

**KLASIFIKASI PENYAKIT PADA TANAMAN PADI BERBASIS CITRA
DAUN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

SKRIPSI

**Oleh:
MOH. HERI SUSANTO
NIM. 200605110087**



**PROGRAM STUDI DARI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**KLASIFIKASI PENYAKIT PADA TANAMAN PADI BERBASIS CITRA
DAUN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

SKRIPSI

Diajukan kepada:

Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :

**Moh. Heri Susanto
NIM. 200605110087**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

HALAMAN PERSETUJUAN

KLASIFIKASI PENYAKIT PADA TANAMAN PADI BERBASIS CITRA DAUN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*


SKRIPSI

Oleh:

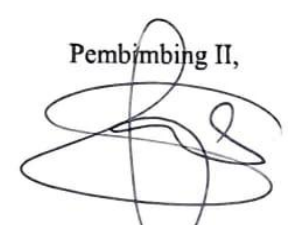
Moh. Heri Susanto
NIM. 200605110087

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 14 Februari 2025

Pembimbing I,


Dr. Irwan Budi Santoso, M. Kom
NIP. 19770103 201101 1 004

Pembimbing II,



Dr. M. Amin Hariyadi, M.T
NIP. 19670018 200501 1 001

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. H. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

KLASIFIKASI PENYAKIT PADA TANAMAN PADI BERBASIS CITRA DAUN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

SKRIPSI

Oleh :
Moh. Heri Susanto
NIM. 200605110087

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 07 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: <u>Prof. Dr. Suhartono, S.Si M.Kom</u> NIP. 19680519 200312 1 001
Anggota Penguji I	: <u>Ahmad Fahmi Karami, M.Kom</u> NIP. 19870909 202012 1 001
Anggota Penguji II	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004
Anggota Penguji III	: <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670018 200501 1 001

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




D.L. Fachrud Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Moh. Heri Susanto

NIM : 200605110087

Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Klasifikasi Penyakit Pada Tanaman Padi Berbasis
Citra Daun Menggunakan *Convolutional Neural Network*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 16 Mei 2025

Yang membuat pernyataan,



Moh. Heri Susanto
NIM.200605110087

HALAMAN MOTTO

"Langkah ini takkan sampai jika bukan karena kasih dan pertolongan-Nya."

" Dalam tiap lelah yang nyaris mematahkan, Dia hadir sebagai kekuatan."

" Saat harap terasa redup, doa menjadi cahaya yang menuntun kembali."

" Bukan kepintaran atau kemampuan, tapi ridha-Nya yang membuka jalan."

" Dan pada akhirnya, segala pencapaian ini hanyalah bentuk kecil dari besarnya cinta Allah kepada hamba-Nya yang terus berusaha."

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, saya mempersembahkan karya ini kepada orang tua tercinta yang selalu memberikan dukungan, kasih sayang, dan motivasi tanpa henti, serta menjadi sumber kekuatan yang luar biasa sepanjang perjalanan ini. Semoga Allah senantiasa melimpahkan rahmat dan kebahagiaan kepada mereka. Saya juga ingin mengucapkan terima kasih yang sebesar-besarnya kepada dosen dan pembimbing saya yang dengan sabar memberikan arahan, bimbingan, dan inspirasi, serta memperkaya pengetahuan saya dalam bidang ini. Semoga Allah SWT membalas segala kebaikan mereka dengan pahala yang berlipat. Tidak lupa, saya haturkan terima kasih kepada teman-teman dan rekan-rekan akademik yang telah berbagi semangat, ide, dan kerja keras dalam setiap langkah penelitian ini. Semoga Allah SWT memberkahi setiap usaha kita. Semoga karya ini dapat memberikan kontribusi yang bermanfaat bagi perkembangan ilmu pengetahuan dan teknologi, serta memberi inspirasi bagi penelitian-penelitian selanjutnya yang bermanfaat untuk kemajuan umat.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh,

Bismillahirrahmanirrahim, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, karena dengan limpahan rahmat, taufik, dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi ini yang berjudul "Klasifikasi Penyakit Pada Tanaman Padi Berbasis Citra Daun Menggunakan *Convolutional Neural Network*", sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Negeri Maulana Malik Ibrahim Malang.

Shalawat serta salam semoga selalu tercurah kepada Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman. Penulis menyadari sepenuhnya bahwa penyusunan skripsi ini tidak akan terwujud tanpa dukungan dari berbagai pihak.

Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU., selaku Ketua Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Dr. Irwan Budi Santoso, M. Kom, selaku Dosen Pembimbing I, yang telah memberikan bimbingan, arahan, dan masukan yang sangat berarti dalam setiap tahapan penelitian ini.
5. Dr. M. Amin Hariyadi, M.T., selaku Dosen Pembimbing II, yang telah memberikan motivasi, saran yang konstruktif, serta membantu penulis dalam memperdalam pemahaman terhadap topik penelitian.
6. Prof. Dr. Suhartono, M.Kom., selaku Ketua Penguji, yang telah memberikan evaluasi dan pertanyaan kritis yang sangat membangun dalam proses ujian skripsi.
7. Ahmad Fahmi Karami, M.Kom., selaku Anggota Penguji I, yang telah memberikan masukan dan pandangan yang memperkaya hasil akhir penelitian ini.
8. Seluruh Dosen dan Jajaran Staf Program Studi Teknik Informatika atas ilmu, bimbingan, dan semangat yang telah diberikan selama perkuliahan.
9. Seluruh staf administrasi dan laboran Teknik Informatika, atas bantuan dan pelayanan yang sangat membantu dalam kelancaran proses studi.
10. Kedua orang tua saya, yaitu Bapak Mohammad Sadik dan Ibu Isti Ika Rohaniya atas segala doa, cinta, dukungan, dan pengorbanan yang tidak pernah berhenti.
11. Teman-teman Angkatan 2020 Teknik Informatika “INTEGER”, yang telah menjadi bagian penting dalam perjalanan akademik ini, melalui kebersamaan, semangat, dan kerja sama.
12. Dan semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan dukungan, baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun sangat penulis harapkan demi penyempurnaan ke depannya. Semoga karya ini dapat memberikan manfaat dan menjadi kontribusi kecil dalam pengembangan ilmu pengetahuan, serta mendapat ridha dari Allah SWT.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Malang, 15 April 2025

Moh. Heri Susanto

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	7
2.1 Klasifikasi Penyakit Padi	7
2.2 Penyakit Daun Tanaman Padi	14
2.2.1 <i>Blight</i>	14
2.2.2 <i>Blast</i>	15
2.2.3 <i>Tungro</i>	16
2.3 <i>Convolutional Neural Network (CNN)</i>	17
BAB III DESAIN DAN IMPLEMENTASI	20
3.1 Pengumpulan Data	20
3.2 Input Citra	21
3.3 Pra-proses	22
3.4 Arsitektur CNN Usulan	24
3.4.1 Convolutional Layer	25
3.4.2 Rectified Linear Unit (ReLU)	28
3.4.3 Pooling Layer	28
3.4.4 Flatten Layer	30
3.4.5 Fully Connected Layer	30
3.4.6 Softmax	30
3.4.7 Negative Log-Likelihood Loss	31
3.4.8 Backpropagation dan Perhitungan Gradien	32
3.4.9 Evaluasi	34
3.4.10 Skema Uji Coba	37
BAB IV HASIL DAN PEMBAHASAN	40

4.1 Hasil Uji Coba	40
4.1.1 Hasil Uji Coba A1	40
4.1.2 Hasil Uji Coba A2	43
4.1.3 Hasil Uji Coba A3	45
4.1.4 Hasil Uji Coba A4	48
4.1.5 Hasil Uji Coba B1	51
4.1.6 Hasil Uji Coba B2	53
4.1.7 Hasil Uji Coba B3	56
4.1.8 Hasil Uji Coba B4	58
4.1.9 Hasil Uji Coba Keseluruhan	60
4.1.10 Hasil Penggunaan Parameter Terbaik pada Arsitektur Baru	61
4.2 Pembahasan	63
BAB V KESIMPULAN DAN SARAN	68
5.1 Kesimpulan	68
5.2 Saran	68
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Blight Pada Daun Padi	14
Gambar 2.2 Blast Pada Daun Padi	15
Gambar 2.3 Tungro Pada Daun Padi.....	16
Gambar 2.4 Arsitektur MLP Sederhana.....	18
Gambar 3.1 Skema Data Training dan data Testing	22
Gambar 3.2 Gambar Yang Telah Dilakukan Pra-proses.....	23
Gambar 3.3 Arsitektur CNN Usulan.....	24
Gambar 3.4 Arsitektur Baru 1: 3 Hidden Layer dan Jumlah Filter 8, 16, 32.....	38
Gambar 3.5 Arsitektur Baru 2: 4 Hidden Layer dan Jumlah Filter 8, 8, 16, 32....	38
Gambar 3.6 Arsitektur Baru 3: 4 Hidden Layer dan Jumlah Filter 8, 16, 16, 32..	38
Gambar 3.7 Arsitektur Baru 4: 4 Hidden Layer dan Jumlah Filter 8, 16, 32, 32..	38
Gambar 4.1 Grafik Akurasi dan Loss Uji Coba A1 Selama Pelatihan	40
Gambar 4.2 Confusion Matrix A1	40
Gambar 4.3 Matriks Evaluasi A1	41
Gambar 4.4 Grafik Akurasi dan Loss Uji Coba A2 Selama Pelatihan	42
Gambar 4.5 Confusion Matrix A2	43
Gambar 4.6 Matrik Evaluasi A2	44
Gambar 4.7 Grafik Akurasi dan Loss Uji Coba A3 Selama Pelatihan	45
Gambar 4. 8 Confusion Matrix A2	46
Gambar 4.9 Matrik Evaluasi A3	47
Gambar 4.10 Grafik Akurasi dan Loss Uji Coba A4 Selama Pelatihan	48
Gambar 4.11 Confusion Matrix A4	49
Gambar 4.12 Matrik Evaluasi A4	50
Gambar 4.13 Grafik Akurasi dan Loss Uji Coba B1 Selama Pelatihan.....	51
Gambar 4.14 Confusion Matrix B1.....	51
Gambar 4.15 Matrik Evaluasi	52
Gambar 4.16 Grafik Akurasi dan Loss Uji Coba B2 Selama Pelatihan.....	53
Gambar 4.17 Confusion Matrix B2.....	54
Gambar 4.18 Matrik Evaluasi B2	55
Gambar 4.19 Grafik Akurasi dan Loss Uji Coba B3 Selama Pelatihan.....	56
Gambar 4.20 Confusion Matrix B3.....	56
Gambar 4.21 Matrik Evaluasi B3	57
Gambar 4.22 Grafik Akurasi dan Loss Uji Coba B4 Selama Pelatihan.....	58
Gambar 4.23 Confusion Matrix B4.....	58
Gambar 4.24 Matrik Evaluasi B4	59

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	12
Tabel 3.1 Contoh Sampel Gambar Dari Dataset Yang Digunakan.....	20
Tabel 3.2 Confusion Matrix	35
Tabel 3.3 Skema Uji Coba	36
Tabel 4.1 Hasil Uji Coba Keseluruhan	59

ABSTRAK

Susanto, Moh. Heri. 2025. **Klasifikasi Penyakit Pada Tanaman Padi Berbasis Citra Daun Menggunakan Convolutional Neural Network**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. M. Amin Hariyadi, M.T.

Kata kunci: Penyakit Padi, Arsitektur CNN Khusus, Citra Daun, Optimasi Adam, Evaluasi Model.

Penyakit tanaman padi memiliki dampak yang besar terhadap produktivitas pertanian, sehingga dibutuhkan model klasifikasi untuk membedakan jenis penyakit dari citra daun padi secara akurat. Berbagai pendekatan klasifikasi berbasis citra daun telah dikembangkan sebelumnya, namun performanya masih dapat ditingkatkan. Oleh karena itu, penelitian ini mengusulkan penggunaan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan penyakit daun padi. Dataset yang digunakan dalam penelitian ini mencakup citra daun padi dalam empat kondisi yaitu *blast*, *blight*, *tungro*, dan *healthy*. Tahap awal melibatkan langkah-langkah praproses seperti *resize*, augmentasi, dan normalisasi. Setelah ini, arsitektur CNN khusus dirancang, yang terdiri dari 4 lapisan *convolutional*, 4 lapisan *pooling*, dan 3 lapisan *fully connected*. Setiap lapisan *convolutional* menggunakan kernel 3x3 dengan *stride* 1 dan aktivasi ReLU, sementara lapisan *pooling* menerapkan *max pooling* dengan kernel 3x3 dan *stride* 2. Dengan menggunakan ukuran batch 32 dan optimasi Adam, hasil terbaik dicapai pada 100 epoch pelatihan dengan *learning rate* 0,0002. Model tersebut mencapai akurasi pelatihan 0,9930 dan nilai kerugian 0,0221, sementara akurasi pengujian mencapai 0,9647. Metrik evaluasi menunjukkan kinerja yang seimbang di seluruh presisi, *recall*, dan *f1-score* masing-masing 0,9647 yang menunjukkan bahwa proses klasifikasi berjalan secara efektif tanpa bias terhadap kelas tertentu. Temuan ini menunjukkan bahwa model CNN yang dirancang dengan baik namun sederhana dapat memberikan hasil klasifikasi yang kompetitif tanpa bergantung pada arsitektur yang kompleks atau dengan teknik tambahan. Selain itu, model yang diusulkan mengungguli beberapa arsitektur CNN yang ada, termasuk Inception-ResNet-V2, VGG-16, VGG-19, dan Xception.

ABSTRACT

Susanto, Moh. Heri. 2025. **Leaf Image-Based Classification of Rice Plant Diseases Using Convolutional Neural Network**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisor: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. M. Amin Hariyadi, M.T.

Rice plant diseases have a significant impact on agricultural productivity, so a classification model is needed to accurately distinguish disease types from rice leaf images. Various leaf image-based classification approaches have been developed previously, but their performance can still be improved. Therefore, this study proposes the use of Convolutional Neural Network (CNN) to classify rice leaf diseases. The dataset used in this study includes rice leaf images in four conditions, namely blast, blight, tungro, and healthy. The initial stage involves preprocessing steps such as resizing, augmentation, and normalization. After this, a special CNN architecture is designed, consisting of 4 convolutional layers, 4 pooling layers, and 3 fully connected layers. Each convolutional layer uses a 3x3 kernel with stride 1 and ReLU activation, while the pooling layer applies max pooling with a 3x3 kernel and stride 2. Using a batch size of 32 and Adam optimization, the best results were achieved at 100 training epochs with a learning rate of 0.0002. The model achieved a training accuracy of 0.9930 and a loss value of 0.0221, while the testing accuracy reached 0.9647. The evaluation metrics showed balanced performance across precision, recall, and f1-score of 0.9647 each indicating that the classification process was effective without bias towards a particular class. These findings suggest that a well-designed yet simple CNN model can provide competitive classification results without relying on complex architectures or additional techniques. Furthermore, the proposed model outperforms several existing CNN architectures, including Inception-ResNet-V2, VGG-16, VGG-19, and Xception.

Keywords: Rice Diseases, Custom CNN Architecture, Leaf Image, Adam Optimization, Model Evaluation.

مستخلص البحث

سوسانتو، محمد هيري. ٢٠٢٥. تصنيف أمراض نبات الأرز باستخدام صور الأوراق باستخدام الشبكة العصبية التلافيفية. أطروحة. برنامج هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (أ) د. إروان بودي سانتوسو، م. كوم (ب) د. محمد أمين هريادي، م. ت.

الكلمات المفتاحية: أمراض الأرز، بنية CNN المخصصة، صورة الورقة، تحسين آدم، تقييم النموذج

إن أمراض الأرز لها تأثير كبير على الإنتاجية الزراعية، لذا فإن نماذج التصنيف ضرورية للتمييز بدقة بين أمراض أوراق الأرز. تم اقتراح نماذج تصنيف مختلفة لتصنيف أمراض الأرز استناداً إلى صور الأوراق، ولكن هناك حاجة إلى مزيد من تحسينات الأداء. وعليه، تقترح هذه الدراسة استخدام الشبكة العصبية التلافيفية (CNN) لتصنيف أمراض الأرز بناءً على صور الأوراق. في هذه الدراسة، استخدمنا مجموعة بيانات صور الأوراق التي تحتوي على أوراق مصابة باللفحة، والانفجار، والتونجرو، وأوراق صحية. في العملية الأولية، تتم معالجة البيانات أولاً، مثل تغيير الحجم، والتكبير، والتطبيع. بعد اكتمال المعالجة، يتم تشكيل بنية CNN مخصصة تتكون من 4 طبقات ملتوية، و 4 طبقات تجميع، و 3 طبقات متصلة بالكامل. تستخدم كل طبقة ملتوية نواة 3x3 وخطوة 1 وتنشيط ReLU، بينما تستخدم طبقة التجميع أقصى تجميع مع نواة 3x3 وخطوة 2. إلى جانب استخدام تكوين حجم الدفعة 32 وتحسين Adam، تم الحصول على أفضل نتائج الاختبار في التجربة باستخدام Epoch 100 ومعدل تعلم 0.0002 مما أدى إلى دقة تدريب 0.9930 وخسارة 0.0221 ودقة اختبار 0.9647. تمكنت نتائج تقييم النموذج من الحصول على توازن بين الدقة والاستدعاء ودرجة f1، أي 0.9647، مما يشير إلى أن عملية تصنيف البيانات تسير بشكل جيد للغاية، دون أي تحيز تجاه فئة معينة. تظهر نتائج هذه الدراسة أن نموذج CNN المبسط يمكنه توفير أداء تصنيف تنافسي دون الحاجة إلى نموذج CNN معقد أو تقنيات إضافية أخرى. نموذج CNN المقترح قادر على الأداء بشكل أفضل من نماذج CNN الحالية مثل Inception-ResNet-V2 و VGG-16 و VGG-19 و Xception.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tanaman padi merupakan makanan pokok masyarakat Indonesia. Seiring bertambahnya jumlah penduduk, konsumsi beras oleh masyarakat Indonesia meningkat dari tahun ke tahun. Oleh karena itu, perluasan areal pertanian dan pemanfaatan teknologi pertanian diperlukan untuk meningkatkan produksi padi di Indonesia. Hal ini semakin diperkuat dengan kondisi tanah dan iklim yang mendukung di Indonesia untuk budidaya padi. Tanah yang kaya dan subur di Indonesia sangat ideal untuk menanam tanaman seperti padi, dan iklim juga mendukung pertanian padi. Oleh karena itu kebutuhan pokok seperti tanaman padi dapat tercukupi dengan para petani padi dan juga lahan serta cuaca yang mendukung dalam proses pertanian.

Namun di tengah dukungan yang diberikan oleh tanah, cuaca dan lain – lain yang ada. Ada pula masalah yang timbul di tengah proses pertanian tersebut. Seperti hama dan pola iklim yang tidak menentu. Kondisi ini membuat para petani mengalami gagal panen atau berkurangnya jumlah produksi padi hingga kerugian yang cukup besar (Rijal et al., 2024). Permasalahan yang dihadapi petani adalah sulitnya mendeteksi penyakit sejak awal secara langsung di lapangan. Penelitian yang akan dilakukan bertujuan untuk mengatasi keterbatasan tersebut dengan memanfaatkan dataset citra daun padi yang mencakup fase awal hingga fase penyakit yang sudah parah. Penyakit tersebut biasanya timbul gejala-gejala secara

berkala. Sehingga para petani harus menanggulangi hal tersebut agar tidak mengalami kerugian besar.

