

TESIS

KLASIFIKASI PENYAKIT CITRA DAUN TANAMAN PADI BERBASIS DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)



Disusun oleh:

Nama : Fachri Ayudi Filtrony
NIM : 24.55.1617
Konsentrasi : Teknologi Media Digital

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2026

TESIS

**KLASIFIKASI PENYAKIT CITRA DAUN TANAMAN PADI BERBASIS
DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK (CNN)**

**RICE PLANT LEAF IMAGE DISEASE CLASSIFICATION BASED ON
DEEP LEARNING USING THE CONVOLUTIONAL NEURAL NETWORK
(CNN) METHOD**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Fachri Ayudi Fitrony
NIM : 24.55.1617
Konsentrasi : Teknologi Media Digital

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

HALAMAN PERSETUJUAN

KLASIFIKASI PENYAKIT CITRA DAUN TANAMAN PADI BERBASIS DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

RICE PLANT LEAF IMAGE DISEASE CLASSIFICATION BASED ON DEEP LEARNING USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Dipersiapkan dan Disusun oleh

Fachri Ayudi Fitrony

24.55.1617

Telah disetujui oleh Dosen Pembimbing Tesis
pada pada hari Selasa, 02 Juni

Dosen Pembimbing,



Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

HALAMAN PENGESAHAN

KLASIFIKASI PENYAKIT CITRA DAUN TANAMAN PADI BERBASIS DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

RICE PLANT LEAF IMAGE DISEASE CLASSIFICATION BASED ON DEEP LEARNING USING THE CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Dipersiapkan dan Disusun oleh

Fachri Ayudi Fitriy
24.55.1617

Telah Dibacakan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 02 Juni

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Dr. Kumara Ari Yuana, S.T., M.T.
NIK. 190302035



Robert Marco, S.T., M.T., Ph.D.
NIK. 190302228



Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, tanggal ujian tesis
Direktur Program Pascasarjana



Dr. Kusri, M.Kom.
NIK. 19030210

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Fachri Ayudi Fitrony

NIM : 24.55.1617

Konsentrasi : Teknologi Media Digital

Menyatakan bahwa Tesis dengan judul berikut:

**Klasifikasi Penyakit Citra Daun Tanaman Padi Berbasis Deep Learning
Menggunakan Metode Convolutional Neural Network (CNN)**

Dosen Pembimbing Utama : Prof. Dr. Etna Utami, S.Si., M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 02 Juni

Yang Menyatakan,



Fachri Ayudi Fitrony

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, tiada daya dan upaya melainkan atas pertolongan dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian tugas akhir ini. Penulisan tesis ini penulis persembahkan kepada :

1. Kedua orang tua (Bapak Amirul Kusnadi dan Ibu Yuyun Indahwati) yang senantiasa memberikan semangat dan doa, semoga selalu dalam lindungan-Nya.
2. Keluarga Penulis Istri dan anak-anak yang senantiasa memberikan semangat dan doa, semoga selalu dalam lindungan-Nya.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. dan . yang telah memberikan bimbingan dan dorongan selama pelaksanaan penelitian tugas akhir ini.
4. Bapak Ibu dosen Magister Informatika Universitas Amikom Yogyakarta yang telah memberikan ilmu dan arahan yang sangat bermanfaat.
5. Seluruh keluarga, teman-teman MTI Amikom, dan semua pihak yang tidak bisa saya sebutkan satu persatu.
6. Seluruh civitas akademika Universitas Nahdlatul Ulama Pasuruan dan khususnya Dekan Fakultas Ilmu Komunikasi Universitas Nahdlatul Ulama Pasuruan.

HALAMAN MOTTO

"Al-muhafazhatsu 'alal qadimis shalih wal akhdzu bil jadidil ashlah"
(Memelihara tradisi lama yang baik dan mengambil tradisi baru yang lebih baik)

"Ilmu tanpa adab laksana api tanpa kayu bakar, dan adab tanpa ilmu laksana ruh tanpa jasad." (KH. Hasyim Asy'ari)

"Ngaji iku sing penting istiqomah, ora ketang sethithik asal ajeg, suwe-suwe dadi bukit." — (KH Maimun Zubair)

"Ilmu tanpa amal itu gila, dan amal tanpa ilmu itu sia-sia." — Prinsip Dasar Ahlussunnah wal Jama'ah

"Ilmu pengetahuan dan teknologi adalah alat. Nasib manusia sangat tergantung pada siapa yang menggunakannya." (KH Abdurrahman Wahid)



KATA PENGANTAR

Alhamdulillah, puji syukur senantiasa kita panjatkan kepada Allah SWT yang telah memberikan karunia dan anugerah yang tak terhingga, sehingga penulis dapat menyelesaikan tesis yang berjudul "Klasifikasi Penyakit Citra Daun Tanaman Padi Berbasis Deep Learning Menggunakan Metode Convolutional Neural Network (CNN)". Penulisan tesis ini diajukan untuk memenuhi salah satu syarat kelulusan dalam jenjang perkuliahan Strata 2 Universitas Amikom Yogyakarta.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. M. Suyanto, M.M. selaku Rektor Universitas Amikom Yogyakarta.
2. Ibu Prof. Dr. Kusriani, M.Kom. selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. dan. selaku dosen pembimbing yang telah sabar memberikan bantuan, masukan, dan saran dalam menyelesaikan tesis ini.
4. Bapak Ibu dosen di Pascasarjana Universitas Amikom Yogyakarta yang telah memberikan banyak ilmu dan pengalaman.
5. Kedua orang tua keluarga istri anak yang tak pernah lelah dalam memberikan *support* dan doa.

Serta semua pihak yang telah membantu penyelesaian penelitian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu. Penulis juga sadar bahwa penyusunan tesis ini masih jauh dari kata sempurna, sehingga dengan senang hati menerima kritik dan saran yang membangun. Semoga laporan tesis ini dapat menambah pengetahuan dan memberikan manfaat bagi penulis maupun para pembaca, serta dapat digunakan sebagai salah satu referensi penelitian yang lain.

Yogyakarta, 02 Juni 2026

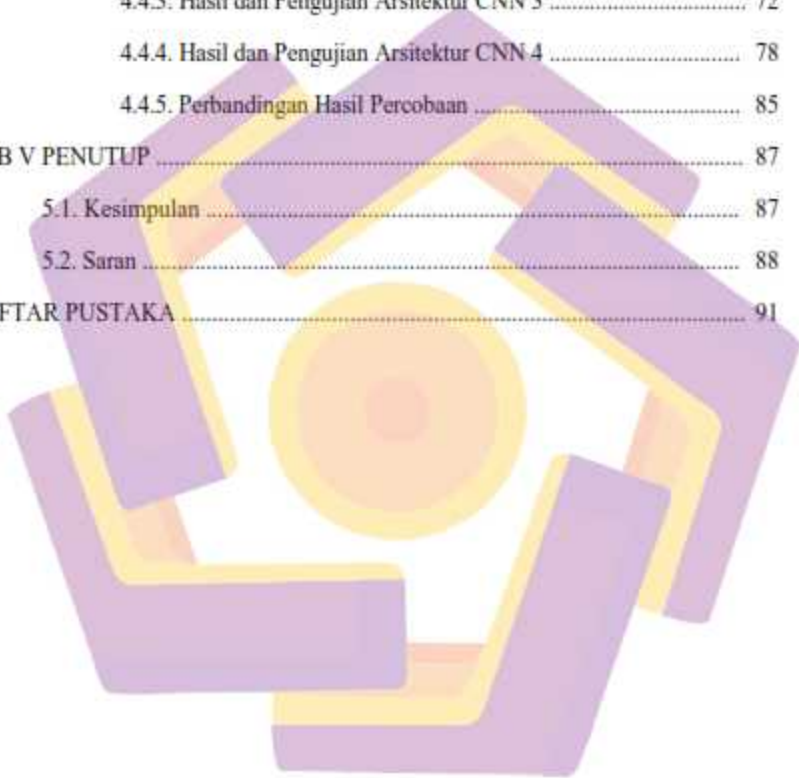
Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
INTISARI	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	5
1.3. Batasan Masalah	5
1.4. Tujuan Penelitian	6
1.5. Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA	8
2.1. Tinjauan Pustaka	8
2.2. Keaslian Penelitian	12
2.3. Landasan Teori	20

2.3.1. Penyakit Tanaman Padi	20
2.3.2. Citra Digital	22
2.3.3. Artificial Intelligence	22
2.3.4. Deep Learning	23
2.3.5. Convolutional Neural Network (CNN)	24
2.3.6. Tahap Feature Learning	25
2.3.7. Tahap Klasifikasi	29
2.3.8. Data Augmentation	31
2.3.9. Normalization Layer	32
2.3.10. Optimizer	32
2.3.11. Confusion Matrix	32
BAB III METODE PENELITIAN	35
3.1. Jenis Penelitian dan Data	35
3.2. Metode Pengumpulan Data	35
3.3. Pengolahan Data	37
3.4. Metode Analisis Data	40
3.5. Alur Penelitian	40
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	44
4.1. Pengumpulan Data	44
4.2. Analisis Data	45
4.2.1. Augmentasi Data	45
4.2.2. Preprocessing Data	48
4.3. Persiapan Percobaan CNN	49
4.3.1. VGG16	51
4.3.2. Xception	54

4.3.3. MobileNetV2	55
4.3.4. InceptionV3	57
4.4. Analisis Hasil Penelitian	58
4.4.1. Hasil dan Pengujian Arsitektur CNN 1	59
4.4.2. Hasil dan Pengujian Arsitektur CNN 2	65
4.4.3. Hasil dan Pengujian Arsitektur CNN 3	72
4.4.4. Hasil dan Pengujian Arsitektur CNN 4	78
4.4.5. Perbandingan Hasil Percobaan	85
BAB V PENUTUP	87
5.1. Kesimpulan	87
5.2. Saran	88
DAFTAR PUSTAKA	91



DAFTAR TABEL

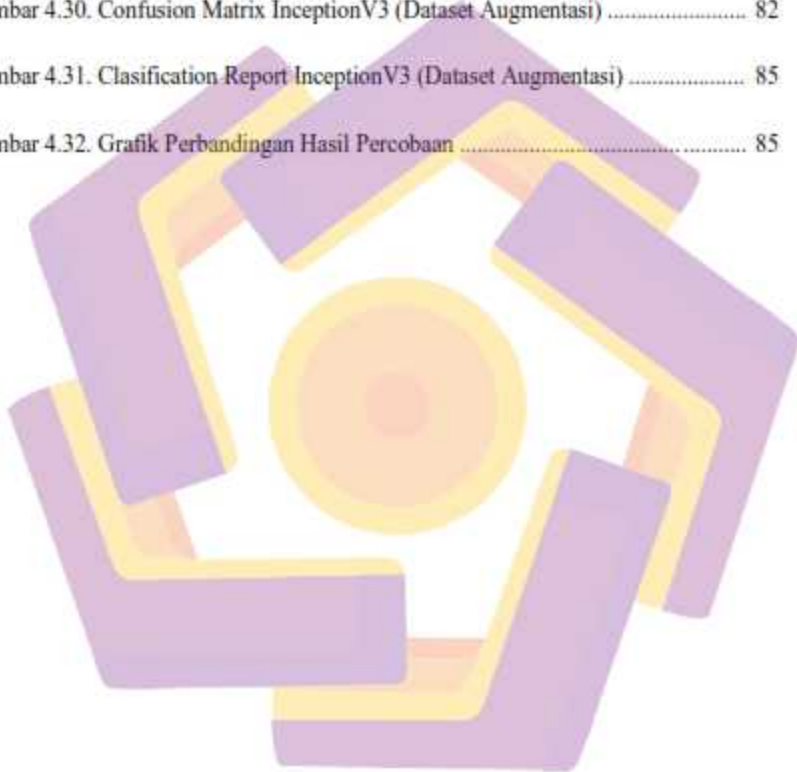
Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian	12
Tabel 2.2. Parameter Confusion Matrix	33
Tabel 3.1. Bentuk Kondisi Daun	36
Tabel 4.1. Jumlah Dataset Original	44
Tabel 4.2 Jumlah Dataset Augmentasi	46
Tabel 4.3. Skenario Percobaan Arsitektur CNN	50
Tabel 4.4. Konfigurasi Hyperparameter Pelatihan Model	51
Tabel 4.5. Perbandingan Hasil Percobaan	86
Tabel 4.6. Analisis Metrik Evaluasi MobileNetV2 Augmentasi	86

DAFTAR GAMBAR

Gambar 2.1. Matriks Citra Digital	22
Gambar 2.2. Arsitektur Deep Learning	24
Gambar 2.3. Tahapan Convolutional Neural Network (CNN)	25
Gambar 2.4. Convolution Operation	26
Gambar 2.5. Convolutional Operation dengan kernel, stride dan padding	27
Gambar 2.6. Fungsi Aktivasi ReLU	28
Gambar 2.7. Max dan Average Pooling Layer	28
Gambar 2.8. Fully Connected Layer dan Flatten	29
Gambar 2.9. Fungsi Aktivasi Softmax	30
Gambar 2.10. Dropout	31
Gambar 3.1. Alur Preprocessing Data	38
Gambar 3.2. Kerangka Alur Penelitian	41
Gambar 4.1. Sampel Dataset Original	45
Gambar 4.2. Sampel Dataset Augmentasi Original	48
Gambar 4.3. Model Arsitektur VGG16	52
Gambar 4.4. Arsitektur VGG16 Setelah Fine Tuning	53
Gambar 4.5. Model Arsitektur Xception	55
Gambar 4.6. Model Arsitektur MobileNetV2	56

Gambar 4.7. Model Arsitektur InceptionV3	57
Gambar 4.8. Grafik Training & Validation VGG16 (Dataset Original)	59
Gambar 4.9. Confusion Matrix VGG16 (Dataset Original)	59
Gambar 4.10. Clasification Report VGG16 (Dataset Original)	62
Gambar 4.11. Grafik Training & Validation VGG16 (Dataset Augmentasi)	62
Gambar 4.12. Confusion Matrix VGG16 (Dataset Augmentasi)	61
Gambar 4.13. Clasification Report VGG16 (Dataset Augmentasi)	65
Gambar 4.14. Grafik Training & Validation Xception (Dataset Original)	66
Gambar 4.15. Confusion Matrix Xception (Dataset Original)	66
Gambar 4.16. Clasification Report Xception (Dataset Original)	67
Gambar 4.17. Grafik Training & Validation Xception (Dataset Augmentasi).....	69
Gambar 4.18. Confusion Matrix Xception (Dataset Augmentasi)	69
Gambar 4.19. Clasification Report Xception (Dataset Augmentasi)	72
Gambar 4.20. Grafik Training & Validation MobileNetV2 (Dataset Original).....	72
Gambar 4.21. Confusion Matrix MobileNetV2 (Dataset Original)	73
Gambar 4.22. Clasification Report MobileNetV2 (Dataset Original)	75
Gambar 4.23. Grafik Training & Validation MobileNetV2 (Dataset Augmentasi) ...	75
Gambar 4.24. Confusion Matrix MobileNetV2 (Dataset Augmentasi)	76
Gambar 4.25. Clasification Report MobileNetV2 (Dataset Augmentasi)	78

Gambar 4.26. Grafik Training & Validation InceptionV3 (Dataset Original)	79
Gambar 4.27. Confusion Matrix InceptionV3 (Dataset Original)	79
Gambar 4.28. Clasification Report InceptionV3 (Dataset Original)	81
Gambar 4.29. Grafik Training & Validation InceptionV3 (Dataset Augmentasi)	82
Gambar 4.30. Confusion Matrix InceptionV3 (Dataset Augmentasi)	82
Gambar 4.31. Clasification Report InceptionV3 (Dataset Augmentasi)	85
Gambar 4.32. Grafik Perbandingan Hasil Percobaan	85



INTISARI

Padi merupakan sumber makanan pokok bagi sebagian besar masyarakat di dunia, termasuk Indonesia yang merupakan negara agraris dimana sebagian besar penduduknya menanam dan mengonsumsi padi. Tentu saja tanaman padi juga memiliki berbagai kondisi penyakit terutama pada daun yaitu diantaranya *Brownspot*, *Healthy*, *Leafblight* dan *Hispa*. Jika infeksi atau penyakit pada tanaman padi tidak segera diidentifikasi sejak dini maka akan menurunkan produksi dan merugikan petani. Untuk menangani masalah ini dapat memanfaatkan bidang informatika dalam mengidentifikasi penyakit dengan menggunakan *image processing* dan *image classification*.

Tahapan penelitian dimulai dari identifikasi masalah, studi literatur, penentuan algoritma dan metode. Setelah itu mulai mengumpulkan dataset publik, melakukan augmentasi dataset, kemudian eksperimen *training* data dengan beberapa percobaan arsitektur CNN, lalu mengevaluasi hasil percobaan, dan terakhir menarik kesimpulan serta menulis laporan. Dataset diambil dari repository publik kaggle dan augmentasi data juga digunakan dalam penelitian ini untuk memperbanyak dataset guna meningkatkan akurasi data *training*.

Dengan latar belakang tersebut maka diusulkan pendekatan sistem pendeteksian penyakit pada daun padi dengan metode *deep learning* menggunakan *Convolutional Neural Network* (CNN) dan beberapa perbandingan arsitektur yaitu VGG16, MobileNetV2, Xception, dan InceptionV3. Hasil eksperimen paling baik diperoleh menggunakan arsitektur MobileNetV2 yang menghasilkan akurasi *validation* 97.15%.

Kata kunci: cnn, deep learning, augmentasi data, penyakit daun, tanaman padi

ABSTRACT

Rice is a staple food for most people worldwide, including Indonesia, an agricultural country where the majority of the population grows and consumes rice. Rice plants are also susceptible to various diseases, particularly those affecting the leaves, including brownspot, leaf blight, leaf blight, and Hispa. If infections or diseases in rice plants are not identified early, they can reduce production and cause losses for farmers. To address this issue, informatics can be utilized to identify diseases using image processing and image classification.

The research phase begins with problem identification, literature review, and algorithm and method determination. Following this, public datasets are collected, dataset augmentation is conducted, training data experiments are conducted using various scenarios and architectures, and the results are evaluated. Finally, conclusions are drawn and a report is written. The dataset is taken from the Kaggle public repository, and data augmentation is also used in this research to expand the dataset and improve the accuracy of the training data.

Against this background, a deep learning approach to detecting rice leaf diseases using a Convolutional Neural Network (CNN) and several architecture comparisons are proposed: VGG16, MobileNetV2, Xception, and InceptionV3. The best experimental results were obtained using the MobileNetV2 architecture which produced a validation accuracy of 97.15%.

Keywords: cnn, deep learning, data augmentation, leaf disease, rice plants

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Di dunia yang serba modern ini banyak perkembangan yang telah dialami oleh umat manusia dalam memenuhi kebutuhan hidup, baik perkembangan pada bidang sosial budaya politik hukum dan ham, ekonomi, kesehatan, pertanian, dan tentunya teknologi yang semakin canggih. Perkembangan yang dirasakan manusia tidak semuanya memiliki dampak positif dikarenakan timbul juga perkembangan yang berdampak negatif salah satunya dalam bidang pertanian. Permasalahan pada perkembangan pertanian sangatlah meresahkan karena pertanian mempunyai peranan penting dalam kehidupan manusia dan berkontribusi penuh dalam ketahanan pangan maupun ekonomi. (Mohidem et al., 2022). Beberapa permasalahan yang sering terjadi dalam perkembangan pertanian adalah rusaknya hasil pertanian akibat penyakit yang menyerang pada daun tumbuhan, dimana fungsi daun bagi tumbuhan sangat vital yaitu melakukan proses fotosintesis atau pembuatan makanan pada tumbuhan itu sendiri. Penyakit pada daun tanaman menyerang hampir semua jenis tanaman tak terkecuali tanaman komoditas pangan utama seperti padi.

Padi atau dengan nama latin *Oryza sativa* merupakan salah satu tanaman budidaya terpenting dalam peradaban. Padi merupakan tanaman pangan yang menyediakan makanan bagi sekitar setengah dari populasi dunia, bahkan padi merupakan bahan makanan pokok yang dikonsumsi masyarakat Indonesia. Hal ini menunjukkan bahwa Indonesia adalah negara agraris. (Oktaviana., 2021). Pada tahun 2024, sebesar 10,05 juta hektar lahan padi di panen dengan berat total yang dihasilkan adalah 52,66 juta ton. Hasil tersebut mengalami kenaikan pada luas lahan panen sebesar 167,25 ribu hektar dan berat 1,32 juta ton dari tahun 2023. Sebesar 30,34 juta ton padi menjadi konsumsi pangan penduduk Indonesia pada tahun 2024 (Badan Pusat Statistik, 2024). Peningkatan permintaan terhadap produksi padi semakin meningkat seiring bertambahnya tahun. Terdapat tantangan dalam memenuhi aspek permintaan dan ketersediaan padi. Masalah yang ada adalah produktivitas petani dipengaruhi oleh cuaca yang kadang tidak menentu serta munculnya berbagai penyakit pada padi yang bisa mengakibatkan

gagal panen. Apabila ini terjadi maka akan berpotensi kepada penurunan ketahanan pangan negara. (Kurniawan Effendi, 2020).

Permasalahan penyakit pada tanaman padi merupakan masalah yang cukup krusial dan banyak dialami oleh petani, apalagi sebagian besar petani tidak dapat mendeteksi secara dini jika terjadi serangan hama atau penyakit pada tanaman padi. Hasil menunjukkan bahwa penyakit yang umum menyerang daun tanaman padi disebabkan oleh bakteri dan jamur. Salah satu penyakit yang menyerang pada daun tumbuhan adalah penyakit busuk leher blast, hawar daun blight, dan tungro pada tanaman padi yang dapat merusak daun dan mengakibatkan penurunan hasil produksi padi. Penyakit tungro adalah satu dari penyakit padi yang paling merusak, dimana epidemic penyakit ini telah terjadi sejak pertengahan tahun 1960an. Mulai yang terserang jarang menghasilkan gabah, menjadi pendek dan steril atau hanya sebagian yang berisi dengan gabah yang berubah warna. Pembungaan dari tanaman yang terserang jadi tertunda dan pembentukan malai sering tidak sempurna. Penyakit akibat bakteri yaitu hawar daun (blight) yang disebabkan oleh *Xanthomonas campestris* dengan gejala daun mengering, melipat, dan berwarna kelabu. Serta penyakit akibat jamur yaitu blast daun yang disebabkan oleh *Magnaporthe grisea* dengan gejala bercak runcing dikelilingi warna kuning pucat (Walascha et al., 2021).

Serangan hama dan penyakit perlu penanganan yang cepat, akurat dan tepat supaya kegagalan panen dapat di hindari. Pemantauan secara berkala harus dilakukan oleh petani untuk mengetahui perkembangan pertumbuhan dari musim tanam hingga memasuki masa panen. (Muhammad Sayuthi, 2020). Untuk mengidentifikasi penyakit pada tanaman padi, salah satu aspek yang bisa dilakukan adalah pemeriksaan pada daun padi. Organ daun pada tanaman padi sangat tepat untuk dijadikan sebagai sumber informasi dalam mengidentifikasi sehat atau tidaknya tumbuhan padi (Amanda Caecilia Milano, 2024). Penanganan masalah pada penyakit daun tanaman dalam mengurangi dampak kerusakan telah banyak dilakukan dalam beberapa penelitian, tidak hanya penelitian dalam bidang pertanian saja tetapi pada bidang teknologi pun turut andil dalam permasalahan ini, salah satunya pemanfaatan bidang informatika dalam mengidentifikasi penyakit yang ada pada daun tanaman dengan menggunakan citra atau bisa kita sebut image processing. Pemanfaatan image processing dan image

classification pada deteksi penyakit tanaman dapat membantu para pengelola pertanian dalam mengambil langkah yang efektif dan efisien (Rozaqi, Sunyoto, & Arief, 2021). Deep learning telah menjadi salah satu topik hangat dalam dunia machine learning karena kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti citra dan suara. Metode deep learning yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah Convolutional Neural Network (CNN). Metode Convolutional Neural Network atau CNN berhasil mengungguli metode machine learning lain pada kasus klasifikasi gambar dikarenakan CNN memiliki cara kerja menyerupai fungsi otak pada manusia, dimana komputer akan diberikan data citra untuk dipelajari, dilatih mengenali setiap elemen visual pada citra serta memahami setiap pola citranya, hingga nantinya komputer mampu mengidentifikasi citra tersebut.

CNN merupakan salah satu metode pengembangan Multi-Layer Perception (MLP) yang memiliki sedikit parameter karena tidak memerlukan nilai pre- processing, segmentasi dan ekstraksi fitur. (Dicki Irfansyah, 2021). Dengan metode Convolutional Neural Network (CNN) ini para peneliti secara empiris telah terbukti mampu mengklasifikasikan citra dengan baik seperti yang dilakukan oleh Mohtar Khoiruddin et al (2022) dengan judul "Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network" mendapatkan hasil untuk epoch 25, 50, 75 dan 100 memiliki akurasi yang beragam. Hasil akurasi training terbaik ada di epoch 100 dengan akurasi 98% dan dilakukan testing menggunakan confusion matrix mendapatkan nilai rata-rata akurasi sebesar 98%. Penelitian lain dengan judul "Classification of Rice Plant Diseases Using the Convolutional Neural Network Method" yang dilakukan oleh A A JE Veggy Priyanka dan I Made Surya Kumara pada tahun 2021 dengan merancang sebuah aplikasi untuk mengklasifikasikan penyakit padi berdasarkan warna dan tekstur daun menggunakan metode CNN dengan augmentasi data pada dataset. Didapatkan hasil model CNN pada percobaan keempat yang dilakukan pada citra penyakit padi mendapatkan akurasi sebesar 95.24%.

