 foto lapangan yang diuji, terdapat 8 foto yang memperoleh hasil prediksi sama dari CNN Kustom dengan *Dropout* dan MobileNetV2. Karena foto lapangan tidak memiliki label *ground truth* yang terverifikasi, hasil pengujian ini tidak digunakan untuk menghitung metrik evaluasi, tetapi digunakan sebagai analisis tambahan secara kualitatif untuk melihat respons model terhadap citra baru di luar dataset utama.

Secara keseluruhan, model terbaik pada penelitian ini adalah **MobileNetV2 *fine-tuning* 50 *layer* pada dataset *balanced quality-hybrid***. Model tersebut dipilih karena memperoleh

nilai *recall macro*, *F1-score macro*, dan *accuracy* tertinggi, serta memiliki ukuran model yang lebih kecil dibandingkan CNN Kustom. Dengan demikian, MobileNetV2 *fine-tuning* 50 *layer* lebih sesuai digunakan pada penelitian klasifikasi penyakit daun padi berbasis citra digital.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan penelitian selanjutnya.

Pertama, proses penyusunan dataset sebaiknya dilakukan dengan jumlah citra yang lebih banyak dan lebih merata pada setiap kelas. Dataset yang digunakan pada penelitian ini masih memiliki variasi visual yang cukup besar, baik dari sisi pencahayaan, latar belakang, sudut pengambilan gambar, maupun kualitas citra. Penelitian selanjutnya dapat menggunakan dataset yang lebih terkontrol atau melakukan validasi kualitas citra secara lebih sistematis sebelum proses pelatihan model.

Kedua, proses *balancing* dataset dapat dikembangkan dengan metode yang lebih beragam. Pada penelitian ini, dataset *balanced* dibuat menggunakan *quality-based filtering* pada kelas mayoritas dan augmentasi pada kelas minoritas. Penelitian selanjutnya dapat membandingkan beberapa pendekatan lain, seperti *oversampling* berbasis augmentasi yang lebih bervariasi, *class weighting*, *focal loss*, atau pendekatan *hybrid* lain untuk melihat metode penanganan *imbalance dataset* yang paling sesuai.

Ketiga, percobaan *fine-tuning* MobileNetV2 dapat diperluas dengan skenario ablasi yang lebih detail. Penelitian ini menggunakan skenario 0, 10, 20, 30, dan 50 *layer* akhir. Penelitian selanjutnya dapat menambahkan variasi jumlah *layer* lain, *learning rate* berbeda, *optimizer* berbeda, atau strategi *fine-tuning* bertahap agar diperoleh konfigurasi yang lebih optimal.

Keempat, pengujian menggunakan foto lapangan sebaiknya dilakukan dengan jumlah data yang lebih banyak dan dilengkapi label *ground truth* dari ahli atau pengamat yang memahami penyakit daun padi. Dengan adanya label yang terverifikasi, hasil pengujian lapangan dapat dihitung menggunakan metrik evaluasi seperti *accuracy*, *recall*, *precision*, dan *F1-score*, sehingga analisis terhadap kemampuan model pada kondisi nyata menjadi lebih kuat.

Kelima, penelitian selanjutnya dapat menambahkan analisis aspek komputasional yang lebih lengkap, seperti waktu pelatihan dalam satuan menit, penggunaan memori, konsumsi CPU atau GPU, serta waktu inferensi pada berbagai perangkat. Analisis ini penting apabila model akan dikembangkan menjadi sistem klasifikasi yang digunakan secara langsung pada perangkat dengan keterbatasan komputasi.

Keenam, model dapat dikembangkan lebih lanjut ke dalam bentuk aplikasi sederhana, baik berbasis web maupun *mobile*. Dengan demikian, hasil penelitian tidak hanya berhenti pada evaluasi model, tetapi juga dapat digunakan sebagai dasar pengembangan sistem bantu identifikasi penyakit daun padi berbasis citra digital.