Penyakit padi biasanya dideteksi secara manual dengan mengamati gejala-gejala seperti daun rusak, pertumbuhan terhambat, atau kegagalan menghasilkan benih. Namun, proses ini memakan waktu karena sering kali harus menunggu hingga gejalanya terlihat sebelum mengidentifikasi penyakit spesifik yang menyerang tanaman padi. Kesadaran masyarakat Indonesia terhadap hama dan penyakit padi masih rendah, termasuk pengetahuan mengenai cara pengobatan yang tepat. Hama serangga, hewan pengerat, dan berbagai penyakit padi masih belum sepenuhnya terkendali, dan para petani sangat bergantung pada pestisida. Kesalahan pengobatan yang sering terjadi bukan karena salah diagnosis, tetapi lebih karena kurangnya perhatian terhadap penyakit dan cara penggunaan pestisida yang tidak disesuaikan dengan hama dan penyakit yang menyerang tanaman padi.

Penelitian tentang klasifikasi penyakit padi menggunakan berbagai metode menunjukkan pendekatan yang berbeda untuk meningkatkan diagnosis. Satu penelitian menggunakan *Local Binary Pattern* (LBP) dan *Gray-Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur, mencapai akurasi 90% dengan pengklasifikasi *Decision Tree* (Alwy et al., 2023). Penelitian lain menggunakan model *Attention Based Neural Network* yang dioptimalkan dengan teknik Bayesian, mencapai akurasi 94,65% pada 2.370 sampel daun padi (Wang et al., 2021). Sementara itu, pendekatan yang berbeda menggunakan metode *Naive Bayes* pada dataset dari Kaggle mencapai akurasi 76%, dengan saran untuk perbaikan dengan melalui data yang lebih besar (Rahmi & Saputra, 2024).

Dalam agama Islam, perintah kepada manusia tentang pentingnya mempelajari ilmu tentang tanaman telah dianjurkan. Berikut firman Allah (ayat Al-Qur'an) yang berkaitan dengan hal tersebut selaras dengan surah Al-Hujurat ayat 13.

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۚ إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتْقَاكُمْ ۚ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

“Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa-bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia di antara kamu di sisi Allah ialah orang yang paling takwa di antara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal.” (Q.S Al-Hujurat:13)

Berdasarkan tafsir Tahlili, dalam ayat ini dijelaskan bahwa Allah menciptakan manusia dari seorang laki-laki (Adam) dan seorang perempuan (Hawa) dan menjadikannya berbangsa-bangsa, bersuku-suku, dan berbeda-beda warna kulit bukan untuk saling mencemoohkan, tetapi supaya saling mengenal dan menolong. Allah tidak menyukai orang-orang yang memperlihatkan kesombongan dengan keturunan, kepangkatan, atau kekayaannya karena yang paling mulia di antara manusia pada sisi Allah hanyalah orang yang paling bertakwa kepada-Nya. Kebiasaan manusia memandang kemuliaan itu selalu ada sangkut-pautnya dengan kebangsaan dan kekayaan. Padahal menurut pandangan Allah, orang yang paling mulia itu adalah orang yang paling takwa kepada-Nya.

Inti dari penafsiran tersebut, menegaskan pentingnya mengenali dan memahami perbedaan antarmanusia yang diciptakan dalam berbagai suku dan bangsa agar mereka dapat saling mengenal (ta'aruf). Konsep ini dapat diaplikasikan

dalam ranah sains melalui proses klasifikasi, di mana berbagai jenis penyakit padi diidentifikasi dan dibedakan berdasarkan ciri-ciri yang terdapat pada daunnya. Dalam konteks penelitian ini, klasifikasi penyakit padi merupakan pengenalan dan pemahaman yang lebih mendalam tentang keragaman alamiah yang diciptakan Allah, sejalan dengan tujuan yang digariskan dalam ayat tersebut.

Dengan demikian, penelitian ini mengusulkan penggunaan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan penyakit padi berbasis citra daun. Penelitian ini akan difokuskan pada penerapan metode CNN untuk mengevaluasi seberapa baik kinerjanya dalam menganalisis dan menginterpretasi citra daun padi. CNN dipilih karena dapat secara otomatis mempelajari dan mengekstraksi fitur-fitur penting dari data citra mentah. Ini berarti CNN dapat secara efektif mengidentifikasi dan memahami pola dalam gambar tanpa memerlukan ekstraksi fitur manual. Dan metode CNN telah terbukti berhasil mengklasifikasikan objek dalam gambar dan mengungguli metode pembelajaran mesin lainnya seperti SVM (Suartika E. P et al., 2016). Meskipun penelitian sebelumnya mungkin telah mencapai akurasi yang baik, akan selalu ada ruang untuk perbaikan. Kemampuan ini menjadikan CNN sebagai alat yang ampuh untuk mengklasifikasikan penyakit daun padi berdasarkan karakteristik visualnya.

1.2 Rumusan Masalah

Bagaimana performa Metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan penyakit daun tanaman padi berbasis citra daun.

1.3 Batasan Masalah

Mengingat luasnya cakupan masalah yang akan diteliti, maka penulis membatasi penelitian ini yaitu:

1. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) yang sederhana untuk klasifikasi penyakit padi. Dalam penelitian ini, tidak digunakan teknik tambahan seperti *transfer learning*, *fine-tuning*, atau metode lainnya yang sering digunakan untuk meningkatkan kinerja model. Pendekatan ini dipilih untuk fokus pada kemampuan dasar CNN dalam mengenali pola visual dari citra daun padi secara langsung, sehingga hasil yang diperoleh lebih merefleksikan efektivitas dasar arsitektur CNN itu sendiri tanpa pengaruh dari teknik eksternal.
2. Dataset penyakit padi yang digunakan dalam penelitian ini akan terbatas pada kumpulan data yang tersedia dari kaggle (publik) yang diunggah oleh pengunggah sekitar 8 bulan lalu dari tanggal 14 Agustus 2024.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengklasifikasikan penyakit padi dengan menggunakan *Convolutional Neural Network* (CNN) pada objek gambar berbasis citra daun.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini diharapkan dapat memberi beberapa manfaat yaitu:

1. Petani dapat dengan mudah mendeteksi penyakit padi lebih cepat dan akurat. Ini menghemat waktu dan biaya serta meningkatkan hasil panen dengan mencegah kerusakan yang lebih luas pada tanaman.
2. Dinas pertanian dapat menggunakan dan mengembangkan model CNN ini untuk membangun sistem pemantauan otomatis untuk deteksi penyakit padi. Sistem ini akan mempercepat proses pengumpulan data dan memberikan informasi yang lebih tepat waktu kepada petani. Dengan pemanfaatan model ini, dinas pertanian dapat mengembangkan teknologi yang lebih canggih untuk pengelolaan penyakit padi yang berbasis data, serta meningkatkan efisiensi dan efektivitas program-program pertanian

BAB II

STUDI PUSTAKA

2.1 Klasifikasi Penyakit Padi

Klasifikasi, yang berasal dari kata "classificatie" dalam bahasa Belanda dan "classification" dalam bahasa Prancis, adalah metode sistematis untuk mengorganisasikan data menurut aturan yang telah ditetapkan. Kamus Besar Bahasa Indonesia (KBBI) mengatakan bahwa klasifikasi adalah penempatan informasi yang teratur ke dalam kelompok atau kategori berdasarkan aturan tertentu. Dalam konteks klasifikasi penyakit padi, proses ini melibatkan penggunaan teknik canggih untuk mengidentifikasi dan mengkategorikan berbagai penyakit yang menyerang tanaman padi. Komputer melakukan klasifikasi melalui pembelajaran mendalam dengan menganalisis gambar daun padi dan mendeteksi pola atau anomali yang terkait dengan berbagai penyakit. Algoritma pembelajaran mendalam dilatih untuk mengekstrak fitur dan pola dari gambar ini, yang kemudian digunakan untuk mengkategorikan penyakit ke dalam kelas tertentu (Pratama et al., 2024). Dalam bidang pertanian, metode ini sangat penting untuk menyortir dan mengkategorikan tanaman, hama/penyakit, jenis tanah, dan elemen pertanian lainnya. Klasifikasi membantu memahami pola, mengelola sumber daya secara lebih efektif, dan meningkatkan proses pengambilan keputusan tentang pengelolaan tanaman dan pertanian.

Seperti yang dilakukan oleh Chaudhary & Kumar, (2024) studi ini mengumpulkan data ekstensif untuk memastikan temuan yang kuat dan efektif. Dengan menggunakan pendekatan ekstraksi fitur gabungan menggunakan metode

Gray-level Co-occurrence Matrix (GLCM) dan *Intensity-Level Based Multi-Fractal Dimension* (ILMFD), diikuti oleh klasifikasi dengan teknik *Support Vector Machine* (SVM), yang dikenal dengan akurasi tinggi. Fase pra-pemrosesan mencakup teknik seperti GLCM, *Histogram of Oriented Gradients* (HOG), dan *color moments*, yang digabungkan untuk membuat fitur hibrida. Berbagai model klasifikasi, termasuk *Artificial Neural Network* (ANN), *Neuro-Genetic Algorithm* (Neuro-GA), dan SVM, dibandingkan untuk akurasi. Model *Neuro-GA* lebih baik daripada model ANN, tetapi SVM terbukti lebih unggul, mencapai akurasi 96,7% dalam mendeteksi penyakit bercak coklat pada padi. Pendekatan terintegrasi ini, yang diuji dalam MATLAB, menunjukkan harapan untuk memajukan deteksi penyakit pertanian.

Dalam penelitian Sunandar & Sutopo, (2024) ini menggunakan 120 foto JPG dari daun padi dari Universitas Dharmasinh Desai di Gujarat, India. Gambar-gambar yang menunjukkan Penyakit Bercak Daun Bakteri, Penyakit Bercak Daun, dan Bercak Cokelat diberi label dan dibagi menjadi set pelatihan yang terdiri dari 56 data dan set pengujian yang terdiri dari 24 data. Model *Artificial Neural Network* (ANN) yang dilatih menggunakan *Google Colaboratory* dengan 100 *epoch* ini mencapai akurasi 83%, presisi 84%, *recall* 84%, dan F1-skor 83%, yang menunjukkan sistem yang tangguh untuk diagnosis penyakit padi secara otomatis.

Studi yang dilakukan oleh Setiawan et al., (2023) melihat klasifikasi penyakit daun padi menggunakan algoritma *Nu-Support Vector Machine* (Nu-SVM) dengan pendekatan validasi silang lima kali lipat. Kemampuan prediktif model sedang, dengan akurasi 52,12% hingga 53,81%. Nilai presisi sangat

bervariasi di seluruh lipatan, dengan nilai tertinggi 51,95% dan terendah 37,66%. *Recall* relatif konsisten, menunjukkan kemampuan yang stabil untuk menemukan positif yang sebenarnya. *F1-Score*, yang menyeimbangkan presisi dan *recall*, 39–43%. Skor ini menunjukkan kesulitan dalam mencapai presisi tinggi tanpa mengorbankan *recall*. Hasil menunjukkan bahwa penyempurnaan tambahan diperlukan untuk meningkatkan akurasi dan presisi model. Penelitian ini menunjukkan potensi dan kekurangan metode klasifikasi saat ini, berkontribusi pada bidang pertanian presisi.

Studi yang dilakukan oleh Abasi et al., (2023) melibatkan pengumpulan dataset N citra yang mewakili berbagai penyakit daun padi, seperti *tungro*, *blas*, hawar daun, bercak coklat, dan penyakit busuk daun. Citra diresize menjadi 224×224 piksel dan dinormalisasi ke $[0, 1]$. Untuk menilai efektivitas model, data dibagi menjadi set pelatihan, validasi, dan pengujian (80%, 10%, dan 10%). *Convolutional Neural Network* (CNN) yang disesuaikan memiliki tingkat kehilangan yang rendah dan overfitting yang rendah, dengan akurasi 0,914. Untuk menilai kinerja, matriks kebingungan digunakan. Studi ini menunjukkan bahwa dataset yang beragam sangat penting untuk keberhasilan pembelajaran mendalam dan menyarankan bahwa penelitian di masa depan harus berkonsentrasi pada dataset yang lebih bervariasi untuk meningkatkan generalisasi model.

Penelitian tentang prediksi penyakit daun padi yang dilakukan oleh N et al., (2021) menggunakan *deep neural networks* dengan *transfer learning* menyajikan metodologi yang komprehensif, yang mengintegrasikan pengumpulan data, praproses citra, dan pelatihan model untuk mencapai akurasi yang tinggi. Citra

RGB daun padi dikumpulkan, dinormalisasi ke rentang 0-1, diubah ukurannya menjadi 224x224 piksel, dan ditambah melalui rotasi dan pembalikan. Penelitian ini menggunakan jaringan saraf konvolusional InceptionResNetV2 dengan *transfer learning*, pelatihan pada 4.000 citra, dan pengujian pada 300 citra. Pendekatan ini mencapai akurasi 95,67% yang mengesankan, yang menyoroti kinerja pembelajaran mendalam yang unggul dibandingkan metode tradisional dan menekankan peran penting praproses citra dalam manajemen penyakit pertanian.

Studi yang dilakukan oleh Saputra et al., (2020) bertujuan untuk mengklasifikasikan tiga penyakit daun padi: *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Untuk mencapai tujuan ini, mereka menggunakan *Gray-Level Co-occurrence Matrix* (GLCM) dan algoritma *K-Nearest Neighbor* (KNN). Mereka menemukan fitur seperti kontras, energi, entropi, homogenitas, dan korelasi dari 120 citra yang diambil dari repositori UCI. Dengan $k = 11$, algoritma KNN mencapai akurasi tertinggi sebesar 65,83%. Ini diuji dengan nilai k dari 1 hingga 20. Studi ini menunjukkan bahwa metode GLCM dan KNN efektif dalam mendeteksi penyakit daun padi sejak dini. Ini sangat penting untuk mengurangi kemungkinan gagal panen di daerah seperti Indonesia di mana beras adalah makanan pokok.

Terakhir, Gunawan et al., (2021) melakukan sebuah penelitian tentang klasifikasi penyakit daun padi menggunakan *Artificial Neural Network* (ANN) melibatkan pendekatan sistematis yang mengintegrasikan pengumpulan data, metode pemrosesan, dan hasil. Penelitian ini memanfaatkan data citra daun padi dalam format warna RGB untuk mengidentifikasi karakteristik penyakit bercak

coklat dan penyakit bercak daun. Normalisasi data dilakukan untuk memastikan konsistensi dalam nilai fitur, yang penting untuk aktivasi yang efektif dalam arsitektur *Artificial Neural Network*. Kumpulan data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian, dengan model dilatih menggunakan algoritma *backpropagation*. Penelitian ini melaporkan akurasi klasifikasi sebesar 66,3% selama pelatihan dan 76,7% selama pengujian. Untuk meningkatkan akurasi, rekomendasinya meliputi peningkatan fitur input, penambahan lapisan tersembunyi, dan bereksperimen dengan fungsi aktivasi yang berbeda. Penelitian ini memberikan contoh pendekatan komprehensif dalam menggunakan ANN untuk klasifikasi penyakit pertanian, dengan menekankan metodologi dan hasil. Berikut ringkasan penelitian terkait dapat dilihat pada Tabel 2.1:

Tabel 2.1 Penelitian Terkait

Kriteria	Penelitian 1	Penelitian 2	Penelitian 3	Penelitian 4	Penelitian 5	Penelitian 6	Penelitian 7	Penelitian ini
Peneliti (tahun)	(Chaudhary & Kumar, 2024)	(Sunandar & Sutopo, 2024)	(Setiawan et al., 2023)	(Abasi et al., 2023)	(N et al., 2021)	(Saputra et al., 2020)	(Gunawan et al., 2021)	Moh. Heri Susanto
Metode	Kombinasi fitur (<i>Gray-level Co-occurrence Matrix</i>) GLCM dan (<i>Intensity-Level Based Multi-Fractal Dimension</i>) ILMFD + <i>Support Vector Machine</i> (SVM)	<i>Artificial Neural Network</i> (ANN)	<i>Nu-Support Vector Machine</i> (Nu-SVM)	<i>Convolutional Neural Network</i> (CNN)	<i>InceptionResNetV2</i> + <i>transfer learning</i>	<i>Gray-Level Co-occurrence Matrix</i> (GLCM) + <i>K-Nearest Neighbor</i> (KNN)	<i>Artificial Neural Network</i>	<i>Convolutional Neural Network</i> (CNN)
Dataset	Daun sehat dan bercak coklat. Memiliki 2011 dataset.	<i>bacterial leaf blight</i> , <i>brown spots</i> , dan <i>left smut</i> . Total keseluruhan data adalah 120.	<i>Brown spot</i> dan <i>leaf blast</i> dengan data latih 90% dan data uji 10%.	<i>Tungro</i> , <i>blast</i> , <i>leaf blight</i> , dan <i>brown spots</i> . Memiliki total keseluruhan 320 data.	<i>Leaf blast</i> , <i>bacterial blight</i> , <i>brown spots</i> , dan <i>healthy</i> . Dengan 4000 data latih dan 300 data uji.	<i>leaf blight</i> , <i>brown spot</i> , dan <i>leaf smut</i> .	<i>brown spot</i> dan <i>leaf smut</i> dengan data latih 80% dan data uji 20%	<i>Blight</i> , <i>blast</i> , <i>healthy</i> dan <i>tungro</i> . Dengan keseluruhan dataset mencapai 710.
Hasil/Performa	96.92%	83%	53.81%	91.4%	95.67%	65.83%	66.3%	-
Keterbatasan	Hanya mengklasifikasikan penyakit padi pada daun bercak coklat dan daun sehat.	Jumlah dataset kecil dan tidak melakukan augmentasi data.	Varian daun penyakit padi sedikit dan akurasi lemah.	Dataset relatif kecil dan tidak ada penanganan terhadap <i>overfitting</i> seperti <i>augmentasi/cross validation</i> .	Data berisi objek tumpang tindih dan tidak dilakukan proses ROI. Dimana pada kelas <i>blast</i> dan <i>blight</i> data memiliki latar belakang padi	Performa model kurang baik dan tidak memiliki	Jumlah dataset kecil dan tidak memiliki data daun	Hanya difokuskan pada penggunaan metode CNN dan tidak melakukan

Kriteria	Penelitian 1	Penelitian 2	Penelitian 3	Penelitian 4	Penelitian 5	Penelitian 6	Penelitian 7	Penelitian ini
					<p>yang lain, untuk <i>brown spots</i> berlatar belakang hitam polos dan <i>healthy</i> berlatar belakang putih polos. Walaupun akurasi bagus, ini tidak selalu menjamin bahwa model benar-benar memahami fitur penting dari gambar. Ada kemungkinan model tersebut tidak belajar dari pola yang benar tetapi malah mengeksploitasi noise atau fitur yang tidak diinginkan.</p>	data daun padi sehat.	padi yang sehat.	<p>teknik tambahan untuk meningkatkan performa seperti <i>transfer learning</i>, <i>fine tuning</i>, dll.</p>

2.2 Penyakit Daun Tanaman Padi

Menurut Agustiani et al., (2022) penyakit daun padi merupakan salah satu musuh utama dalam pertanian, hal tersebut sangat mempengaruhi kualitas beras yang dihasilkan. Tanaman sakit menunjukkan gejala berupa bercak dengan pola dan warna tertentu. Gejala tersebut dapat diamati pada bagian daun padi. Beberapa penyakit padi antara lain:

2.2.1 *Blight*



Gambar 2.1 Blight Pada Daun Padi

Penyakit *Blight* atau hawar daun bakteri. Penyakit yang disebabkan oleh bakteri *Xanthomonas Oryzae Pvoryzae* (Xoo) merupakan salah satu penyakit terpenting yang mempengaruhi produksi padi. Penyakit ini dapat menginfeksi tanaman padi pada tahap pertumbuhan dan reproduksi serta menurunkan produksi padi sebesar 30-40%. Sebagaimana ditunjukkan pada Gambar 2.1, penyakit ini memiliki ciri-ciri sebagai berikut:

- a) Bagian tanaman yang terserang penyakit pada umumnya terdapat pada daun tanaman padi.
- b) Bentuk gejalanya ialah mengandung lesi memanjang di ujung daun padi.
- c) Warna lesi pada penyakit ini berwarna kuning ke putih dikarenakan pengaruh bakteri.