Ada juga penelitian yang dilakukan oleh Syed Md. Minhaz Hossain et al (2020) dengan judul "Rice Leaf Diseases Recognition Using Convolutional Neural Networks" dimana peneliti mengusulkan model baru berbasis Convolutional Neural Network untuk mengenali penyakit daun padi dengan mengurangi parameter jaringan serta

melakukan augmentasi data citra. Model yang diusulkan mencapai akurasi training tertinggi 99,78% dan akurasi validasi 97,35%. Efektivitas model yang diusulkan dievaluasi pada satu set gambar penyakit daun padi independen dengan akurasi terbaik 97,82% dengan area under curve (AUC) 0,99. Untuk membuat sebuah model algoritma yang mempunyai nilai akurasi yang tinggi dengan metode Convolutional Neural Network maka diperlukan jumlah data yang banyak terlebih citra yang diambil juga harus terklasifikasi terlebih dahulu. Beberapa penelitian menyebutkan bahwa untuk membuat sebuah model mempunyai akurasi tinggi dan mengurangi overfitting pada model, salah satunya adalah penelitian yang dilakukan oleh (Enkvetchakul & Surinta, 2021) menyebutkan dalam penelitiannya dengan menggunakan eksperimen pada ribuan dataset penyakit daun pada tanaman dimana pada augmentasi data beberapa teknik kombinasi dilakukan menghasilkan 15 dataset baru dari setiap 1 data asli dan berhasil meningkatkan akurasi menjadi 76,15% dari dataset yang dilatih dengan tanpa melakukan augmentasi data yang nilai akurasinya hanya 63,07%.

Selanjutnya, menurut penelitian sebelumnya yang mengenai klasifikasi penyakit tanaman padi menggunakan *Convolutional Neural Network (CNN)*, mengungkapkan bahwa klasifikasi yang dilakukan menggunakan *Convolutional Neural Network (CNN)* dengan model arsitektur VGG19 untuk mendeteksi penyakit padi menghasilkan tingkat akurasi 90.24% (Priyanka & Kumara, 2021). Hal ini sejalan dengan penelitian klasifikasi padi bagus dan padi rusak menggunakan *Convolutional Neural Network (CNN)* serta menghasilkan tingkat akurasi sebesar 82% (Indra et al., 2022). Pada penelitian lain, Klasifikasi pada daun padi menggunakan model Arsitektur CNN *EfficientNetB0* menghasilkan akurasi sebesar 92,6%. (W. Ahdi, 2023).

Di samping keunggulannya dalam menghasilkan akurasi yang tinggi, klasifikasi menggunakan Convolutional Neural Network (CNN) pada arsitektur kompleks seperti VGG16 dan InceptionV3 sering kali membutuhkan waktu eksekusi yang lama jika dijalankan menggunakan CPU. Banyak peneliti berusaha memitigasi kelemahan ini melalui pendekatan augmentasi data, penggunaan pooling layer, hingga optimasi hyperparameter. Di antara berbagai strategi tersebut, proses fine-tuning memegang peranan paling penting dalam memangkas beban komputasi. Dengan melakukan fine-

tuning yang berfokus pada modifikasi lapisan akhir (classification head) dan mempertahankan bobot pre-trained, model tidak perlu melatih ulang seluruh jaringan dari awal. Atas dasar itu, diperlukan skema fine-tuning pada arsitektur alternatif yang lebih efisien agar dapat menghasilkan model dengan jumlah parameter terlatih yang minimal namun memiliki waktu pemrosesan yang jauh lebih cepat (Beheshti & Johnsson, 2020).

Muncul pula tantangan teknis terkait kualitas data visual; citra penyakit padi di lapangan seringkali memiliki latar belakang yang kompleks, kondisi pencahayaan yang tidak konsisten, serta kemiripan visual antar jenis penyakit (misalnya antara *Brownspot* dan fase awal *Leaf Blight*). Hal ini menjadi isu krusial dalam pengembangan model *Deep Learning*, karena data yang kotor atau tidak representatif akan mengakibatkan model mengalami *overfitting* atau gagal melakukan generalisasi fitur pada kondisi riil. (Islam et al., 2021)

Berdasarkan uraian latar belakang dan studi literatur yang telah dilakukan pada beberapa kasus diatas, masih ada peluang penelitian terkait deteksi penyakit daun tanaman padi dan juga penggunaan salah satu metode *deep learning* yaitu *Convolutional Neural Network* (CNN), seperti contoh dengan membandingkan beberapa arsitektur yang berbeda dan eksperimen jumlah dataset yang berbeda, maka peneliti melakukan analisis dan penelitian dengan judul “Klasifikasi Penyakit Citra Daun Tanaman Padi Berbasis Deep Learning Menggunakan Metode *Convolutional Neural Network* (CNN)”. Peneliti juga akan menggunakan dataset publik penyakit daun tanaman padi dengan melakukan teknik – teknik augmentasi data citra yang ada. Dengan adanya penelitian ini di harapkan dapat diketahui tingkat akurasi model yang di peroleh serta seberapa mampu model CNN mengklasifikasi gambar penyakit daun tanaman padi dengan lebih baik dan akurat.

1.2. Rumusan Masalah

Berdasarkan permasalahan yang dibahas dalam latar belakang, maka rumusan masalah pada penelitian ini yaitu:

- a. Berapa tingkat akurasi klasifikasi penyakit daun padi menggunakan berbagai model arsitektur *Convolutional Neural Network* (CNN)?

- b. Bagaimana pengaruh teknik augmentasi data terhadap tingkat akurasi pada masing-masing arsitektur CNN yang diuji?
- c. Manakah model arsitektur CNN yang menghasilkan performa terbaik dan paling optimal untuk deteksi penyakit daun padi?

1.3. Batasan Masalah

Batasan masalah yang digunakan penulis pada penyusunan laporan tesis ini ditujukan untuk membatasi pokok permasalahan yang akan dibahas agar tidak menyimpang pada permasalahan pokok, maka batasan yang dibahas oleh penulis antara lain:

- a. Permasalahan yang diangkat penulis hanya berkaitan dengan masalah penyakit pada tanaman padi, tidak termasuk hama yang menyerang tanaman padi.
- b. Data yang digunakan oleh penulis merupakan data *public* yang sampelnya diambil dari *Kaggle*.
- c. Dataset yang digunakan sebanyak 1.150 citra daun tanaman padi dengan dibagi menjadi 4 kelas yaitu *Brownspot*, *Leafblight*, *Hispa* dan *Healthy*.
- d. Format citra berkas yang diklasifikasi adalah *JPG*.
- e. Target penelitian hanya modeling akurasi uji dengan *python* tidak sampai implementasi pembuatan aplikasi *end-user*.

1.4. Tujuan Penelitian

Tujuan yang ingin di capai dari penelitian ini adalah:

- a. Dapat mengetahui bagaimana implementasi metode *Convolutional Neural Network* (CNN) dengan arsitektur yang berbeda untuk melakukan klasifikasi penyakit pada daun padi.
- b. Mencari rancangan arsitektur model klasifikasi dengan tingkat akurasi terbaik pada identifikasi penyakit daun tanaman padi menggunakan metode *Convolutional Neural Network* (CNN).
- c. Model dapat digunakan sebagai bahan penelitian lanjutan yang harapannya dapat membantu para petani dalam mengendalikan penyakit pada tanaman padi secara cepat, tepat dan akurat.

1.5. Manfaat Penelitian

Berdasarkan deskripsi dari tujuan penelitian, dapat diketahui manfaat dari penelitian ini adalah sebagai berikut:

- a. Penelitian ini diharapkan dapat menjadi referensi atau acuan dalam melakukan klasifikasi penyakit pada daun padi menggunakan metode *Convolutional Neural Network* (CNN)
- b. Memberikan kontribusi penelitian pada penerapan arsitektur *Convolutional Neural Network* (CNN) yang cocok digunakan dan menghasilkan tingkat akurasi yang tinggi.
- c. Rancangan model yang diterapkan dapat dijadikan pembelajaran sebagai tambahan ilmu pengetahuan baru.
- d. Hasil dari penelitian ini diharapkan dapat menjadi salah satu syarat penyelesaian studi
- e. Penelitian ini diharapkan dapat menjadi referensi atau acuan dalam melakukan klasifikasi penyakit pada daun padi menggunakan metode *Convolutional Neural Network* (CNN)
- f. Memberikan kontribusi penelitian pada penerapan arsitektur *Convolutional Neural Network* (CNN) yang cocok digunakan dan menghasilkan tingkat akurasi yang tinggi.
- g. Rancangan model yang diterapkan dapat dijadikan pembelajaran sebagai tambahan ilmu pengetahuan baru.
- h. Hasil dari penelitian ini diharapkan dapat menjadi salah satu syarat penyelesaian studi.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Sebelum melakukan penelitian terkait identifikasi penyakit pada daun tanaman padi, peneliti terlebih dahulu melakukan peninjauan pustaka terkait penelitian - penelitian terdahulu yang memiliki relevansi dengan judul sebagai referensi dan bahan acuan untuk melakukan penelitian. Beberapa penelitian tersebut adalah sebagai berikut :

1. Penelitian dengan judul "Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation" yang dilakukan oleh (Suseno, Minamo, & Azhar, 2023) mempunyai tujuan untuk mengklasifikasikan penyakit daun padi dan teknik segmentasi citra pada dataset daun padi untuk mengubah citra menjadi bentuk yang lebih mudah untuk dianalisis. Data yang digunakan terdiri dari 3 kelas bacterial leaf blight, brown spot, dan leaf smut. Kemudian dilakukan segmentasi menggunakan dua teknik, yaitu metode threshold dan k-means. Selanjutnya, dilakukan augmentasi data untuk membuat dataset yang digunakan memiliki jumlah yang besar dan bervariasi. Pelatihan menggunakan model VGG16 dengan nilai hiperparameter optimizer RMSProp dan tingkat pembelajaran 0,001. Hasil yang diperoleh adalah akurasi sebesar 91,66% pada skenario dengan dataset kmeans. Metode klasifikasi menggunakan model yang telah di-pretrained VGG16 dengan teknik segmentasi citra belum dapat mengungguli beberapa penelitian sebelumnya. Oleh karena itu, diperlukan penelitian lanjutan dengan melakukan augmentasi data dan metode segmentasi lainnya.
2. Penelitian lain dilakukan oleh (Aggarwal et al., 2022) dengan judul "Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improve AgroBusiness" dengan tujuan menganalisis tren penelitian dalam domain identifikasi penyakit padi menggunakan kecerdasan buatan menggunakan analisis global, kutipan per tahun dan per negara, dan sebagainya untuk mendukung berbagai peneliti yang bekerja di domain ini. Didapatkan kesimpulan bahwa teknik artificial intelligence dan machine learning dapat mengimprovisasi agrobisnis dengan spesifik

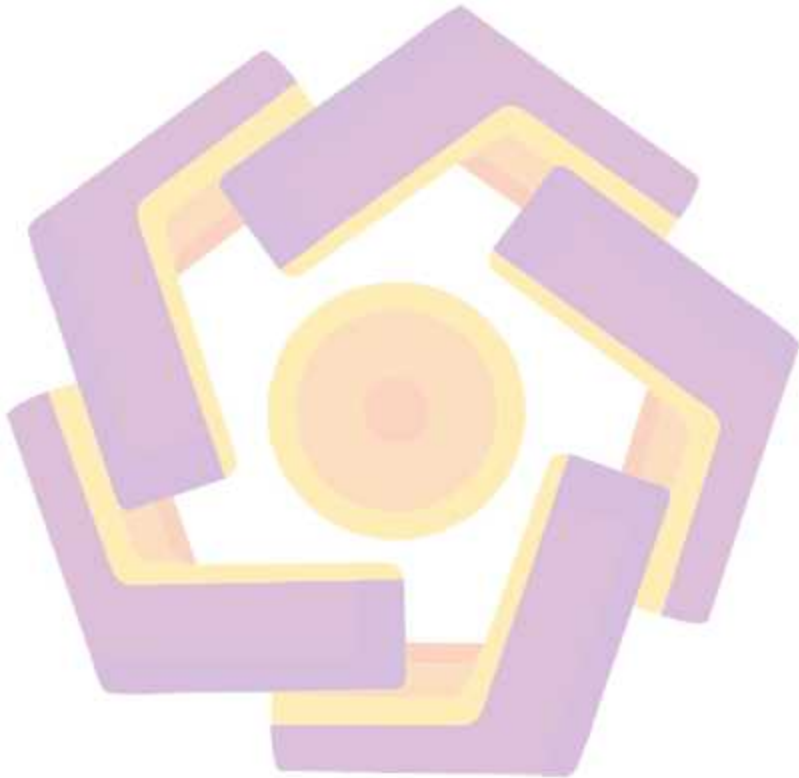
dalam hal deteksi penyakit padi dengan hasil yang berbeda-beda. Terdapat 29 penelitian yang direview dan terakhir tahun 2020 sebanyak 4 paper. sebaiknya ditambah lagi referensi dengan penelitian-penelitian yang lebih baru dan beragam.

3. Artikel yang ditulis oleh (Jatmika & Saputra, 2022) membahas tentang penerapan deep learning untuk mengklasifikasikan atau mengidentifikasi penyakit pada daun padi menggunakan aplikasi mobile. Sistem ini akan memudahkan pengguna dalam mendiagnosa penyakit dengan menampilkan hasil diagnosa berupa nama penyakit beserta taksonominya, deskripsi penyakit dan rekomendasi obat untuk solusi penyakitnya. Dari pengujian yang telah dilakukan, disimpulkan bahwa sistem dengan model yang dibuat menggunakan pendekatan transfer learning menghasilkan akurasi yang baik sebesar 90%. Dataset yang digunakan berjumlah 800 gambar serta model lain hanya memperoleh akurasi 62% karena tanpa menggunakan metode transfer learning sehingga mengalami overfitting.
4. Pada penelitian yang dilakukan oleh (Budi Setiyono, 2023), penelitian menggunakan metode Convolutional Neural Network (CNN) sebagai model dasarnya. Eksperimen dengan citra tanaman riil dilakukan sebanyak 1019 percobaan. Model tersebut cukup baik dalam mengklasifikasikan jenis tanaman herbal pada dataset yang telah diberikan, nilai akurasi rata-rata 96%. Hal ini menjadikan CNN memiliki akurasi terbaik dalam melakukan klasifikasi menggunakan *labelled data* termasuk *image processing*. (Budi Setiyono, 2023).
5. Pada penelitian yang dilakukan oleh (Dwiretno Istiyadi Swasono, 2023), Penelitian ini menggunakan metode CNN dengan menggunakan arsitektur Alexnet dan VGG 16 pada klasifikasi citra buah jeruk dengan empat kelas yaitu *blackspot*, *canker*, *fresh*, dan *grenning* dapat bekerja dengan sangat baik. Arsitektur AlexNet tanpa *batch normalization* dengan pemisahan dataset 90% *data training* (latih) dan 10% data validasi memberikan kinerja yang paling baik dengan akurasi sebesar 92,34% dan rata-rata nilai *precision* sebesar 91,0%, nilai *recall* sebesar 91,0%, dan *F1-score* sebesar 91,0%. Model arsitektur pada metode CNN mampu membantu dalam meningkatkan akurasi hasil klasifikasi citra gambar. (Dwiretno Istiyadi Swasono, 2023)
6. Dalam Penelitian (Adi Fajaryanto Cobantoro, 2023), memaparkan mengenai layer dan arsitektur pada CNN. Arsitektur CNN terbagi menjadi 2 yaitu model arsitektur klasik

dan modern. Perbedaan kedua arsitektur ini ada pada *layer* yang digunakan. *Layer* pada model arsitektur CNN klasik lebih banyak karena parameter yang digunakan juga lebih banyak. Hal ini menyebabkan penggunaan memori dan *runtime* yang berlebih. Beberapa model arsitektur klasik yang ada adalah LeNet, AlexN, dan VGG. Sedangkan arsitektur modern memiliki lebih sedikit parameter namun dengan akurasi yang setara. Beberapa model arsitektur modern adalah EfficientNet GoogleNet, ResNet, dan MobileNetV2. (Adi Fajaryanto Cobantoro, 2023).

7. Pada penelitian (Aditya Rajbongshi, 2022), dilakukan komparasi semua model arsitektur pada CNN. Percobaan ini dilakukan menggunakan 2 skenario yaitu dengan menggunakan optimasi dan tidak. Hasil dari komparasi tersebut mengungkapkan bahwa VGG menjadi yang terbaik dari segi akurasi. Namun, ada hal unik saat melihat hasil percobaan pada model Xception. Model arsitektur Xception dengan penggunaan memori paling sedikit memiliki akurasi yang baik bahkan sempat menyaingi akurasi VGG dan AlexNet saat percobaan menggunakan optimasi. Menurut (Aditya Rajbongshi, 2022), Hal ini dapat terjadi karena akurasi model dapat dipengaruhi oleh kemampuan yang tinggi dalam melakukan pengenalan dan pembelajaran fitur meskipun dengan parameter yang lebih sedikit.
8. Pada penelitian yang dilakukan oleh (R. Surya, 2022), mengenai klasifikasi penyakit daun singkong. Metode yang digunakan adalah metode CNN. Jenis daun singkong yang diklasifikasi pada penelitian ini diantaranya: bacterial leaf blight, rice blast, dan rice tungro virus. Jumlah dataset yang digunakan 6000 yang terbagi menjadi data training 80%, data validasi 10%, dan data testing 10%. Hasil yang didapatkan untuk epoch 25, 50, 75 dan 100 memiliki akurasi yang beragam. Hasil akurasi training terbaik ada di epoch 100 dengan akurasi 96%, hal ini menghasilkan akurasi yang cukup baik tentunya tingkat akurasi dapat dipengaruhi oleh penggunaan arsitektur dan *hyper-parameter* yang diterapkan.
9. Penelitian mengenai deteksi penyakit daun padi selanjutnya dengan metode Convolutional Neural Network (CNN) yang dilakukan oleh (L. Y. Win Lwin, 2023) dengan menggunakan arsitektur Xception. Data citra yang digunakan berupa data asli sebanyak 440 citra dan data augmentasi sebanyak 1320 citra. Nilai akurasi menggunakan citra augmentasi tertinggi yang diperoleh yaitu sebesar 94.31%,

precision 88.00%, recall 95.0%, dan fi score 93.61%. Nilai akurasi tanpa augmentasi tertinggi yang diperoleh yaitu accuracy 93.18%, precision 95%, recall 81.81%, dan fi score 89.99%. Hasil penelitian menunjukkan bahwa augmentasi dan penggunaan hyperparameter dapat meningkatkan hasil akurasi.



2.2.Keaslian Penelitian

Tabel 2.1 Matriks Literatur Review dan Posisi Penelitian

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1.	Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi	Dicki Irfansyah, Metty Mustikasari, Amat Suroso, <i>Jurnal Informatika: Jurnal pengembangan IT JPIT</i> . (2021)	Penelitian ini bertujuan untuk mengembangkan program dengan platform MATLAB. Program ini menggunakan arsitektur CNN Alexnet untuk klasifikasi penyakit daun pada tanaman kopi. Uji coba dilakukan guna mengetahui akurasi dari arsitektur CNN dalam klasifikasi penyakit pada tanaman kopi.	Berdasarkan penelitian yang dilakukan dapat disimpulkan arsitektur CNN Alexnet dapat diimplementasikan dengan platform pemrograman MATLAB. Nilai akurasi klasifikasi jaringan terhadap data testing mencapai 0.816 sehingga dapat disimpulkan arsitektur Alexnet akurat untuk klasifikasi hama pada daun tanaman kopi.	Klasifikasi hama pada penelitian ini fokus pada daun tanaman kopi, kedepan dapat dikembangkan klasifikasi hama melalui bagian lain tanaman kopi misalnya batang, akar dan buah. Kelemahan dari penelitian ini memiliki dataset yang sedikit sekitar 300 dan data test 250.	Perbedaan dari penelitian yang terdahulu terletak di arsitektur yang digunakan, peneliti menggunakan arsitektur yang lebih modern serta jumlah dataset yang digunakan, penelitian terbaru menggunakan google collab (Python) untuk mendapatkan akurasi yang lebih tinggi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2.	Efficient deep learning approach for enhancing plant leaf disease classification	BELMIR Meroua, DIFALLAH Wafa, GHAZLI Abdelkader, <i>Indonesian Journal of Electrical Engineering and Computer Science</i> , (2025)	Untuk deteksi penyakit tanaman di lapangan melalui pengenalan gambar otomatis dalam skala luas dan mengklasifikasikan berbagai penyakit tanaman.	Penelitian ini menggunakan arsitektur (Xception) dan (DenseNet-121) sebagai pengklasifikasi. Menggunakan data (RGB) dari 14 jenis daun tanaman yang berbeda, baik yang sehat maupun yang terkena penyakit, diklasifikasikan ke dalam 38 kelas kategori penyakit tanaman. Xception yang diusulkan dengan model DenseNet-121 mencapai tingkat akurasi 90,22%	Untuk kedepannya dapat menggunakan pendekatan transfer learning agar mampu meminimalisir terjadinya overfitting dalam proses pelatihan model	Perbedaan dari penelitian yang terdahulu terletak di metode yang di gunakan , untuk penelitian yang akan di lakukan menggunakan CNN dengan arsitektur yang berbeda untuk menghindari overfitting dan mencari nilai klasifikasi yang lebih akurat

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3.	Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode <i>Convolutional Neural Network (CNN)</i>	Rendra Soekarta, NirwanaNurdjan, Ardian Syah, <i>INSECT</i> , (2023)	Tujuan penelitian ini yaitu membangun sistem yang dapat digunakan untuk melakukan klasifikasi penyakit tanaman tomat dari 10 jenis penyakit tanaman yaitu, Bacterial spot, Early blight, Late blight, Leaf mold, Septoria leaf spot, Spider mite, Target spot, Mosaic virus, Yellow leaf virus	Hasil klasifikasi penyakit tanaman tomat dengan menggunakan Convolutional Neural Network (CNN) dengan model VGG 16 dapat mengklasifikasikan penyakit tanaman tomat dengan tingkat <i>accuracy training</i> mencapai 98 % dan <i>accuracy validation</i> mencapai 82 %	Menambah dataset dari bagian dari tanaman tomat selain daun. Menambah kelas dataset agar lebih bervariasi dalam melakukan klasifikasi dan dapat mengambil dataset dari sumber selain <i>kaggle</i>	Perbedaan dari penelitian yang terdahulu adalah metode yang di gunakan, penelitian yang akan dilakukan menggunakan CNN dengan arsitektur yang berbeda dalam mencari akurasi yang lebih tinggi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4.	Convolutional neural networks breast cancer classification using Palestinian mammogram dataset	Hanin Saadah , Amani Yousef Owda , Majdi Owda, <i>Indonesian Journal of Electrical Engineering and Computer Science</i> , (2024)	Untuk mendeteksi dan mengklasifikasikan penyakit kelainan kanker payudara. Penelitian ini menerapkan metodologi untuk menyelidiki efek daya dan efektivitas pembelajaran mesin dalam mendeteksi kelainan payudara pada citra mamimogram lokal	Berdasarkan hasil yang diperoleh dari penelitian ini, DenseNet121 mencapai akurasi pengujian tertinggi sebesar 0,83, dan AUC sebesar 0,85, dan mengungguli semua model lainnya. Berdasarkan hasil tersebut, karya yang disajikan telah membuktikan kekuatan dan efektivitasnya dalam mendeteksi dugaan kelainan payudara.	Saran dari penulis, kedepannya dapat menggunakan data pasien lain seperti informasi genetik, faktor gaya hidup, dan riwayat medis, untuk menilai risiko seseorang terkena penyakit tertentu.	Perbedaan dari penelitian yang terdahulu adalah metode yang di gunakan, penelitian yang akan dilakukan menggunakan CNN dengan arsitektur yang berbeda dalam mencari akurasi yang lebih tinggi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5.	Recognition of mango leaf disease using convolutional neural network models: a transfer learning approach.	Aditya Rajbongshi, Anik Pramanik, Thaharim Khan. <i>Indonesian Journal of Electrical Engineering and Computer Science</i> , (2022)	Penelitian ini bertujuan untuk pengelompokan dan pengklasifikasian penyakit daun mangga melalui proses CNN. DenseNet201 , InceptionResNetV2 , InceptionV3 , ResNet50 , ResNet152V2 , MobileNetV2 untuk mendapatkan akurasi yang lebih baik dari kumpulan data yang ditargetkan.	Dalam penelitian, enam model CNN digunakan untuk lima kelas unik penyakit mangga. Menghasilkan bahwa akurasi 97,67% untuk identifikasi penyakit daun mangga menggunakan model Xception. Skor F1, akurasi, sensitivitas, spesifisitas, FNR, dan FPR juga telah diukur. Skor F1, presisi, dan sensitivitas tertinggi untuk antraknosa, embun tepung, dan karat merah adalah 98,33%.	Di penelitian yang akan mendatang dapat menggunakan lebih banyak mendeteksi penyakit daun tanaman lain dengan teknik CNN ini untuk mendapatkan akurasi yang lebih baik untuk penyakit yang ditargetkan tersebut.	Perbandingan terletak pada metode penelitian arsitektur yang akan di gunakan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6.	Erformance Analysis of EfficientNet Convolutional Neural Network (CNN) Architecture with Image Objects of Rice Plant Leaves	Adi Fajaryanto Cobantoro, Fauzan Masykur, Kelik Sussolaikah, JITK , (2023)	Tujuan akhir dari penelitian ini adalah untuk mengklasifikasikan citra daun tanaman padi dengan 4 kelas. Klasifikasi citra menggunakan arsitektur Efficientnet dan proses pelatihan menggunakan google colab. Analisis proses pelatihan klasifikasi menggunakan matriks konfusi yang terdiri dari akurasi, presisi, recall, F1 score, dan loss.	Hasil dari penelitian ini adalah setelah melewati tahapan-tahapan penelitian didapatkan hasil penelitian yakni akurasi senilai 89%, presisi rata-rata senilai 0,24 dan loss senilai 0,05.	Mendapatkan akurasi sebesar 89%. Saran dari penulis untuk preprocessing data lebih giat lagi agar mendapatkan akurasi yang lebih tinggi dan mendapatkan runtime yang lebih cepat	Perbandingan dari penelitian ini adalah jumlah arsitektur yang digunakan untuk penelitian yang akan dilakukan menggunakan arsitektur yang berbeda dan mencari nilai klasifikasi yang lebih akurat

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7.	Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation	Jody Ririt Krido Suseno, Yufiz Azhar, Agus Eko Minarno. <i>Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control</i> (2023)	Mengklasifikasikan penyakit daun padi dan teknik segmentasi citra pada dataset daun padi untuk mengubah citra menjadi bentuk yang lebih mudah untuk dianalisis. Data yang digunakan terdiri dari 3 kelas bacterial leaf blight, brown spot, dan leaf smut. Kemudian dilakukan segmentasi menggunakan dua teknik, yaitu metode threshold dan k-means. Selanjutnya, dilakukan augmentasi data untuk membuat dataset yang digunakan memiliki	Pelatihan menggunakan model VGG16 dengan nilai hiperparameter optimizer RMSProp dan tingkat pembelajaran 0,001. Hasil yang diperoleh adalah akurasi sebesar 91,66% pada skenario dengan dataset k-means.	Metode klasifikasi menggunakan model yang telah di-pretrained VGG16 dengan teknik segmentasi citra belum dapat mengungguli beberapa penelitian sebelumnya. Oleh karena itu, diperlukan penelitian lanjutan dengan melakukan augmentasi data dan metode segmentasi lainnya.	Penelitian Suseno et al 2023 menggunakan dataset dari sumber terbuka UCI Machine Learning dengan nama Leaf Rice Diseases berisi 120 data gambar yang dibagi dalam 3 kelas. Penelitian yang akan dilakukan menggunakan dataset dari kaggle dengan augmentasi data dan pada proses CNN tidak hanya menggunakan arsitektur VGG16.