DAFTAR PUSTAKA

Costales, H., Arpee, C.-A., & Noel, R. (2020). Development of a Prototype Application for Rice Disease Detection Using Convolutional Neural Networks. *International Journal of Emerging Trends in Engineering Research*, *8*(10), 7076–7081. https://doi.org/10.30534/ijeter/2020/708102020

Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., & Qi, L. (2021). Automatic Diagnosis of Rice Diseases Using *Deep learning*. *Frontiers in Plant Science*, *12*, 701038. https://doi.org/10.3389/fpls.2021.701038

Gunawan, R., & Darmansyah, D. (2024). Diagnosa Penyakit BrownSpot dan LeafBlast Pada Tanaman Padi dengan MobileNetV2 dan TensorFlow-Lite. *Jurnal Informatika Upgris*, *10*(1), 20–25. https://doi.org/10.26877/jiu.v10i1.19133

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* (arXiv:1502.03167). arXiv. https://doi.org/10.48550/arXiv.1502.03167

Jisan, M. K. (2023). *Rice Leafs Disease Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/maimunulkjisan/rice-leaf-dataset-from-mendeley-data

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on *deep learning* with class imbalance. *Journal of Big Data*, *6*(1), 27. https://doi.org/10.1186/s40537-019-0192-5

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017). *On Large-Batch Training for Deep learning: Generalization Gap and Sharp Minima* (arXiv:1609.04836). arXiv. https://doi.org/10.48550/arXiv.1609.04836

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980

Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do Better ImageNet Models Transfer Better? *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2656–2666. https://doi.org/10.1109/CVPR.2019.00277

Liu, R. (2023). Paddy Disease Classification Based on the Lightweight MobileNet-V2. *Highlights in Science, Engineering and Technology*, *57*, 126–133. https://doi.org/10.54097/hset.v57i.9990

M., A. (2021). *Rice Disease Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/anshulm257/rice-disease-dataset

Mao, A., Mohri, M., & Zhong, Y. (2023). *Cross-Entropy Loss Functions: Theoretical Analysis and Applications* (arXiv:2304.07288). arXiv. https://doi.org/10.48550/arXiv.2304.07288

Miftahuddin, A., Lutfi, Moch., & Saadah, Z. nikmatus. (2025). Comparison of MobileNetv2 and MobileNetv3 architectures in rice leaf disease classification using transfer learning. *Jurnal Mandiri IT, Lecture Notes in Networks and Systems*, *14*(2), 195–202. https://doi.org/10.35335/mandiri.v14i2.459

Mustofa, M. B., Arifin, M., & Zulfia, S. (2026). *Multi-Class Classification of Rice Leaf Diseases Using MobileNetV2 Architecture with Transfer Learning*. *10*(2).

Paemo, N. D., Ismail, A. R., & Harun, R. (2025). Rice Leaf Disease Classification Based on Convolutional Neural Network Transfer Learning Using MobileNetV2 and EfficientNetB0. *Journal of Computer Science and Informatics Engineering*, *5*(1), 56–68. https://doi.org/10.55537/cosie.v5i1.1371

Prechelt, L. (1998). *Early Stopping | but when?*

Rahman, M. M., Yan, Z., Aziz, M. T., Siddick, M. A. B., Truong, T., Sharif, Md. M., Datta, N., Mahmud, T., Pecho, R. D. C., & Farid, S. M. (2024). Explainable Deep Transfer Learning Framework for Rice Leaf Disease Diagnosis and Classification. *International*

Journal of Advanced Computer Science and Applications, *15*(12). https://doi.org/10.14569/IJACSA.2024.0151287

Saddami, K., Nurdin, Y., Zahramita, M., & Safiruz, M. S. (2024). *Advancing Green AI: Efficient and Accurate Lightweight CNNs for Rice Leaf Disease Identification*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). *MobileNetV2: Inverted Residuals and Linear Bottlenecks* (arXiv:1801.04381). arXiv. https://doi.org/10.48550/arXiv.1801.04381

Seelwal, P., & Rohilla, T. R. (2023). Diagnosis of Rice Diseases using Canny Edge K-means Clustering and Convolutional Neural Network based Transfer Learning. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(9s), 164–172. https://doi.org/10.17762/ijritcc.v11i9s.7408

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for *Deep learning*. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*.

Utami, T. W., Novita, M., & Latifa, K. (2025). *Implementation and Comparative Analysis of CNN and Transfer Learning Models (EfficientNetB0, MobileNetV2, and ResNet50) for Rice Leaf Disease Detection Based on Digital Images*. *9*(6).

Veeramreddy, R., Gnanasekaran, A., & Kunchala, S. B. (2024). Detection of diseases in rice leaf using convolutional neural network with transfer learning based on ResNeXt. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(2), 1739. https://doi.org/10.11591/ijece.v14i2.pp1739-1749

Zhou, H., Deng, J., Cai, D., Lv, X., & Wu, B. M. (2022). Effects of Image Dataset Configuration on the *Accuracy* of Rice Disease Recognition Based on Convolution

Neural Network. *Frontiers in Plant Science*, *13*, 910878. https://doi.org/10.3389/fpls.2022.910878

LAMPIRAN