2.2.2 Blast



Gambar 2.2 Blast Pada Daun Padi

Penyakit blas yang disebabkan oleh *Pyricularia grisea* merupakan penyakit penting dalam budidaya padi di Indonesia, khususnya padi gogo di lahan kering. Terjadinya smut pada padi diduga terkait secara khusus dengan teknik budidaya padi. Penggunaan pupuk N dosis tinggi dan penanaman kultivar yang tidak tahan terhadap penyakit ini. Fenomena serupa juga terjadi di berbagai negara seperti Jepang, Filipina, Vietnam, dan Korea.

Gambar 2.2 menunjukkan bahwa penyakit ini memiliki ciri-ciri dan gejala diantaranya sebagai berikut:

- a) Warna gejalanya abu-abu agak putih dan tepi pinggir berwarna coklat sedikit orange.
- b) Bentuknya belah ketupat, bagian tepi agak runcing.
- c) Biasanya bercak-bercak ini mulai tumbuh pada masa vegetative.

- d) Bercak-bercak bisa menyatu sehingga tampak seperti terbakar.
- e) Disisi lain malai bisa mengkerut dan padi bisa terisi setengah bahkan kosong.

2.2.3 *Tungro*



Gambar 2.3 *Tungro* Pada Daun Padi

Penyakit *tungro* pada padi adalah salah satu penyakit virus yang paling merusak tanaman padi. Penyakit ini dipicu oleh dua jenis virus, yaitu *Rice Tungro Bacilliform Virus* (RTBV) dan *Rice Tungro Spherical Virus* (RTSV). Virus ini disebarkan oleh serangga vektor, khususnya wereng hijau (*Nephotettix virescens*) dan wereng daun (*Nephotettix nigropictus*). Gambar 2.3 mengilustrasikan gejala penyakit tungro pada tanaman padi, yaitu sebagai berikut:

- a) Kerdil: Tanaman yang terinfeksi mengalami kerdil dan pertumbuhan terhambat.
- b) Klorosis: Daun menjadi kuning hingga oranye, terutama pada daun muda.
- c) Pengguguran bunga dan biji: Tanaman yang terinfeksi parah cenderung menghasilkan sedikit atau bahkan tidak menghasilkan biji.

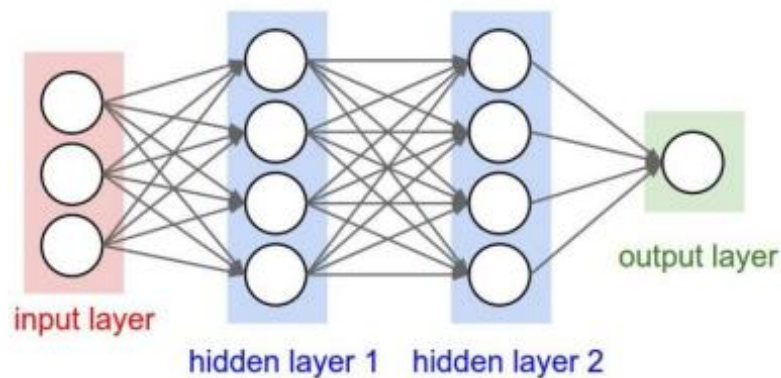
- d) Malformasi daun: Daun dapat mengalami perubahan bentuk, menjadi lebih sempit dan pendek.

2.3 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah evolusi dari *multilayer perceptron* (MLP) yang dirancang untuk memproses data dua dimensi. Karena memiliki struktur lapisan yang berlapis atau yang mendalam, CNN dikategorikan sebagai *deep learning* dan cocok untuk diterapkan pada data gambar (Ihtada et al., 2025). Untuk klasifikasi citra, MLP tidak cocok karena tidak memiliki mekanisme untuk menangkap atau memanfaatkan informasi spasial (pola dan bentuk) dari data citra.

Awal perkembangannya, CNN bernama NeoCognitron dilakukan oleh Kunihiko Fukushima, seorang peneliti di NHK Science and Technology Research Laboratories yang berlokasi di Kinuta, Setagaya-ku, Tokyo. Konsep tersebut diselesaikan oleh peneliti Yann LeChun dari AT&T Bell Laboratories di Holmdel, New Jersey, AS. LeChun berhasil mengerjakan pengenalan digit dan tulisan tangan menggunakan model CNN yang disebut LeNet. Kemenangan Alex Krizhevsky tahun 2012 di ImageNet *Large Scale Visual Recognition Challenge* 2012 menggunakan aplikasi CNN menjadi ajang pembuktian metode deep learning khususnya CNN.

CNN beroperasi mirip dengan *Multi-Layer Perceptron* (MLP), tetapi dalam CNN, setiap neuron direpresentasikan dalam dua dimensi, tidak seperti MLP yang setiap neuronnya hanya satu dimensi Septiandi et al., (2021).



Gambar 2.4 Arsitektur MLP Sederhana

Seperti yang ditunjukkan pada Gambar 2.4, MLP terdiri dari lapisan-i (kotak merah dan biru) dan setiap lapisan berisi neuron (lingkaran putih). MLP menerima input data satu dimensi dan mengirimkannya melalui jaringan untuk menghasilkan output. Setiap penghubung antara neuron dalam dua lapisan yang berdekatan memiliki parameter bobot satu dimensi yang menentukan kualitas ruang. Setiap lapisan data masukan menerapkan operasi linier ke nilai bobot yang ditetapkan dan kemudian menyesuaikan hasil yang dihitung menggunakan operasi nonlinier yang dikenal sebagai fungsi aktivasi.

Di CNN, data yang beredar online adalah dua dimensi, sehingga fungsi linear dan bobot parameter berbeda di CNN. Di CNN, operasi linier dilakukan menggunakan operasi konvolusi, yang bobotnya tidak satu dimensi melainkan terstruktur dalam empat dimensi, yang terdiri dari sekumpulan kernel konvolusional. Dimensi bobot CNN adalah (h, w, d_{in}, d_{out}) . Dimana h adalah tinggi dari kernel konvolusi, w adalah lebar dari kernel konvolusi, d_{in} adalah jumlah saluran input (misalnya, 3 untuk gambar RGB), dan d_{out} adalah jumlah saluran output atau jumlah filter di lapisan konvolusi. Contoh dimensi bobotnya

adalah (3,3,3,64). Berarti lapisan konvolusi memiliki kernel berukuran 3x3, 3 saluran untuk RGB, dan 64 filter (kernel).

CNN (*Convolutional Neural Networks*) dirancang untuk bekerja dengan data yang memiliki dua dimensi, seperti gambar dan audio. Ini karena proses konvolusi, yaitu cara CNN memproses data, sangat efektif dalam menemukan pola dalam format dua dimensi. Konvolusi menggunakan filter kecil yang bergerak di seluruh area data untuk mencari pola seperti tepi atau sudut. Salah satu kelebihan dari konvolusi adalah kemampuannya untuk mendeteksi pola yang sama di berbagai bagian dari data. Artinya, jika pola tertentu muncul di satu bagian gambar atau audio, CNN dapat mengenalinya di bagian lain, tanpa peduli di mana pola itu berada. Inilah sebabnya mengapa CNN sangat cocok untuk menganalisis gambar dan audio, di mana pola penting bisa muncul di berbagai lokasi dalam data.


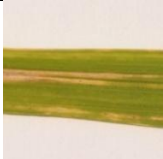


BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Dalam penelitian ini, mengumpulkan berbagai dataset daun padi dari Kaggle untuk mengklasifikasikan jenis penyakit yang menyerang tanaman tersebut. Untuk mengklasifikasikan jenis penyakit pada daun padi, daun yang digunakan dalam penelitian ini dikelompokkan menjadi 4 kelas diantaranya 3 jenis penyakit padi berupa 220 data *blight*, 200 data *blast* dan 80 data *tungro*. Satu kelasnya lagi untuk data daun padi *healthy* terdiri 210 data. Jumlah keseluruhan citra daun yang berhasil dikumpulkan adalah 710 citra. Berikut contoh sample data penyakit daun padi yang digunakan dalam penelitian dapat dilihat pada Tabel 3.1:

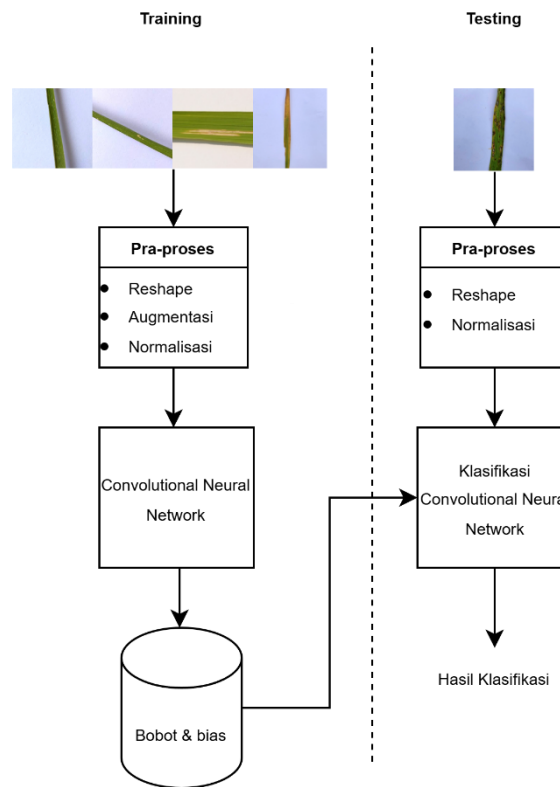
Tabel 3.1 Contoh Sampel Gambar Dari Dataset Yang Digunakan

No.	Nama Kondisi Daun Padi	Gambar
1.	<i>Blast</i>	
2.	<i>Blight</i>	
3.	<i>Tungro</i>	
4.	<i>Healthy</i>	

3.2 Input Citra

Dalam penelitian ini, citra daun padi yang digunakan sebagai input untuk model CNN dibagi menjadi dua kelompok utama yaitu data latih dan data uji. Data latih, yang merupakan 80% dari kumpulan data, digunakan untuk melatih model agar mengenali pola dalam citra yang terkait dengan penyakit tanaman padi. Proses pelatihan melibatkan langkah-langkah pra-proses, seperti *reshape*, augmentasi, dan normalisasi. Setelah itu data akan diproses pada model *Convolutional Neural Network*. Dan parameter yang dilatih (bobot dan bias) disimpan untuk penggunaan selanjutnya.

Setelah model dilatih, 20% kumpulan data yang tersisa digunakan sebagai data uji. Proses pengujian dilakukan dengan cara yang sama *reshape* dan normalisasi. Gambar yang diproses ini kemudian dimasukkan ke dalam CNN, yang menggunakan bobot dan bias yang dipelajari sebelumnya untuk mengklasifikasikan data uji, menghasilkan hasil klasifikasi akhir. Berikut skema proses data *training* dan *testing* pada Gambar 3.1 di bawah ini:



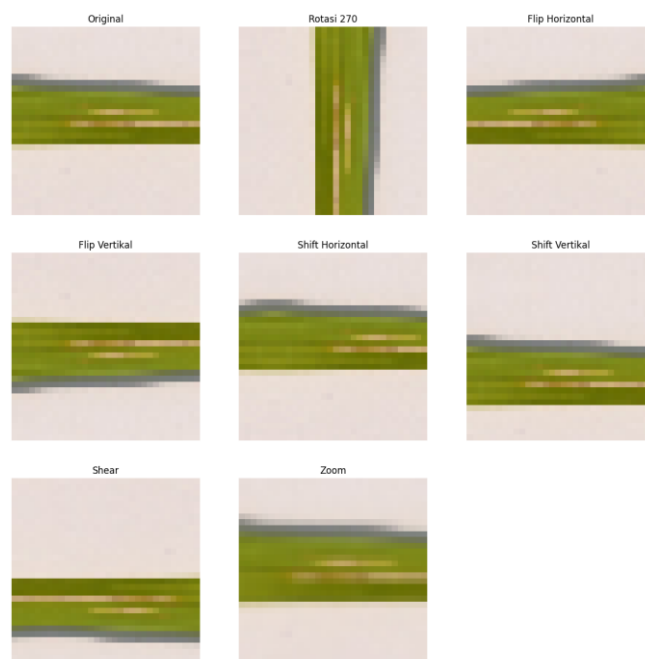
Gambar 3.1 Skema Data *Training* dan data *Testing*

3.3 Pra-proses

Pra-proses adalah langkah penting dalam mempersiapkan data untuk model CNN, dan melibatkan beberapa teknik utama, seperti *resize*, dan *augmentation*. Karena fitur warna dianggap penting untuk klasifikasi, konversi warna tidak dilakukan. Pertama, *resize* gambar dilakukan untuk memastikan bahwa semua gambar memiliki ukuran yang konsisten sebelum diproses oleh model. Teknik ini penting untuk menjaga keseragaman dimensi input, sehingga model dapat memproses data secara efisien. Dalam penelitian ini, dilakukan *resize* dengan ukuran 64 x 64 piksel.

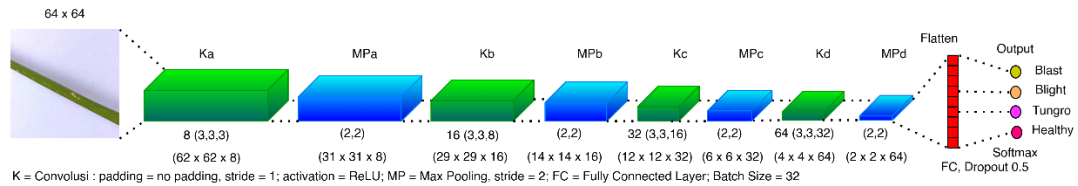
Selanjutnya, augmentasi gambar digunakan untuk meningkatkan keragaman data latih dan memperbaiki kemampuan generalisasi model. Rotasi gambar pada sudut 270 derajat memungkinkan model untuk belajar objek meskipun posisinya berbeda. Selain itu, *flip* horizontal-vertikal, *shift* horizontal-vertikal, *shear* dan *zoom* menambah variasi data latih dengan menciptakan cermin dari gambar asli, yang memperkaya representasi data dan mengurangi risiko overfitting. Teknik-teknik ini secara signifikan meningkatkan kemampuan model untuk menangani variasi gambar dalam data uji dan meningkatkan akurasi klasifikasi.

Setelah augmentasi, normalisasi data diterapkan pada data pelatihan dan pengujian. Normalisasi menskalakan nilai piksel ke rentang standar, antara 0 dan 1, untuk memastikan konsistensi dan membantu model belajar lebih efektif. Berikut hasil pra-proses data dengan menampilkan salah satu data yang dilampirkan pada Gambar 3.2:



Gambar 3.2 Gambar Yang Telah Dilakukan Pra-proses

3.4 Arsitektur CNN Usulan



Gambar 3.3 Arsitektur CNN Usulan

Gambar 3.3 merupakan arsitektur usulan untuk model klasifikasi penyakit padi menggunakan CNN dirancang untuk mengoptimalkan deteksi fitur pada gambar berukuran 64x64 piksel. Model ini dimulai dengan lapisan konvolusi pertama yang menerapkan 8 filter berukuran 3x3 dengan 3 input *channel* (RGB), lapisan *max pooling* pertama dengan menggunakan filter 2x2, lapisan konvolusi kedua menggunakan 16 filter yang berukuran 3x3 dengan 8 input *channel*, lapisan *max pooling* kedua dengan menggunakan filter berukuran 2x2, lapisan konvolusi ketiga dengan menggunakan 32 filter berukuran 3x3 dengan input *channel* 16, lapisan *max pooling* ketiga dengan filter 2x2, lapisan konvolusi keempat menggunakan 64 filter berukuran 3x3 dengan 32 input *channel*, dan lapisan *max pooling* keempat menerapkan ukuran filter 2x2. Seluruh lapisan konvolusi dilengkapi dengan aktivasi *ReLU* untuk memperkenalkan non-linearitas. Setelah fitur diekstrak, selanjutnya hasilnya diratakan menjadi vektor atau proses *flatten* dan dimasukkan ke dalam lapisan *fully connected*. Disini jaringan belajar untuk mengklasifikasikan gambar berdasarkan fitur yang diekstraksi. Outputnya berupa data mentah (logit). Akhirnya, lapisan *softmax* digunakan untuk mengonversi keluaran dari lapisan *fully connected* menjadi probabilitas klasifikasi akhir,

memungkinkan model untuk menentukan jenis penyakit padi dengan akurasi tinggi berdasarkan gambar resolusi kecil.

3.4.1 Convolutional Layer

Pertama, lapisan konvolusi (*convolution layer*). *Convolutional Layer* adalah komponen inti dari *Convolutional Neural Networks* (CNN). Fungsi utamanya adalah untuk mengekstraksi fitur dari data input, seperti gambar atau sinyal lainnya, melalui operasi konvolusi Azmi et al., (2023).

Convolutional Layer bekerja dengan cara menerapkan serangkaian filter (juga dikenal sebagai kernel) ke gambar masukan untuk mengekstraksi fitur tingkat tinggi. Setiap filter adalah matriks kecil yang meluncur (berkonvolusi) melintasi gambar masukan, menghitung produk titik antara elemen filter dan *patch* gambar masukan yang tumpang tindih dengannya. Hasil dari operasi ini adalah peta fitur, yang menyoroti pola tertentu dalam gambar, seperti tepi, tekstur, atau bentuk yang lebih kompleks. Berikut rumus untuk menghitung output dari lapisan konvolusi pada Persamaan 3.1 (Santoso et al., 2022):

$$y_a = g(W_a X + b_a) \quad (3.1)$$

Keterangan:

y_a = output dari neuron ke-a

g = fungsi aktivasi pada konvolusi

W_a = filter bobot ke-a dari proses konvolusi.

X = input gambar

b_a = filter bias ke-a dari proses konvolusi

a = Indeks a menunjukkan banyaknya filter dalam satu lapisan konvolusi, di mana setiap filter menghasilkan satu feature map.

Pada saat convolutional bekerja ada beberapa atribut/parameter di dalamnya. Pertama, *stride* adalah langkah atau perpindahan yang digunakan saat

filter diterapkan pada data input. Bisa satu kolom atau lebih. Kedua, *padding* adalah proses menambahkan satu baris/satu kolom lagi yang berisi 0 bertujuan untuk mempertahankan dimensi tinggi dan lebar gambar input tetap sama setelah operasi konvolusi. *Padding* ini bisa dilakukan atau tidak dilakukan karena disesuaikan dengan kebutuhan. Jika tanpa *padding*, tujuan utamanya adalah untuk mengurangi dimensi tinggi dan lebar gambar input sesuai dengan ukuran filter (kernel) yang digunakan dan *stride* (langkah) yang diterapkan.