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8.	Cucumis Melo L. Leaf Diseases Identification Using a Convolutional Neural Network	Minh-Dung Lam, Quoc-Bao Truong, Chanh-Nghiem Nguyen and Tan-Kiet Nguyen Thanh, <i>Vietnam Journal of Computer Science</i> (2025)	Mengembangkan dan menguji model Convolutional Neural Network (CNN) yang mampu secara otomatis dan akurat mengidentifikasi tiga penyakit daun pada tanaman Cucumis Melo L., yaitu anthracnose, downy mildew, dan powdery mildew, serta menentukan kesehatan daun secara umum, produksi tanaman cucumis melo.	Model CNN yang dikembangkan mampu secara efektif membedakan antara daun sehat dan daun yang terinfeksi oleh tiga penyakit utama pada Cucumis Melo L., yaitu anthracnose, downy mildew, dan powdery mildew, dengan tingkat akurasi sekitar 91.58%. Model ini tidak memerlukan preprocessing yang rumit dan memiliki keunggulan dalam kecepatan serta performa pelatihan yang baik dibandingkan model-model standar seperti VGG 16, VGG 19 dan InceptionV3.	Pengembangan model CNN dapat dilakukan dengan menambah jumlah data gambar untuk meningkatkan tingkat akurasi dan generalisasi model. Kelemahan dari penelitian ini adalah terbatasnya jumlah data gambar yang digunakan, yang dapat mempengaruhi akurasi dan ketahanan model saat diterapkan di kondisi nyata yang bervariasi.	Dalam penelitian Minh-Dung Lam et al, menggunakan platform Matlab dan dataset daun Cucumis Melo L. serta membuat model sendiri tanpa menggunakan arsitektur yang sudah ada, Sedangkan peneliti menggunakan platform Google collab dan dataset daun tanaman padi serta menggunakan arsitektur yang sudah di sediakan

2.3. Landasan Teori

2.3.1. Penyakit Tanaman Padi

Padi atau dalam nama latin biasa disebut *Oryza sativa* ini adalah salah satu jenis tanaman yang biasa dibudidayakan dan sangat penting bagi peradaban dunia, terutama Indonesia. Diduga bahwa padi adalah tanaman yang berasal dari India dan dibawa masuk ke Indonesia pada kisaran tahun 1500 sebelum masehi. (Arifa, 2021). Terdapat dua tipe padi budidaya yaitu padi gogo dan padi rawa. Padi gogo biasa dibudidaya di daerah bersawah yang memiliki tempat untuk tadah air dan cenderung kering (Januarti et al., 2021).

Padi memiliki beberapa jenis penyakit yang dapat menyerang di beberapa bagian seperti batang, daun dan biji selama proses pertumbuhan. Pada daerah daun, umumnya penyakit yang ada disebabkan oleh mikroorganisme kecil. Bisa juga disebabkan oleh gangguan cuaca, kekurangan kandungan gizi, obat-obatan yang tidak cocok dan jenis tanah (Nuryanto, 2019).

Pengendalian penyakit pada padi sebenarnya dapat dilakukan dengan berbagai cara tergantung dengan akar permasalahan. Namun hal yang paling penting adalah dengan mengetahui jenis tanaman, jenis tanah, jenis penyakit yang menyerang dan cara penanggulangannya. Penyebaran penyakit dapat diminimalisir dengan beberapa cara seperti penanaman varietas yang lebih tahan, penanaman yang berjarak, pemupukan berimbang, dan pestisida (Nuryanto, 2019).

Penyakit pada daun padi memiliki beberapa jenis tipe dan gejala sesuai dengan penyebab dan kondisi daerah tempat penanaman padi. Berikut adalah penjelasan dari penyakit daun padi yang digunakan pada penelitian ini:

1. Hispa

Penyakit *hispa* pada daun padi adalah salah satu penyakit pada padi yang disebabkan oleh hama yang berkembang biak dari gulma rumput di sekitar sawah. Pemupukan secara berlebih dapat memperparah keadaan dari daun yang terinfeksi. Keadaan cuaca hujan juga sangat menguntungkan bagi hama untuk berkembang biak dan menyebar lebih luas pada padi (Adi Fajaryanto

Cobantoro, 2023).

Penyakit ini dapat diidentifikasi dengan munculnya bercak putih memanjang dengan posisi yang sejajar dengan urat daun. Penyebabnya adalah pengikisan hingga jaringan epidermis oleh larva yang ditaruh oleh hama sehingga daun menjadi layu dan rusak. Pengendalian dari penyakit ini dapat dilakukan dengan menghindari pemupukan yang berlebihan. Mengatur jarak tanam yang berdekatan juga meminimalisir perkembangbiakan hama. Pemotongan ujung pucuk juga dapat mengurangi populasi hama sebesar 70-90% (Muhammad Sayuthi, 2020)

2. *Brown Spot*

Penyakit *Brown Spot* ini sering disebut bercak coklat dimana penyakit ini disebabkan oleh jamur *helminthosporium oryzae*. Penyakit ini dapat beresiko menyebabkan kematian pada tanaman muda dan menurunkan kualitas gabah. Bercak daun berwarna coklat, berbentuk oval hingga bulat berukuran sebesar biji wijen pada permukaan daun menjadi gejala yang paling umum pada penyakit ini.

3. *Leafblight*

Penyakit ini adalah penyakit paling berbahaya bagi tanaman padi. *Leafblight* disebabkan oleh cendawan yang menyebabkan munculnya lesi pada daun. Penyakit ini dapat terjadi pada seluruh bagian tanaman secara bersamaan. Penyakit ini disebabkan oleh jamur *Magnaporthe oryzae* (Nasution, 2014).

Identifikasi penyakit ini adalah dengan melihat bentuk bercak berbentuk berlian di area daun. Pada daun, penyakit ini dapat bervariasi sesuai dengan keadaan lingkungan, umur dan ketahanan varietas tanaman. Pada varietas yang rentan, penyakit ini mulanya akan berwarna abu kehijauan dan meluas dengan cepat menjadi warna coklat sepanjang beberapa sentimeter (Nasution, 2014).

4. Healthy

Daun padi yang sehat dan tidak terinfeksi oleh penyakit apapun adalah yang memiliki warna daun hijau tua. Tidak layu dan tampak segar adalah ciri lain dari daun padi yang sehat. Juga tidak terdapat kondisi yang ganjil atau perubahan warna yang aneh pada daun (Haris, 2020).

2.3.2. Citra Digital

Citra merupakan representasi visual dari suatu objek, seseorang, atau lokasi peristiwa yang dihasilkan oleh mesin optis seperti lensa atau kamera. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra monokrom (keabuan) setiap nilai merepresentasikan intensitas dan piksel dalam rentang $[0, 255]$. Pada citra berwarna direpresentasikan oleh 3 nilai warna dengan intensitas komponen merah (R), hijau (G), dan biru (B) (Siahaan & Sianipar, 2020)

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Gambar 2.1. Matriks Citra Digital

Pada Gambar 2.1. merupakan contoh penulisan citra digital ke dalam suatu matriks. Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$ di mana x dan y adalah koordinat spasial (bidang), dan amplitudo dari f pada setiap pasangan koordinat (x, y) disebut intensitas atau skala keabuan dari bayangan pada titik tersebut. Jika nilai x , y , dan intensitas f semuanya berhingga, besaran diskrit, kita menyebut bayangan itu sebagai bayangan digital (Gonzalez & Woods, 2008).

2.3.3. Artificial Intelligence

Artificial Intelligence (AI) atau yang biasa disebut sebagai kecerdasan buatan adalah salah satu rancangan dimana kecerdasan ditambahkan ke dalam sistem yang memiliki konfigurasi sebagai bentuk kecerdasan ilmiah. AI

dimasukkan ke dalam sistem komputer dengan tujuan untuk melakukan tindakan layaknya manusia secara natural dengan dibekali kemampuan penalaran. (Huawei Technologies Co., 2021)

Dalam dunia komputasi, AI mampu membentuk cabang yang berkaitan dengan perilaku, pembelajaran dan adaptasi pada suatu sistem. Dalam dunia penelitian, AI biasa digunakan untuk mengotomasi sistem dalam menyelesaikan masalah dan menjadi disiplin ilmu yang memiliki fokus untuk menyediakan solusi riil pada kehidupan nyata (Adib Bin Rashid, 2024). Dewasa ini, kebanyakan AI digunakan pada beberapa bidang seperti militer, ekonomi, farmasi, dan *game*.

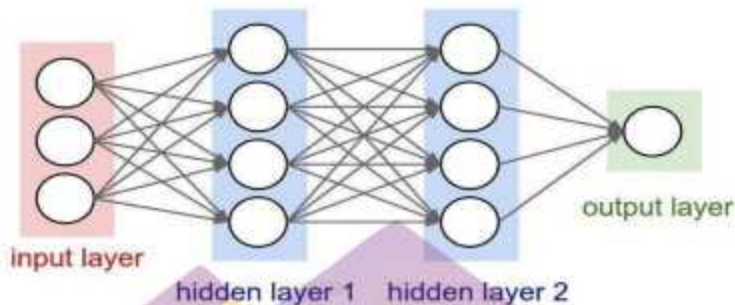
Garis besar dalam AI dibagi menjadi dua yaitu AI Konvensional dan *Computational Intelligence* (CI). AI Konvensional banyak melibatkan pembelajaran berbasis analisa statistik dengan beberapa metode seperti *bayesian network*, sistem pakar dan sebagainya. Sedangkan CI lebih melibatkan pembelajaran iteratif yang berasal dari data empiris dengan menggunakan beberapa metode seperti *fuzzy*, *neural network* dan komputasi evolusioner (Jones, 2009)

2.3.4. Deep Learning

Deep learning merupakan cabang ilmu dari *machine learning* yang lebih memiliki basis kepada jaringan syaraf tiruan untuk melatih sistem agar mampu meniru perilaku manusia secara natural. Contoh dari tindakan yang dimaksud adalah seperti mesin melakukan pembelajaran dari data yang diinputkan sehingga mampu melakukan pengambilan keputusan. *Deep learning* biasa digunakan untuk melakukan pemrosesan dari berbagai media seperti teks, gambar dan suara (Sharma et al., 2021).

Layer pada *deep learning* sendiri terdapat 3 bagian yaitu *input layer*, *hidden layer* dan *output layer*. *Input layer* berisikan *node* atau titik yang memiliki nilai. Lalu *hidden layer* berisikan algoritma untuk meminimalisasi terjadinya *error* waktu *output*. Sedangkan *output layer* bertugas untuk menampilkan hasil perhitungan matematis dari *hidden layer* sebesar *input* yang diterima. Semakin banyak *layer* dan lengkap label dalam model pembelajaran, maka model

pembelajaran akan semakin baik (Sarker, 2022).



Gambar 2.2. Arsitektur *Deep Learning*
Sumber: (J. Kim et al., 2022)

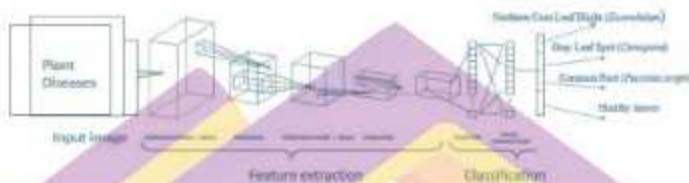
Pada gambar 2.2. di atas dapat diketahui bahwa bahwa setiap *node* pada setiap *layer* di *deep learning* akan langsung terhubung dengan *node* lain pada *layer* berikutnya. Arsitektur tersebut dapat disebut sebagai *Multi Layer Perceptron* (MLP). Semakin banyak *layer* yang terdapat pada *hidden layer* akan membantu optimasi dalam melakukan proses klasifikasi.

Oleh sebab itu, *deep learning* mampu mempelajari berbagai model abstraksi pada data dengan menggunakan proses pada model komputasi yang terdiri dari beberapa *layer*. Hal ini tentu sangat membantu peneliti dalam melakukan pengenalan objek, suara, deteksi objek dan lain sebagainya (Ilin et al., 2017). Berikut adalah beberapa algoritma *deep learning* dan kegunaannya antara lain *Deep Neural Network* (DNN) untuk *voice recognition*, *Convolutional Neural Network* (CNN) untuk *image processing*, *Recurrent Neural Network* (RNN) untuk translasi dan lain sebagainya (Anggraeni et al., 2022).

2.3.5. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu metode dalam *deep learning* untuk melakukan pengolahan citra yang dirancang dari cara kerja sistem saraf manusia. Metode ini digunakan untuk melakukan ekstraksi fitur dan pengklasifikasian data berupa gambar. CNN mempelajari *input* dari suatu citra dengan melakukan perhitungan bobot terlebih dahulu untuk dilakukan

pengelompokan. (Zewen Li, 2021). Oleh sebab itu, CNN telah digunakan oleh beberapa orang untuk melakukan komputasi pada bidang segmentasi, pengenalan pola, deteksi dan klasifikasi (Taye M.M, 2023). CNN memiliki dua tahap dalam proses pembelajarannya, yaitu proses feature learning dan tahap klasifikasi seperti pada gambar 2.3.



Gambar 2.3. Tahapan *Convolutional Neural Network* (CNN)

Sumber : (Alzubaidi, (2022).

Gambar di atas memperlihatkan bahwa terdapat tiga tahap dalam CNN, yaitu *input image*, *feature extraction* dan *classification*. Pada *feature extraction*, terdapat beberapa *layer* di dalamnya seperti *convolution layer*, *ReLU*, dan *pooling layer*. *Layer* ini akan menghasilkan vektor yang akan dijadikan data *input* untuk melakukan klasifikasi. Lalu pada tahap *classification*, CNN akan menggunakan metode *neural network* untuk mencari nilai probabilitas pada setiap klasifikasi dengan bantuan dari *fully connected layer*, fungsi aktivasi *softmax*, dan *dropout*. (D. Zahirah, 2024).

2.3.6. Tahap *Feature Learning*

Tahap *feature learning* adalah tahap dimana mesin mempelajari fitur pada citra dengan mengolah nilai pixel dan menghasilkan vektor sebagai bahan input pada layer selanjutnya (K. R. Prilianti., 2024). Tahap ini memiliki beberapa jenis layer di dalamnya, yaitu:

1. *Convolution Layer*

Convolution Layer ini adalah tempat dimana input dari citra akan melalui proses translasi untuk mempelajari fitur tepi, warna, dan bentuk menjadi fitur-fitur

berdasarkan pada citra. Operasi ini akan melalui dua matriks yaitu matriks pixel pada citra dan matriks *kernel* atau *filter*. Matriks *filter* pada CNN berisikan nilai random antara -1 hingga 1 yang berfungsi untuk melakukan pembelajaran pada area kecil di dalam citra sesuai ukuran *filter*. Ukuran *filter* ini tergantung pada jenis arsitektur apa yang dipakai. Citra yang telah diinputkan akan diolah hingga tercipta *output* berupa tumpukan *feature map* dari semua lapisan matriks *filter*. Operasi ini dapat dilihat pada gambar 2.4.

$$\begin{bmatrix} 2 & 4 & 9 & 1 & 4 \\ 2 & 1 & 4 & 4 & 6 \\ 1 & 1 & 2 & 9 & 2 \\ 7 & 3 & 5 & 1 & 3 \\ 2 & 3 & 4 & 8 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ -4 & 7 & 4 \\ 2 & -5 & 1 \end{bmatrix} = \begin{bmatrix} 51 & 66 & 20 \\ 31 & 49 & 101 \\ 15 & 53 & -2 \end{bmatrix}$$

Image Filter / Kernel Feature

Gambar 2.4. Convolution Operation

Sumber: (Patel, 2020)

Untuk mencari nilai dalam *feature map*, perhitungan yang dilakukan adalah perkalian setiap elemen dalam pixel input dengan elemen dan ukuran kernel. Pada perhitungan jumlah *feature map*, terdapat pertimbangan perhitungan lain bernama *stride* atau jumlah pergeseran pixel dan *padding* atau parameter penentu jumlah pixel yang berisikan nilai 0 dan ditambahkan di setiap sisi dari citra yang di *input*. Contoh perhitungan dari operasi pada *convolution layer* ini dapat dilihat pada gambar 2.5 Dibawah.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & 9 & 1 & 4 & 0 & 0 \\ 0 & 2 & 1 & 4 & 4 & 6 & 0 & 0 \\ 0 & 1 & 1 & 2 & 9 & 2 & 0 & 0 \\ 0 & 7 & 3 & 5 & 1 & 3 & 0 & 0 \\ 0 & 2 & 3 & 4 & 8 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ -4 & 7 & 4 \\ 2 & -5 & 1 \end{bmatrix} = \begin{bmatrix} 21 & 59 & 37 & -19 & 2 \\ 30 & 51 & 66 & 20 & 43 \\ -14 & 31 & 49 & 101 & -19 \\ 59 & 15 & 53 & -2 & 21 \\ 49 & 57 & 64 & 76 & 10 \end{bmatrix}$$

Image Filter / Kernel Feature

Gambar 2.5. *Convolutional Operation* dengan *kernel*, *stride* dan *padding*

Sumber : (Patel, 2020)

Input citra yang mulanya berukuran 5x5 berubah menjadi 7x7 dikarenakan penambahan *padding* sebesar 1 pada setiap sisi dari citra. Lalu ukuran *filter* yaitu 3x3 dan *stride* sebesar 1 sehingga ditemukan ukuran *feature map* yaitu 5x5 dengan persamaan :

$$H = \left(\frac{I + 2P - K}{S} \right) + 1$$

Penjelasan:

I = ukuran *input* citra gambar

P = nilai *padding*

K = ukuran *kernel*

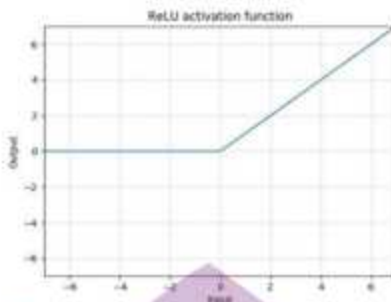
S = nilai *stride*

H = ukuran *feature map*

2. *Rectified Linear Unit* (ReLU)

ReLU merupakan salah satu fungsi aktivasi yang berfungsi untuk merubah nilai *feature map* pada *convolution layer* dalam range 0 hingga tak terbatas. Sederhananya, ReLU adalah fungsi yang menggantikan nilai negatif pada *feature map* menjadi 0 dan nilai selain negatif atau lebih besar sama dengan 0 akan tetap. Perhitungan ini dapat dilihat pada persamaan berikut: $f(x) = \max(0, x)$

$f(x)$ memiliki arti hasil dari nilai 0 dan x . x adalah nilai dari *feature map*. Ketika x memiliki nilai yang lebih kecil daripada 0, maka $f(x)$ akan bernilai 0. Ketika x bernilai lebih atau sama dengan 0, maka $f(x)$ bernilai x . Hal ini dapat dilihat pada gambar 2.6. Di bawah.

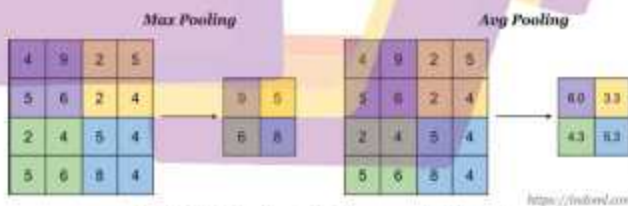


Gambar 2.6. Fungsi Aktivasi ReLU

Sumber: (Patel, 2020)

3. Pooling Layer

Pooling layer berfungsi sebagai pengurang ukuran matriks input dengan mereduksi data spasial. Perhitungan *output* pada *layer* ini menggunakan ukuran *filter* dan nilai pergeseran *stride*. Terdapat 2 jenis operasi dalam *pooling layer*, yaitu *Max pooling* dan *average pooling*. *Max pooling* memiliki cara kerja dengan menghitung nilai maksimum pada baris kolom sesuai dengan ukuran *filter* dan nilai *stride*. Sedangkan *average pooling* melakukan perhitungan dengan cara mencari rata-rata pada baris-kolom sesuai ukuran *filter* dan nilai *stride*. Hal ini dapat dilihat pada gambar 2.7 Di bawah.



Gambar 2.7. Max dan Average Pooling Layer

Sumber: (Patel, 2020)

Dengan adanya *pooling layer*, proses perhitungan komputasi akan menjadi lebih cepat karena proses ini melakukan perhitungan dengan mengurangi volume tumpukan *feature map* tanpa mengabaikan informasi penting di

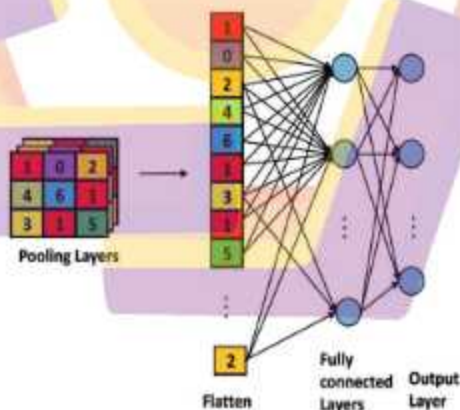
dalamnya untuk melakukan pengendalian *overfitting*. CNN menggunakan *pooling layer* dengan alasan agar ukuran dari citra gambar dapat direduksi sehingga dapat diganti dengan *convolutional layer* dengan jumlah *stride* yang sama secara mudah.

2.3.7. Tahap Klasifikasi

Tahap ini adalah tahap dimana menggunakan metode *neural network* untuk mencari nilai probabilitas pada setiap klasifikasi (Zewen Li, 2021). Tahap ini memiliki beberapa jenis layer di dalamnya, yaitu:

1. Fully Connected Layer

Layer ini menjadi tempat dimana *feature* dianalisis untuk menentukan hubungan atau korelasi antar kelas dan *feature*. Layer ini juga menjadi tempat dimana *feature* akan dibentuk ulang (*reshape feature map*) menggunakan *flatten* menjadi suatu vektor agar dapat menjadi *input* bagi *fully connected layer*. Hal ini dikarenakan *feature map* hasil dari *feature learning* masih berupa multidimensional array. Ilustrasi dari tahap ini dapat dilihat pada gambar 2.8. di bawah.



Gambar 2.8. Fully Connected Layer dan Flatten

Sumber: (Lasky, 2022)

Perhitungan dari *fully connected layer* ini dapat dilihat pada persamaan

berikut berikut.

$$y = f(W^T x + b)$$

Dimana y adalah nilai dari vektor satu dimensi yang berisikan fungsi W sebagai matriks sederhana dengan T atau bobot yang dikalikan dengan x sebagai vektor input lalu ditambah dengan b sebagai vektor bias.

2. Softmax

Softmax adalah suatu fungsi aktivasi yang melakukan fungsi probabilitas eksponensial yang dinormalisasi pada kelas yang diamati. Tujuan dari *layer* ini adalah untuk memprediksi *output* hasil klasifikasi menjadi suatu probabilitas terbesar sebagai hasil dari prediksi. Keuntungan dari *softmax* adalah dapat mengubah vektor skor nilai riil menjadi rentang 0 hingga 1 yang berjumlah 1. Oleh sebab itu, fungsi ini dapat membantu mengembalikan peluang dari setiap kelas sehingga probabilitas target akan lebih tinggi. Perhitungan dari fungsi aktivasi ini dapat dilihat pada persamaan berikut:

$$\frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)}$$

Dimana y adalah vektor input dan y_i adalah elemen ke- i pada vektor *input*. $\exp(y_i)$ memiliki arti bahwa fungsi standar eksponensial yang diaplikasikan ke y_i dan $\sum_{j=1}^n \exp(y_j)$ memiliki arti bahwa bentuk normalisasi dari $\exp(y_i)$. Ilustrasi dari fungsi aktivasi softmax dapat dilihat pada gambar 2.9, berikut.



Gambar 2.9. Fungsi Aktivasi *Softmax*

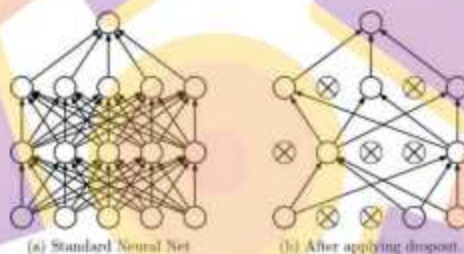
Sumber: (Koech, 2022)

Gambar di atas menjelaskan cara kerja dari fungsi aktivasi ini adalah dengan

merubah input citra menjadi angka logis atau vektor yang akan dimasukkan ke dalam perhitungan komputasi dan diurutkan berdasarkan label dengan probabilitas paling tinggi.

3. Dropout

Layer ini berfungsi untuk mengatasi *overfitting* akibat kompleksitas dari susunan *hidden layer* pada CNN. Teknik ini dapat membantu kinerja selama proses *testing*. Layer ini memiliki cara kerja dengan menghilangkan beberapa *node* acak dengan probabilitas pada setiap iterasi selama proses *training*. Akibatnya, jaringan akan menjadi lebih tipis sehingga proses *testing* akan menjadi lebih cepat dengan kemungkinan *overfitting* yang kecil. Ilustrasi dari *dropout* dapat dilihat pada gambar 2.10 berikut.



Gambar 2.10. Dropout

Sumber: (Novindasari, 2021)

2.3.8 Data Augmentation

Data Augmentation merupakan salah satu teknik yang digunakan untuk memecahkan keterbatasan pada suatu penelitian. Hal ini dikarenakan dapat mempengaruhi kinerja sistem dalam melakukan proses klasifikasi (Mikołajczyk & Grochowski, 2018). Teknik ini juga membantu dalam mengatasi *overfitting* atau kekompleksan data yang menyebabkan rendahnya akurasi waktu dilakukan pengujian (Romero Aquino et al., 2019). Secara umum, *data augmentation* dibagi menjadi dua yaitu modifikasi citra dan transformasi geometri. Modifikasi citra yang dimaksud adalah melakukan modifikasi pada warna citra dengan memainkan kontras, dan sebagainya. Sedangkan maksud dari transformasi geometri adalah

melakukan modifikasi pada citra berupa rotasi, refleksi, *resize* dan sebagainya (Kim et al., 2020).

2.3.9. Normalization Layer

Normalization Layer adalah salah satu jenis *data pre-process* yang biasa digunakan untuk mendistribusikan data secara baik agar dapat meningkatkan performa model dan menghindari *overfitting* pada data. Menurut (Widneh, 2025), *normalization layer* memiliki tujuan untuk mengatasi masalah pada pelatihan model dengan mengurangi kovariat internal pada *neural network* agar dapat menggali lebih banyak informasi pada data hingga berpengaruh pada peningkatan akurasi. Proses ini cocok digunakan pada jenis *dataset* yang memiliki variasi yang banyak karena dapat meningkatkan kecepatan pelatihan yang relevan dengan data (Ba et al., 2018).

2.3.10. Optimizer

Optimizer atau optimasi dalam *deep learning* adalah fungsi atau algoritma yang mengadaptasi atribut dalam sistem *neural network* seperti *learning rate* dan *weight*. Hal ini dikarenakan saat mesin melakukan pelatihan pada model, penyesuaian setiap *epoch* harus dilakukan untuk meminimalisir *loss function*. Oleh sebab itu, optimasi ini memiliki peran dalam meningkatkan akurasi dan mengurangi total *loss*. Beberapa optimasi yang memiliki kemampuan baik adalah seperti Adam, RMSprop dan SGD (Mehmood et al., 2023). Beberapa modifikasi yang dilakukan oleh peneliti untuk mengoptimalkan kinerja optimasi ini juga menghasilkan beberapa terobosan baru seperti modifikasi dari Adam yang menghasilkan AdaMax, Nadam, EAdam dan beberapa modifikasi lain pada optimasi yang berbeda.

2.3.11. Confusion Matrix

Confusion matrix merupakan salah satu teknik yang sering digunakan untuk mengetahui tingkat kesalahan pada proses klasifikasi data. *Confusion matrix* biasanya diinterpretasi dengan angka 0 hingga 1. Semakin kecil angka yang didapat, maka semakin baik pula sistem klasifikasi yang digunakan (J.

Erbani, 2024). Dengan begitu peneliti akan mampu mengetahui keakuratan dan tingkat *error* sistem klasifikasi. Cara *confusion matrix* dalam menentukan akurasi adalah dengan menilai beberapa parameter berikut:

Tabel 2.2. Parameter *Confusion Matrix*

Kelas riil / Klasifikasi	<i>Predicted</i>	<i>Predicted</i>
<i>Actual</i>	True Positive (TP)	True Negative (TN)
<i>Actual</i>	False Positive (FP)	False Negative (FN)

Penjelasan:

- TP (*True Positive*) = data bernilai positif yang diklasifikasi dengan benar
 TN (*True Negative*) = data bernilai negatif yang diklasifikasi dengan benar
 FN (*False Negative*) = data bernilai negatif yang tidak diklasifikasi dengan benar
 FP (*False Positive*) = data bernilai positif yang tidak diklasifikasi dengan benar

Teknik ini menilai suatu klasifikasi berdasarkan *accuracy*, *sensitivity* dan *specificity* berdasarkan nilai dari parameter di atas (Rainio, O, 2024). Berikut adalah penjelasan dari tiap taraf evaluasi tersebut:

1. *Accuracy*

Nilai yang didapat dengan membandingkan data lalu diklasifikasi dalam jumlah semua *dataset* yang menunjukkan keakuratan dari suatu klasifikasi. Semakin banyak data yang digunakan dalam melakukan klasifikasi, semakin baik pula nilai akurasi. Rumus dari perhitungan akurasi dapat dilihat pada persamaan di bawah:

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN} \right) \times 100\%$$

2. *Specificity*

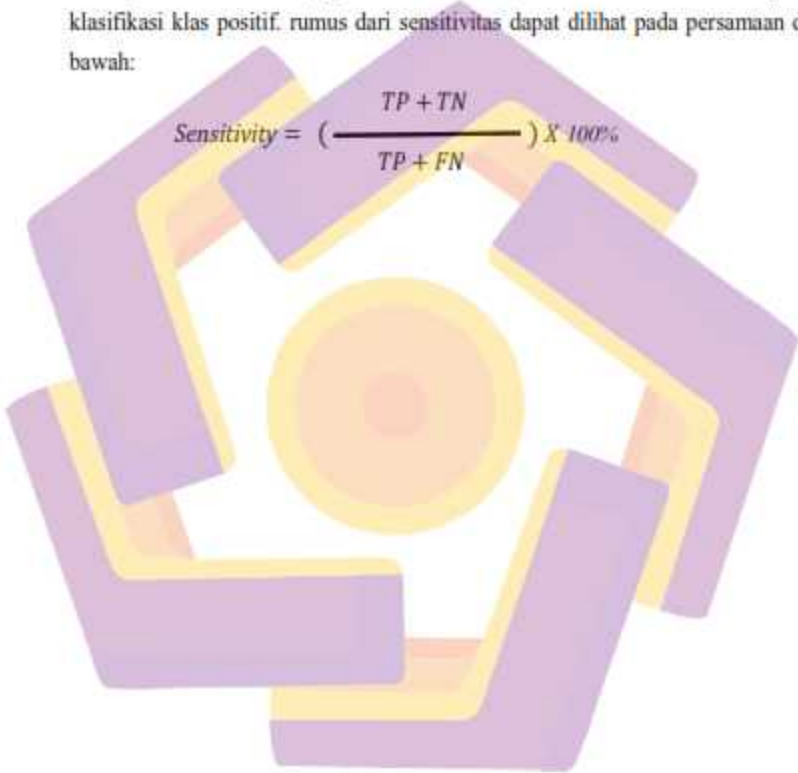
Nilai yang merepresentasi jumlah data bernilai negatif yang terklasifikasi sebagai kelas negatif. Semakin besar nilai spesifitas, maka semakin baik pula proses klasifikasi pada kelas negatif suatu data. rumus dari spesifitas dapat dilihat pada persamaan di bawah :

$$Specificity = \left(\frac{TN}{TN + FP} \right) \times 100\%$$

3. Sensitivity

Nilai yang merepresentasi jumlah data bernilai positif dan terklasifikasi sebagai kelas positif. Semakin tinggi nilai sensitivitas, maka akan semakin baik pula klasifikasi kelas positif. rumus dari sensitivitas dapat dilihat pada persamaan di bawah:

$$Sensitivity = \left(\frac{TP}{TP + FN} \right) \times 100\%$$



BAB III

METODE PENELITIAN

3.1. Jenis Penelitian dan Data

Penelitian ini adalah jenis penelitian kuantitatif dikarenakan data yang diperoleh diukur atau diproses dengan menggunakan teknik matematis. Pada penelitian ini data berupa citra yang berisikan informasi dalam bentuk matriks yang kemudian dilakukan perhitungan matematis. Tujuan dari proses tersebut adalah untuk melakukan klasifikasi sehingga mengetahui jenis penyakit pada daun padi yang diteliti.

Jenis data pada penelitian ini adalah menggunakan tipe jenis data yang bersifat sekunder. Jenis data ini biasa diperoleh melalui media, perantara atau pihak lain. Maksud dari perantara adalah media yang mampu mendukung segala bentuk keperluan dari kredibilitas suatu data seperti buku, literatur dan website.

3.2. Metode Pengumpulan Data

Data pada penelitian ini diperoleh dari situs [www.kaggle.com](https://www.kaggle.com/datasets/shayanriyaz/riceleafs) yaitu <https://www.kaggle.com/datasets/shayanriyaz/riceleafs> dengan jumlah data sekunder yang digunakan adalah 900 berkas dengan masing-masing pembagian 716 berkas untuk *training* dan 184 berkas untuk *validation*. ini memiliki 4 label untuk setiap kondisi daun yang ada yaitu:

1. *Brown Spot*

Kondisi daun dimana mengalami perubahan warna bercak coklat memanjang yang sejajar dengan ibu daun tulang yang disebabkan oleh jamur *Drechslera oryzae*.

2. *Hispa*

Kondisi pada daun padi dimana muncul garis paralel berwarna putih di sepanjang sumbu padi. Gejala lain adalah daun yang menjadi layu. Hal ini disebabkan oleh kumbang yang memakan bagian luar epidermis atas daun padi.

3. *Leafblight*

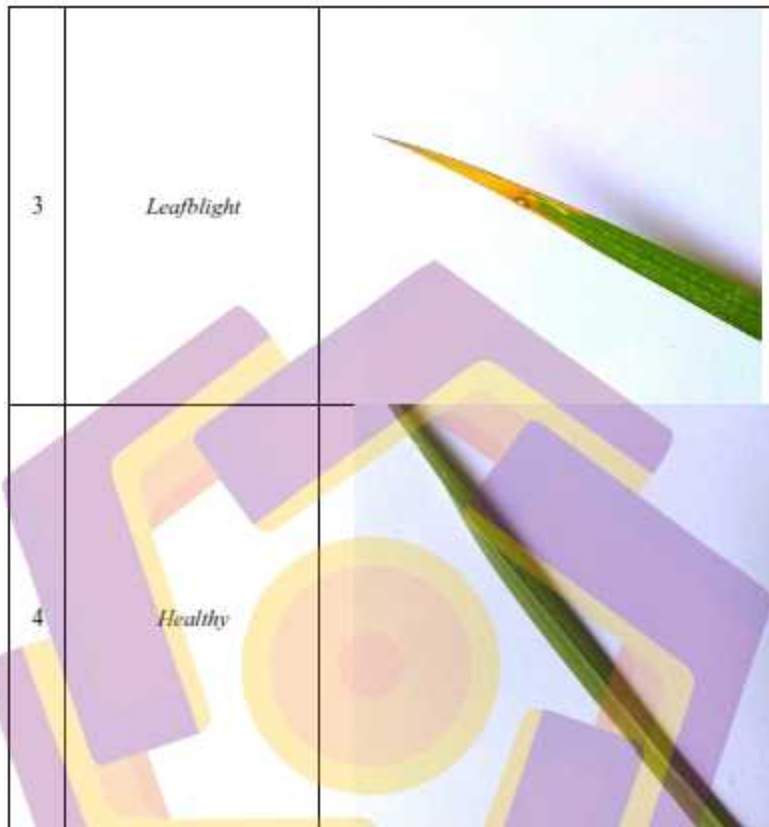
Kondisi dimana daun terinfeksi oleh jamur *Pyricularia Grisea* yang menyebabkan munculnya bercak coklat pada daun berbentuk belah ketupat.

4. *Healthy*

Kondisi dimana daun padi tidak mengalami kelainan apapun dan dinyatakan sehat. Hal ini ditunjukkan dengan warna daun yang tidak terdapat perubahan dan kondisi fisik daun yang normal. Untuk lebih jelasnya, bentuk dan gambar dari setiap kondisi daun yang dipaparkan dapat dilihat pada tabel 3.

Tabel 3.1. Bentuk Kondisi Daun

No	Nama Kondisi Daun	Gambar
1	<i>Brown Spot</i>	
2	<i>Hispa</i>	



3.3. Pengolahan Data

Setelah berhasil memperoleh data yang diinginkan, langkah selanjutnya dalam penelitian ini adalah melakukan pengolahan dari data yang meliputi 3 tahap yaitu *pre-processing*, *modelling* dan *evaluating*.

1. *Pre-Processing*

Tahap ini adalah tahap awal dimana data akan dirapikan sedemikian rupa untuk kelancaran pada tahap selanjutnya. Tahap ini akan menentukan kelayakan data yang diproses. Penelitian ini menggunakan teknik *data augmentation* dan *normalization layer*. Tujuan dari proses ini adalah untuk menyiapkan data agar siap digunakan pada tahap *modelling* serta mengurangi resiko terjadinya *overfitting* pada data selama masa *training*. Langkah-langkah dalam tahap

pre-processing ini dapat dilihat pada gambar 3.1.



Gambar 3.1. Alur *Preprocessing* Data

Teknik pertama yang digunakan pada tahap *pre-processing* yaitu *data augmentation*. Jenis teknik *data augmentation* yang digunakan adalah *rotation*, *translation*, *flip* dan *contrast*. Teknik ini digunakan untuk mengurangi resiko adanya *overfitting* dengan cara membuat data tambahan dengan melakukan perubahan pada data asli. Lalu setelah itu ditambahkan *normalization layer*. Teknik ini digunakan untuk menyiapkan *dataset* agar dapat digunakan pada tahap *modelling*.

2. *Modeling*

Tahap ini adalah tahap dimana data yang sudah dilakukan *pre-process* dimasukkan ke dalam *deep learning* untuk dilakukan klasifikasi menggunakan model arsitektur *Convolutional Neural Network* (CNN). Komposisi perbandingan rasio *split* data yaitu 80% untuk *training* dan 20% untuk *validation*.

Selama pemrosesan pada model, dilakukan penambahan fungsi optimasi untuk meningkatkan akurasi prediksi dan mengurangi *loss* pada model dengan melakukan *update* pada setiap iterasi yang disesuaikan dengan *learning rate*. Untuk model yang digunakan adalah model arsitektur VGG16, Xception,

MobileNetV2 dan InceptionV3. Menurut Wan, G., & Yao, L model arsitektur tersebut telah terbukti efektif dan dapat digunakan dalam pemilihan model CNN. Sedangkan pemrosesan model dilakukan dengan menggunakan 3 proses yaitu *scratch*, *pre-train* dan *fine-tuning*, dan *unfreeze layer*. *Scratch Model* adalah proses awal pada tahap pelatihan model dimana model dibuat tanpa menggunakan bobot yang sudah tersedia sebelumnya agar memiliki fleksibilitas pada pengaturan arsitektur yang sesuai dengan *task*.

Proses *pre-trained* dan *fine tuning* adalah proses dimana model di *load* dengan bobot yang sudah dilatih sebelumnya untuk membantu model dalam melakukan ekstraksi fitur serta agar dapat disesuaikan dengan *dataset* yang digunakan untuk studi kasus. Pada umumnya, hasil akurasi pada saat validasi selama training akan menjadi lebih baik daripada hasil akurasi training dikarenakan proses ini sangat membantu pengenalan fitur dan pengenalan pada data baru.

Sedangkan *Unfreeze layer* adalah proses lanjutan dari proses *fine tuning* yang dilakukan untuk menyesuaikan model dengan kecepatan pembelajaran. Hal ini ditujukan agar ekstraksi fitur bekerja lebih baik dengan model yang digunakan. Menurut peneliti, proses ini dirasa penting karena penelitian ini menggunakan kumpulan data yang agak berbeda dari kumpulan data dalam *dataset* "imagenet". Oleh sebab itu, proses tersebut akan membantu proses ekstraksi fitur lebih optimal dikarenakan memiliki data penyesuaian yang lebih banyak.

Seluruh proses ini dikendalikan melalui Hyperparameter Tuning, termasuk penggunaan fungsi optimasi Adam dan pengaturan *learning rate* sebesar 0.0001 yang adaptif untuk meminimalkan nilai *loss* secara konsisten pada setiap iterasi.

3. Evaluating

Tahap ini dilakukan untuk mengukur bagaimana akurasi, konsistensi dan kinerja model dalam melakukan prediksi data. Proses ini dilakukan dengan melalui pengujian secara objektif kepada model agar dapat menjamin bahwa model memiliki kemampuan dalam memecahkan permasalahan sesuai kebutuhan. Evaluasi hasil klasifikasi dilakukan menggunakan teknik *cross validation* dan *confusion matrix* untuk menilai tingkat akurasi dari model arsitektur yang

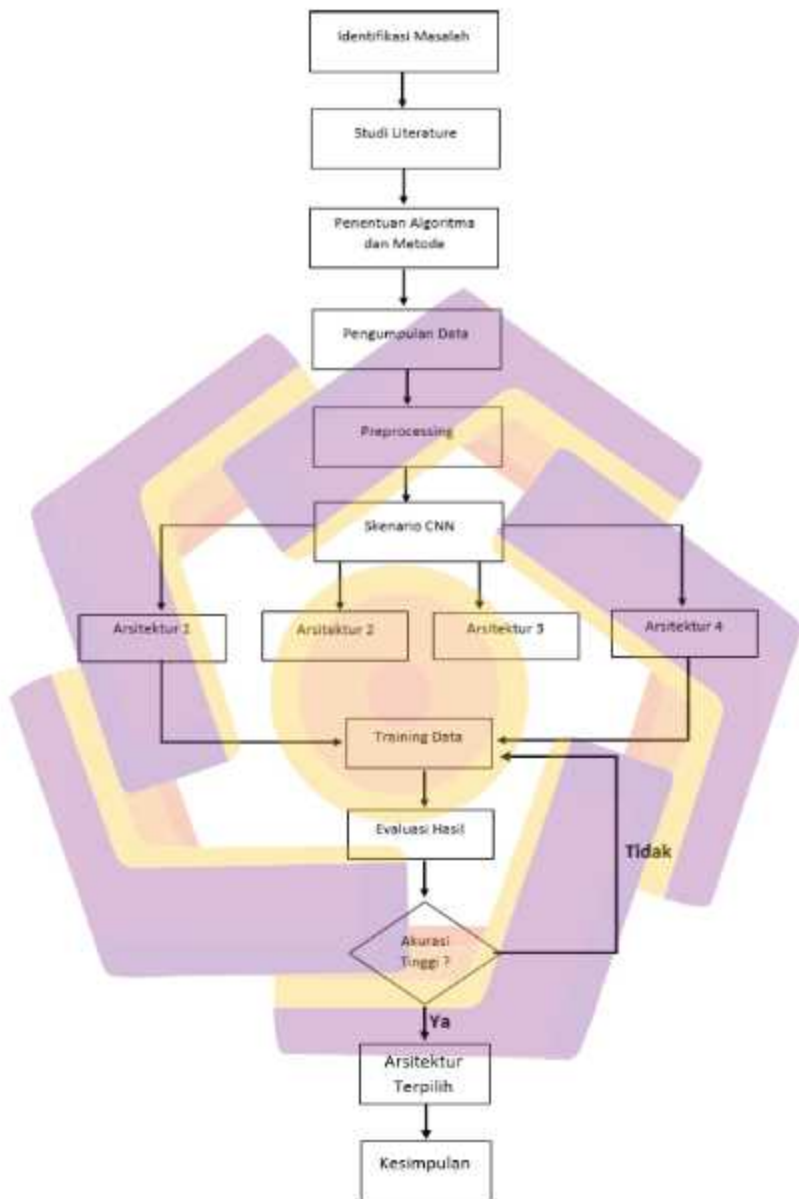
diterapkan. Pengukuran evaluasi akan didasarkan pada nilai yang dihasilkan. Teknik evaluasi *cross validation* dilakukan dengan cara membagi *dataset* menjadi beberapa bagian. Bagian data yang sudah dibagi kemudian dilakukan iterasi. Hasil akhir evaluasi diambil dari rata-rata hasil dari setiap iterasi dari proses yang dilakukan. Sedangkan *confusion matrix* ini digunakan untuk mengukur performa dengan melakukan perbandingan prediksi dari model dan label yang sebenarnya dari *dataset*. Untuk memudahkan dalam melakukan analisa, pada umumnya hasil dari *confusion matrix* akan divisualisasikan. Terdapat beberapa percobaan pengolahan data yang dilakukan pada penelitian ini yaitu percobaan normal yang berisikan konfigurasi *kode modelling* dan evaluasi tanpa *pre-process*, lalu dilanjut dengan penambahan teknik *pre-processing data augmentation* untuk percobaan *data augmentation*, dan penambahan teknik *pre-process normalizational layer* untuk percobaan *normalizational layer*. Dilakukan juga percobaan perbandingan fungsi optimasi pada model menggunakan tiga jenis fungsi optimasi umum yaitu SGD, RMSProp, dan Adam untuk melihat pengaruh fungsi ini terhadap kinerja model.

3.4. Metode Analisis Data

Analisa yang dilakukan adalah dengan melihat hasil evaluasi menggunakan *confusion matrix* dengan menilai berdasarkan tingkat akurasi klasifikasi menggunakan model arsitektur *Convolutional Neural Network (CNN)*. Sajian dari *confusion matrix* adalah berupa tabel perhitungan klasifikasi dengan pengukuran evaluasi. Hasil analisa ini nantinya akan dapat melihat seberapa baik metode yang diterapkan untuk melakukan proses klasifikasi.

3.5. Alur Penelitian

Alur penelitian akan dipaparkan pada media berbentuk diagram alur di bawah. Hal ini dilakukan untuk lebih memudahkan dalam memahami setiap langkah atau proses penelitian secara rinci. Alur penelitian dapat dilihat pada gambar 12.



Gambar 3.2. Kerangka Alur Penelitian

Penyelesaian penelitian dalam mengidentifikasi penyakit pada daun tanaman padi dilakukan dalam beberapa kegiatan yang dilakukan secara bertahap. Kerangka alur penelitian yang akan dilaksanakan ditunjukkan pada Gambar 3.2. Terdapat beberapa tahapan untuk menyelesaikan penelitian tentang identifikasi penyakit pada daun tanaman padi. Tahapan-tahapan penelitian ini dapat dijabarkan lebih lanjut sebagai berikut:

a. Identifikasi Masalah

Proses ini merupakan tahap dimana penulis mencari tahu permasalahan yang ada dari keadaan di lapangan maupun informasi berupa artikel terkait atau berita publik dengan sumber terpercaya.

b. Studi Literatur

Pada proses ini penulis mencari tahu informasi dengan membaca berbagai jurnal penelitian dan buku yang dianggap relevan dengan permasalahan yang akan diangkat. Proses ini juga sebagai bahan rujukan penelitian untuk memilih metode atau algoritma yang dianggap sesuai dengan permasalahan.

c. Penentuan Algoritma dan Metode

Setelah membaca berbagai jurnal yang berkaitan dengan permasalahan, pada proses ini penulis menentukan algoritma dan metode yang paling cocok digunakan untuk menyelesaikan permasalahan yang ada. Algoritma yang digunakan pada penelitian ini ialah Convolutional Neural Network (CNN) untuk membantu proses deteksi dan klasifikasi.

d. Pengumpulan Data

Tahap berikutnya adalah mengumpulkan dataset yang diperlukan dan relevan dengan algoritma dan metode yang akan digunakan. Data pada penelitian ini merupakan data sekunder yang mana sumber didapat dari pihak ketiga atau tidak secara langsung melainkan open data yang dapat diakses oleh umum.

e. Preprocessing

Pada tahap ini dilakukan proses augmentasi data guna menambah jumlah dataset yang ada serta dilakukan pemilihan dan proses enhancement data lebih lanjut, serta pembagian dataset citra penyakit daun padi yang akan digunakan sebagai data training dan data testing dengan presentase tertentu.

f. Percobaan Arsitektur CNN

Karena jenis penelitian bersifat eksperimental maka alur penelitian ini terdapat langkah percobaan. Pada tahapan percobaan CNN ini penulis ingin mencari akurasi tertinggi. Maka untuk mencari hasil tersebut dilakukanlah 4 percobaan menggunakan metode Convolutional Neural Network dengan arsitektur yang dibedakan pada lapisannya.

g. Training Data

Setelah melakukan percobaan yang ada maka mulai dilakukan percobaan menggunakan dataset citra daun padi yang telah disiapkan sebelumnya. Training data bertujuan untuk melatih keempat percobaan yang ada agar saat tahap evaluasi dapat menghasilkan nilai akurasi yang tinggi.

h. Evaluasi Hasil

Pada tahap ini model yang telah melalui tahap sebelumnya akan dilakukan pengujian model evaluasi. Dalam tahapan evaluasi akan mendapatkan nilai confusion matrix dari setiap percobaan untuk menentukan nilai akurasi presisi dan recall.

i. Kesimpulan

Setelah mendapatkan hasil penelitian berupa data obyektif yang dihasilkan oleh beberapa percobaan, tahap selanjutnya adalah membuat kesimpulan dengan menyajikan hasil dari percobaan yang telah dilakukan dengan beberapa fakta terkait arsitektur Convolutional Neural Network (CNN) terhadap tingkat akurasi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

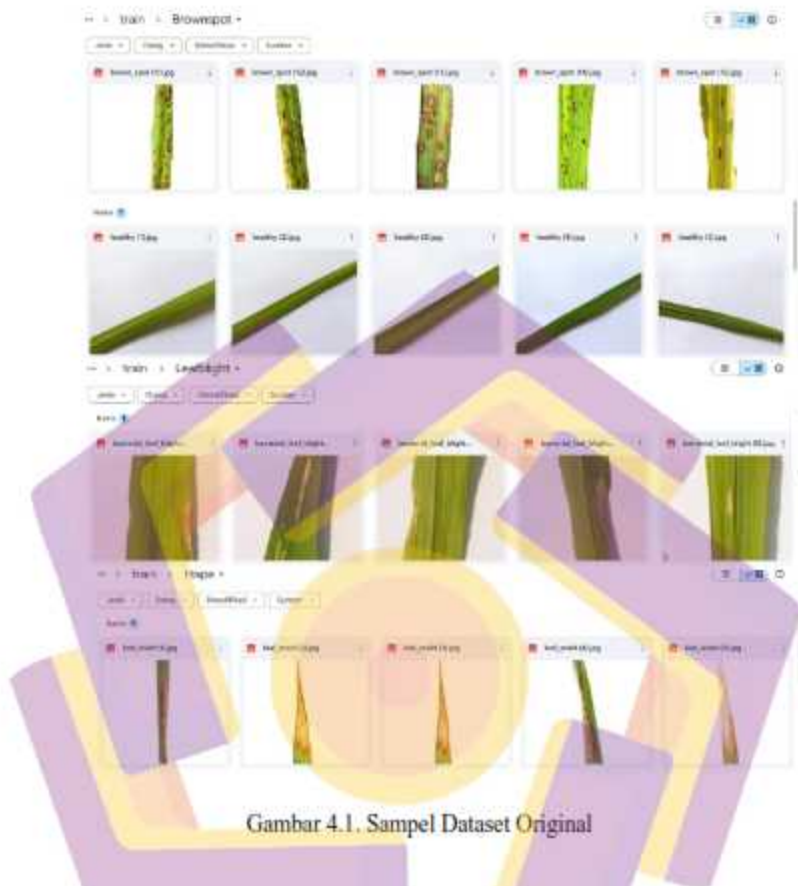
4.1. Pengumpulan Data

Metode pengumpulan data pada penelitian ini adalah dengan melakukan pencarian data pada website penyedia data publik yang bisa diakses oleh semua orang. Data yang digunakan pada penelitian ini didapatkan dari website Kaggle dengan spesifik alamat <https://www.kaggle.com/datasets/shayanriyaz/riceleafs>. Kumpulan data (dataset) ini bernama Rice Leafs yang diambil dari wilayah Sulawesi Tenggara, Indonesia. Keterangan yang ada pada dataset tersebut adalah data ini diambil dengan menggunakan kertas putih sebagai latar belakang dan data ini diambil pada bulan September 2020. Dataset ini dipilih karena data citra diambil langsung dari padi di wilayah Indonesia serta beberapa penelitian sebelumnya juga mengolahnya seperti yang dilakukan oleh (Khoiruddin et al, 2022) dan juga (Julianto et al, 2022).