Berikut rumus konvolusi dengan *padding* pada persamaan 3.2 dan 3.3 (Bhat et al., 2023):

$$W_2 = \frac{F_1 - 1}{2} \quad (3.2)$$

$$H_2 = \frac{F_2 - 1}{2} \quad (3.3)$$

Berikut jika rumus konvolusi tanpa *padding* pada persamaan 3.4 dan 3.5:

$$W_2 = \frac{W_1 - F_1}{S} + 1 \quad (3.4)$$

$$H_2 = \frac{H_1 - F_2}{S} + 1 \quad (3.5)$$

Keterangan:

W_1 dan H_1 : Lebar dan Tinggi dari input matriks

F_1 dan F_2 : Lebar dan Tinggi Filter/Kernel

P: *Padding*

S: *Stride*

Dalam studi ini, konvolusi dilakukan tanpa *padding* karena beberapa alasan utama. Pertama, konvolusi tanpa *padding* menyederhanakan proses komputasi dan memberikan visualisasi yang lebih jelas tentang cara kerja filter pada citra input

tanpa menambahkan piksel tambahan di tepinya. Kedua, pendekatan ini secara alami mengurangi dimensi citra input setelah setiap operasi konvolusi, yang membantu mengurangi kompleksitas data dan jumlah parameter yang perlu dipelajari jaringan. Pengurangan dimensi ini juga membantu mengurangi risiko overfitting dengan memaksa jaringan untuk fokus pada fitur yang lebih ringkas dan penting. Selain itu, untuk kasus pengklasifikasian penyakit padi berdasarkan citra daun, sangat penting untuk menangkap fitur inti yang sering kali terletak di tengah citra, sehingga membuat *padding* tidak diperlukan dan berpotensi menimbulkan noise. Proses algoritma pada *convolutional layer* tertera pada Lampiran I.

3.4.2 Rectified Linear Unit (ReLU)

Dalam studi ini, fungsi aktivasi *Rectified Linear Unit* (ReLU) digunakan dalam CNN untuk mengklasifikasikan penyakit padi berdasarkan gambar daun. ReLU dipilih karena kemampuannya untuk menangani non-linearitas dalam data secara efektif. Dengan menetapkan semua nilai negatif ke nol dan mempertahankan nilai positif, ReLU memperkenalkan non-linearitas penting ke dalam model, yang memungkinkannya untuk mempelajari dan merepresentasikan pola dan fitur kompleks dari gambar daun. Kemampuan ini sangat penting untuk mendeteksi dan mengklasifikasikan berbagai pola penyakit yang ada di daun secara akurat, yang sering kali menunjukkan karakteristik non-linier dalam tekstur, bentuk, dan warna. Selain itu, efisiensi komputasi ReLU mempercepat proses pelatihan, menjadikannya pilihan yang optimal untuk mengelola kumpulan data ekstensif

yang biasanya terlibat dalam tugas klasifikasi penyakit berbasis gambar. Untuk rumus ReLU disajikan dengan persamaan 3.6 berikut (Santoso & Purnama, 2020):

$$\widehat{y}_a(y_a) = \begin{cases} 0, & y_a < 0 \\ y_a, & y_a \geq 0 \end{cases} \quad (3.6)$$

Dimana y_a adalah output dari lapisan konvolusi. Dan untuk algoritma ReLU telah dilampirkan pada Lampiran 2.

3.4.3 Pooling Layer

Pooling merupakan operasi penting dalam *Convolutional Neural Network* (CNN) yang mengurangi dimensi spasial peta fitur, yang mengarah pada penurunan beban komputasi dan lebih sedikit parameter dalam jaringan (Rahman et al., 2023). Ada beberapa jenis penggabungan, dengan yang paling umum adalah *Max Pooling* dan *Average Pooling*. *Max Pooling* memilih nilai maksimum dalam setiap jendela penggabungan, sementara *Average Pooling* menghitung nilai rata-rata.

Dalam studi ini, *Max Pooling* digunakan untuk mengklasifikasikan penyakit padi berdasarkan gambar daun. *Max Pooling* dipilih karena kemampuannya untuk menangkap fitur yang paling menonjol dengan memilih nilai maksimum dalam setiap jendela penggabungan. Pendekatan ini efektif dalam mempertahankan informasi yang paling penting sambil membuang detail yang kurang signifikan, yang membantu dalam mempertahankan ketahanan representasi fitur. Penggunaan penggabungan maksimum juga meningkatkan kemampuan model untuk mengenali dan mengklasifikasikan berbagai pola penyakit, karena mempertahankan sinyal aktivasi terkuat yang sesuai dengan karakteristik utama penyakit daun. Selain itu, dengan mengurangi dimensionalitas data, penggabungan maksimum berkontribusi

pada waktu pelatihan yang lebih cepat dan generalisasi yang lebih baik, menjadikannya pilihan yang tepat untuk menangani gambar beresolusi tinggi yang terlibat dalam tugas klasifikasi ini.

Sementara, rumus untuk menemukan peta fitur keluaran setelah melakukan proses *max pooling* dapat dilihat pada persamaan 3.7 dan 3.8 (Ghosh et al., 2019):

$$W_2 = \frac{W_1 - F_1}{S} \quad (3.7)$$

$$H_2 = \frac{H_1 - F_2}{S} \quad (3.8)$$

Algoritma *pooling* dapat dilihat pada lampiran 3.

3.4.4 Flatten Layer

Flatten adalah proses mengubah peta fitur multidimensi menjadi vektor satu dimensi, yang memungkinkan transisi dari ekstraksi fitur ke klasifikasi. Disini, algoritma flatten telah dilampirkan pada Lampiran 4.

3.4.5 Fully Connected Layer

Fully Connected Layer adalah komponen kunci yang memproses dan mengintegrasikan fitur yang diekstraksi oleh lapisan sebelumnya. Yang bertugas untuk mengolah dan menggabungkan fitur yang diekstraksi menjadi output mentah (logits). Contoh, katakanlah output dari *fully connected layer* terakhir adalah [3,0; 1,5; 2,0; 0,7]. Ini adalah skor mentah untuk masing-masing kelas *Blast*, *Blight*, *Tungro*, dan *Healthy*. Untuk algoritma *fully connected layer*, dapat dilihat pada Lampiran 5.

3.4.6 Softmax

Setelah *Fully Connected Layer* menghasilkan skor keluaran mentah (logit) untuk setiap kelas, fungsi *softmax* diterapkan pada logit ini untuk mengubahnya menjadi probabilitas. Pengklasifikasi *Softmax* merupakan perluasan dari algoritma *Regresi Logistik* yang mampu mengklasifikasikan kelas menjadi lebih dari dua kategori (Wulandari et al., 2020). Fungsi *softmax* mengubah skor mentah menjadi distribusi probabilitas, memastikan bahwa setiap nilai tidak negatif dan jumlah semua probabilitas sama dengan satu. Proses normalisasi ini memungkinkan model mengeluarkan sekumpulan probabilitas yang dapat diartikan sebagai kemungkinan setiap kelas. Misalnya, jika logit untuk suatu gambar adalah [2,0; 1,0; 0,1], fungsi *softmax* akan mengonversinya menjadi probabilitas seperti [0,7; 0,2; 0,1]. Probabilitas ini menunjukkan keyakinan model di setiap kelas dan penting untuk penghitungan kesalahan selanjutnya. Berikut rumus Softmax seperti pada persamaan 3.9 (Shrestha et al., 2022):

$$S(\widetilde{y}_{ak}) = \frac{e^{\widetilde{y}_{ak}}}{\sum_{i=1}^K e^{\widetilde{y}_{ai}}} \quad (3.9)$$

Dimana, S adalah keluaran softmax di kelas ke- k , \widetilde{y}_a adalah nilai keluaran dari *fully connected layer*, e adalah bilangan euler sekitar 2,718 dan K adalah total kelas.

3.4.7 Negative Log-Likelihood Loss

Kerugian *Log-Likelihood Negatif* (NLL), juga dikenal sebagai *cross entropy loss*, digunakan untuk mengukur perbedaan antara probabilitas yang diprediksi dan label kelas sebenarnya. Tidak seperti *cross entropy loss*, yang menghitung

probabilitas softmax secara terpisah dari perhitungan kerugian, NLL mengintegrasikan fungsi softmax secara langsung ke dalam perhitungan kerugian. Fungsi kerugian ini sangat cocok untuk tugas klasifikasi karena memberikan penalti terhadap prediksi yang salah berdasarkan keyakinan model. Untuk label kelas sebenarnya tertentu, kerugian NLL dihitung sebagai logaritma negatif dari prediksi probabilitas untuk kelas tersebut. Misalnya, jika label sebenarnya sesuai dengan kelas A dan probabilitas prediksi untuk kelas A adalah 0,7, kerugian NLL dihitung sebagai $-\log(0,7)$. Ukuran ini memberikan nilai skalar yang mewakili kesalahan, yang diminimalkan selama pelatihan untuk meningkatkan akurasi model.

Berikut rumus NLL pada persamaan 3.10 (Shrestha et al., 2022):

$$E = - \hat{y} \log \widetilde{y}_{ak} \quad (3.10)$$

Dimana \hat{y} adalah label sebenarnya (bernilai 1 untuk benar dan 0 jika salah) dan \widetilde{y}_{ak} adalah probabilitas prediksi model untuk kelas ke-k. Disini, keluaran NLL dapat dikatakan prediksi probabilitasnya lebih baik apabila nilai keluarannya lebih rendah. Dan penerapan algoritma NLL dapat dilihat pada Lampiran 6.

3.4.8 Backpropagation dan Perhitungan Gradien

Backpropagation atau propagasi mundur adalah proses yang digunakan untuk memperbarui parameter model (bobot dan bias) berdasarkan kerugian yang dihitung. Setelah kerugian NLL ditentukan, gradien fungsi kerugian terhadap setiap parameter dihitung. Proses ini melibatkan penerapan aturan rantai kalkulus untuk menyebarkan gradien kesalahan ke belakang melalui jaringan. Gradien menunjukkan seberapa banyak setiap parameter perlu disesuaikan untuk

mengurangi kerugian. Penurunan gradien atau variannya (seperti *Adam* atau *RMSprop*) kemudian digunakan untuk memperbarui parameter yang sesuai. Proses berulang ini berlanjut melalui beberapa periode, memungkinkan model belajar dari data dan meningkatkan performanya.

Penelitian ini akan dilakukan penggunaan optimasi *Adam* untuk mendapatkan bobot dan bias baru. Adam optimization adalah sebuah algoritma yang mengatur laju pembelajaran secara adaptif, yang dirancang untuk efisiensi dalam melatih *deep neural network* (Wardianto et al., 2023). Jadi, ADAM merupakan sebuah teknik yang digunakan untuk memperbaiki dan mempercepat proses pembaruan parameter dalam algoritma optimasi. Berikut rumus optimasi Adam yang tertera pada persamaan 3.11 sampai 3.18 (Bhattacharjee et al., 2024):

a. Untuk menghitung bobot baru

$$m_w = \beta_1 * m_{w-1} + (1 - \beta_1) * \left(\frac{\partial E}{\partial w}\right) \quad (3.11)$$

$$\frac{\partial E}{\partial w} = \delta \cdot X \quad (3.12)$$

$$v_w = \beta_2 * v_{w-1} + (1 - \beta_2) * \left(\frac{\partial E}{\partial w}\right)^2 \quad (3.13)$$

$$w = w - (\alpha * m_w / \sqrt{(v_w + \epsilon)}) \quad (3.14)$$

b. Untuk menghitung bias baru

$$m_b = \beta_1 * m_{b-1} + (1 - \beta_1) * \left(\frac{\partial E}{\partial b}\right) \quad (3.15)$$

$$\frac{\partial E}{\partial b} = \delta \quad (3.16)$$

$$v_b = \beta_2 * v_{b-1} + (1 - \beta_2) * \left(\frac{\partial E}{\partial b}\right)^2 \quad (3.17)$$

$$b = b - (\alpha * m_b / \sqrt{(v_b + \epsilon)}) \quad (3.18)$$

Keterangan:

m_w dan m_b = momentum term pada bobot dan bias

v_w dan v_b = velocity term pada bobot dan bias

β_1 = parameter untuk momen pertama (biasanya 0,9)

β_2 = parameter untuk momen kedua (biasanya 0,999)

$\frac{\partial E}{\partial w}$ = gradien loss function terhadap bobot

$\frac{\partial E}{\partial b}$ = gradien loss function terhadap bias

m_{w-1} = momen bobot pada waktu sebelumnya

m_{b-1} = momen bias pada waktu sebelumnya

v_{w-1} = velocity bobot pada waktu sebelumnya

v_{b-1} = velocity bias pada waktu sebelumnya

w = parameter bobot yang diperbarui

b = parameter bias yang diperbarui

α = laju pembelajaran (learning rate)

ϵ = konstanta kecil untuk mencegah pembagian dengan nol (biasanya 10^{-7})

δ = loss untuk neuron keluaran

x = input yang terkait dengan bobot tersebut dalam jaringan neural. Jika berada di *hidden layer* bisa output dari ReLU. Namun, jika di *output layer* bisa hasil dari lapisan terakhir sebelum softmax.

Momentum merupakan rata-rata eksponensial dari gradien, yang membantu menentukan arah pembaruan parameter. Sementara *velocity* merupakan rata-rata eksponensial dari kuadrat gradien, yang membantu menormalkan ukuran langkah pembaruan, membuatnya lebih stabil.

3.4.9 Evaluasi

Selama pelatihan, proses propagasi maju (termasuk lapisan *Fully Connected* dan fungsi *softmax*), komputasi kerugian, dan propagasi mundur diulangi untuk setiap kumpulan data melalui beberapa periode. Suatu zaman terdiri dari satu lintasan penuh melalui kumpulan data pelatihan. Setelah mencapai *epoch* yang ditentukan dan menyelesaikan proses pelatihan, artinya telah mendapatkan model yang terlatih. Model ini dapat diterapkan untuk memprediksi data yang belum pernah dilihat sebelumnya.

Saat mengevaluasi seberapa baik model klasifikasi bekerja, akurasi merupakan metrik utama. Akurasi menunjukkan seberapa sering prediksi model tersebut benar. Untuk menghitung akurasi, di sini menggunakan rata-rata jumlah prediksi yang benar atas jumlah total prediksi (*Arithmetic Mean*). Rumusnya seperti pada Persamaan 3.19 (Takacs & Bourrat, 2022):

$$\text{Akurasi} = \frac{1}{N} \sum_{i=1}^N I(\text{Prediction}_i = \text{True}_i) \quad (3.19)$$

Di sini, N adalah jumlah total prediksi, dan I adalah fungsi yang mengembalikan 1 jika prediksi tersebut benar dan 0 jika tidak. Dengan merata-ratakan nilai-nilai ini, akan diperoleh persentase yang menunjukkan seberapa sering model tersebut benar.

Selain akurasi, penting untuk mempertimbangkan matrik evaluasi lainnya seperti *precision*, *recall*, *F1-Score* dan *Confusion Matrix*.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.20)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.21)$$

$$\text{F1-Score} = 2 \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (3.22)$$

Precision atau presisi pada Persamaan 3.20 mengukur prediksi positif yang benar dari keseluruhan kelas positif. *Recall* pada Persamaan 3.21 mengukur seberapa banyak data positif yang berhasil diprediksi dari keseluruhan data yang sebenarnya positif (Syahrnita et al., 2023). *F1-Score* pada Persamaan 3.22 mengukur rata-rata dari *recall* dan *precision*. Dan confusion matrix merupakan sebuah tabel yang merangkum kinerja model klasifikasi seperti *True Positive* (TP),

False Positive (FP), *True Negative (TN)* dan *False Negative (FN)*. Tabel Confusion matrix ditampilkan pada Tabel 3.2.

Tabel 3.2 Confusion Matrix

Aktual	Kelas Prediksi			
	<i>Blight (A)</i>	<i>Blast (B)</i>	<i>Tungro (C)</i>	<i>Healthy (D)</i>
<i>Blight (A)</i>	AA	AB	AC	AD
<i>Blast (B)</i>	BA	BB	BC	BD
<i>Tungro (C)</i>	CA	CB	CC	CD
<i>Healthy (D)</i>	DA	DB	DC	DD

Berikut penentuan TP, TN, FP, FN untuk setiap kelas:

a. Kelas “Blight”

- TP: *Blight* benar diprediksi sebagai *blight* (AA)
- FN: Semua element di baris *blight* yang bukan TP (AB, AC, AD)
- FP: Semua element di kolom *blight* yang bukan TP (BA, CA, DA)
- TN: Semua element yang bukan di baris dan kolom *blight* (BB, BC, BD, CB, CC, CD, DB, DC, DD)

b. Kelas “Blast”

- TP: *Blast* benar diprediksi sebagai *blast* (BB)
- FN: Semua element di baris *blast* yang bukan TP (BA, BC, BD)
- FP: Semua element di kolom *blast* yang bukan TP (AB, CB, DB)
- TN: Semua element yang bukan di baris dan kolom *blast* (AA, AC, AD, CA, CC, CD, DA, DC, DD)

c. Kelas “Tungro”

- TP: *Tungro* benar diprediksi sebagai *tungro* (CC)

- FN: Semua element di baris *tungro* yang bukan TP (CA, CB, CD)
- FP: Semua element di kolom *tungro* yang bukan TP (AC, BC, DC)
- TN: Semua element yang bukan di baris dan kolom *tungro* (AA, AB, AD, BA, BB, BD, DA, DB, DD)

d. Kelas “Healthy”

- TP: *Healthy* benar diprediksi sebagai *healthy* (DD)
- FN: Semua element di baris *healthy* yang bukan TP (DA, DB, DC)
- FP: Semua element di kolom *healthy* yang bukan TP (AD, BD, CD)
- TN: Semua element yang bukan di baris dan kolom *healthy* (AA, AB, AC, BA, BB, BC, CA, CB, CC)

3.4.10 Skema Uji Coba

Untuk mengevaluasi kinerja model CNN dalam klasifikasi penyakit padi berbasis citra daun, dilakukan skema uji coba dengan fokus pada pengaturan parameter-parameter pelatihan, khususnya *epoch*, dan *learning rate*. Berikut adalah rincian skema uji coba yang diimplementasikan dapat dilihat pada Tabel 3.3:

Tabel 3.3 Skema Uji Coba

Uji Coba	<i>Epoch</i>	<i>Learning Rate</i>
A1	30	0,001
A2	30	0,0001
A3	30	0,0002
A4	30	0,0003
B1	100	0,001

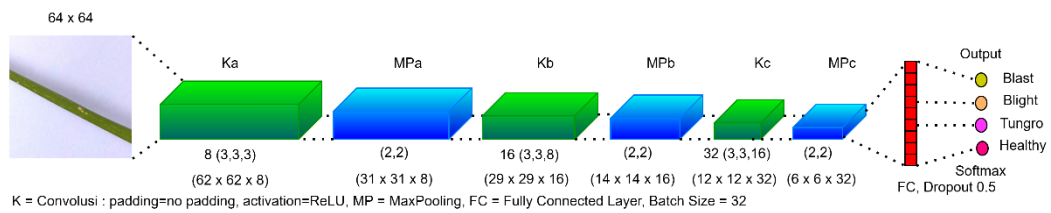
B2	100	0,0001
B3	100	0,0002
B4	100	0,0003

Tahapan ujicoba *epoch* dan *learning rate* dilakukan dengan tujuan untuk mendapatkan hasil yang lebih cepat dan akurasi yang lebih baik. Dalam percobaan yang diuraikan dalam tabel, terdapat jumlah *epoch* yang berkisar antara 30 hingga 200. Memvariasikan jumlah *epoch* ini dapat membantu pemahaman pengaruh *epoch* yang diperbesar terhadap performa model. Pada penelitian ini, variasi jumlah *epoch* yang digunakan adalah 30 dan 100. Pemilihan *epoch* 30 dan 100 didasarkan pada penelitian yang dilakukan oleh Abasi et al., (2023) yang menggunakan jumlah *epoch* serupa pada dataset dengan ukuran di bawah 1000 serta karakteristik data yang mirip. Penelitian tersebut menunjukkan bahwa penggunaan *epoch* 30 dan 100 memberikan performa yang baik, sehingga diadopsi sebagai acuan dalam penelitian ini.