Tabel 4.1. Jumlah Dataset Original

Sampel Data	Jumlah Data	Sumber Data
<i>Brownspot</i>	226	Kaggle (Rice Leafs)
<i>Healthy</i>	226	
<i>Leafblight</i>	226	
<i>Hispa</i>	226	
Total	904	

Detail dari jumlah dataset awal yang digunakan pada penelitian ini di tampilan pada tabel 4.1. Total dari keseluruhan data yang digunakan sebelum dilakukan proses preprocessing selanjutnya adalah 904 citra, dimana sebarannya dibagi ke dalam 4 kelas dengan perincian Brownspot, Healthy, Leafblight dan Hispa masing- masing berjumlah 226 buah. Contoh dataset original per kelas dari website kaggle sebelum dilakukan preprocessing dapat dilihat pada Gambar 4.1.



Gambar 4.1. Sampel Dataset Original

4.2. Analisis Data

Data yang telah didapatkan akan dilakukan analisis data untuk siap digunakan pada tahap model klasifikasi. Tabel 4.1 menunjukkan persebaran data daun padi yang digunakan pada penelitian ini sebagai data yang sedikit, maka pada tahap ini akan melakukan beberapa perlakuan untuk dataset guna membuat data yang digunakan menjadi cukup ideal untuk sebuah model klasifikasi.

4.2.1. Augmentasi Data

Augmentasi data merupakan teknik untuk menambahkan jumlah citra

yang digunakan dengan cara mengubah atau memodifikasi citra. Augmentasi data juga memiliki fungsi untuk menambah keberagaman data sehingga data yang digunakan untuk klasifikasi model lebih kaya akan informasi. Augmentasi data citra dapat dilakukan dengan cara transisi, transformasi, penambahan noise, rotasi, pembesaran, dan lain sebagainya. Pada penelitian ini proses augmentasi data dilakukan pada seluruh kelas menggunakan beberapa teknik seperti memutar searah jarum jam, pembalikan vertikal, pergeseran warp, gambar buram, dan lain sebagainya. Implementasi augmentasi data citra dilakukan menggunakan tools *jupyter notebook* dengan library *keras*.

Tabel 4.2 Jumlah Dataset Augmentasi

Sampel Data	Jumlah Data	Sumber Data
<i>Brownspot</i>	438	Kaggle (Rice Leafs)
<i>Healthy</i>	438	
<i>Leafblight</i>	438	
<i>Hispa</i>	438	
Total	1752	

Jumlah dataset yang diperoleh dari Kaggle telah melalui proses augmentasi untuk meningkatkan variasi dan kuantitas data. Sebagaimana tersaji pada Tabel 4.2, setiap kelas (*Brownspot*, *Healthy*, *Hispa*, dan *Leafblight*) dilakukan augmentasi hingga mencapai jumlah yang seragam, yaitu 438 gambar per kelas, sehingga total keseluruhan dataset menjadi 1.752 gambar. Perlu ditekankan bahwa angka tersebut merupakan hasil akhir setelah dilakukan seleksi manual (*manual curation*). Pada awalnya, proses augmentasi menghasilkan jumlah citra yang lebih besar, namun dilakukan eliminasi terhadap citra yang mengalami distorsi berlebihan guna menjaga kualitas fitur dan meminimalkan noise pada saat proses pelatihan model. Berikut adalah potongan kode augmentasi data pada daun padi untuk menambah datanya.

```

def h_flip(image):
    return np.fliplr(image)

def v_flip(image):
    return np.flipud(image)

def warp_shift(image):
    transform = AffineTransform(translation=(0,40))
    warp_image = warp(image, transform, mode='wrap')
    return warp_image

def add_noise(image):
    return random_noise(image)

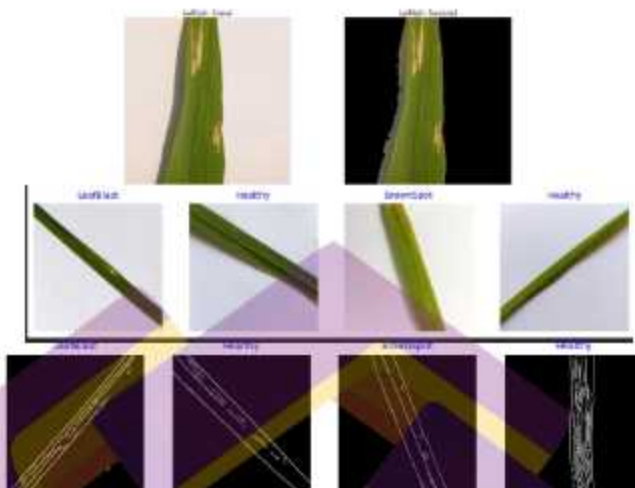
def blur_image(image):
    return cv2.GaussianBlur(image, (0, 0), 0)

transformations = \
    ( 'horizontal flip': h_flip,
      'vertical flip': v_flip,
      'warp shift': warp_shift,
      'adding noise': add_noise,
      'blurring image': blur_image
    )

```

Pada potongan kode di atas kita lihat ada sekitar 5 pendefinisian *function* untuk melakukan augmentasi data. Teknik augmentasi dilakukan secara acak (*random*) pada setiap iterasi latih menggunakan kombinasi transformasi antara lain memutar gambar searah jarum jam, *h_flip* atau singkatan dari *horizontal flip* untuk penempatan gambar secara horizontal, *v_flip* atau singkatan dari *vertical flip*, untuk membalikkan gambar secara vertikal atau tegak lurus dari atas ke bawah, *warp_shift* untuk membelokkan dan menggeser gambar, *add_noise* untuk menambahkan *noise* atau derau acak ke dalam gambar, dan terakhir *blur_image* untuk mengaburkan atau menghaluskan gambar dengan cara mengurangi detail tajam dan meredam tingkat kontras dalam gambar. Kemudian baris terakhir adalah pemanggilan masing-masing *function* yang disimpan pada variable *transformations* untuk digunakan pada *script* lanjutan. Contoh hasil augmentasi data citra pada masing-masing kelas dapat kita lihat pada Gambar 4.2.





Gambar 4.2. Sampel Dataset Augmentasi Original

Proses augmentasi yang dilakukan menghasilkan jumlah citra sintetis yang secara kuantitatif lebih besar dari dataset asli. Namun, dalam penelitian ini dilakukan pemilihan manual (*manual curation*) terhadap hasil augmentasi sebelum masuk ke tahap pelatihan model. Pemilihan manual ini bertujuan untuk mengeliminasi citra hasil augmentasi yang mengalami distorsi berlebihan (seperti rotasi atau *cropping* yang menghilangkan fitur utama penyakit) yang dapat menyebabkan model mempelajari fitur yang menyesatkan (*noise*). Dengan demikian, meskipun jumlah dataset akhir tetap lebih banyak dari dataset original, setiap citra yang digunakan telah divalidasi secara visual untuk memastikan karakteristik penyakit dapat digunakan untuk proses berikutnya.

4.2.2. Preprocessing Data

Preprocessing data merupakan proses untuk menghindari data yang digunakan menjadi tidak ideal untuk digunakan pada model klasifikasi. Langkah pertama adalah memastikan persebaran jumlah data pada masing-masing kelas sudah seimbang atau bisa dibilang ideal. Setelah itu dilakukan proses augmentasi data guna menambah jumlah data sehingga menjadi lebih bervariasi. Tahap selanjutnya adalah pembagian data untuk digunakan pada model klasifikasi.

Pembagian data disini adalah melakukan pembagian data menjadi *data training* dan *validasi data*. *Data training* adalah yang digunakan untuk melatih model klasifikasi dan *validasi data* merupakan data yang digunakan untuk melihat apakah hasil klasifikasi model pelatihan sudah baik atau memiliki kendala terlalu pas / kurang pas. *Overfitting* terjadi ketika model pembelajaran mesin terlalu kompleks dalam pelatihan data, bukan hanya pola yang sebenarnya, model ini "menghafal" data pelatihan dari pada "mengerti" pola di dalamnya. Sedangkan *underfitting* terjadi ketika model pembelajaran mesin terlalu sederhana untuk memahami kompleksitas data pelatihan, model ini tidak dapat menangkap pola yang ada di dalam data dengan baik. Pembagiannya adalah *data training* memiliki jumlah 80% data dan *data validation* sebanyak 20%.

4.3. Persiapan Percobaan CNN

Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dalam melakukan klasifikasi penyakit pada citra daun padi. CNN memiliki beberapa lapisan arsitektur yang dapat disesuaikan untuk melakukan klasifikasi. Kelebihan yang dimiliki oleh CNN adalah lapisan *convolutional*-nya yang dapat digunakan untuk ekstraksi ciri atau ekstraksi tekstur dari citra secara mendalam. Lapisan yang tidak kalah berguna adalah lapisan *pooling* yang digunakan untuk mengurangi dimensi yang dimiliki oleh citra dengan tetap mempertahankan informasi penting yang dimiliki citra. Pelaksanaan penelitian dilakukan sesuai dengan jenis penelitian yang bersifat eksperimental dimana penulis akan mencoba bereksperimen menggunakan algoritma dan metode yang telah ditetapkan sebelumnya dengan beberapa arsitektur yang berbeda. Arsitektur dari CNN yang akan digunakan pada penelitian kali ini adalah beberapa arsitektur lapisan yang telah ada sebelumnya yaitu VGG16, Xception, MobileNetV2 dan InceptionV3. VGG16 dipilih sebagai *baseline* arsitektur konvensional yang mengandalkan kedalaman *layer* melalui tumpukan konvolusi 3 X 3 untuk mengevaluasi performa model tradisional pada dataset ini. Sementara itu, InceptionV3 dan Xception dipilih karena kemampuannya dalam menangani variasi ukuran objek (lesi penyakit) melalui mekanisme *multi-scale processing* dan *Depthwise Separable Convolution*, yang secara teori sangat efektif dalam menangkap detail spasial pada

bercak daun yang beragam. Di sisi lain, MobileNetV2 disertakan sebagai representasi arsitektur *lightweight* yang dirancang untuk efisiensi komputasi tinggi.

VGG16 sendiri dipilih karena cukup populer dan dari penelitian Priyanka & Kumara tahun 2021 dengan judul *Classification of Rice Plant Diseases Using the Convolutional Neural Network Method* mendapatkan tes akurasi lumayan tinggi sebesar 92,24%. Selanjutnya Xception dipilih karena Tejaswini et al (2022) melalui penelitiannya yang berjudul *Rice Leaf Disease Classification Using CNN* juga menggunakan *Xception* yang mendapatkan hasil cukup baik sebesar 72,2%. Serta penelitian Shah et al (2023) dan Dutta et al (2024) menggunakan CNN dengan mengkomparasi arsitektur termasuk arsitektur MobileNetV2 dan InceptionV3. Selain itu dari web Keras dengan alamat <https://keras.io/api/applications> yang memiliki banyak pilihan arsitektur, keduanya mendapatkan akurasi yang cukup tinggi dan size yang cukup ringan.

Tabel 4.3. Skenario Percobaan Arsitektur CNN

No	Arsitektur	Percobaan	Dataset	Jumlah Data
1	VGG 16 (Arsitektur 1)	1	Data Original	904
2		2	Data Augmentasi	1752
3	Xception (Arsitektur 2)	1	Data Original	904
4		2	Data Augmentasi	1752
5	MobileNetV2 (Arsitektur 3)	1	Data Original	904
6		2	Data Augmentasi	1752
7	InceptionV3 (Arsitektur 4)	1	Data Original	904
8		2	Data Augmentasi	1752

Terdapat total 8 percobaan dari 4 arsitektur CNN yang telah dibuat, dimana masing-masing arsitektur dibagi menjadi 2 percobaan dengan olah dataset yang berbeda dari sumber yang sama, detailnya dapat kita lihat pada Tabel 4.3. Implementasi percobaan CNN dilakukan dengan bahasa *python* di *Jupyter notebook* menggunakan *library-library* yang dibutuhkan seperti *keras*, *tensorflow*, *numpy*, *pandas*, *sklearn*, *matplotlib*, dan lain-lain.

Untuk memastikan proses pelatihan berjalan secara terstruktur dan objektif, pengaturan konstanta serta penalaan hyperparameter dikelompokkan ke dalam sebuah tabel referensi tunggal. Pengaturan ini mencakup parameter proses latihan yang dibuat seragam sebagai variabel kontrol, serta konfigurasi arsitektur kepala (classification head) yang disesuaikan secara spesifik demi mengakomodasi karakteristik kompleksitas masing-masing model. Detail pengaturan parameter tersebut disajikan pada Tabel 4.4. berikut.

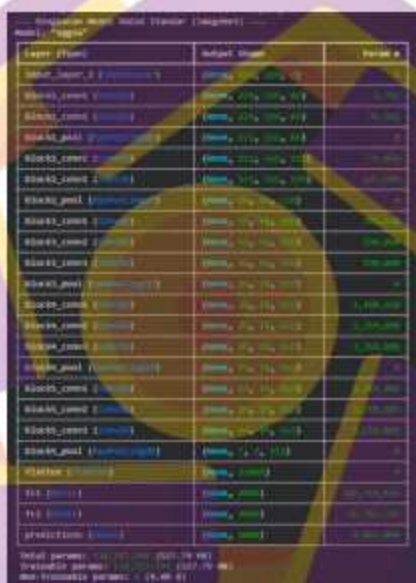
Tabel 4.4. Konfigurasi Hyperparameter Pelatihan Model

Nama Parameter	MobileNetV2	VGG16	Xception	InceptionV3
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.0001 ($1e-4$)	0.0001 ($1e-4$)	0.0001 ($1e-4$)	0.0001 ($1e-4$)
Loss Function	Categorical Cross-Entropy	Categorical Cross-Entropy	Categorical Cross-Entropy	Categorical Cross-Entropy
Input Shape	150 X 150	150 X 150	150 X 150	150 X 150
Epoch Pertahap	30	30	30	30
Pooling Layer	Global Average Pooling	Flatten	Global Average Pooling	Global Average Pooling
Neuron Dense Layer	1024	128	128	1024
Aktivasi Dense	ReLU	ReLU	ReLU	ReLU
Dropout Rate	0.25	0.50	0.50	0.50

4.3.1. VGG16

Pada percobaan CNN 1 ini penulis menggunakan arsitektur yang telah tersedia sebelumnya yaitu VGG16 (*Visual Geometry Group 16*) dengan beberapa olah dataset yang berbeda dari sumber yang sama yaitu dataset original dan dataset augmentasi. VGG16 adalah sebuah arsitektur *deep convolutional neural network* yang terdiri atas 16 layer dan merupakan salah satu arsitektur paling awal yang digunakan dalam *transfer learning*. VGG16 dikembangkan oleh Karen

Simonyan dan Andrew Zisserman dan dipublikasikan pada tahun 2014 dalam salah satu paper mereka (Simonyan & Zisserman, 2014). Model ini kemudian memenangkan kompetisi 2014 ILSVRC Challenge atas perolehan top-5 test accuracy mencapai 92,7% pada klasifikasi 1,4 juta gambar dalam 1000 kelas. Alasan lain penggunaan VGG16 karena salah satu arsitektur terbaik dengan telah memenangkan kompetisi ILSVR (*imagenet*) pada tahun 2014. Arsitektur VGG16 berisikan feature extraction yang berisikan convolutional layer dan pooling layer yang saling berhubungan. Setiap layer tersusun terdiri dari 16 blok layer sesuai dengan namanya.



Layer Name	Layer Type	Params
conv_1_conv_1	conv_1	128,000
conv_1_conv_2	conv_1	128,000
conv_1_conv_3	conv_1	128,000
conv_1_conv_4	conv_1	128,000
conv_1_conv_5	conv_1	128,000
conv_1_conv_6	conv_1	128,000
conv_1_conv_7	conv_1	128,000
conv_1_conv_8	conv_1	128,000
conv_1_conv_9	conv_1	128,000
conv_1_conv_10	conv_1	128,000
conv_1_conv_11	conv_1	128,000
conv_1_conv_12	conv_1	128,000
conv_1_conv_13	conv_1	128,000
conv_1_conv_14	conv_1	128,000
conv_1_conv_15	conv_1	128,000
conv_1_conv_16	conv_1	128,000
conv_2_conv_1	conv_2	128,000
conv_2_conv_2	conv_2	128,000
conv_2_conv_3	conv_2	128,000
conv_2_conv_4	conv_2	128,000
conv_2_conv_5	conv_2	128,000
conv_2_conv_6	conv_2	128,000
conv_2_conv_7	conv_2	128,000
conv_2_conv_8	conv_2	128,000
conv_2_conv_9	conv_2	128,000
conv_2_conv_10	conv_2	128,000
conv_2_conv_11	conv_2	128,000
conv_2_conv_12	conv_2	128,000
conv_2_conv_13	conv_2	128,000
conv_2_conv_14	conv_2	128,000
conv_2_conv_15	conv_2	128,000
conv_2_conv_16	conv_2	128,000
pool_1	pool_1	0
pool_2	pool_2	0
softmax	softmax	1000
Total Params		14,744,816

Gambar 4.3. Model Arsitektur VGG16

Susunan model dari arsitektur VGG16 dapat dilihat pada Gambar 4.3 di atas, dimana input citra sudah di-set otomatis menjadi 150x150 pixel, untuk ukuran filter berukuran 3x3 dengan *stride* 1 yang berada pada setiap layer yang ada pada *convolutional layer*. *Pooling layer* menggunakan *max pooling* dengan jumlah *filter* 2x2. Tetapi lapisan di atas belum bisa dijadikan arsitektur untuk proses selanjutnya karena hanya lapisan bawaan dan masih belum mempunyai lapisan sesuai dengan percobaan yang dibutuhkan, maka langkah selanjutnya yang

dilakukan setelah mendapatkan layer arsitektur VGG16 adalah *fine tuning*. *Fine tuning* adalah proses pengambilan bobot dari arsitektur yang telah dilatih sebelumnya menggunakan *imagenet*. Teknik ini juga dapat digunakan untuk menyesuaikan jumlah kelas pada *convolutional layer* terhadap objek yang akan dilatih, berikut potongan kode dari proses *fine tuning*.

```
# Load pretrained vgg16
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(150,150,3))

# freeze the base model layers
for layer in base_model.layers:
    layer.trainable = False

# Define the VGG16 model
VGG16_model = Sequential([
    base_model,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')
])

# Compile VGG16 Model
VGG16_model.compile(optimizer=Adam(learning_rate=0.0001),
                    loss='categorical_crossentropy',
                    metrics=['accuracy'])
```

Tahap ini peneliti akan menyisipkan *convolutional layer* dengan *activation relu*, *dropout* untuk mengurangi *overfitting*, fungsi *flatten* untuk membuat citra yang berbentuk array menjadi bentuk vektor, serta terakhir yaitu fungsi *dense* yang menambahkan layer pada *fully connected layer* dengan *activation softmax* untuk menghubungkan fitur-fitur yang telah diekstraksi oleh lapisan-lapisan sebelumnya agar menghasilkan output akhir seperti kelas prediksi, selanjutnya model tersebut di-*compile*. Hasil setelah *fine tuning* dapat dilihat pada Gambar 4.4.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 128)	14,714,400
flatten_1 (Flatten)	(None, 8192)	0
dense_1 (Dense)	(None, 128)	1,048,704
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 5)	640

Total params: 15,763,104 (60.13 MB)
 Trainable params: 1,049,120 (4.00 MB)
 Non-trainable params: 14,714,984 (56.13 MB)

Gambar 4.4. Arsitektur VGG16 Setelah *Fine Tuning*

4.3.2. Xception

Pada eksperimen selanjutnya, penulis menggunakan arsitektur pre-trained lainnya, yaitu Xception, dengan mengujinya pada dua kondisi data yang berbeda, yakni dataset original dan dataset augmentasi. Arsitektur Xception merupakan inovasi berbasis Convolutional Neural Networks (CNN) yang diperkenalkan oleh Google Research sebagai pengembangan dari arsitektur Inception melalui peningkatan yang signifikan. Xception berfokus pada pemisahan fitur spasial dan fitur lintas kanal secara ekstrem melalui implementasi Depthwise Separable Convolutions. Secara struktural, susunan jaringan Xception diawali oleh input layer dengan dimensi citra tertentu, diikuti oleh entry flow yang berfungsi sebagai gerbang awal jaringan melalui kombinasi convolutional layer, batch normalization untuk stabilitas pelatihan, serta fungsi aktivasi ReLU. Bagian inti dari arsitektur ini terdiri dari blok-blok separable convolution yang membagi proses konvolusi menjadi dua tahap, yaitu konvolusi terpisah pada setiap kanal (depthwise) dan konvolusi 1×1 (pointwise) untuk memadukan fitur lintas kanal, yang diperkuat dengan batch normalization, aktivasi, serta residual connections. Proses ekstraksi fitur ini kemudian diulang beberapa kali pada bagian middle flow. Struktur diakhiri oleh exit flow yang mereduksi dimensi fitur menggunakan global average pooling sebelum diteruskan ke fully connected layer dan softmax activation untuk menghasilkan probabilitas kelas penyakit daun padi. Dalam penelitian ini, strategi *fine-tuning* diterapkan secara spesifik dengan membekukan (*freezing*) seluruh lapisan bobot bawaan ImageNet pada tahap entry flow, middle flow, hingga sebagian exit flow (`layer.trainable = False`). Langkah pembekuan ini bertujuan untuk mempertahankan kemampuan Xception dalam mengenali karakteristik geometri dasar dan tekstur bercak daun universal. Proses fine-tuning kemudian dipusatkan secara penuh pada lapisan klasifikasi kustom di ujung exit flow. Pada bagian inilah dilakukan tuning parameter aktif melalui penempatan Dense layer baru, penentuan tingkat regulasi Dropout guna menekan risiko overfitting, serta pembaruan bobot (*weights updating*) menggunakan optimizer Adam dengan laju pembelajaran (*learning rate*) sebesar 0,0001. Skema fine-tuning ini terbukti efektif memangkas jumlah parameter terlatih, sehingga model

Xception dapat mencapai konvergensi fitur penyakit padi secara lebih cepat dan efisien. Susunan modelnya tercantum pada Gambar 4.5.

Layer (Type)	Output Shape	Param #	Connected to
Input Layer (Conv2D)	(None, 256, 256, 3)	0	-
Block1_conv1 (Conv2D)	(None, 128, 128, 3)	105	Input Layer[0][1-3]
Block1_conv1_bn (BatchNormalisation)	(None, 128, 128, 3)	0	Block1_conv1[1-3]
Block1_conv1_act (Activation)	(None, 128, 128, 3)	0	Block1_conv1_bn[1-3]
Block1_conv2 (Conv2D)	(None, 128, 128, 3)	105	Block1_conv1_act[1-3]
Block1_conv2_bn (BatchNormalisation)	(None, 128, 128, 3)	0	Block1_conv2[1-3]
Block1_conv2_act (Activation)	(None, 128, 128, 3)	0	Block1_conv2_bn[1-3]
Block1_maxpool (MaxPooling2D)	(None, 64, 64, 3)	0	Block1_conv2_act[1-3]
Block1_maxpool_bn (BatchNormalisation)	(None, 64, 64, 3)	0	Block1_maxpool[1-3]
Block1_maxpool_act (Activation)	(None, 64, 64, 3)	0	Block1_maxpool_bn[1-3]
Block12_maxpool (MaxPooling2D)	(None, 32, 32, 3)	0	Block1_maxpool_act[1-3]
Block12_maxpool_bn (BatchNormalisation)	(None, 32, 32, 3)	0	Block12_maxpool[1-3]
Block12_maxpool_act (Activation)	(None, 32, 32, 3)	0	Block12_maxpool_bn[1-3]
Block12_maxpool (MaxPooling2D)	(None, 16, 16, 3)	0	Block12_maxpool_act[1-3]
Block12_maxpool_bn (BatchNormalisation)	(None, 16, 16, 3)	0	Block12_maxpool[1-3]
Block12_maxpool_act (Activation)	(None, 16, 16, 3)	0	Block12_maxpool_bn[1-3]
conv2d_2 (Conv2D)	(None, 16, 16, 3)	105	act_5[1-3]
Block13_pool (MaxPooling2D)	(None, 8, 8, 3)	0	conv2d_2[1-3]
batch_normalization (BatchNormalisation)	(None, 8, 8, 3)	0	conv2d_2[1-3]
act_11 (Act)	(None, 8, 8, 3)	0	Block13_pool[1-3] batch_normalization[1-3]
Block14_maxpool (MaxPooling2D)	(None, 4, 4, 3)	0	act_11[1-3]
Block14_maxpool_bn (BatchNormalisation)	(None, 4, 4, 3)	0	Block14_maxpool[1-3]
Block14_maxpool_act (Activation)	(None, 4, 4, 3)	0	Block14_maxpool_bn[1-3]
Block14_maxpool (MaxPooling2D)	(None, 2, 2, 3)	0	Block14_maxpool_act[1-3]
Block14_maxpool_bn (BatchNormalisation)	(None, 2, 2, 3)	0	Block14_maxpool[1-3]
Block14_maxpool_act (Activation)	(None, 2, 2, 3)	0	Block14_maxpool_bn[1-3]
Block14_maxpool (MaxPooling2D)	(None, 1, 1, 3)	0	Block14_maxpool_act[1-3]
Block14_maxpool_bn (BatchNormalisation)	(None, 1, 1, 3)	0	Block14_maxpool[1-3]
Block14_maxpool_act (Activation)	(None, 1, 1, 3)	0	Block14_maxpool_bn[1-3]

Total params: 30,000,000 (79.36 MB)
 Trainable params: 6,000,000 (20.97 MB)
 Non-trainable params: 24,000,000 (58.39 MB)

Gambar 4.5. Model Arsitektur Xception

4.3.3. MobileNetV2

Pada eksperimen selanjutnya, penulis menggunakan arsitektur pre-trained lainnya, yaitu MobileNetV2, dengan mengujinya pada dua kondisi data yang berbeda

dari sumber yang sama, yakni dataset original dan dataset augmentasi. Secara struktural, arsitektur MobileNetV2 terdiri dari serangkaian komponen utama yang diawali oleh lapisan konvolusi awal (*Initial Convolution Layer*) berupa konvolusi standar dengan stride 2, dilanjutkan oleh batch normalization dan fungsi aktivasi ReLU. Bagian utama dari jaringan ini didominasi oleh serangkaian blok residual terbalik (*Inverted Residual Blocks*) sebanyak kurang lebih 17 blok, yang memiliki karakteristik faktor ekspansi (r), jumlah kanal luaran (c), dan nilai stride (s) yang bervariasi. Aliran data kemudian diteruskan menuju lapisan konvolusi akhir (*Final Convolution Layer*) berbentuk konvolusi pointwise untuk menghasilkan fitur berdimensi tinggi, sebelum akhirnya direduksi melalui mekanisme Global Average Pooling (GAP) yang mengambil rata-rata spasial dari setiap kanal fitur. Tahap akhir dari model dasar ini ditutup dengan *Classification Layer* berupa Fully Connected Layer (*Dense*) berkombinasi fungsi aktivasi Softmax untuk kebutuhan klasifikasi multi-kelas. Dalam penelitian ini, strategi *fine-tuning* diimplementasikan pada MobileNetV2 dengan membekukan seluruh parameter di dalam blok konvolusi awal hingga lapisan konvolusi akhir bawaan ImageNet (`layer.trainable = False`). Pembekuan ini bertujuan mengunci kemampuan ekstraksi fitur esensial dari MobileNetV2 agar tidak rusak oleh dataset baru yang relatif kecil. Selanjutnya, proses *fine-tuning* difokuskan secara penuh pada lapisan klasifikasi kustom setelah lapisan GAP. Pada bagian *classification head* baru inilah penyesuaian hyperparameter dilakukan, yang mencakup penempatan Dense layer sebesar 1024 neuron, penerapan regulasi Dropout 0.25 untuk memitigasi risiko overfitting, serta pembaruan bobot (*weights updating*) menggunakan optimizer Adam dengan laju pembelajaran learning rate sebesar 0.0001. Melalui pendekatan *fine-tuning* ini, model mampu memanfaatkan efisiensi arsitektur MobileNetV2 secara maksimal untuk menghasilkan konvergensi pelatihan yang cepat dan akurasi tinggi, sebagaimana detail susunan model dari arsitektur MobileNetV2 terdapat pada Gambar 4.6.

Model: "mobilenetv2_1.0b_224"

Layer (type)	Output Shape	Param #	Connected to
Input Layer [InputLayer]	(None, 224, 224, 3)	0	-
Conv1 (conv2d)	(None, 112, 112, 32)	964	Input_Layer[0][0]
bn_conv1 [BatchNormalisation]	(None, 112, 112, 32)	128	Conv1[0][0]
Conv2_relu (conv2d)	(None, 112, 112, 64)	0	bn_conv1[0][0]
expanded_conv_dept. [BatchNormalisation]	(None, 112, 112, 32)	960	Conv2_relu[0][0]
expanded_conv_dept. [BatchNormalisation]	(None, 112, 112, 32)	128	expanded_conv_dept.
expanded_conv_dept. [ReLU]	(None, 112, 112, 32)	0	expanded_conv_dept.
expanded_conv_dept. [conv2d]	(None, 112, 112, 32)	960	expanded_conv_dept.
expanded_conv_dept. [BatchNormalisation]	(None, 112, 112, 32)	128	expanded_conv_dept.
block_1_expand [conv2d]	(None, 112, 112, 32)	960	expanded_conv_dept.
block_1_expand_relu [BatchNormalisation]	(None, 112, 112, 32)	128	block_1_expand[0][0]
block_1_expand_relu [ReLU]	(None, 112, 112, 32)	0	block_1_expand_relu

Gambar 4.6. Model Arsitektur MobileNetV2

4.3.4. InceptionV3

Pada percobaan CNN 4 penulis menggunakan arsitektur yang telah tersedia lainnya yaitu InceptionV3 dengan beberapa *dataset* yang berbeda dari sumber yang sama yaitu *dataset* original dan *dataset augmentasi*. Inception V3 adalah arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh Google yang sangat terkenal karena efisiensinya dalam penggunaan parameter dan komputasi, sambil mencapai akurasi yang sangat tinggi. Inti dari arsitektur InceptionV3 adalah Modul Inception yang secara paralel menerapkan konvolusi berukuran berbeda pada input yang sama, dengan inovasi kunci berupa faktorisasi konvolusi asimetris dan mekanisme pengurangan ukuran grid yang efisien untuk meminimalkan biaya komputasi dan parameter, serta memanfaatkan klasifikasi pembantu pada lapisan tengah untuk mempercepat konvergensi dan mengatasi *vanishing gradient* pada jaringan yang sangat dalam. Strategi *fine-tuning* diimplementasikan pada MobileNetV2 dengan membekukan seluruh parameter di

dalam blok konvolusi awal hingga lapisan konvolusi akhir bawaan ImageNet (layer.trainable = False). Detail susunan model dari arsitektur InceptionV3 terdapat pada Gambar 4.7.]

```
base_model.summary()
```

Model: "inception_v3"

Layer (type)	Output Shape	Param #	Connected to
input_layer_3 (Conv2D)	(None, 224, 224, 3)	0	-
conv2d_34 (Conv2D)	(None, 128, 128, 32)	964	input_layer_3[1]
batch_normalization_34 (Batch Normalization)	(None, 128, 128, 32)	0	conv2d_34[0][2]
activation_34 (ReLU)	(None, 128, 128, 32)	0	batch_normalization_34[0]
conv2d_35 (Conv2D)	(None, 128, 128, 32)	964	activation_34[1]
batch_normalization_35 (Batch Normalization)	(None, 128, 128, 32)	0	conv2d_35[0][2]
activation_35 (ReLU)	(None, 128, 128, 32)	0	batch_normalization_35[0]
conv2d_36 (Conv2D)	(None, 128, 128, 32)	964	activation_35[1]
batch_normalization_36 (Batch Normalization)	(None, 128, 128, 32)	0	conv2d_36[0][2]
activation_36 (ReLU)	(None, 128, 128, 32)	0	batch_normalization_36[0]

Gambar 4.7. Model Arsitektur InceptionV3

4.4. Analisis Hasil Penelitian

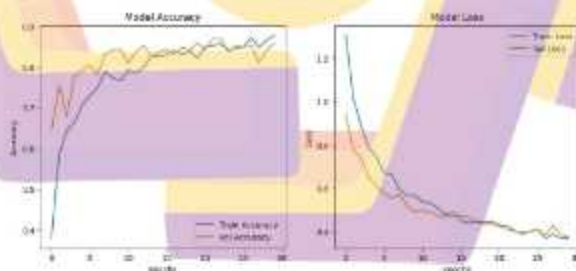
Proses klasifikasi pada daun tanaman padi dilakukan dengan menggunakan beberapa percobaan yang telah ditentukan pada penjelasan sebelumnya. Hasil dari percobaan yang telah dilakukan akan dianalisis dan dijelaskan lebih rinci. Metode pengujian yang digunakan pada penelitian ini adalah *Confusion Matrix*, dengan metode ini tidak hanya menunjukkan akurasi dari model klasifikasi saja akan tetapi memberikan nilai *precision*, *recall*, *f1-score*, dan *support* yang bisa digunakan untuk analisa lainnya. *Precision* mengukur seberapa akurat model dalam mengidentifikasi kelas positif. *Recall* juga disebut *sensitivity* mengukur sejauh mana model dapat mendeteksi semua instans positif yang seharusnya ditemukan. *F1-score* adalah perpaduan atau rata-rata antara *precision* dan *recall* dalam satu metrik tunggal. Serta *support* memberikan informasi tentang seberapa banyak data yang termasuk dalam kelas tertentu.

Training loss dan *validation loss*, serta *training accuracy* dan *validation*

accuracy adalah metrik-metrik yang digunakan untuk mengukur kinerja model pada saat pelatihan dan evaluasi pada *machine learning* atau *deep learning*. *Training loss* adalah metrik yang mengukur sejauh mana model telah belajar dari data pelatihan. Ini adalah ukuran seberapa baik model memprediksi data pelatihan. Tujuan utama selama pelatihan adalah mengurangi *training loss* sekecil mungkin. *Validation loss* adalah metrik yang digunakan untuk mengukur sejauh mana model dapat menggeneralisasi dari data pelatihan ke data yang belum pernah dilihat sebelumnya (data validasi). Ini adalah ukuran seberapa baik model dapat melakukan prediksi pada data baru. Contoh implementasi generalisasi fitur pada penelitian ini terletak pada model memahami bahwa fitur Brownspot adalah "bercak cokelat berbentuk oval dengan tepian kuning". Jadi, bagaimanapun posisi daun atau cahayanya, model tetap tahu itu adalah Brownspot. *Training accuracy* adalah metrik yang mengukur sejauh mana model benar-benar memprediksi kelas atau label yang benar pada data pelatihan. *Validation accuracy* adalah metrik yang digunakan untuk mengukur sejauh mana model benar-benar memprediksi kelas atau label yang benar pada data validasi, yang belum pernah dilihat selama pelatihan.

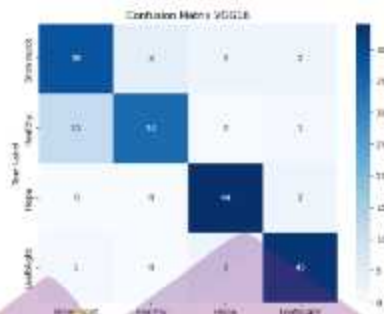
4.4.1 Hasil dan Pengujian Arsitektur CNN 1

1. Arsitektur VGG16 pada Dataset Original



Gambar 4.8. Grafik Training & Validation VGG16 (Dataset Original)

Pada Gambar 4.8 terlihat bahwa grafik *Model loss* arsitektur VGG16 pada dataset original dengan epoch terbaik terdapat pada urutan 30. Sedangkan grafik *Model accuracy* menunjukkan epoch terbaik pada urutan 30 dengan nilai *train_loss* 0,3606, *train_accuracy* 0,8829, *val_loss* 0,3758, dan *val_accuracy* 0,8587.



Gambar 4.9. Confusion Matrix VGG16 (Dataset Original)

Gambar 4.9. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur VGG16 pada dataset original. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.9 jumlah TP adalah $38 + 34 + 44 + 43 = 159$, kemudian jumlah data validation secara keseluruhan adalah 184 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(159 / 184) * 100 = 86,51 \%$ atau jika dibulatkan menjadi 87%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus *confusion matrix multi-class* dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 38 / (38 + 12) = 38 / 50 = 0,7600 * 100 = 76 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 34 / (34 + 4) = 34 / 38 = 0,8947 * 100 = 89 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 44 / (44 + 4) = 44 / 48 = 0,9167 * 100 = 92 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 43 / (43 + 5) = 43 / 48 = 0,8958 * 100 = 90 \%$$

Hasil *precision* secara keseluruhan (*Macro Average*) didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari arsitektur VGG16 dataset original adalah $(0,7600 + 0,8947 + 0,9167 + 0,8958) / 4 = 3,4672 / 4 = 0,8668 * 100 = 87 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada *confusion matrix multi-class* dalam menentukan nilai *recall* dihitung pada masing-masing kelas.

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 38 / (38 + 8) = 38 / 46 = 0,8261 * 100 = 83 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 34 / (34 + 12) = 34 / 46 = 0,7391 * 100 = 74 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 44 / (44 + 2) = 44 / 46 = 0,9565 * 100 = 96 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 43 / (43 + 3) = 43 / 46 = 0,9348 * 100 = 93 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari arsitektur VGG16 dataset original adalah $(0,7391 + 0,8261 + 0,9565 + 0,9348) / 4 = 3,4348 / 4 = 0,8648 * 100 = 86 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 1 VGG16 pada dataset original.

a. Nilai F1-Score Kelas Brownspot

F1-Score = $2 * (0,7600 * 0,8261) / (0,7600 + 0,8261) = 0,7917 * 100 = 79\%$

b. Nilai F1-Score Kelas Healthy

F1-Score = $2 * (0,8947 * 0,7391) / (0,8947 + 0,7391) = 0,8095 * 100 = 81\%$

c. Nilai F1-Score Kelas Hispa

F1-Score = $2 * (0,9167 * 0,9565) / (0,9167 + 0,9565) = 0,9362 * 100 = 94\%$

d. Nilai F1-Score Kelas Leafblight

F1-Score = $2 * (0,8958 * 0,9348) / (0,8958 + 0,9348) = 0,9149 * 100 = 91\%$

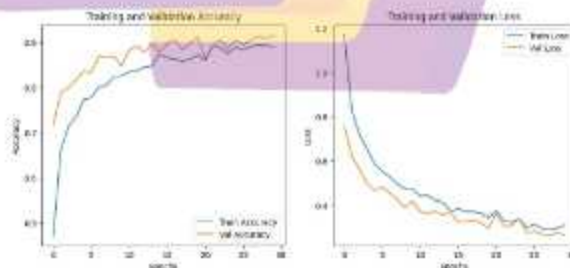
Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,7917 + 0,8095 + 0,9362 + 0,9149) / 4 = 3,4523 / 4 = 0,8631 * 100 = 86\%$.

Classification Report:				
	precision	recall	F1-score	support
Brownspot	0.76	0.83	0.79	40
Healthy	0.89	0.74	0.81	46
Hispa	0.92	0.96	0.94	46
Leafblight	0.90	0.93	0.91	46
accuracy			0.86	164
macro avg	0.87	0.86	0.86	164
weighted avg	0.87	0.86	0.86	164

Gambar 4.10. Clasification Report VGG16 (Dataset Original)

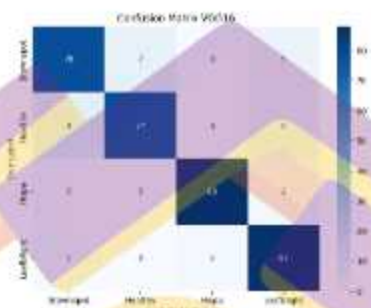
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.10.

2. Arsitektur VGG16 pada Dataset Augmentasi



Gambar 4.11. Grafik Training & Validation VGG16 (Dataset Augmentasi)

Pada Gambar 4.11 terlihat bahwa grafik *training and validation loss* model arsitektur VGG16 pada dataset augmentasi semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 27. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik dimana *epoch* terbaik pada urutan 27 dengan nilai *train_loss* 0,2784, *train_accuracy* 0,9021, *val_loss* 0,2535, dan *val_accuracy* 0,9167.



Gambar 4.12. Confusion Matrix VGG16 (Dataset Augmentasi)

Gambar 4.12. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur VGG16 pada dataset augmentasi. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.12 jumlah TP adalah $78 + 77 + 88 + 87 = 330$, kemudian jumlah data validation secara keseluruhan adalah 360 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(330 / 360) * 100 = 91.76 \%$ atau jika dibulatkan menjadi 92 %.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus confusion matrix multi-class dalam menentukan nilai

precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 78 / (78 + 9) = 78 / 87 = 0,8966 * 100 = 90 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 77 / (77 + 7) = 77 / 84 = 0,9167 * 100 = 92 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 88 / (88 + 2) = 88 / 90 = 0,9778 * 100 = 98 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 87 / (87 + 12) = 87 / 99 = 0,8788 * 100 = 88 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari arsitektur VGG16 dataset augmentasi adalah $(0,8966 + 0,9167 + 0,9778 + 0,8788) / 4 = 3,6699 / 4 = 0,9175 * 100 = 92 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada *confusion matrix multi-class* dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 78 / (78 + 12) = 78 / 90 = 0,8667 * 100 = 87 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 77 / (77 + 13) = 77 / 90 = 0,8556 * 100 = 86 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 88 / (88 + 2) = 88 / 90 = 0,9778 * 100 = 98 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 87 / (87 + 3) = 87 / 90 = 0,9667 * 100 = 97 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari VGG16 dataset augmentasi adalah $(0,8667 + 0,8556 + 0,9778 + 0,9667) / 4 = 3,668 / 4 = 0,917 * 100 = 92 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 1 pada dataset augmentasi.

a. Nilai F1-Score Kelas Brownspot

$$\text{F1-Score} = 2 * (0,8966 * 0,8667) / (0,8966 + 0,8667) = 0,8814 * 100 = 88\%$$

b. Nilai F1-Score Kelas Healthy

$$\text{F1-Score} = 2 * (0,9167 * 0,8556) / (0,9167 + 0,8556) = 0,8851 * 100 = 89\%$$

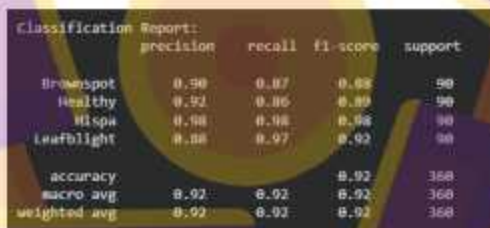
c. Nilai F1-Score Kelas Hispa

$$\text{F1-Score} = 2 * (0,9778 * 0,9778) / (0,9778 + 0,9778) = 0,9778 * 100 = 98\%$$

d. Nilai F1-Score Kelas Leafblight

$$\text{F1-Score} = 2 * (0,8788 * 0,9667) / (0,8788 + 0,9667) = 0,9206 * 100 = 92\%$$

Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8814 + 0,8851 + 0,9778 + 0,9206) / 4 = 3,6649 / 4 = 0,9162 * 100 = 92\%$.



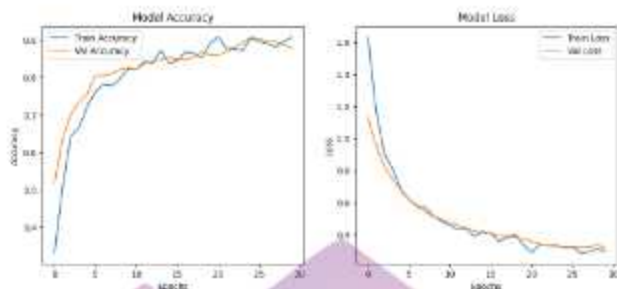
Classification Report:				
	precision	recall	f1-score	support
Brownspot	0.90	0.87	0.88	90
Healthy	0.92	0.86	0.89	90
Hispa	0.98	0.98	0.98	90
Leafblight	0.88	0.97	0.92	90
accuracy			0.92	360
macro avg	0.92	0.92	0.92	360
weighted avg	0.92	0.92	0.92	360

Gambar 4.13. Clasification Report VGG16 (Dataset Augmentasi)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.13.

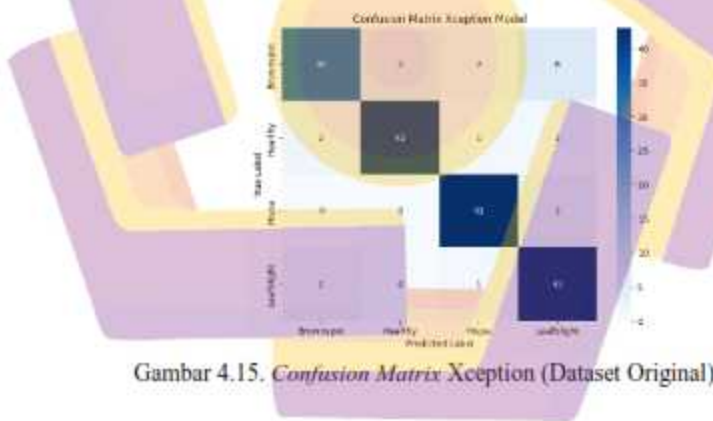
4.4.2 Hasil dan Pengujian Arsitektur CNN 2

1. Arsitektur Xception pada Dataset Original



Gambar 4.14. Grafik *Training & Validation Xception (Dataset Original)*

Pada Gambar 4.14. terlihat bahwa grafik *Model loss* arsitektur Xception pada dataset original dengan epoch terbaik terdapat pada urutan 27. Sedangkan grafik *Model accuracy* menunjukkan epoch terbaik pada urutan 21 dengan nilai *train_loss* 0,2674, *train_accuracy* 0,9030, *val_loss* 0,3180, dan *val_accuracy* 0,9061.



Gambar 4.15. *Confusion Matrix Xception (Dataset Original)*

Gambar 4.15. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur Xception pada dataset origina. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.15 jumlah TP adalah $34 + 42 + 43 + 43 = 162$, kemudian jumlah data validation secara keseluruhan adalah 184 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(158 / 184) * 100 = 88,04 \%$ atau jika dibulatkan menjadi 88%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus confusion matrix multi-class dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 34 / (34 + 4) = 34 / 38 = 0,8947 * 100 = 89 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 42 / (42 + 4) = 42 / 46 = 0,9130 * 100 = 91 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 43 / (43 + 4) = 43 / 47 = 0,9149 * 100 = 91 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 43 / (43 + 10) = 43 / 53 = 0,8113 * 100 = 81 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari Xception dataset original adalah $(0,8947 + 0,9130 + 0,9149 + 0,8113) / 4 = 3,5339 / 4 = 0,8835 * 100 = 88 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada confusion matrix multi-class dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 34 / (34 + 12) = 34 / 46 = 0,7391 * 100 = 73 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 42 / (42 + 4) = 42 / 46 = 0,9130 * 100 = 91 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 43 / (43 + 3) = 43 / 46 = 0,9348 * 100 = 93 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 43 / (43 + 3) = 43 / 46 = 0,9348 * 100 = 93 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari Xception dataset original adalah $(0,7391 + 0,9130 + 0,9348 + 0,9348) / 4 = 3,5217 / 4 = 0,8804 * 100 = 88 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 2 Xception pada dataset original.

a. Nilai F1-Score Kelas Brownspot

$$\text{F1-Score} = 2 * (0,8947 * 0,7391) / (0,8947 + 0,7391) = 0,8093 * 100 = 81\%$$

b. Nilai F1-Score Kelas Healthy

$$\text{F1-Score} = 2 * (0,9130 * 0,9130) / (0,9130 + 0,9130) = 0,9130 * 100 = 91\%$$

c. Nilai F1-Score Kelas Hispa

$$\text{F1-Score} = 2 * (0,9149 * 0,9348) / (0,9149 + 0,9348) = 0,9247 * 100 = 92\%$$

d. Nilai F1-Score Kelas Leafblight

$$\text{F1-Score} = 2 * (0,8113 * 0,9348) / (0,8113 + 0,9348) = 0,8685 * 100 = 87\%$$

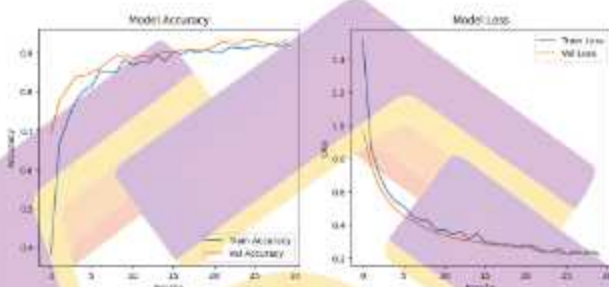
Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8093 + 0,9130 + 0,9247 + 0,8685) / 4 = 3,5155 / 4 = 0,8789 * 100 = 88\%$.

Classification Report:				
	precision	recall	f1-score	support
Brownspot	0.89	0.74	0.81	46
Healthy	0.91	0.91	0.91	46
Hispa	0.91	0.93	0.92	46
Leafblight	0.81	0.93	0.87	46
accuracy			0.88	184
macro avg	0.88	0.88	0.88	184
weighted avg	0.88	0.88	0.88	184

Gambar 4.16. Clasification Report Xception (Dataset Original)

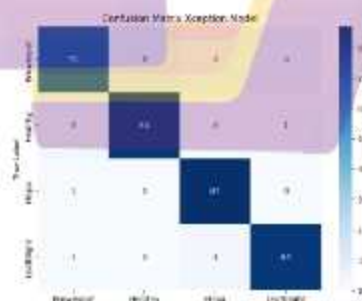
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.16.

2. Arsitektur Xception pada Dataset Augmentasi



Gambar 4.17. Grafik *Training & Validation* Xception (Dataset Augmentasi)

Pada Gambar 4.17, terlihat bahwa grafik *Model loss* arsitektur Xception pada dataset augmentasi dengan epoch terbaik terdapat pada urutan 28. Sedangkan grafik *Model accuracy* menunjukkan epoch terbaik pada urutan 26 dengan nilai *train_loss* 0,2137, *train_accuracy* 0,9314, *val_loss* 0,2205, dan *val_accuracy* 0,9347.