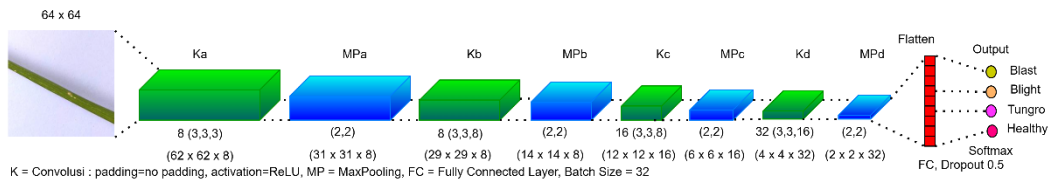
Penggunaan *learning rate* 0,001 dan 0,0001 digunakan untuk menilai apakah nilai tersebut dapat menghasilkan hasil pelatihan yang baik. Sementara, untuk *learning rate* 0,0002 hingga 0,0003 digunakan untuk memberikan pembandingan tambahan bagaimana *learning rate* tersebut berpengaruh terhadap performa model. Penelitian ini memilih awal *learning rate* 0,001 dikarenakan nilai tersebut memiliki nilai yang relatif kecil dan memungkinkan langkah-langkah pembelajaran lebih kecil dan halus, yang memungkinkan membantu model melakukan konvergensi lebih stabil.

Setelah didapatkan kombinasi parameter terbaik, dilakukan pengujian tambahan dengan arsitektur baru dengan menggunakan kombinasi parameter

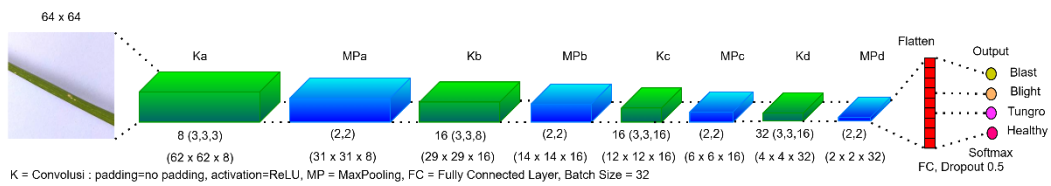
terbaik tersebut. Penambahan arsitektur baru ini bertujuan untuk mengevaluasi apakah dengan memodifikasi arsitektur dapat meningkatkan kinerja model. Dengan adanya arsitektur baru ini akan memberikan pemahaman pengaruh kompleksitas arsitektur terhadap kinerja model. Struktur arsitektur baru yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3.4 hingga Gambar 3.7.



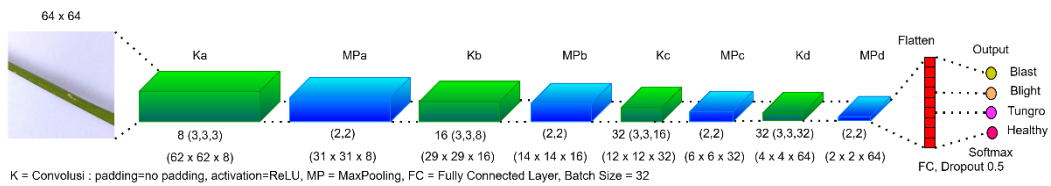
Gambar 3.4 Arsitektur Baru 1: 3 *Hidden Layer* dan Jumlah Filter 8, 16, 32



Gambar 3.5 Arsitektur Baru 2: 4 *Hidden Layer* dan Jumlah Filter 8, 8, 16, 32



Gambar 3.6 Arsitektur Baru 3: 4 *Hidden Layer* dan Jumlah Filter 8, 16, 16, 32



Gambar 3.7 Arsitektur Baru 4: 4 *Hidden Layer* dan Jumlah Filter 8, 16, 32, 32

BAB IV

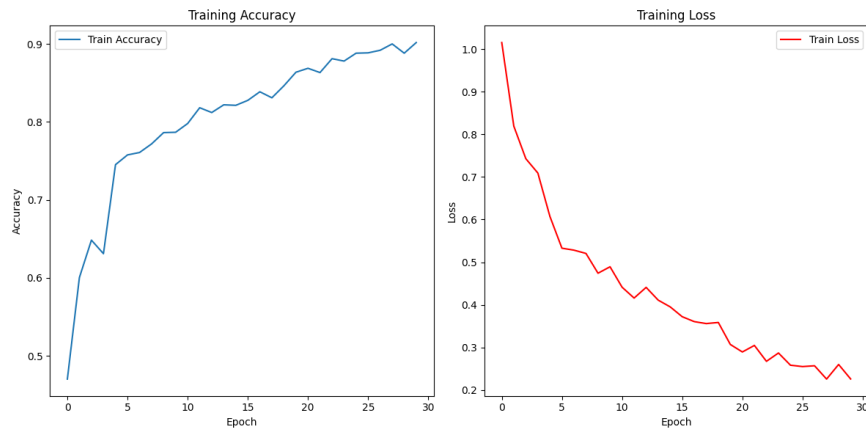
HASIL DAN PEMBAHASAN

4.1 Hasil Uji Coba

Bab ini memuat hasil uji coba serta pembahasan hasil Klasifikasi Penyakit Padi Menggunakan *Convolutional Neural Network* Berbasis Citra Daun.

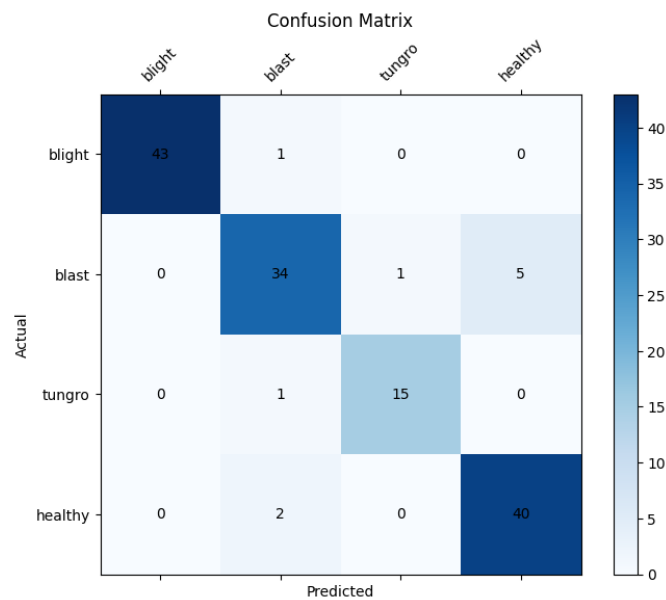
4.1.1 Hasil Uji Coba A1

Pada uji coba A1 dengan *epoch* 30 dan *learning rate* 0,001 didapatkan akurasi *training* sebesar 0,9018, *loss* 0,2258, waktu pelatihan sekitar 112 menit 0,8 detik dan akurasi *testing* sebesar 0,9295. Gambar 4.1 menunjukkan grafik akurasi dan *loss* selama proses pelatihan model. Pada grafik akurasi, terlihat adanya kenaikan nilai akurasi yang cukup signifikan di awal pelatihan, yang menunjukkan bahwa model mulai berhasil mempelajari pola dari data dengan sangat baik. Namun, terdapat sedikit fluktuasi atau ketidakstabilan di beberapa *epoch*, yang kemungkinan terjadi karena adanya data baru yang sebelumnya belum terlihat oleh model, sehingga proses penyesuaian terhadap pola-pola tersebut masih berlangsung. Sementara itu, pada grafik *loss* terlihat bahwa pada awal pelatihan, nilai *loss* mendekati angka 1, yang menandakan bahwa model masih dalam tahap awal menyesuaikan bobot-bobotnya terhadap data latih. Seiring bertambahnya *epoch*, nilai *loss* secara bertahap menurun dan mendekati nol, yang menunjukkan bahwa model semakin mampu menyesuaikan bobot-bobotnya dan mengurangi kesalahan prediksi secara efektif. Kombinasi antara peningkatan akurasi dan penurunan *loss* ini menjadi indikator bahwa proses pelatihan berjalan dengan baik.



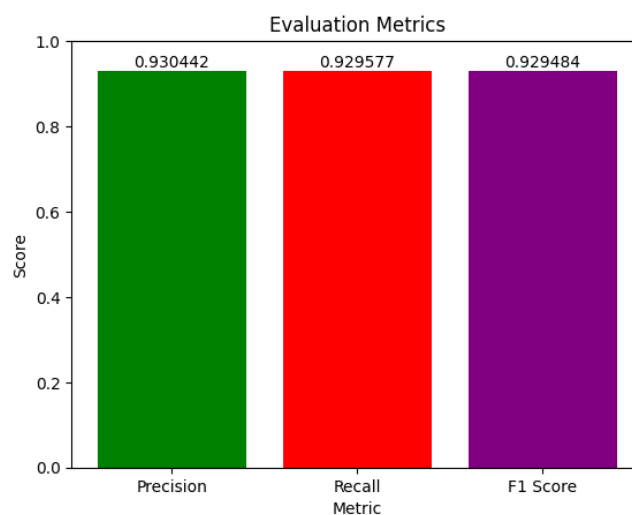
Gambar 4.1 Grafik Akurasi dan *Loss* Uji Coba A1 Selama Pelatihan

Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix pada Gambar 4.2. Terlihat prediksi benar ditunjukkan dengan kolom yang berwarna lebih terang dan berjejer secara diagonal. Sementara sisi kolom yang lain adalah prediksi yang salah, dengan total kesalahan 10 data.



Gambar 4.2 Confusion Matrix A1

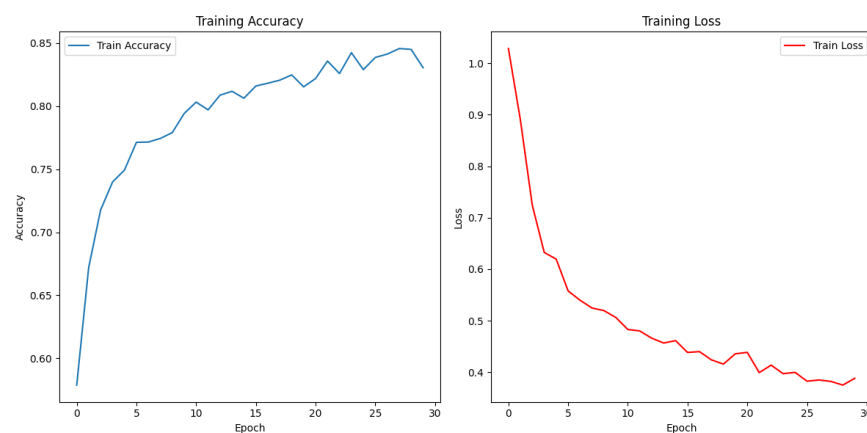
Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision*, *recall*, dan *F1-Score* yang ditampilkan secara visual pada Gambar 4.3. Nilai *precision* yang diperoleh adalah 0,930442, *recall* sebesar 0,929577, dan F1-score sebesar 0,929484. Meskipun ketiga metrik ini memiliki nilai yang tidak identik, perbedaannya sangat kecil dan masih berada dalam kisaran yang konsisten. Hal ini menunjukkan bahwa model memiliki kinerja klasifikasi yang baik dan seimbang, tanpa adanya kecenderungan bias yang signifikan terhadap salah satu metrik. Nilai *precision* yang sedikit lebih tinggi dari *recall* menunjukkan bahwa model cenderung sedikit lebih akurat dalam prediksi positif dibandingkan dalam mendeteksi seluruh data positif, namun perbedaan ini masih dalam batas wajar. Secara keseluruhan, metrik evaluasi ini memperkuat bahwa model mampu mengenali pola data dengan stabil dan dapat diandalkan.



Gambar 4.3 Matriks Evaluasi A1

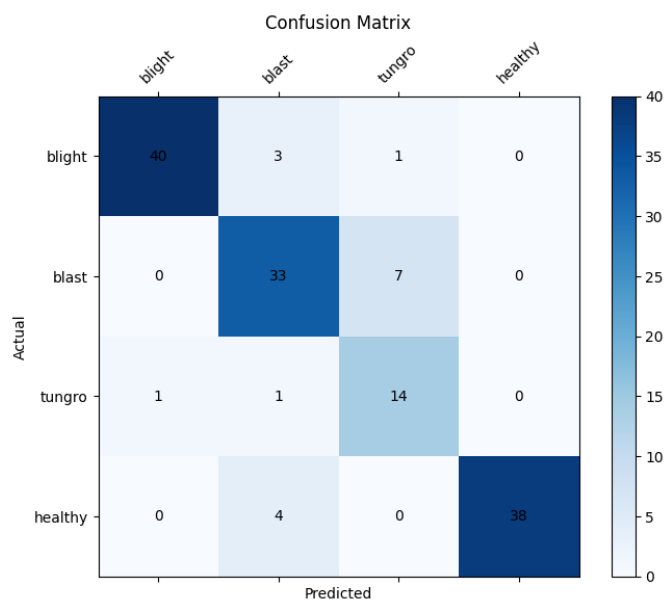
4.1.2 Hasil Uji Coba A2

Uji coba A2 dengan *epoch* 30, *learning rate* 0,0001, memiliki akurasi pelatihan 0,8305, *loss* 0,3880, waktu pelatihan sekitar 128 menit 7 detik, dan akurasi tes 0,8802. Gambar 4.4 menunjukkan grafik akurasi dan *loss* selama proses pelatihan model. Pada grafik akurasi, terlihat bahwa nilai akurasi meningkat cukup tajam di awal pelatihan, menandakan bahwa model mulai mengenali pola data dengan baik. Seiring bertambahnya *epoch*, akurasi terus meningkat meskipun terdapat sedikit fluktuasi, yang bisa terjadi karena model masih menyesuaikan diri terhadap data yang lebih kompleks atau bervariasi. Sementara itu, grafik *loss* menunjukkan penurunan yang konsisten dari awal hingga akhir pelatihan. Pada awal *epoch*, nilai *loss* berada di atas 1, yang berarti model masih melakukan banyak kesalahan dan bobot-bobot belum optimal. Namun seiring pelatihan berlangsung, *loss* menurun hingga mendekati 0,3, yang menandakan bahwa model telah semakin baik dalam meminimalkan kesalahan prediksi. Secara keseluruhan, grafik ini menunjukkan bahwa proses pelatihan model berlangsung secara efektif dengan kecenderungan performa yang terus membaik.



Gambar 4.4 Grafik Akurasi dan *Loss* Uji Coba A2 Selama Pelatihan

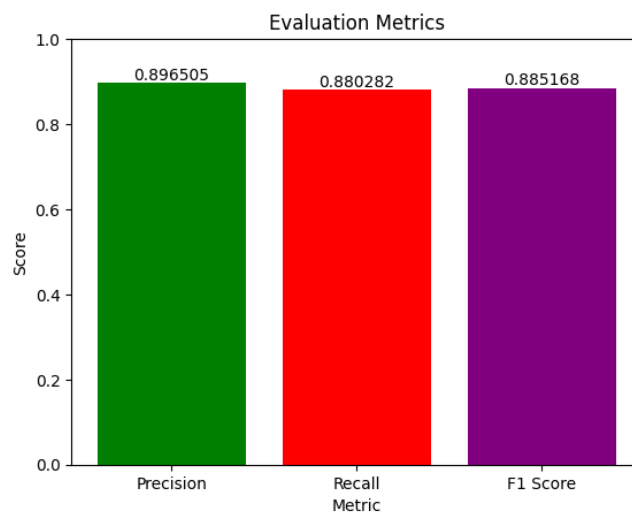
Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix yang ditampilkan dalam Gambar 4.5. Terlihat hasil confusion matrix pada pengujian A2 menunjukkan bahwa jumlah kesalahan prediksi lebih tinggi, yaitu sebanyak 17 data yang diklasifikasikan secara keliru. Jumlah ini lebih besar dibandingkan dengan pengujian A1 yang hanya mengalami 10 kesalahan.



Gambar 4.5 Confusion Matrix A2

Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision*, *recall*, dan *F1-Score* yang ditampilkan secara visual di Gambar 4.6. Hasil evaluasi metrik menunjukkan berupa *precision* sebesar 0,8965, *recall* sebesar 0,8803, dan *F1-score* sebesar 0,8852. Nilai-nilai ini masih tergolong tinggi dan menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengklasifikasikan data. Meskipun demikian, terdapat sedikit penurunan dibandingkan pengujian sebelumnya, yang tercermin dari nilai *recall* dan *F1-score* yang lebih rendah. Hal

ini sejalan dengan hasil confusion matrix, di mana jumlah kesalahan prediksi pada pengujian A2 lebih tinggi dibandingkan A1. Dengan demikian, meskipun model masih tergolong efektif, performanya pada pengujian ini sedikit menurun dan menunjukkan bahwa model tidak sepenuhnya konsisten dalam mengenali seluruh kelas secara seimbang.

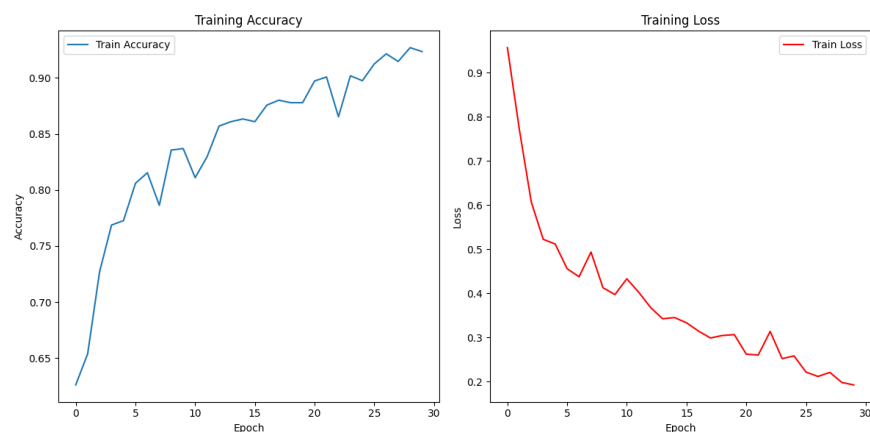


Gambar 4.6 Matrik Evaluasi A2

4.1.3 Hasil Uji Coba A3

Pada uji coba A3 dengan *epoch* 30, *learning rate* 0,0002, didapatkan akurasi pelatihan 0,9234, *loss* 0,1921, dan waktu pelatihan sekitar 112 menit 6,4 detik, dan akurasi tes 0,9436. Berdasarkan Gambar 4.7, terlihat pada grafik akurasi terjadi peningkatan yang tidak terlalu drastis tapi secara bertahap. Hal ini menunjukkan bahwa model memerlukan waktu untuk mulai mengenali pola dari data pelatihan secara efektif. Namun, peningkatan ini tidak selalu berlangsung secara stabil; terdapat beberapa fluktuasi pada titik-titik tertentu yang menunjukkan bahwa akurasi sempat mengalami penurunan kecil sebelum kembali naik. Fluktuasi ini merupakan hal yang wajar dalam proses pelatihan, terutama ketika model sedang

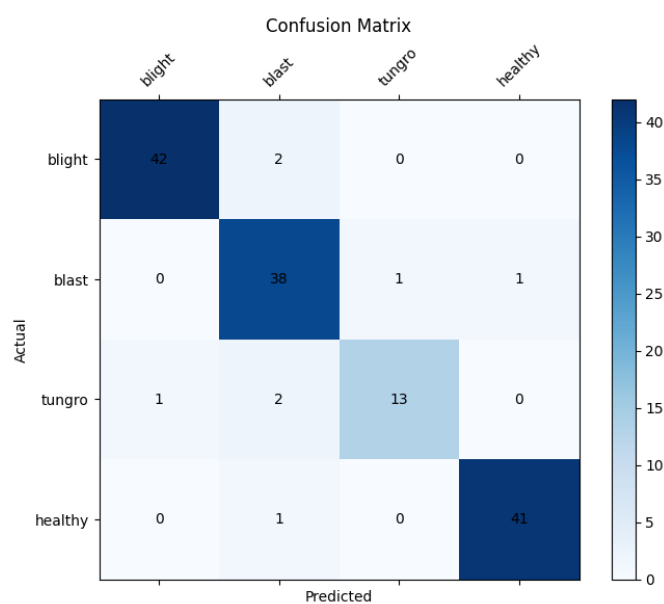
berusaha menyesuaikan bobot untuk mempelajari pola yang kompleks pada data pelatihan. Di sisi lain, grafik *loss* menunjukkan penurunan nilai *loss* yang cukup konsisten, dari nilai awal yang tinggi hingga mendekati 0,2, yang mengindikasikan bahwa kesalahan prediksi terhadap data pelatihan semakin berkurang. Secara keseluruhan, tren peningkatan akurasi dan penurunan *loss* ini menunjukkan bahwa model berhasil mempelajari data pelatihan dengan baik, meskipun mengalami sedikit ketidakstabilan selama proses pelatihan.



Gambar 4.7 Grafik Akurasi dan *Loss* Uji Coba A3 Selama Pelatihan

Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix pada Gambar 4.8. Confusion matrix tersebut, model mampu mengklasifikasikan sebagian besar data dengan benar, ditunjukkan oleh nilai-nilai diagonal yang tinggi. Kelas *blight* diklasifikasikan dengan sangat baik, di mana 42 sampel berhasil dikenali dengan benar, dan hanya 2 sampel yang salah diklasifikasikan sebagai *blast*. Untuk kelas *blast*, terdapat 38 prediksi yang benar, sementara masing-masing satu sampel salah diklasifikasikan sebagai *tungro* dan *healthy*. Sementara itu, kelas *tungro* menunjukkan performa yang kurang optimal dibanding kelas lainnya, dengan hanya 13 sampel yang diklasifikasikan

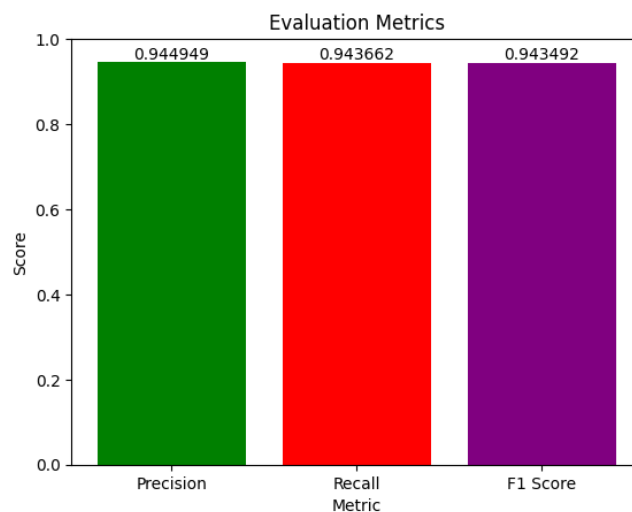
dengan benar, sedangkan 3 lainnya keliru diklasifikasikan sebagai *blight* dan *blast*. Kelas *healthy* juga menunjukkan akurasi tinggi, dengan 41 prediksi yang benar dan hanya satu sampel yang salah diklasifikasikan sebagai *blast*. Secara keseluruhan, confusion matrix ini memperlihatkan bahwa model memiliki kinerja klasifikasi yang baik dari pada uji coba sebelumnya (A1 dan A2), namun seharusnya pada data *healthy* model dapat memprediksi semua datanya.



Gambar 4. 8 Confusion Matrix A2

Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision*, *recall*, dan *F1-Score* yang ditampilkan secara visual di Gambar 4.9. Terlihat nilai evaluasi model pada *precision* diperoleh sebesar 0,9449, yang menunjukkan bahwa model memiliki tingkat ketepatan yang tinggi dalam memprediksi label yang benar dari seluruh prediksi positif yang dibuat. Sementara itu, nilai *recall* sebesar 0,9437 menunjukkan bahwa model mampu mengidentifikasi sebagian besar data positif secara akurat dari keseluruhan data positif yang sebenarnya ada. *F1-score*, yang

merupakan rata-rata harmonis dari *precision* dan *recall*, juga menunjukkan nilai yang tinggi, yaitu sebesar 0,9435. Ketiga metrik tersebut memiliki nilai yang sangat berdekatan, yang mengindikasikan bahwa model memiliki kinerja yang seimbang dalam hal ketepatan dan sensitivitas. Secara umum, hasil evaluasi ini menunjukkan bahwa model bekerja dengan baik dan tidak menunjukkan adanya bias yang signifikan terhadap salah satu metrik. Namun demikian, jika mengacu pada confusion matrix sebelumnya, masih terdapat kesalahan pada kelas tertentu seperti *tungro*, sehingga dapat disimpulkan bahwa meskipun performa rata-rata model tinggi, masih diperlukan evaluasi lebih lanjut terhadap performa tiap kelas secara spesifik.

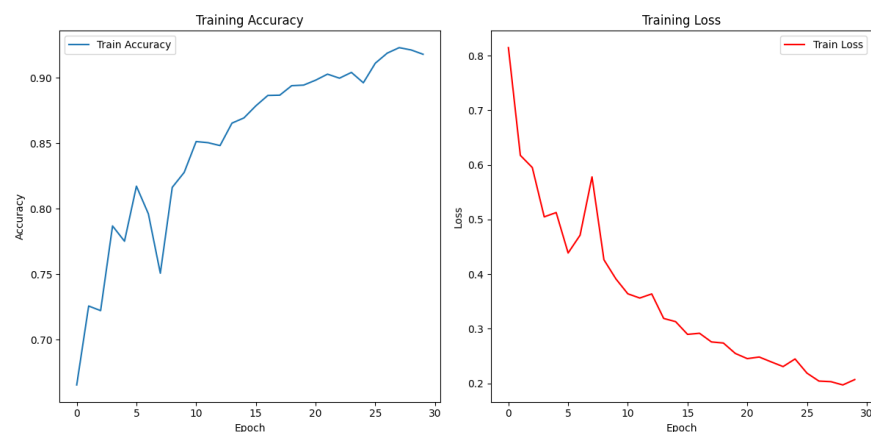


Gambar 4.9 Matrik Evaluasi A3

4.1.4 Hasil Uji Coba A4

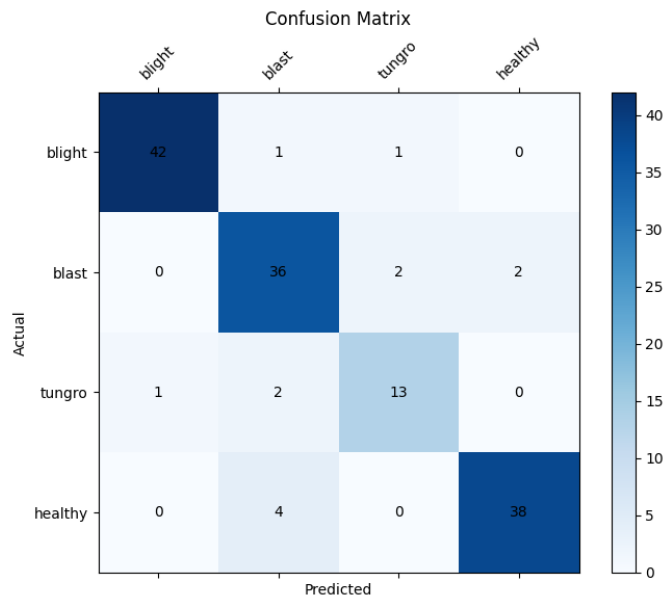
Pada uji coba A4, dengan *epoch* 30 dan *learning rate* 0,0003, didapatkan akurasi pelatihan 0,9179, *loss* 0,2069, waktu pelatihan sekitar 112 menit 3,7 detik, dan akurasi tes 0,9084. Gambar 4.10 menunjukkan grafik akurasi dan *loss* pada saat proses pelatihan uji coba A4. Terlihat bahwa nilai akurasi terus meningkat hingga

mencapai 91%, yang menandakan bahwa model berhasil mempelajari pola-pola dari data pelatihan dengan baik. Meski demikian, terdapat fluktuasi atau ketidakstabilan kecil pada grafik akurasi, terutama pada pertengahan *epoch*, yang bisa diartikan bahwa model sedang menyesuaikan bobot terhadap data yang baru terlihat. Sementara itu, grafik training *loss* secara konsisten menurun hingga mendekati nilai 0,2, menandakan bahwa model berhasil meminimalkan kesalahan prediksi selama pelatihan.



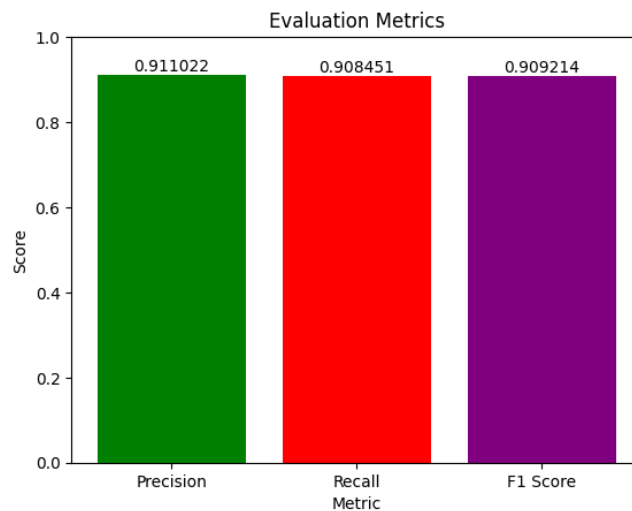
Gambar 4.10 Grafik Akurasi dan *Loss* Uji Coba A4 Selama Pelatihan

Gambar 4.11 adalah penjabaran untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix. Meskipun performa klasifikasi secara umum sudah cukup baik, jumlah kesalahan prediksi pada A4 lebih tinggi dibandingkan dengan A3. Hal ini terlihat dari sebaran nilai prediksi yang keliru, terutama pada kelas *healthy* yang salah diprediksi sebagai *blast* sebanyak 4 kali, serta kelas *tungro* yang memiliki 3 kesalahan. Meskipun nilai diagonal (prediksi benar) masih dominan, peningkatan jumlah kesalahan ini menunjukkan bahwa penyesuaian parameter (epoch dan learning rate) pada A4 tidak mampu mengungguli model pada A3 dari segi akurasi klasifikasi pada data uji.



Gambar 4.11 Confusion Matrix A4

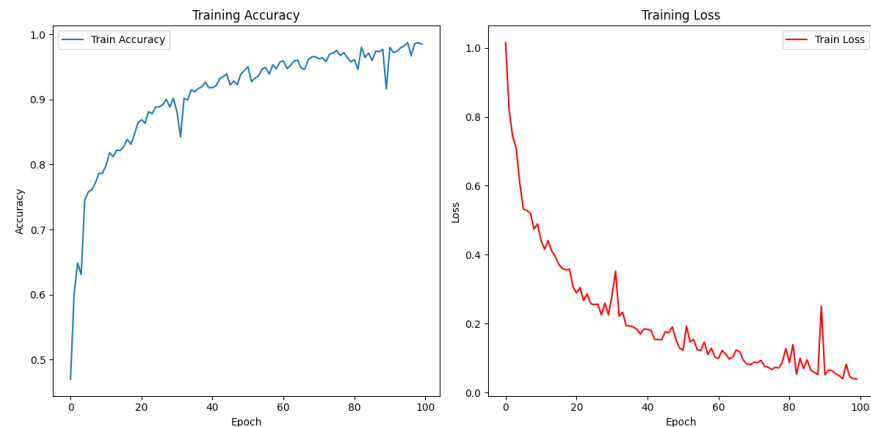
Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision*, *recall*, dan F1-Score yang ditampilkan secara visual pada Gambar 4.12. Terlihat nilai *precision* sebesar 0,911, *recall* sebesar 0,908, dan *F1-score* sebesar 0,909. Nilai-nilai ini tergolong tinggi dan seimbang, yang mengindikasikan bahwa model memiliki performa klasifikasi yang stabil serta tidak menunjukkan bias yang signifikan terhadap salah satu kelas. Selisih antar metrik sangat kecil, memperkuat bahwa model memiliki keseimbangan antara ketepatan dan sensitivitas dalam melakukan prediksi.



Gambar 4.12 Matrik Evaluasi A4

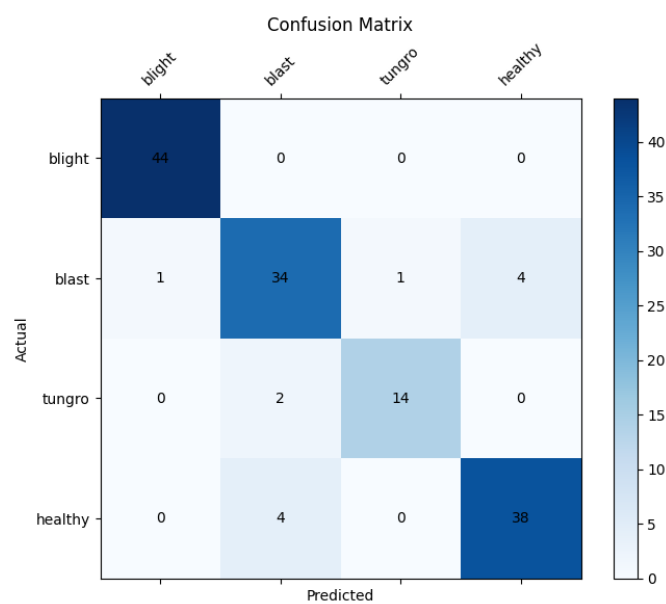
4.1.5 Hasil Uji Coba B1

Pada uji coba B1 dengan *epoch* 100 dan *learning rate* 0,001 didapatkan akurasi *training* sebesar 0,9850, *loss* 0,0393, waktu pelatihan sekitar 377 menit 24,3 detik dan akurasi *testing* sebesar 0,9154. Terlihat pada Gambar 4.13 akurasi mengalami progres pelatihan yang sangat baik ditandai dengan akurasi yang terus meningkat hingga akhir *epoch*. Ini menunjukkan bahwa model belajar dengan baik dan mampu mengenali pola dalam data pelatihan. Meskipun terjadi fluktuasi kecil yang mungkin disebabkan oleh *noise* pada data atau *learning rate*, tetapi secara umum model berhasil belajar dengan efektif. Sementara pada grafik *loss*, terjadi penurunan *loss* secara konsisten hingga di akhir *epoch*, *loss* mencapai nilai yang sangat rendah, menunjukkan konvergensi model.



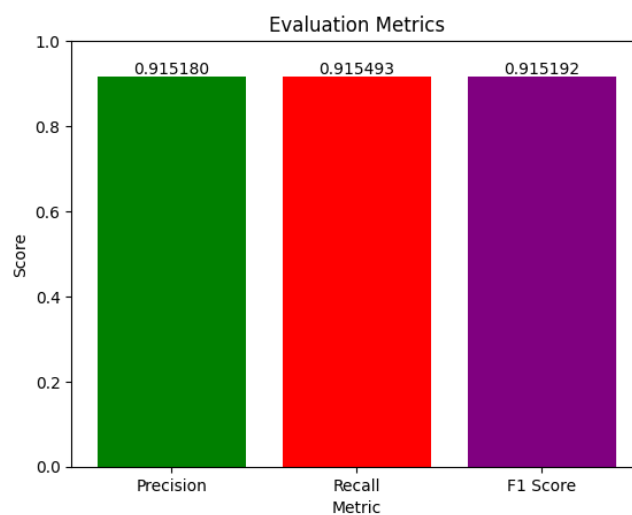
Gambar 4.13 Grafik Akurasi dan *Loss* Uji Coba B1 Selama Pelatihan

Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix pada Gambar 4.14. Terlihat total kesalahan prediksi pada uji coba B1 masih jauh lebih tinggi dibandingkan uji coba A3. Meskipun pada akurasi *training* uji B1 lebih unggul, akan tetapi pada data *testing* mengalami penurunan yang menyebabkan *overfitting*. Maka, dari hasil ini dapat dikatakan bahwa A3 lebih layak dipilih dibandingkan B1, karena kinerja pada data *testing* lebih akurat.



Gambar 4.14 Confusion Matrix B1

Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision* sebesar 0,915180, *recall* sebesar 0,915493, dan *F1-Score* sebesar 0,915192 yang ditampilkan secara visual pada Gambar 4.15. Ketiga metrik ini cukup tinggi dan seimbang, menunjukkan bahwa model B1 memiliki performa yang stabil dan tidak terlalu bias terhadap kelas tertentu. Meskipun demikian, hasil *evaluation metrics* yang didapatkan juga belum lebih baik dibandingkan uji coba A3.

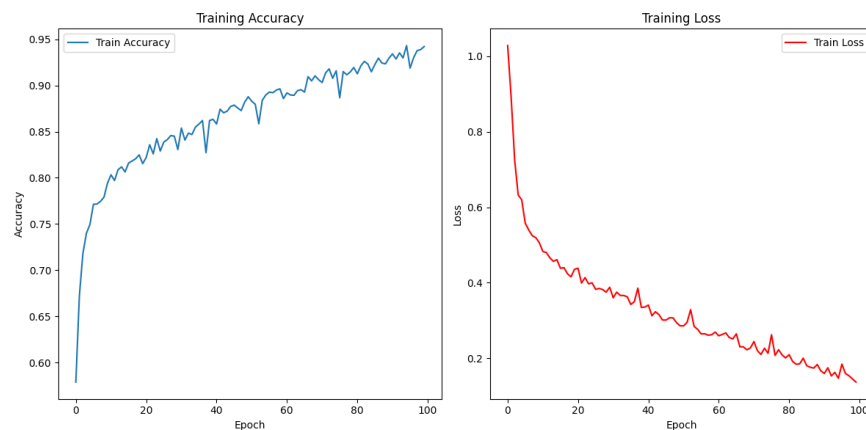


Gambar 4.15 Matrik Evaluasi

4.1.6 Hasil Uji Coba B2

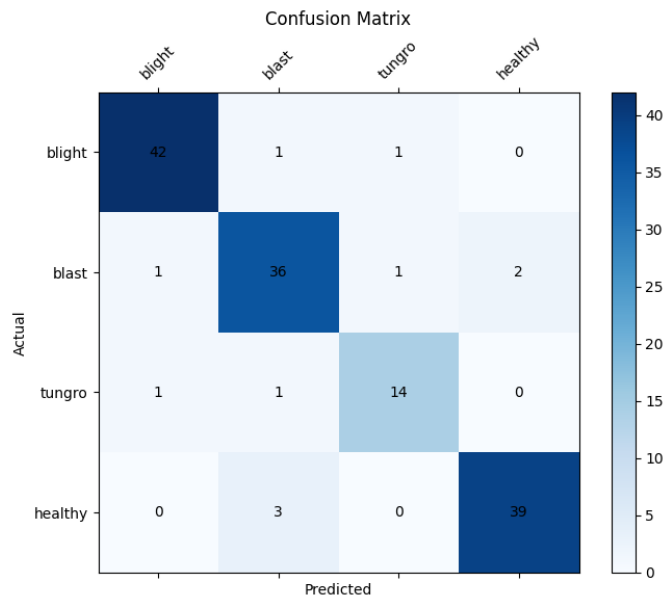
Uji coba B2 dengan *epoch* 100, *learning rate* 0,0001, memiliki akurasi pelatihan 0,9421, *loss* 0,1366, waktu pelatihan sekitar 364 menit 8,7 detik, dan akurasi tes 0,9225. Grafik akurasi dan *loss* pada saat proses pelatihan uji coba B2 ditunjukkan pada Gambar 4.16. Pada grafik akurasi terlihat dalam beberapa *epoch* pertama terjadi peningkatan akurasi yang sangat tinggi, yang menandakan bahwa model dengan cepat mempelajari pola dasar dari data latih. Seiring bertambahnya *epoch*, akurasi terus meningkat hingga menyentuh sekitar 0,94 pada data pelatihan,

sementara *loss* menurun secara konsisten hingga mendekati 0,1. Pola ini menunjukkan proses pelatihan yang stabil dan konvergen tanpa indikasi *overfitting* yang mencolok, meskipun akurasi tidak setinggi model B1.