Gambar 4.18. *Confusion Matrix* Xception (Dataset Augmentasi)

Gambar 4.18. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur Xception pada dataset augmentasi. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.18 jumlah TP adalah $74 + 84 + 87 + 84 = 329$, kemudian jumlah data validation secara keseluruhan adalah 352 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(329 / 352) * 100 = 93,47\%$ atau jika dibulatkan menjadi 93%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus *confusion matrix multi-class* dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 74 / (74 + 5) = 74 / 79 = 0,9367 * 100 = 94 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 84 / (84 + 6) = 84 / 90 = 0,9333 * 100 = 93 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 87 / (87 + 7) = 87 / 94 = 0,9255 * 100 = 93 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 84 / (84 + 5) = 84 / 89 = 0,9438 * 100 = 94 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari Xception dataset augmentasi adalah $(0,9367 + 0,9333 + 0,9255 + 0,9438) / 4 = 3,7393 / 4 = 0,9348 * 100 = 93 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada *confusion matrix multi-class* dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 74 / (74 + 14) = 74 / 88 = 0,8409 * 100 = 84 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 84 / (84 + 4) = 84 / 88 = 0,9545 * 100 = 95 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 87 / (87 + 1) = 87 / 88 = 0,9886 * 100 = 99 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 84 / (84 + 4) = 84 / 88 = 0,9545 * 100 = 95 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari Xception dataset augmentasi adalah $(0,8409 + 0,9545 + 0,9886 + 0,9545) / 4 = 3,7385 / 4 = 0,9346 * 100 = 93 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 2 Xception pada dataset augmentasi.

a. Nilai F1-Score Kelas Brownspot

$$\text{F1-Score} = 2 * (0,9367 * 0,8409) / (0,9367 + 0,8409) = 0,8862 * 100 = 89\%$$

b. Nilai F1-Score Kelas Healthy

$$\text{F1-Score} = 2 * (0,9333 * 0,9545) / (0,9333 + 0,9545) = 0,9438 * 100 = 94\%$$

c. Nilai F1-Score Kelas Hispa

$$\text{F1-Score} = 2 * (0,9255 * 0,9886) / (0,9255 + 0,9886) = 0,9560 * 100 = 96\%$$

d. Nilai F1-Score Kelas Leafblight

$$\text{F1-Score} = 2 * (0,9438 * 0,9545) / (0,9438 + 0,9545) = 0,9491 * 100 = 95\%$$

Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8862 + 0,9438 + 0,9560 + 0,9491) / 4 = 3,7351 / 4 = 0,9338 * 100 = 93\%$.

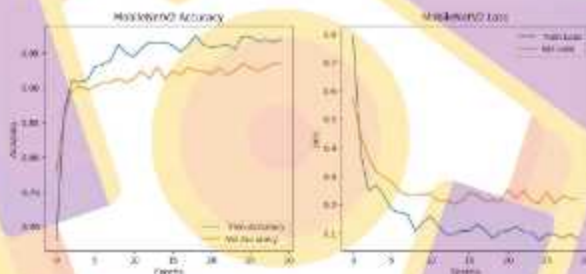
Classification Report:				
	precision	recall	F1-score	support
Brownspeck	0.94	0.84	0.89	88
Healthy	0.93	0.95	0.94	88
Itispa	0.93	0.99	0.96	88
leafblight	0.94	0.95	0.95	88
accuracy			0.93	352
macro avg	0.93	0.93	0.93	352
weighted avg	0.93	0.93	0.93	352

Gambar 4.19. Classification Report Xception (Dataset Augmentasi)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.19.

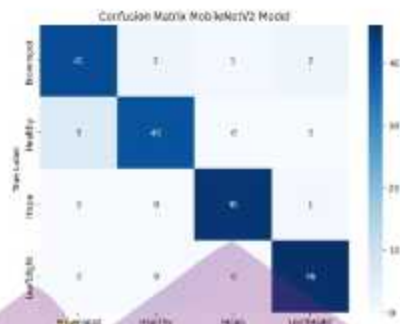
4.4.3 Hasil dan Pengujian Arsitektur CNN 3

1. Arsitektur MobileNetV2 pada Dataset Original



Gambar 4.20. Grafik *Training & Validation* MobileNetV2 (Dataset Original)

Pada Gambar 4.20. terlihat bahwa grafik *MobileNetV2 loss* arsitektur MobileNetV2 pada dataset original dengan epoch terbaik terdapat pada urutan 28. Sedangkan grafik *MobileNetV2 accuracy* menunjukkan epoch terbaik pada urutan 29 dengan nilai *train_loss* 0,0577, *train_accuracy* 0,9760, *val_loss* 0,2036, dan *val_accuracy* 0,9348.



Gambar 4.21. *Confusion Matrix MobileNetV2 (Dataset Original)*

Gambar 4.21. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur MobileNetV2 pada dataset original. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.21 jumlah TP adalah $41 + 40 + 45 + 46 = 172$, kemudian jumlah data validation secara keseluruhan adalah 184 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(172 / 184) * 100 = 93,87\%$ atau jika dibulatkan menjadi 94%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus *confusion matrix multi-class* dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 41 / (41 + 6) = 41 / 47 = 0.8723 * 100 = 87\%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 40 / (40 + 2) = 40 / 42 = 0.9524 * 100 = 95\%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 45 / (45 + 1) = 45 / 46 = 0.9783 * 100 = 98 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 46 / (46 + 3) = 46 / 49 = 0.9388 * 100 = 94 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari MobileNetV2 dataset original adalah $(0.8723 + 0.9524 + 0.9783 + 0.9388) / 4 = 3.7418 / 4 = 0.9354 * 100 = 94 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada *confusion matrix multi-class* dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 41 / (41 + 5) = 41 / 46 = 0.8913 * 100 = 89 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 40 / (40 + 6) = 40 / 46 = 0.8696 * 100 = 87 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 45 / (45 + 1) = 45 / 46 = 0.9783 * 100 = 98 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 46 / (46 + 0) = 46 / 46 = 1.0 * 100 = 100 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari MobileNetV2 dataset original adalah $(0.8913 + 0.8696 + 0.9783 + 1.0) / 4 = 3.7392 / 4 = 0.9348 * 100 = 93 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 3 MobileNetV2 pada dataset original.

a. Nilai F1-Score Kelas Brownspot

F1-Score = $2 * (0.8723 * 0.8193) / (0.8723 + 0.8193) = 0,8817 * 100 = 88\%$

b. Nilai F1-Score Kelas Healthy

F1-Score = $2 * (0.9524 * 0.8696) / (0.9524 + 0.8696) = 0,9091 * 100 = 91\%$

c. Nilai F1-Score Kelas Hispa

F1-Score = $2 * (0.9783 * 0.9783) / (0.9783 + 0.9783) = 0,9783 * 100 = 98\%$

d. Nilai F1-Score Kelas Leafblight

F1-Score = $2 * (0.9388 * 1.000) / (0.9388 + 1.000) = 0,9684 * 100 = 97\%$

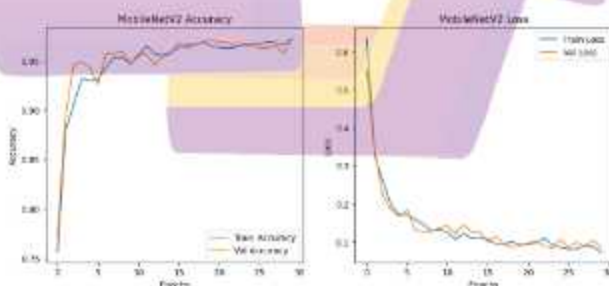
Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8817 + 0,9091 + 0,9783 + 0,9684) / 4 = 3,7375 / 4 = 0,9344 * 100 = 93\%$.

Classification Report:				
	precision	recall	f1-score	support
Brownspot	0.87	0.89	0.88	46
Healthy	0.95	0.87	0.91	46
Hispa	0.98	0.98	0.98	46
Leafblight	0.94	1.00	0.97	46
accuracy			0.93	184
macro avg	0.94	0.93	0.93	184
weighted avg	0.94	0.93	0.93	184

Gambar 4.22. Clasification Report MobileNetV2 (Dataset Original)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.23.

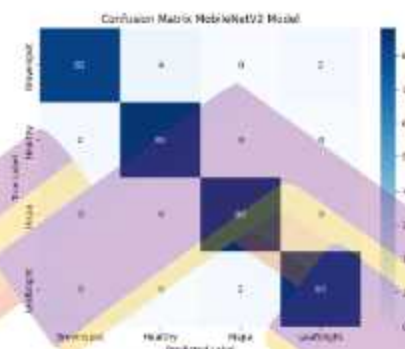
2. Arsitektur MobileNetV2 pada Dataset Augmentasi



Gambar 4.23. Grafik *Training & Validation* MobileNetV2 (Dataset Augmentasi)

Pada Gambar 4.23. terlihat bahwa grafik *MobileNetV2 loss* arsitektur

MobileNetV2 pada dataset augmentasi dengan epoch terbaik terdapat pada urutan 26. Sedangkan grafik *MobileNetV2 accuracy* menunjukkan epoch terbaik pada urutan 27 dengan nilai *train_loss* 0,0754, *train_accuracy* 0,9704, *val_loss* 0,0844, dan *val_accuracy* 0,9716.



Gambar 4.24. *Confusion Matrix* MobileNetV2 (Dataset Augmentasi)

Gambar 4.24. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur MobileNetV2 pada dataset augmentasi. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.24 jumlah TP adalah $82 + 86 + 88 + 86 = 342$, kemudian jumlah data validation secara keseluruhan adalah 352 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(342 / 352) * 100 = 97,15 \%$ atau jika dibulatkan menjadi 97%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus *confusion matrix multi-class* dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 82 / (82 + 2) = 82 / 84 = 0,9762 * 100 = 98 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 86 / (86 + 4) = 86 / 90 = 0,9556 * 100 = 96 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 88 / (88 + 2) = 88 / 90 = 0,9778 * 100 = 98 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 86 / (86 + 2) = 86 / 88 = 0,9773 * 100 = 98 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari MobileNetV2 dataset augmentasi adalah $(0,9762 + 0,9556 + 0,9778 + 0,9773) / 4 = 3,8869 / 4 = 0,9717 * 100 = 97 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada *confusion matrix multi-class* dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 82 / (82 + 6) = 82 / 88 = 0,9318 * 100 = 93 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 86 / (86 + 2) = 86 / 88 = 0,9773 * 100 = 97 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 88 / (88 + 0) = 88 / 88 = 1,0 * 100 = 100 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 86 / (86 + 2) = 86 / 88 = 0,9773 * 100 = 98 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari MobileNetV2 dataset augmentasi adalah $(0,9318 + 0,9773 + 1,0 + 0,9773) / 4 = 3,8864 / 4 = 0,9716 * 100 = 97 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 3 MobileNetV2 pada dataset augmentasi.

a. Nilai F1-Score Kelas Brownspot

$$F1\text{-Score} = 2 * (0,9762 * 0,9318) / (0,9762 + 0,9318) = 0,9535 * 100 = 95\%$$

b. Nilai F1-Score Kelas Healthy

$$F1\text{-Score} = 2 * (0,9556 * 0,9773) / (0,9556 + 0,9773) = 0,9663 * 100 = 97\%$$

c. Nilai F1-Score Kelas Hispa

$$F1\text{-Score} = 2 * (0,9778 * 1,00) / (0,9778 + 1,00) = 0,9888 * 100 = 99\%$$

d. Nilai F1-Score Kelas Leafblight

$$F1\text{-Score} = 2 * (0,9773 * 0,9773) / (0,9773 + 0,9773) = 0,9773 * 100 = 98\%$$

Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,9535 + 0,9663 + 0,9888 + 0,9773) / 4 = 3,8859 / 4 = 0,9715 * 100 = 97\%$.



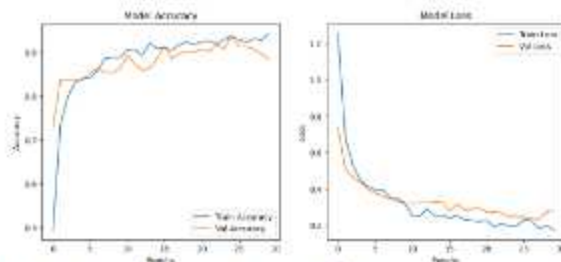
	precision	recall	f1-score	support
Brownspot	0.98	0.93	0.95	88
Healthy	0.96	0.98	0.97	88
Hispa	0.98	1.00	0.99	88
Leafblight	0.98	0.98	0.98	88
accuracy			0.97	352
macro avg	0.97	0.97	0.97	352
weighted avg	0.97	0.97	0.97	352

Gambar 4.25. Clasification Report MobileNetV2 (Dataset Augmentasi)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.25.

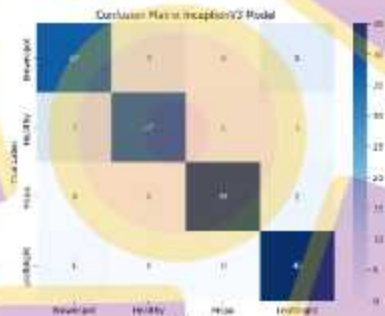
4.4.4 Hasil dan Pengujian Arsitektur CNN 4

1. Arsitektur InceptionV3 pada Dataset Original



Gambar 4.26. Grafik Training & Validation InceptionV3 (Dataset Original)

Pada Gambar 4.26, terlihat bahwa grafik *InceptionV3 loss* arsitektur *InceptionV3* pada dataset original dengan epoch terbaik terdapat pada urutan 30. Sedangkan grafik *InceptionV3 accuracy* menunjukkan epoch terbaik pada urutan 30 dengan nilai *train_loss* 0,1765, *train_accuracy* 0,9429, *val_loss* 0,2805, dan *val_accuracy* 0,8859.



Gambar 4.27. Confusion Matrix InceptionV3 (Dataset Original)

Gambar 4.27. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur *InceptionV3* pada dataset original. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.27 jumlah TP adalah $37 + 37 + 44 + 45 = 163$, kemudian jumlah data validation secara keseluruhan adalah 184

data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(163 / 184) * 100 = 88,58 \%$ atau jika dibulatkan menjadi 89%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus confusion matrix multi-class dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 37 / (37 + 8) = 37 / 45 = 0,822 * 100 = 82 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 37 / (37 + 4) = 37 / 41 = 0,902 * 100 = 90 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 44 / (44 + 1) = 44 / 45 = 0,977 * 100 = 97 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 45 / (45 + 5 + 1 + 2) = 45 / 53 = 0,849 * 100 = 84 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari InceptionV3 dataset original adalah $(0.822 + 0.902 + 0.977 + 0.849) / 4 = 3.55 / 4 = 0.8875 * 100 = 89 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada confusion matrix multi-class dalam menentukan nilai recall dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 37 / (37 + 9) = 37 / 46 = 0,804 * 100 = 80 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 37 / (37 + 9) = 37 / 46 = 0,804 * 100 = 80 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 44 / (44 + 2) = 44 / 46 = 0,956 * 100 = 96 \%$$

d. Nilai Recall Kelas Leafblight

$\text{Recall} = 45 / (45 + 1) = 45 / 46 = 0,978 * 100 = 98 \%$.

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari InceptionV3 dataset original adalah $(0.804 + 0.804 + 0.956 + 0.978) / 4 = 3,543 / 4 = 0,8858 * 100 = 89 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 4 InceptionV3 pada dataset original.

a. Nilai F1-Score Kelas Brownspot

$\text{F1-Score} = 2 * (0.8222 * 0.8043) / (0.8222 + 0.8043) = 0,8131 * 100 = 81\%$

b. Nilai F1-Score Kelas Healthy

$\text{F1-Score} = 2 * (0.9024 * 0.8043) / (0.9024 + 0.8043) = 0,8505 * 100 = 86\%$

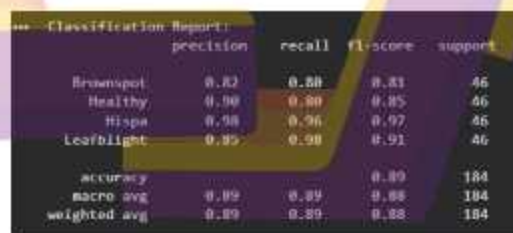
c. Nilai F1-Score Kelas Hispa

$\text{F1-Score} = 2 * (0.9777 * 0.9565) / (0.9777 + 0.9565) = 0,9669 * 100 = 97\%$

d. Nilai F1-Score Kelas Leafblight

$\text{F1-Score} = 2 * (0.8490 * 0.9782) / (0.8490 + 0.9782) = 0,9090 * 100 = 91\%$

Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8131 + 0,8505 + 0,9669 + 0,9090) / 4 = 3,537 / 4 = 0,8840 * 100 = 88\%$.

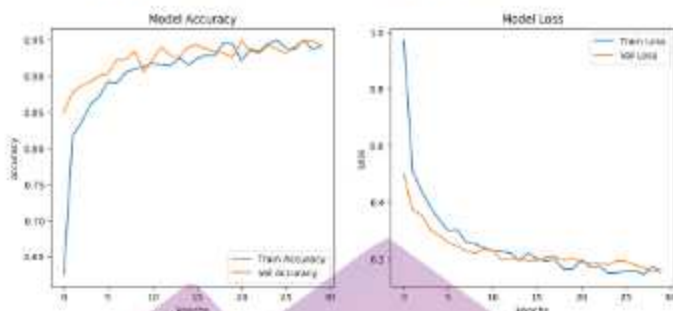


	precision	recall	f1-score	support
Brownspot	0.82	0.80	0.81	46
Healthy	0.90	0.80	0.85	46
Hispa	0.98	0.96	0.97	46
Leafblight	0.89	0.98	0.91	46
accuracy			0.89	184
macro avg	0.89	0.89	0.88	184
weighted avg	0.89	0.89	0.88	184

Gambar 4.28. Clasification Report InceptionV3 (Dataset Original)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.28.

2. Arsitektur InceptionV3 pada Dataset Augmentasi



Gambar 4.29. Grafik *Training & Validation* InceptionV3 (Dataset Augmentasi)

Pada Gambar 4.29, terlihat bahwa grafik *InceptionV3 loss* arsitektur *InceptionV3* pada dataset augmentasi dengan epoch terbaik terdapat pada urutan 24. Sedangkan grafik *InceptionV3 accuracy* menunjukkan epoch terbaik pada urutan 25 dengan nilai *train_loss* 0,1381, *train_accuracy* 0,9561, *val_loss* 0,1550, dan *val_accuracy* 0,9489.



Gambar 4.30. *Confusion Matrix* InceptionV3 (Dataset Augmentasi)

Gambar 4.30. adalah representasi hasil pengujian *confusion matrix* dari percobaan arsitektur *InceptionV3* pada dataset augmentasi. Klasifikasi penyakit daun padi memiliki 4 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* di mana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah True Positive dari masing-masing kelas. Pada Gambar 4.30 jumlah TP adalah $76 + 80 + 88 + 86 = 330$, kemudian jumlah data validation secara keseluruhan adalah 352 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(330 / 352) * 100 = 93,75 \%$ atau jika dibulatkan menjadi 94%.

2. Precision

Precision didapatkan dengan menghitung nilai TP dibagi dengan TP + FP, namun untuk kasus confusion matrix multi-class dalam menentukan nilai precision dihitung pada masing-masing kelas.

a. Nilai Precision Kelas Brownspot

$$\text{Precision} = 76 / (76 + 8) = 76 / 84 = 0,904 * 100 = 90 \%$$

b. Nilai Precision Kelas Healthy

$$\text{Precision} = 80 / (80 + 6) = 80 / 86 = 0,930 * 100 = 93 \%$$

c. Nilai Precision Kelas Hispa

$$\text{Precision} = 88 / (1 + 1 + 88 + 3) = 88 / 93 = 0,946 * 100 = 95 \%$$

d. Nilai Precision Kelas Leafblight

$$\text{Precision} = 86 / (86 + 3) = 86 / 89 = 0,966 * 100 = 97 \%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari InceptionV3 dataset augmentasi adalah $(0,904 + 0,930 + 0,946 + 0,966) / 4 = 3,747 / 4 = 0,9368 * 100 = 94 \%$.

3. Recall

Recall didapatkan dengan menghitung nilai TP dibagi dengan TP + FN, namun untuk kasus pada confusion matrix multi-class dalam menentukan nilai *recall* dihitung pada masing-masing kelas

a. Nilai Recall Kelas Brownspot

$$\text{Recall} = 76 / (6 + 1 + 3) = 76 / 86 = 0,8837 * 100 = 88 \%$$

b. Nilai Recall Kelas Healthy

$$\text{Recall} = 80 / (80 + 9) = 80 / 89 = 0,898 * 100 = 90 \%$$

c. Nilai Recall Kelas Hispa

$$\text{Recall} = 88 / (88 + 0) = 88 / 88 = 1,0 * 100 = 100 \%$$

d. Nilai Recall Kelas Leafblight

$$\text{Recall} = 86 / (86 + 3) = 86 / 89 = 0,9662 * 100 = 97 \%$$

Hasil *recall* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *recall* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *recall* dari InceptionV3 dataset augmentasi adalah $(0,8837 + 0,898 + 1,0 + 0,9662) / 4 = 3,748 / 4 = 0,9372 * 100 = 94 \%$.

4. F1-Score

F1 Score adalah gabungan dari dua metrik lain, yaitu Precision dan Recall. Berikut adalah perhitungan nilai F1-Score percobaan CNN 4 InceptionV3 pada dataset augmentasi.

a. Nilai F1-Score Kelas Brownspot

$$\text{F1-Score} = 2 * (0,9047 * 0,8837) / (0,9047 + 0,8837) = 0,8941 * 100 = 89\%$$

b. Nilai F1-Score Kelas Healthy

$$\text{F1-Score} = 2 * (0,9302 * 0,8988) / (0,9302 + 0,8988) = 0,9142 * 100 = 91\%$$

c. Nilai F1-Score Kelas Hispa

$$\text{F1-Score} = 2 * (0,9462 * 1,00) / (0,9462 + 1,00) = 0,9723 * 100 = 97\%$$

d. Nilai F1-Score Kelas Leafblight

$$\text{F1-Score} = 2 * (0,9662 * 0,9662) / (0,9662 + 0,9662) = 0,9662 * 100 = 97\%$$

Nilai keseluruhan performa model didapatkan dengan merata-rata F1 Score dari keempat kelas adalah $(0,8941 + 0,9142 + 0,9723 + 0,9662) / 4 = 3,7467 / 4 = 0,9367 * 100 = 94\%$.

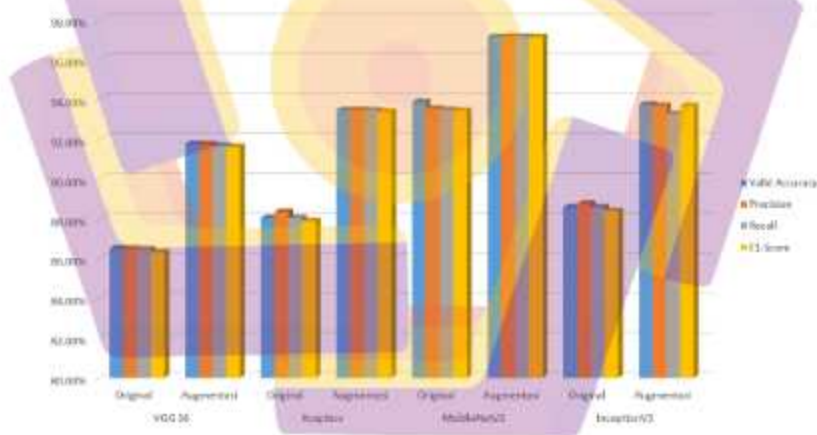
Classification Report:				
	precision	recall	f1-score	support
Brownspot	0.90	0.88	0.89	86
Healthy	0.93	0.90	0.91	89
Hispa	0.95	1.00	0.97	88
Leafblight	0.97	0.97	0.97	89
accuracy			0.94	352
macro avg	0.94	0.94	0.94	352
weighted avg	0.94	0.94	0.94	352

Gambar 4.31. Clasification Report InceptionV3 (Dataset Augmentasi)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program python seperti yang terangkum dalam Gambar 4.31.

4.4.5 Perbandingan Hasil Percobaan

Sebagai bahan perbandingan hasil percobaan pada masing-masing model arsitektur *Convolutional Neural Network (CNN)* dengan olah dataset berbeda yang telah dilakukan, Tabel 4.4 mencakup Validation Accuracy, Precision, Recall dan F1-Score. Guna memberikan gambaran performa model yang representatif pada dataset yang memiliki sebaran kelas berbeda, nilai Precision, Recall dan F1-Score dihitung menggunakan metode *Macro Average*. Guna memberikan representasi visual yang lebih jelas mengenai keunggulan setiap model, Gambar 4.32 menyajikan perbandingan akurasi antar arsitektur.



Gambar 4.32. Grafik Perbandingan Hasil Percobaan

Grafik diatas memperlihatkan bobot rata-rata yang proporsional terhadap jumlah sampel pada masing-masing kelas (Brownspot, Healthy, Hispa, dan Leafblight) berikut penulis sajikan Tabel 4.5. sebagai detail nilai yang didapatkan pada masing-masing arsitektur CNN.