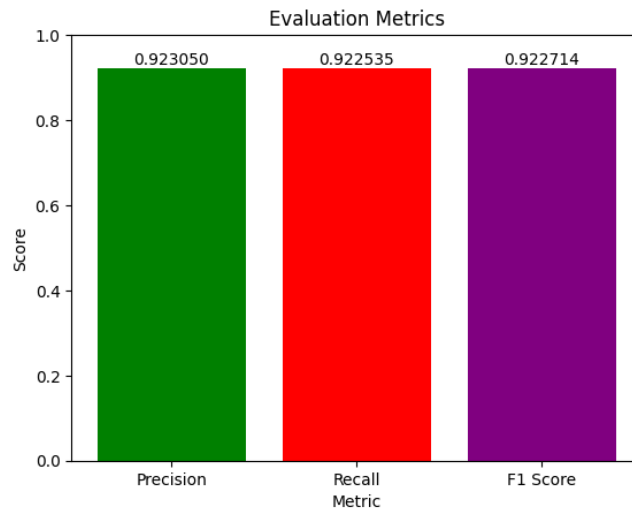
Gambar 4.16 Grafik Akurasi dan *Loss* Uji Coba B2 Selama Pelatihan

Gambar 4.17 adalah confusion matrix untuk mengetahui gambaran secara detail kesalahan prediksi. Secara umum, model berhasil mengklasifikasikan sebagian besar sampel dengan benar, terutama pada kelas *blight* (42 benar dari 44), *blast* (36 benar dari 40), dan *healthy* (39 benar dari 42). Namun, masih terdapat beberapa kesalahan klasifikasi yang cukup signifikan, khususnya pada kelas *healthy* yang diprediksi sebagai *blast* sebanyak 3 kali dan *blast* yang keliru diklasifikasikan sebagai *healthy* sebanyak 2 kali. Jumlah total kesalahan pada uji coba B2 ini lebih sedikit tinggi dibandingkan uji coba A3 yang lebih akurat dalam memprediksi label asli.



Gambar 4.17 Confusion Matrix B2

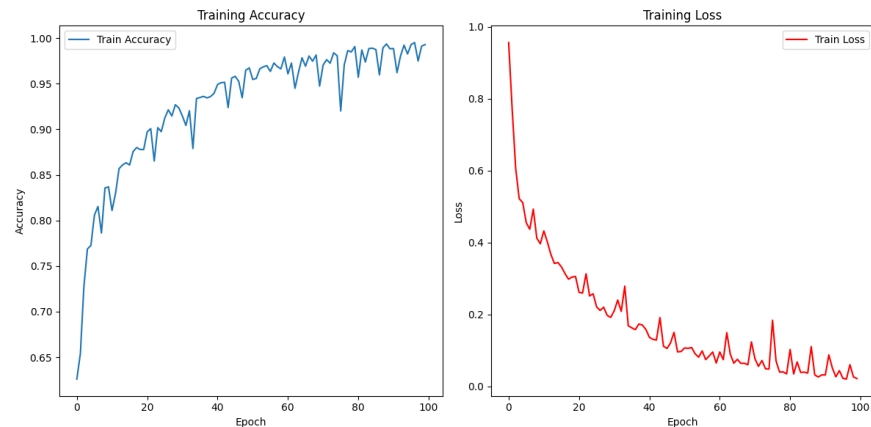
Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision* sebesar 0,923050, *recall* sebesar 0,922535, dan *F1-Score* sebesar 0,922714 yang ditampilkan secara visual pada Gambar 4.18. Meskipun terdapat sedikit perbedaan antar metrik, nilainya tetap konsisten sehingga mencerminkan kinerja model yang baik dan seimbang tanpa menunjukkan bias yang signifikan terhadap metrik tertentu.



Gambar 4.18 Matrik Evaluasi B2

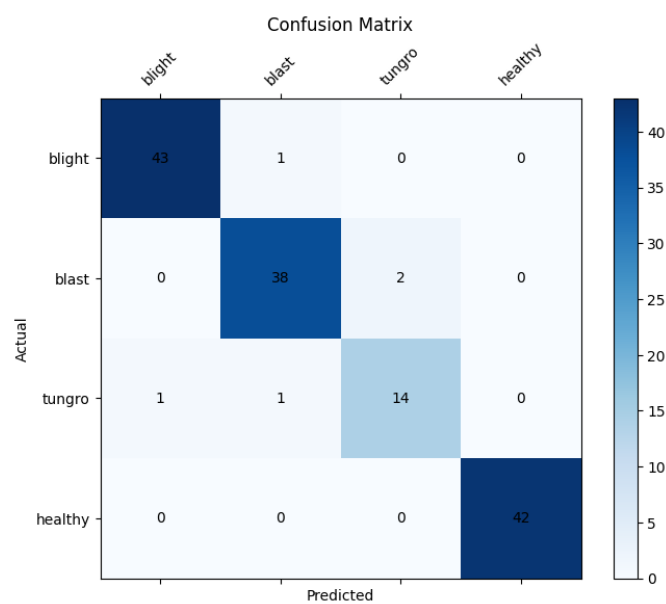
4.1.7 Hasil Uji Coba B3

Pada uji coba B3 dengan *epoch* 100, *learning rate* 0,0002, didapatkan akurasi pelatihan 0,9930, *loss* 0,0221, dan waktu pelatihan sekitar 363 menit 10,8 detik, dan akurasi tes 0,9647. Gambar 4.19 menunjukkan grafik akurasi dan loss selama proses pelatihan uji coba B3, yang menunjukkan peningkatan akurasi yang cukup baik selama pelatihan walaupun ada sedikit terjadi fluktuasi di beberapa *epoch*. Sementara nilai *loss* menunjukkan penurunan yang sangat mendekati nilai 0, artinya model mampu memprediksi data latih dengan tingkat kesalahan yang sangat kecil.



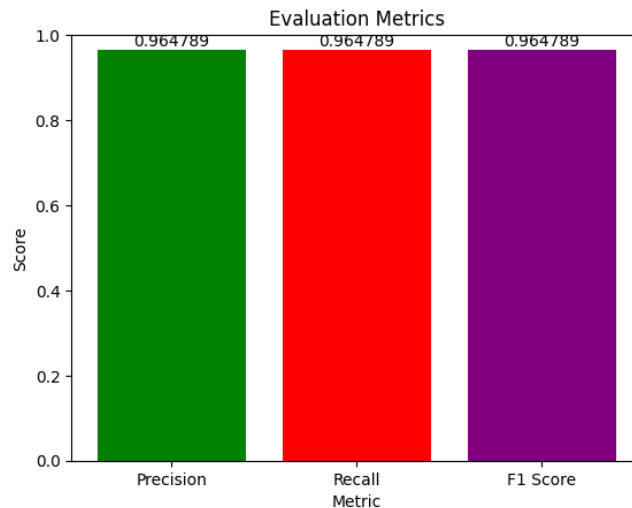
Gambar 4.19 Grafik Akurasi dan *Loss* Uji Coba B3 Selama Pelatihan

Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix pada Gambar 4.20. Secara keseluruhan model mengalami kesalahan prediksi sebanyak 5 data dari seluruh data uji. Jika dibandingkan dengan uji coba sebelumnya, ini merupakan uji coba yang jumlah kesalahan prediksinya paling sedikit. Artinya, model ini mampu mengenali pola dari citra penyakit tanaman padi dengan lebih efektif dan generalisasi terhadap data uji lebih baik.



Gambar 4.20 Confusion Matrix B3

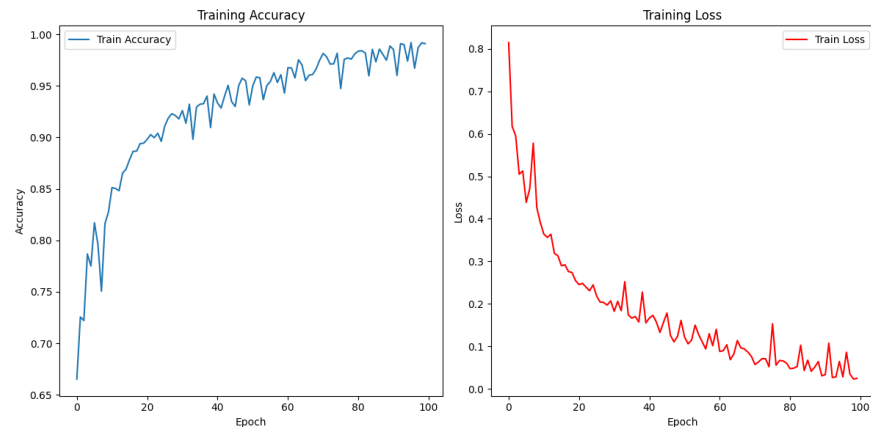
Sementara untuk evaluasi matrix pada Gambar 4.21, terlihat juga nilai dari tiga matrik memiliki nilai yang identik tanpa perbedaan sedikit pun dan nilainya tertinggi dibandingkan uji coba yang lain.



Gambar 4.21 Matrik Evaluasi B3

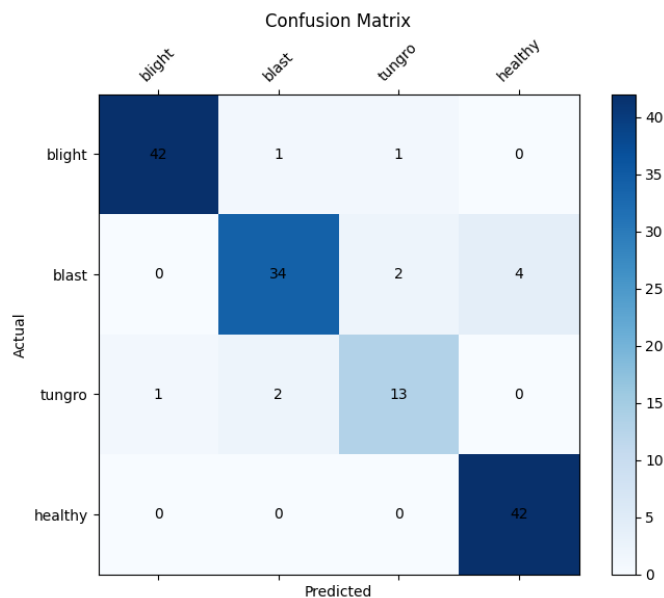
4.1.8 Hasil Uji Coba B4

Pada uji coba B4, dengan *epoch* 100 dan *learning rate* 0,0003, didapatkan akurasi pelatihan 0,9910, *loss* 0,0246, waktu pelatihan sekitar 359 menit 23,2, dan akurasi tes 0,9225. Merujuk pada Gambar 4.22, grafik akurasi menunjukkan peningkatan secara konsisten walaupun juga ada beberapa ketidakstabilan. Namun di akhir *epoch*, model berhasil mendapatkan akurasi yang tinggi. Dan grafik *loss*, menunjukkan penurunan yang sangat mendekati 0 akan tetapi tidak lebih kecil dibandingkan uji coba B3.



Gambar 4.22 Grafik Akurasi dan *Loss* Uji Coba B4 Selama Pelatihan

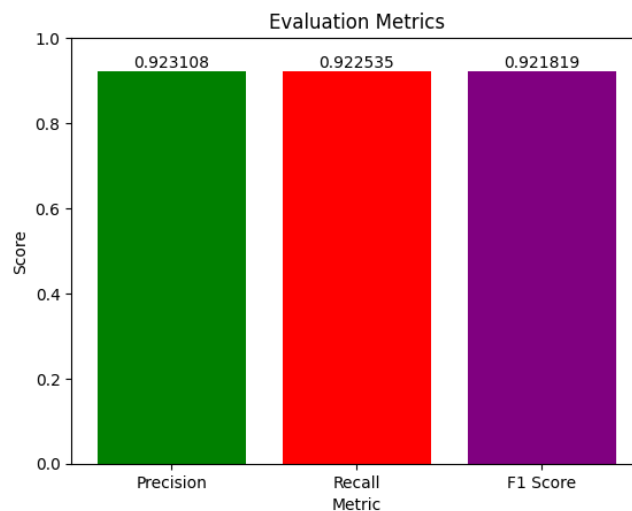
Dan untuk mengetahui gambaran secara detail kesalahan prediksi dijabarkan dalam confusion matrix pada Gambar 4.23. Dengan kesalahan prediksi yang tergolong rendah dengan total kesalahan sebanyak 11, meskipun tidak serendah yang ditunjukkan oleh uji coba B3 yang hanya memiliki 5 kesalahan.



Gambar 4.23 Confusion Matrix B4

Berdasarkan persamaan 3.20, 3.21, dan 3.22 didapatkan nilai *precision*, *recall*, dan *F1-Score* yang ditampilkan secara visual pada Gambar 4.24. Meskipun

ketiga metrik tersebut tidak memiliki nilai yang identik, hasil yang ditunjukkan berada pada kisaran yang serupa dan seimbang, sehingga menggambarkan performa model yang stabil tanpa kecenderungan bias terhadap salah satu metrik.



Gambar 4.24 Matrik Evaluasi B4

4.1.9 Hasil Uji Coba Keseluruhan

Berdasarkan hasil uji coba yang telah dilakukan, untuk mempermudah hasil perbandingan kinerja dari setiap uji coba disajikan dalam Tabel 4.1.

Tabel 4.1 Hasil Uji Coba Keseluruhan

Uji Coba	Epoch	Learning Rate	Akurasi Training	Loss	Durasi Training	Akurasi Testing
A1	30	0,001	0,9018	0,2258	112 menit 0,8 detik	0,929577
A2	30	0,0001	0,8305	0,3880	128 menit 7,8 detik	0,880282
A3	30	0,0002	0,9234	0,1921	112 menit 6,4 detik	0,943662
A4	30	0,0003	0,9179	0,2069	112 menit 3,7 detik	0,908451
B1	100	0,001	0,9850	0,0393	377 menit 24,3 detik	0,915493
B2	100	0,0001	0,9421	0,1366	364 menit 8,7 detik	0,922535
B3	100	0,0002	0,9930	0,0221	363 menit 10,8 detik	0,964789
B4	100	0,0003	0,9910	0,0246	359 menit 23,2 detik	0,922535

Terlihat pada Tabel 4.1, hasil menunjukkan bahwa model yang dilatih dengan 100 *epoch* cenderung menghasilkan akurasi dan *loss* yang lebih baik dibandingkan model yang dilatih dengan 30 *epoch*. Yang dimungkinkan oleh

semakin banyak *epoch*, model memiliki waktu yang lebih lama untuk mempelajari data sehingga mampu mendapatkan *loss* yang lebih rendah.

Pada uji coba A1 hingga A4 dan B1 hingga B4, model mampu mencapai akurasi tertinggi dengan menggunakan *learning rate* 0,0002. Ini menunjukkan *learning rate* yang lebih kecil mampu membantu model belajar lebih stabil dan mendapatkan *loss* yang lebih rendah. Walaupun pada *learning rate* yang lebih kecil dari 0,0002 yaitu 0,0001 belum mampu menghasilkan performa yang lebih baik, ini mungkin disebabkan oleh *learning rate* yang terlalu kecil sehingga terlalu lambat dan menghambat model dalam melakukan pembelajaran.

Jadi performa terbaik diperoleh pada uji coba B3 dengan menggunakan *epoch* 100 dan *learning rate* 0,0002 yang menghasilkan akurasi *training* 0,9930, *loss* 0,0221 dan akurasi *testing* 0,9647. Variasi *epoch* dan *learning rate* secara signifikan memengaruhi kinerja model, yang menunjukkan pentingnya penyetelan parameter dalam mengoptimalkan model CNN untuk klasifikasi penyakit padi.

4.1.10 Hasil Penggunaan Parameter Terbaik pada Arsitektur Baru

Setelah didapatkan kombinasi parameter terbaik, parameter tersebut diterapkan pada arsitektur baru untuk mengamati dampaknya terhadap performa model. Pada uji coba arsitektur baru dengan penggunaan 3 *hidden layer* dan jumlah filter 8, 16, 32 didapatkan hasil akurasi pada data latih sekitar 0,9765 dengan nilai *loss* 0,0613 dan durasi sekitar 336 menit 33 detik. Sementara akurasi pada data uji mencapai 0,9295, hal ini menunjukkan adanya *overfitting* dan penurunan performa pada arsitektur model. Hal ini terjadi karena jumlah *hidden layer* dan jumlah filter yang dikurangi menyebabkan model menjadi lebih sederhana dan memiliki sedikit

kemampuan untuk mengenali pola dalam data. Mengurangi *hidden layer* berarti model kehilangan beberapa tahap pemrosesan. Dan mengurangi jumlah filter berarti model hanya mampu menangkap sedikit fitur dari gambar, sehingga informasi penting bisa terlewat.

Lalu, pada uji coba arsitektur baru dengan penggunaan 4 *hidden layer* dan jumlah filter 8, 8, 16, 32 diperoleh akurasi data latih 0,9208 dan *loss* 0,1889. Durasi yang didapatkan relatif lebih singkat dibandingkan dengan arsitektur usulan dan arsitektur (8, 16, 32) yaitu 265 menit 51,8 detik. Hal ini dimungkinkan karena jumlah filter yang lebih kecil di *hidden layer* kedua. Dan untuk akurasi data uji diperoleh nilai yang cukup tinggi yaitu sekitar 0,9577. Ini menunjukkan bahwa model ini memiliki kemampuan generalisasi yang cukup baik, meskipun kapasitas belajarnya lebih terbatas dibandingkan arsitektur usulan.

Selanjutnya, uji coba arsitektur baru dengan penggunaan 4 *hidden layer* dan jumlah filter 8, 16, 16, 32 diperoleh akurasi data latih 0,9496, *loss* 0,1227, durasi pelatihan 305 menit 31,1 detik dan akurasi data uji 0,9436. Walaupun durasi pelatihan lebih singkat dibandingkan dengan arsitektur usulan, model tetap mengalami penurunan performa. Hal ini dikarenakan dengan jumlah filter yang lebih sedikit, tidak cukup untuk memahami pola dalam data dengan lebih baik.

Dan untuk uji coba arsitektur baru dengan penggunaan 4 *hidden layer* dan jumlah filter 8, 16, 32, 32 menghasilkan akurasi data latih 0,9492 dan *loss* 0,1218. Durasi pelatihan menjadi lebih lama dibandingkan arsitektur (8, 16, 16, 32), yaitu 352 menit 58,4 detik, meskipun masih lebih singkat dibandingkan arsitektur usulan. Namun, akurasi pada data uji menunjukkan adanya penurunan dengan mencapai

0,9154, yang berarti arsitektur model ini kurang generalisasi dibandingkan arsitektur lainnya. Sehingga arsitektur ini dapat dikatakan kurang optimal dikarenakan adanya durasi yang lebih lama tidak diikuti dengan peningkatan akurasi pada data uji.