Tabel 4.5. Perbandingan Hasil Percobaan

No	Arsitektur	Dataset	Valid Accuracy	Precision	Recall	F1-Score
1	VGG 16	Original	86,51%	86,51%	86,48%	86,31%
2		Augmentasi	91,76%	91,75%	91,67%	91,62%
3	Xception	Original	88,04%	88,35%	88,04%	87,89%
4		Augmentasi	93,47%	93,48%	93,46%	93,38%
5	MobileNetV2	Original	93,87%	93,54%	93,48%	93,44%
6		Augmentasi	97,15%	97,17%	97,16%	97,15%
7	InceptionV3	Original	88,58%	88,75%	88,58%	88,40%
8		Augmentasi	93,75%	93,68%	93,27%	93,67%

Dapat dilihat pada Tabel 4.4 bahwa hasil percobaan dengan memanfaatkan augmentasi data untuk memperbanyak dataset pada masing-masing arsitektur mendapatkan nilai akurasi yang lebih tinggi dibandingkan dengan penggunaan data original. Peningkatan tersebut terlihat pada arsitektur VGG16 yang meningkat dari 86,51% menjadi 91,76%, Xception dari 88,04% menjadi 93,47%, MobileNetV2 dari 93,87% menjadi 97,15%, dan InceptionV3 dari 88,58% meningkat menjadi 93,75%. Meskipun akurasi menunjukkan performa umum model, evaluasi mendalam dilakukan menggunakan metrik *Precision*, *Recall*, dan *F1-Score* untuk memastikan model tidak memiliki bias terhadap kelas tertentu. Berdasarkan hasil pengujian, MobileNetV2 dengan dataset augmentasi menjadi arsitektur terbaik dengan nilai *F1-Score* mencapai 97,15%. Adapun analisis lebih lanjut untuk arsitektur tersebut adalah sebagai berikut:

Tabel 4.6. Tabel Analisis Metrik Evaluasi MobileNetV2 Augmentasi

No	Kelas Penyakit	Precision	Recall	F1-Score
1	Brownspot	98%	93%	95%
2	Healthy	96%	98%	97%
3	Hispa	98%	100%	99%
4	Leafblight	98%	98%	98%
Rata-rata (Macro Avg)		97%	97%	97%

Berdasarkan Tabel 4.6, kelas Hispa mencatatkan performa sempurna (Recall 100%). Hal ini dianalisis terjadi karena karakteristik visual penyakit Hispa

yang berupa garis-garis putih memanjang memberikan sinyal fitur yang kuat bagi filter konvolusi MobileNetV2. Sebaliknya, pada kelas *Brownspot*, terjadi *misclassification* sebesar 7% ke kelas lain. Secara fungsional, ini menunjukkan bahwa pada kondisi citra dengan tingkat kepadatan bercak yang rendah, model terkadang mengidentifikasinya sebagai *Healthy* atau *Leafblight* karena adanya kemiripan warna (hue) dan tekstur lesi yang beririsan (overlap).

Hasil pengujian menunjukkan MobileNetV2 secara konsisten unggul dibandingkan arsitektur VGG16, InceptionV3 dan Xception. Hal ini disebabkan oleh kesesuaian antara kapasitas parameter model dengan skala volume dataset dalam penelitian ini. Arsitektur kompleks seperti InceptionV3 dan Xception memiliki struktur jaringan yang sangat dalam dengan jumlah parameter besar (masing-masing lebih dari 22 juta parameter). Karakteristik model berkapasitas besar ini bersifat *data-hungry*, yang membutuhkan variasi data skala masif agar seluruh filter konvolusinya dapat tergeneralisasi dengan optimal. Ketika melatih dataset penyakit daun padi dalam skala kecil, model raksasa tersebut cenderung mengalami performa yang kurang optimal akibat luasnya ruang pencarian bobot. Sebaliknya, MobileNetV2 berhasil unggul karena arsitekturnya dirancang hemat parameter melalui implementasi Depthwise Separable Convolutions (hanya sekitar 2,2 hingga 3,4 juta parameter). Struktur yang ringkas ini menurunkan varians model, sehingga MobileNetV2 jauh lebih tangguh terhadap risiko overfitting dan sangat adaptif dalam mengekstraksi fitur bercak penyakit daun padi secara tajam meskipun dihadapkan pada keterbatasan jumlah dataset.

Temuan dalam penelitian ini sejalan dengan studi yang dilakukan oleh Sandler et al. (2018) selaku pengembang MobileNetV2, yang menyatakan bahwa penggunaan Inverted Residuals dan Linear Bottlenecks berhasil mempertahankan representasi informasi esensial dalam ruang berdimensi rendah tanpa kehilangan fitur penting. Selain itu, dalam konteks klasifikasi penyakit tanaman dengan dataset terbatas, Rahman et al. (2020) serta Ahmed et al. (2021) juga membuktikan secara empiris bahwa arsitektur ringan seperti MobileNetV2 memiliki ketangguhan (robustness) yang jauh lebih stabil dan meminimalkan risiko overfitting dibandingkan arsitektur kompleks seperti VGG16 atau ResNet ketika ukuran sampel data per kelas berada di bawah 1.000 citra.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian pada berbagai skenario arsitektur CNN yang telah dilakukan, dapat disimpulkan bahwa penggunaan dataset augmentasi secara konsisten menghasilkan performa yang lebih unggul dengan rata-rata peningkatan akurasi sebesar 4,78% di seluruh arsitektur. Peningkatan tertinggi ditemukan pada arsitektur Xception sebesar 5,43% dan VGG16 sebesar 5,25%, yang membuktikan bahwa penambahan variasi informasi melalui augmentasi mampu secara efektif menutupi kekurangan jumlah sampel pada dataset original. Arsitektur VGG16 mencapai akurasi tertinggi sebesar 91,76% pada kondisi data augmentasi. Arsitektur Xception memperoleh nilai akurasi tertinggi sebesar 93,47% setelah proses augmentasi. Arsitektur MobileNetV2 mencatatkan performa terbaik di antara semua model dengan nilai akurasi mencapai 97,15% pada dataset augmentasi. Terakhir, arsitektur InceptionV3 mendapatkan nilai akurasi tertinggi sebesar 93,75% juga pada skenario dataset setelah augmentasi.
2. Rancangan arsitektur terbaik pada penelitian ini untuk mendeteksi dan mengklasifikasi citra penyakit pada daun padi adalah MobileNetV2, karena model ini terbukti mengungguli arsitektur lainnya pada seluruh skenario pengujian. Hasil terbaik diperoleh pada skenario ke-6 menggunakan arsitektur MobileNetV2 dengan dataset augmentasi, yang menghasilkan nilai akurasi training tertinggi sebesar 97,15%. Model ini juga terbukti memiliki performa yang sangat stabil dan tidak memiliki bias terhadap kelas tertentu. Hal ini dibuktikan dengan nilai Precision (97,17%), Recall (97,16%), dan F1-Score (97,15%) yang konsisten di angka 97%. Pencapaian metrik tersebut menunjukkan kemampuan model yang sangat baik dalam meminimalkan kesalahan diagnosis pada setiap jenis penyakit daun padi secara presisi.

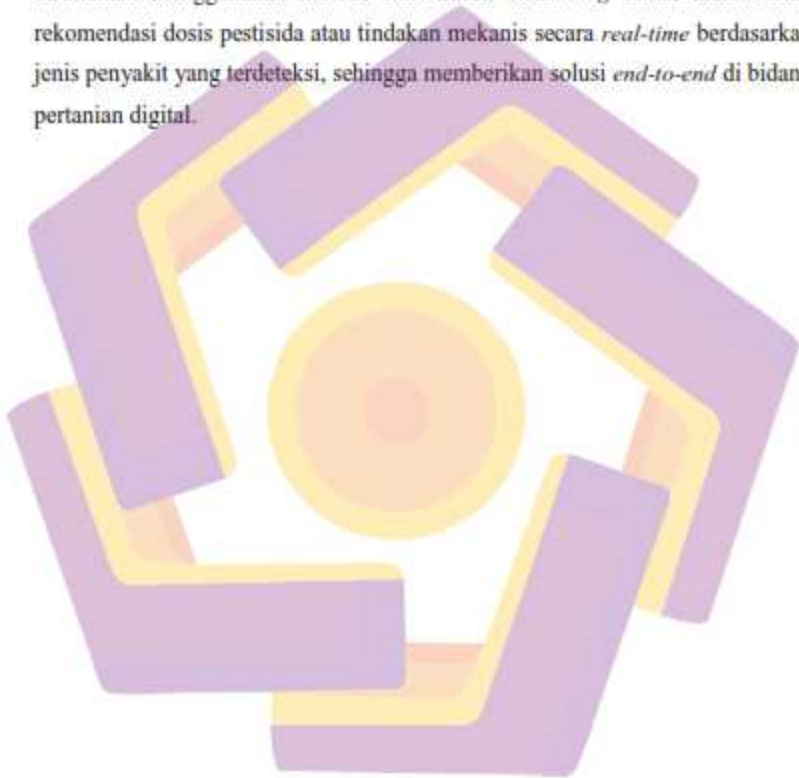
3. MobileNetV2 ditetapkan sebagai arsitektur terbaik dalam penelitian ini dengan capaian akurasi sebesar 97,15%. Keunggulan ini secara ilmiah didukung oleh penggunaan struktur *Depthwise Separable Convolutions* yang memungkinkan model bekerja secara lebih efisien. Dibandingkan dengan arsitektur yang lebih kompleks seperti VGG16 atau InceptionV3, MobileNetV2 memiliki jumlah parameter yang jauh lebih sedikit, sehingga lebih optimal dalam memproses dataset yang terbatas. Meskipun telah dilakukan augmentasi data, model dengan parameter besar cenderung memiliki risiko *overfitting* yang lebih tinggi, sementara MobileNetV2 terbukti lebih efektif dalam melakukan generalisasi fitur dengan peningkatan performa yang signifikan setelah augmentasi yang seperti yang tersaji pada Tabel 4.4 dan meminimalkan beban komputasi tanpa mengurangi performa akurasi. Hal ini memungkinkan integrasi model *deep learning* ke dalam perangkat dengan spesifikasi perangkat keras terbatas.

5.2. Saran

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan penelitian selanjutnya, antara lain :

1. Jumlah dan varian dataset sebaiknya diperkaya atau diperbanyak lagi agar hasil yang didapatkan lebih optimal, sekaligus semakin meyakinkan bahwa model yang dibangun terbukti baik dengan variasi dataset lain.
2. Saran ini diajukan karena berdasarkan hasil pengujian pada Tabel 4.5, model MobileNetV2 masih memiliki nilai *Recall* terendah pada kelas Brownsport (93%) dan *Precision* terendah pada kelas Healthy (96%). Hal ini mengindikasikan adanya kesulitan model dalam membedakan fitur visual yang halus antara bercak kecil Brownsport dengan variasi alami pada daun sehat. Oleh karena itu Eksplorasi Arsitektur *Lightweight* Berbasis *Attention Mechanism*: disarankan mencoba arsitektur yang lebih modern seperti MobileViT atau EfficientNet-B0 untuk membantu model fokus pada bercak kecil penyakit yang seringkali terabaikan pada citra daun dengan latar belakang yang kompleks.
3. Mencoba menggunakan teknik *preprocessing* data dan augmentasi yang lain guna meningkatkan akurasi pada arsitektur-arsitektur yang belum optimal.

4. Mengembangkan penentuan penanganan apa yang akan dilakukan ketika penyakit pada daun tanaman padi sudah diidentifikasi, penanganan yang dilakukan bisa berdasarkan jenis dan level penyakit pada setiap pohonnya.
5. Integrasi ke dalam Sistem Pendukung Keputusan (DSS) Berbasis Mobile: Hasil klasifikasi dari MobileNetV2 dapat diintegrasikan ke dalam arsitektur aplikasi Android yang terhubung dengan Sistem Pendukung Keputusan (DSS). Sistem ini disarankan menggunakan metode *Rule-Based Reasoning* untuk memberikan rekomendasi dosis pestisida atau tindakan mekanis secara *real-time* berdasarkan jenis penyakit yang terdeteksi, sehingga memberikan solusi *end-to-end* di bidang pertanian digital.



DAFTAR PUSTAKA

PUSTAKA BUKU

Huawei Technologies Co., Ltd.(2021). *Artificial Intelligence Technology*. Hangzhou, China: Posts & Telecom Press 2023). <https://doi.org/10.1007/978-981-19-2879-6>

Jones, M. T. (2009). *Artificial intelligence: A systems approach*. Jones and Bartlett Publishers.

PUSTAKA WEBSITE

Badan Pusat Statistik. (2024). Pada 2024, luas panen padi diperkirakan sebesar <https://www.bps.go.id/id/pressrelease/2024/10/15/2376/luas-panen-padi-tahun-2024-diperkirakan-sebesar-10-05-juta-hektare-dengan-produksi-padi-sekitar-52-66-juta-ton-gabah-kering-giling--gkg--html>.

Koech, K. E. (2022, Juli 16). *Cross-Entropy Loss Function*. Medium. <https://towardsdatascience.com/cross-entropy-loss-function-f38e4ec8643e>.

Lasky, A. (2022, Mei 25). Deep Learning Project: Multilayer Perceptron. *Medium*. <https://medium.com/@artjovianprojects/deep-learning-project-multilayer-perceptron-e34017941918>.

Novindasari, I. (2021, Februari 11). Mengapa Diperlukan Regularisasi pada Model Neural Network? *Medium*. <https://idanovinda.medium.com/mengapa-diperlukan-regularisasi-pada-model-neural-network-d622ed98f9a8>.

PUSTAKA JURNAL ILMIAH ATAU PROSIDING

Adi Fajaryanto Cobantoro, F. M. (2023) . Performance analysis of alexnet convolutional neural network (CNN) architecture with image objects of rice plant leaves. *JITK (jurnal ilmu pengetahuan dan teknologi komputer)*, 117-122.

Aditya Rajbongshi, T. K. (2022). Recognition of mango leaf disease using convolutional neural network models: a transfer learning approach. *Indonesian Journal of Electrical Engineering and Computer Science*, 1681-1688.

- Adib Bin Rashid. (2024). AI revolutionizing industries worldwide: A comprehensive overview of its diverse applications. *Hybrid Advances*, Volume 7, 100277, ISSN 2773-207X, <https://doi.org/10.1016/j.hybadv.2024.100277>.
- Ahmed, K., Shahidi, T. R., Alam, S. M. I., & Miah, S. (2021). Rice leaf disease detection using machine learning techniques. 2021 International Conference on Electronics, Information, and Communication (ICEIC), 1-5.
- Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* (2022). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8, 53. <https://doi.org/10.1186/s40537-021-00444-8>.
- Amanda Caecilia Milano, A. Y. (2024). Klasifikasi penyakit daun padi menggunakan model deep learning efficientnet-b6. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, 551-559.
- Anggraeni, S. R., Ranggianto, N. A., Ghozali, I., Fatichah, C., & Purwitasari, D. (2022). Deep Learning Approaches for Multi-Label Incidents Classification from Twitter Textual Information. *Journal of Information Systems Engineering and Business Intelligence*, 8, 31-41. <https://doi.org/10.20473/jisebi.8.1>.
- Aggarwal, S., Suchithra, M., Chandramouli, N., Sarada, M., Verma, A., Vetrithangam, D., Ambachew Adugna, B. (2022). Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improve Agro-Business. *Scientific Programming*, 2022, 1-13. <https://doi.org/10.1155/2022/1757888>.
- Arifa, Amalia Hana; SYAMSIR, Elvira; BUDJANTO, Slamet. Karakterisasi Fisikokimia Beras Hitam (*Oryza sativa* L.) dari Jawa Barat, Indonesia. *agriTECH*, [S.l.], v. 41, n. 1, p. 15-24, mar. 2021. ISSN 2527-3825. doi:<https://doi.org/10.22146/agritech.53307>.
- Beheshti, N., & Johnsson, L. Squeeze U-Net: A Memory and Energy Efficient Image Segmentation Network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1495–1504. <https://doi.org/10.1109/CVPRW50498.2020.00190>.
- Belmir, Meroua; DIFALLAH, Wafia; GHAZLI, Abdelkader, (2025). Efficient deep learning

approach for enhancing plant leaf disease classification. *Indonesian Journal of Electrical Engineering and Computer Science*, [S.l.], v. 37, n. 2, p. 1112-1120, feb. ISSN 2502-4760. <http://doi.org/10.11591/ijeecs.v37.i2.pp1112-1120>.

- Budi Setiyono, M. R. (2023). Identifikasi tanamanobat indonesia melalui citra daun menggunakan metode convolutional neuralnetwork(CNN). *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 385-392.
- Dutta, M., Islam Sujana, M. R., Mojumdar, M. U., Chakraborty, N. R., Marouf, A. A., Rokne, J. G., & Alhaji, R. (2024). Rice Leaf Disease Classification—A Comparative Approach Using Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2 Architectures. *Technologies*, 12(11), 214. <https://doi.org/10.3390/technologies12110214>
- Dwiretno Istiyadi Swasono, M. A. (2023). Klasifikasi Penyakit pada Citra Buah Jeruk Menggunakan Convolutional Neural Networks (CNN) dengan Arsitektur AlexNet. *Informatics Journal*, 68-75.
- Enkvetchakul, P., & Surinta, O. (2021). Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition. *Applied Science and* <https://doi.org/10.14416/j.asep.2021.01.003>
- Haris, N. (2020). Kombinasi Ciri Bentuk dan Ciri Tekstur Untuk Identifikasi Penyakit Pada Tanaman Padi. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 237-250. <https://doi.org/10.35957/jatisi.v7i2.239>.
- Inderpreet Singh, Gulshan Goyal, Anmol Chandel. (2022). AlexNet architecture based convolutional neural network for toxic comments classification, *Journal of King Saud University Computer and Information Sciences*, Volume 34, Issue 9, Pages 7547-7558, ISSN 1319 1578, <https://doi.org/10.1016/j.jksuci.2022.06.007>.
- Indra, D., Fadlillah, H. M., Kasman, K., Ilmawan, L. B., & Lahuddin, H. (2022). Classification of good and damaged rice using convolutional neural network. *Bulletin of Electrical Engineering and Informatics*, 11(2), Art. 2. <https://doi.org/10.11591/eei.v11i2.3385>.

- Islam, M. A., Rahman Shuvo, N., Shamsojjaman, M., Hasan, S., Hossain, S., & Khatun, T. (2021). An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection. *International Journal of Advanced Computer Science and Applications*, 12(1), 280–288. <https://doi.org/10.14569/IJACSA.2021.0120134>
- J. Erhani, P. -É. Portier, E. Egyed-Zsigmond and D. Nurbakova, (2024). "Confusion Matrices: A Unified Theory," in *IEEE Access*, vol. 12, pp. 181372-181419, doi: 10.1109/ACCESS.2024.3507199
- Januarti, I., Junaidi, Y., & Mulyana, E. (2021). ANALISIS USAHATANI PADI GOGO DI LAHAN RAWA LEBAK (Studi Kasus: Desa Talang Dukun, Kecamatan Sungai Pinang, Propinsi Sumatera Selatan). *AGRISAINTEFIKA: Jurnal Ilmu-Ilmu Pertanian*, 5 <https://doi.org/10.32585/ags.v5i1.1109>.
- Jatmika, S., & Saputra, D. E. (2022). Rice Plants Disease Identification Using Deep Learning with Convolutional Neural Network Method. *Sinkron*, 7(3), 2008– 2016. <https://doi.org/10.33395/sinkron.v7i3.11540>.
- Julianto, A., Sunyoto, A., & Wibowo, F. W. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 3(2), 98–105. <https://doi.org/10.46764/teknimedia.v3i2.77>.
- Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, 2(1), 37–45. <https://doi.org/10.20895/dinda.v2i1.341>.
- Kim, E., Lee, H., Kim, J., & Kim, S. (2020). Data Augmentation Method by Applying Color Perturbation of Inverse PSNR and Geometric Transformations for Object Recognition Based on Deep Learning. *Applied Sciences*, 10, 3755. <https://doi.org/10.3390/app10113755>
- Kim, J., Jin, M., Homma, Y., Sim, A., Kroeger, W., & Wu, K. (2022). *Extract Dynamic Information To Improve Time Series Modeling: A Case Study with Scientific Workflow*. <https://doi.org/10.48550/arXiv.2205.09703>.
- Kurniawan Effendi, A. M. (2020). Pengetahuan, Sikap, dan Tindakan Petani Upsus dalam

Mengendalikan Hama dan Penyakit Tanaman Padi, *Jurnal Ilmu Pertanian Indonesia (JIPI)*, 515-523.

L. Y. Win Lwin and A. N. Htwe, "Image Classification for Rice Leaf Disease Using AlexNet Model," *IEEE Conference on Computer Applications (ICCA)*, Yangon, Myanmar, In 2023, pp. 124-129, doi: 10.1109/ICCA51723.2023.10181847.

Mehmood, F., Ahmad, S., & Whangbo, T. K. (2023). An Efficient Optimization Technique for Training Deep Neural Networks. *Mathematics*, 11(6), Article 6. <https://doi.org/10.3390/math11061360>

Mikolajczyk, A., & Grochowski, M. (2018). *Data augmentation for improving deep learning in image classification problem*. 117–122. <https://doi.org/10.1109/IIPHDW.2018.8388338>

Mohidem, N. A., Hashim, N., Shamsudin, R., & Che Man, H. (2022). Rice for Food Security: Revisiting Its Production, Diversity, Rice Milling Process and Nutrient Content. *Agriculture*, 12, 741-755. <https://doi.org/10.3390/agriculture12060741>.

Muhammad Sayuthi, A. H. (2020). Distribusi Hama Tanaman Padi (*Oryza sativa* L.) pada Fase Vegetatif dan Generatif di Provinsi Aceh. *Jurnal Agroecotenia*, 1-10.

M. W. Ahdi, Khalid, A. Kunaefi, B. A. Nugroho and A. Yusuf, "Convolutional Neural Network (CNN) EfficientNet-B0 Model Architecture for Paddy Diseases Classification," *2023 14th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, 2023, pp. 105-110, doi: 10.1109/ICTS58770.2023.10330828.

Nasution, A. (2014). *Penyakit Blas Pyricularia grisea pada Tanaman Padi dan Strategi Pengendaliannya*. 9(2).

Nur Farahin Bt Abdul Halim, R. A. (2022). Enhancement of automatic classification of arcus senilis-nonarcus senilis using convolutional neural network. *Indonesian Journal of Electrical Engineering and Computer Science*, 209-222.

Nuryanto, B. (2019). Pengendalian Penyakit Tanaman Padi Berwawasan Lingkungan Melalui Pengelolaan Komponen Epidemik. *Jurnal Penelitian Dan Pengembangan Pertanian*, 37(1), Art. 1. <https://doi.org/10.21082/jp3.v37n1.2018.p1-8>.

- Priyanka, A. A. J. V., & Kumara, I. M. S. (2021). Classification Of Rice Plant Diseases Using the Convolutional Neural Network Method. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 12(2), Art. 2. <https://doi.org/10.24843/LKJITI.2021.v12.i02.p06>
- Rainio, O., Tamminen, J., Venäläinen, M.S. *et al.* (2024). Comparison of thresholds for a convolutional neural network classifying medical images. *Int J Data Science Analytics*. <https://doi.org/10.1007/s41060-024-00584-z>.
- Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Islam, M. S., & Islam, M. N. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*, 194, 112-120. (Jurnal ini sangat relevan karena membahas penyakit padi dengan CNN).
- Rendra Soekarta, N. N. (2023). Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN). *INSECT*, 143-151.
- Romero Aquino, M., Gutoski, M., Hattori, L., & Lopes, H. (2019, Oktober 30). *The Effect of Data Augmentation on the Performance of Convolutional Neural Networks*. <https://doi.org/10.21528/CBIC2017-51>
- Rozaqi, A. J., Sunyoto, A., & Arief, M. rudyanto. (2021). Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network. *Creative Information Technology Journal*, 8(1), 22. <https://doi.org/10.24076/citec.2021v8i1.263>
- R. Surya and E. Gautama, "Cassava Leaf Disease Detection Using Convolutional Neural Networks," 2020 6th International Conference on Science in Information Technology (ICSITech), Palu, Indonesia, 2020, pp. 97-102, doi: 10.1109/ICSITech49800.2020.9392051.
- Sarker, I. H. (2022). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2(6), 420. <https://doi.org/10.1007/s42979-021-00815-1>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 4510-4520.
- Setiawan, W., Ghofur, A., Hastarita Rachman, F., & Rulaningtyas, R. (2021). Deep

- Convolutional Neural Network AlexNet and Squeezenet for Maize Leaf Diseases Image Classification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 6(4). <https://doi.org/10.22219/kinetik.v6i4.1335>
- Shah, S. R., Qadri, S., Bibi, H., Shah, S. M. W., Sharif, M. I., & Marinello, F. (2023). Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. *Agronomy*, 13(6), 1633. <https://doi.org/10.3390/agronomy13061633>.
- Sharma, N., Sharma, R., & Jindal, N. (2021). Machine Learning and Deep Learning Applications- A Vision. *Global Transitions Proceedings*, 2(1), 24–28. <https://doi.org/10.1016/j.gltp.2021.01.004>.
- Simonyan, K., & Zisserman, A. (2014, September 4). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv:1409.1556S
- Suseno, J. R. K., Minarno, A. E., & Azhar, Y. (2023). Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation. *Kinetik: Game Technology, Information System, Computer Network, Electronics, Computing*, <https://doi.org/10.22219/kinetik.v8i1.1592>
- Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3), 52. <https://doi.org/10.3390/computation11030052>.
- Ulfa Nur Oktaviana (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *JURNAL RESTI Rekayasa Sistem dan Teknologi Informasi*, 1216-1222.
- Wan, G., & Yao, L. (2024). LMFRNet: A Lightweight Convolutional Neural Network Model for Image Analysis. *Electronics*, 13(1), 129. <https://doi.org/10.3390/electronics13010129>
- Walascha, A., Febriana, A., Saputri, D., Haryanti, D. S. N., Tsania, R., Sanjaya, Y., & Priyanti. (2021). Review Artikel: Inventarisasi Jenis Penyakit yang Menyerang Daun Tanaman Padi (*Oryza sativa* L.). *Prosiding Seminar Nasional* 1(2). Biologi, <https://doi.org/10.24036/proseminasbio/vol1/150.471-478>.
- Widneh, Misganaw Aguate, Alemu, Amlakie Aschale, Getie, Dereje Derib, (2025). Exploring

Batch Normalization's Impact on Dense Layers of Multiclass and Multilabel Classifiers, *International Journal of Intelligent Systems*, 2025, 1466655, 14 pages. <https://doi.org/10.1155/int/1466655>

Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

Zahirah, D., Purnawansyah, P., Kurniati, N., & Darwis, H. (2024). Digital Image Classification of Herbal Leaves Using KNN and CNN with GLCM features. *Jurnal Teknik Informatika (Jutif)*, 5(1), 61-67. <https://doi.org/10.52436/1.jutif.2024.5.1.1162>