Berdasarkan hasil uji coba arsitektur baru, dapat dipertimbangkan bahwa arsitektur usulan tetap menjadi pilihan performa terbaik. Hal ini dimungkinkan karena jumlah *hidden layer* yang lebih kompleks dapat membantu model menangkap pola dengan lebih baik. Selain itu juga, jumlah filter dengan tingkatan secara progresif atau bertahap cenderung lebih efektif memahami pola data dibandingkan arsitektur dengan jumlah filternya yang stagnan.

4.2 Pembahasan

Dalam penelitian ini yang berfokus pada klasifikasi penyakit padi, metrik *recall* memiliki peran yang lebih penting dibandingkan *precision* dan *f1-score*. Hal ini disebabkan oleh fungsi *recall* yang mengukur seberapa besar proporsi data positif yang berhasil dikenali dengan benar oleh model dari seluruh data positif yang tersedia. Dengan kata lain, *recall* berperan untuk memastikan bahwa model mampu mengidentifikasi seluruh daun padi yang terinfeksi penyakit.

Dalam konteks pertanian, kesalahan dalam mendeteksi daun yang benar-benar terinfeksi penyakit memiliki dampak yang lebih serius dibandingkan kesalahan dalam mengklasifikasikan daun sehat sebagai terinfeksi. Oleh sebab itu, *recall* menjadi prioritas utama guna mengurangi risiko daun yang berpenyakit tidak terdeteksi oleh sistem.

Berdasarkan hasil terbaik pada pengujian, nilai *precision*, *recall*, dan *f1-score* yang diperoleh, sebagaimana ditunjukkan pada Gambar 4.21, memiliki nilai yang sama. Hal ini menandakan bahwa model memiliki kemampuan klasifikasi yang sangat baik, serta tidak menunjukkan bias terhadap kelas tertentu. Nilai *precision* yang tinggi mengindikasikan bahwa model jarang salah dalam mengidentifikasi daun sehat sebagai daun yang sakit. Nilai *recall* yang tinggi menunjukkan kemampuan model dalam mengenali sebagian besar daun yang benar-benar terkena penyakit. Sementara itu, *f1-score* yang tinggi menunjukkan keseimbangan kinerja model dalam menghindari kesalahan prediksi baik untuk daun sehat maupun daun yang terinfeksi. Kesamaan nilai dari keempat metrik tersebut yaitu akurasi, *precision*, *recall*, dan *f1-score* menunjukkan bahwa model memiliki performa yang optimal dalam mengklasifikasikan seluruh kelas data.

Beberapa penelitian sebelumnya dapat dijadikan pembandingan. Pertama, Inception-ResNet-V2 yang menggunakan pendekatan *transfer learning* dengan model pra-latih dan penyesuaian pada lapisan akhir, menunjukkan hasil akurasi sebesar 92,68% (Islam et al., 2021). Kedua, VGG-16 dengan *transfer learning* dan bobot dari ImageNet menghasilkan akurasi sebesar 92,46% (Ghosal & Sarkar, 2020). Ketiga, VGG-19 sebagai pengembangan dari VGG-16 mampu mencapai akurasi 95,24% (Priyanka & Kumara, 2021). Keempat, model Xception yang menggunakan *depthwise separable convolution* menunjukkan efisiensi komputasi dan akurasi sebesar 89% (Muslikh et al., 2023). Sementara itu, model yang diusulkan dalam penelitian ini berhasil mencapai akurasi sebesar 96,47%, yang

menunjukkan performa kompetitif dengan keunggulan pada sisi kesederhanaan arsitektur.

Secara keseluruhan, model CNN yang diusulkan menunjukkan peningkatan akurasi sekitar 1,23%. Peningkatan ini dipengaruhi oleh rancangan arsitektur yang disesuaikan dengan karakteristik data. Arsitektur CNN kustom dalam penelitian ini dirancang dengan memperhatikan sejumlah aspek seperti jumlah lapisan tersembunyi (*hidden layer*), penggunaan *padding* pada lapisan konvolusi, ukuran dan jumlah kernel, jenis fungsi aktivasi, metode *pooling*, jenis optimasi, teknik penghitungan *loss*, strategi mengatasi *overfitting*, jumlah *epoch*, nilai *learning rate*, dan ukuran *batch size*. Dengan rancangan arsitektur yang optimal, model dapat mengenali fitur kompleks pada data secara lebih efektif, mengatasi *overfitting* atau *underfitting*, serta meningkatkan akurasi. Selain itu, proses prapemrosesan juga berperan penting dalam meningkatkan performa model, menjadikannya lebih unggul dibandingkan model CNN lain yang tidak dioptimalkan secara menyeluruh.

Dengan hasil yang menunjukkan performa tinggi, penelitian ini diharapkan dapat memberikan kontribusi yang berguna dalam mendukung proses identifikasi penyakit tanaman padi secara lebih akurat. Manfaatnya tidak hanya dirasakan oleh peneliti, tetapi juga oleh para petani dan pihak terkait seperti dinas pertanian. Hal ini selaras dengan firman Allah SWT dalam Surah Al-Isra' ayat 36.

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَٰئِكَ كَانَ عَنْهُ مَسْئُولًا ﴿٣٦﴾

“Janganlah engkau mengikuti sesuatu yang tidak kauketahui. Sesungguhnya pendengaran, penglihatan, dan hati nurani, semua itu akan diminta pertanggungjawabannya.” (Q.S Al-Isra’:36)

Dalam Tafsir Kemenag, Allah SWT mengingatkan agar manusia tidak mengikuti sesuatu yang belum diketahui kebenarannya. Setiap tindakan yang melibatkan pendengaran, penglihatan, dan hati akan dimintai pertanggungjawaban karena ketiganya adalah amanah dari Tuhanmu. Hal ini mencerminkan pentingnya kehati-hatian dan dasar pengetahuan yang akurat dalam setiap keputusan dan informasi yang disampaikan. Dalam konteks penelitian ini, prinsip tersebut sangat relevan karena proses identifikasi penyakit padi dengan metode klasifikasi memerlukan tingkat akurasi yang tinggi. Hasil klasifikasi yang tidak akurat dapat menimbulkan kerancuan dalam pengambilan keputusan, baik oleh peneliti, petani, maupun pihak terkait lainnya.

Oleh karena itu, data yang digunakan dalam penelitian ini tidak dikategorikan secara sembarangan, melainkan bersumber dari dataset terpercaya yang tersedia di platform Kaggle. Penggunaan data yang valid dan terverifikasi menjadi langkah penting untuk memastikan bahwa model klasifikasi yang dikembangkan mampu menghasilkan akurasi yang tinggi, sehingga hasilnya dapat dipertanggungjawabkan secara ilmiah maupun etis.

Akurasi bukan hanya sekadar angka dalam hasil evaluasi model, tetapi juga bentuk nyata dari tanggung jawab ilmiah. Penelitian ini berupaya untuk menghindari kesalahan deteksi dan prediksi yang berpotensi menyesatkan, sebagaimana larangan dalam ayat untuk tidak mengatakan atau mengikuti hal yang tidak didasari oleh pengetahuan. Dengan demikian, penelitian ini sejalan dengan nilai-nilai yang diajarkan dalam Al-Qur'an, yaitu menjunjung tinggi kebenaran, kehati-hatian, dan pertanggungjawaban atas ilmu yang dimiliki. Keakuratan sistem

menjadi simbol amanah dalam pengolahan data dan penyampaian informasi, yang pada akhirnya memberi manfaat nyata dalam pengendalian penyakit tanaman dan kesejahteraan petani.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai klasifikasi penyakit pada tanaman padi berbasis citra daun menggunakan *convolutional neural network* (CNN), dapat disimpulkan bahwa model CNN yang diusulkan mampu mengklasifikasikan citra daun padi ke dalam beberapa kategori penyakit seperti *blast*, *blight*, *tungro* dan *healthy* secara akurat. Hasil pengujian menunjukkan model CNN yang diusulkan memperoleh nilai akurasi, *precision*, *recall* dan *f1-score* masing-masing sebesar 0,96. Pencapaian ini menunjukkan bahwa arsitektur CNN yang digunakan dalam penelitian ini mampu memberikan hasil yang unggul dan menunjukkan peningkatan performa dibandingkan pendekatan klasifikasi CNN yang ada, tanpa menggunakan arsitektur yang lebih kompleks atau teknik tambahan.

5.2 Saran

Ada beberapa saran untuk penelitian lebih lanjut. Pertama, penelitian selanjutnya dapat mempertahankan pendekatan dasar CNN seperti yang digunakan dalam penelitian ini, dengan tetap mengevaluasi efektivitasnya pada data yang lebih bervariasi. Hal ini penting mengingat arsitektur CNN sederhana yang digunakan telah menghasilkan akurasi tinggi tanpa bantuan teknik tambahan. Kedua, dianjurkan untuk memperluas kumpulan data dengan sampel yang lebih komprehensif dan beragam, idealnya dikumpulkan dari sawah di dunia nyata, untuk

meningkatkan kemampuan model dalam menggeneralisasi dan beradaptasi dengan kondisi lingkungan yang berbeda.

DAFTAR PUSTAKA

- Abasi, A. K., Makhadmeh, S. N., Alomari, O. A., Tubishat, M., & Mohammed, H. J. (2023). Enhancing Rice Leaf Disease Classification: A Customized Convolutional Neural Network Approach. *Sustainability (Switzerland)*, 15(20). <https://doi.org/10.3390/su152015039>
- Agustiani, S., Tajul Arifin, Y., Junaidi, A., Khotimatul Wildah, S., & Mustopa, A. (2022). Klasifikasi Penyakit Daun Padi menggunakan Random Forest dan Color Histogram 1,*). *Jurnal Komputasi*, 10(1). <https://www.kaggle.com/vbookshelf/rice-leaf->
- Alwy, A. D. P., Wahid, M. S. N., Ag, B. N. N., & Fakhri, M. M. (2023). Klasifikasi Penyakit Pada Padi Dengan Ekstraksi Fitur LBP dan GLCM. *Journal of Deep Learning, Computer Vision and Digital Image Processing*, 1(1). <https://journal.diginus.id/index.php/decoding>
- Azmi, K., Defit, S., & Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, U. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1).
- Bhat, S. S., Ananth, A., & Venugopala, P. S. (2023). Design and Evolution of Deep Convolutional Neural Networks in Image Classification - A Review. *International Journal of Integrated Engineering*, 15(1), 213–225. <https://doi.org/10.30880/ijie.2023.15.01.019>
- Bhattacharjee, A., Popov, A. A., Sarshar, A., & Sandu, A. (2024). *Improving The Adaptive Moment Estimation (ADAM) Stochastic Optimizer Through An Implicit-Explicit (IMEX) Time-Stepping*. <http://arxiv.org/abs/2403.13704>
- Chaudhary, S., & Kumar, U. (2024). Automated Detection and Classification of Rice Crop Diseases Using Advanced Image Processing and Machine Learning Techniques. *Traitement Du Signal*, 41(2), 739–752. <https://doi.org/10.18280/ts.410216>
- Ghosal, S., & Sarkar, K. (2020). Rice Leaf Diseases Classification Using CNN With Transfer Learning. *Proceedings of 2020 IEEE Calcutta Conference (CALCON)*.
- Gunawan, P. A., Kencana, E. N., & Sari, K. (2021). Classification Of Rice Leaf Diseases Using Artificial Neural Network. *Journal of Physics: Conference Series*, 1722(1). <https://doi.org/10.1088/1742-6596/1722/1/012013>
- Ihtada, F. K., Alfianita, R., & Aziz, O. Q. (2025). Aspect-based Multilabel Classification of E-commerce Reviews Using Fine-tuned IndoBERT. *Kinetik: Game Technology, Information System, Computer Network*,

Computing, Electronics, and Control.
<https://doi.org/10.22219/kinetik.v10i1.2088>

- Islam, M. A., Rahman Shuvo, M. N., Shamsojjaman, M., Hasan, S., Hossain, M. S., & Khatun, T. (2021). An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 12(1). www.ijacsa.thesai.org
- Muslikh, A. R., Setiadi, D. R. I. M., & Ojugo, A. A. (2023). Rice Disease Recognition Using Transfer Learning Xception Convolutional Neural Network. *Jurnal Teknik Informatika (Jutif)*, 4(6), 1535–1540. <https://doi.org/10.52436/1.jutif.2023.4.6.1529>
- N, K., Narasimha Prasad, L. V., Pavan Kumar, C. S., Subedi, B., Abraha, H. B., & Sathishkumar, V. E. (2021). Rice Leaf Diseases Prediction Using Deep Neural Networks With Transfer Learning. *Environmental Research*. <https://doi.org/10.1016/j.envres.2021.111275>
- Pratama, M. D., Gustriansyah, R., & Purnamasari, E. (2024). Klasifikasi Penyakit Daun Pisang Menggunakan Convolutional Neural Network (CNN). *Jurnal Teknologi Terpadu*, 10(1), 1–6.
- 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), 123. <https://doi.org/10.24843/lkjiti.2021.v12.i02.p06>
- Rahman, M. A., Islam, M. R., Rafath, M. A. H., & Mhejabin, S. (2023). CNN Based Covid-19 Detection from Image Processing. *Journal of ICT Research and Applications*, 17(1), 99–113. <https://doi.org/10.5614/itbj.ict.res.appl.2023.17.1.7>
- Rahmi, R., & Saputra, R. A. (2024). Klasifikasi Penyakit Padi Melalui Citra Daun Menggunakan Metode Naive Bayes. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(2). <https://doi.org/10.23960/jitet.v12i2.4012>
- Rijal, M., Yani, A. M., & Rahman, A. (2024). Deteksi Citra Daun Untuk Klasifikasi Penyakit Padi Menggunakan Pendekatan Deep Learning Dengan Model CNN. *Jurnal Teknologi Terpadu*, 10(1), 56–62. <https://journal.nurulfikri.ac.id/index.php/jtt>
- Santoso, I. B., Adrianto, Y., Sensusiaty, A. D., Wulandari, D. P., & Purnama, I. K. E. (2022). Ensemble Convolutional Neural Networks With Support Vector Machine for Epilepsy Classification Based on Multi-Sequence of Magnetic Resonance Images. *IEEE Access*, 10(1), 32034–32048. <https://doi.org/10.1109/ACCESS.2022.3159923>
- Santoso, I. B., & Purnama, I. K. E. (2020). Epileptic EEG Signal Classification Using Convolutional Neural Networks Based On Optimum Window Length

And FFT's Length. *ACM International Conference Proceeding Series*, 87–91. <https://doi.org/10.1145/3411174.3411179>

Saputra, R. A., Suharyanto, Wasiyanti, S., Saefudin, D. F., Supriyatna, A., & Wibowo, A. (2020). Rice Leaf Disease Image Classifications Using KNN Based on GLCM Feature Extraction. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012080>

Septiandi, L. A., Yuniarno, E. M., & Zaini, A. (2021). Deteksi Kedipan dengan Metode CNN dan Percentage of Eyelid Closure (PERCLOS). *JURNAL TEKNIK ITS*, 10(1), 2337–3539.

Setiawan, R., Zein, H., & Agramanisti Azdy, R. (2023). Rice Leaf Disease Classification with Machine Learning: An Approach Using Nu-SVM. *Indonesian Journal of Data and Science*, 4. <https://doi.org/10.56705/ijodas>

Shrestha, S. B., Zhu, L., & Sun, P. (2022). Spikemax: Spike-based Loss Methods for Classification. *IEEE International Joint Conference on Neural Network*. <http://arxiv.org/abs/2205.09845>

Suartika E. P, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *JURNAL TEKNIK ITS*, 5(1).

Sunandar, N., & Sutopo, J. (2024). Utilization of Artificial Neural Network in Rice Plant Disease Classification Using Leaf Image. *International Journal of Research In Science & Engineering*, 42, 1–10. <https://doi.org/10.55529/ijrise.42.1.10>

Syahrani, R., Suhartono, & Zaman, S. (2023). Regresi Logistik Multinomial untuk Prediksi Kategori Kelulusan Mahasiswa. *Jurnal Informatika Sunan Kalijaga*, 8(2), 102–111.

Takacs, P., & Bourrat, P. (2022). The Arithmetic Mean Of What? A Cautionary Tale About The Use Of The Geometric Mean As A Measure Of Fitness. *Biology and Philosophy*, 37(2). <https://doi.org/10.1007/s10539-022-09843-4>

Wang, Y., Wang, H., & Peng, Z. (2021). Rice Diseases Detection And Classification Using Attention Based Neural Network And Bayesian Optimization. *Expert Systems with Applications*, 178. <https://doi.org/10.1016/j.eswa.2021.114770>

Wardianto, Farikhin, & Nugraheni, D. M. K. (2023). Analisis Sentimen Berbasis Aspek Ulasan Pelanggan Restoran Menggunakan LSTM Dengan Adam Optimizer. *Journal of Information Technology and Computer Science*, 8(2), 67–76.

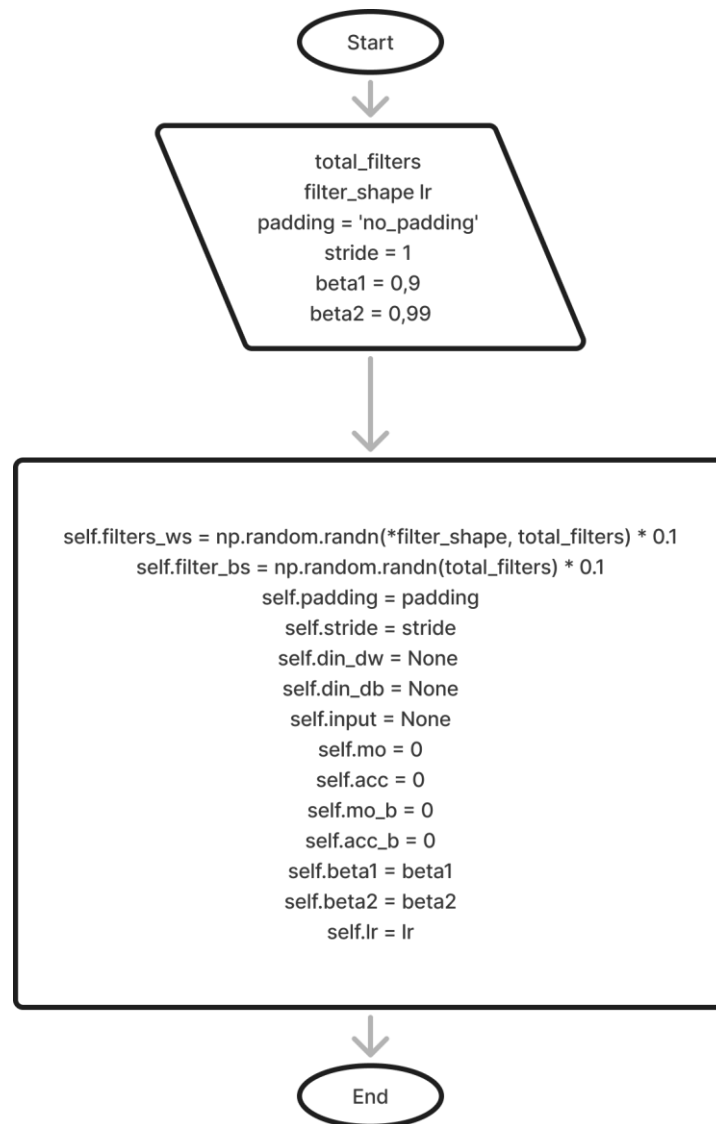
Wulandari, I., Yasin, H., & Widiyarih, T. (2020). Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (CNN).

JURNAL GAUSSIAN, 9(3), 273–282.
<https://ejournal3.undip.ac.id/index.php/gaussian/>

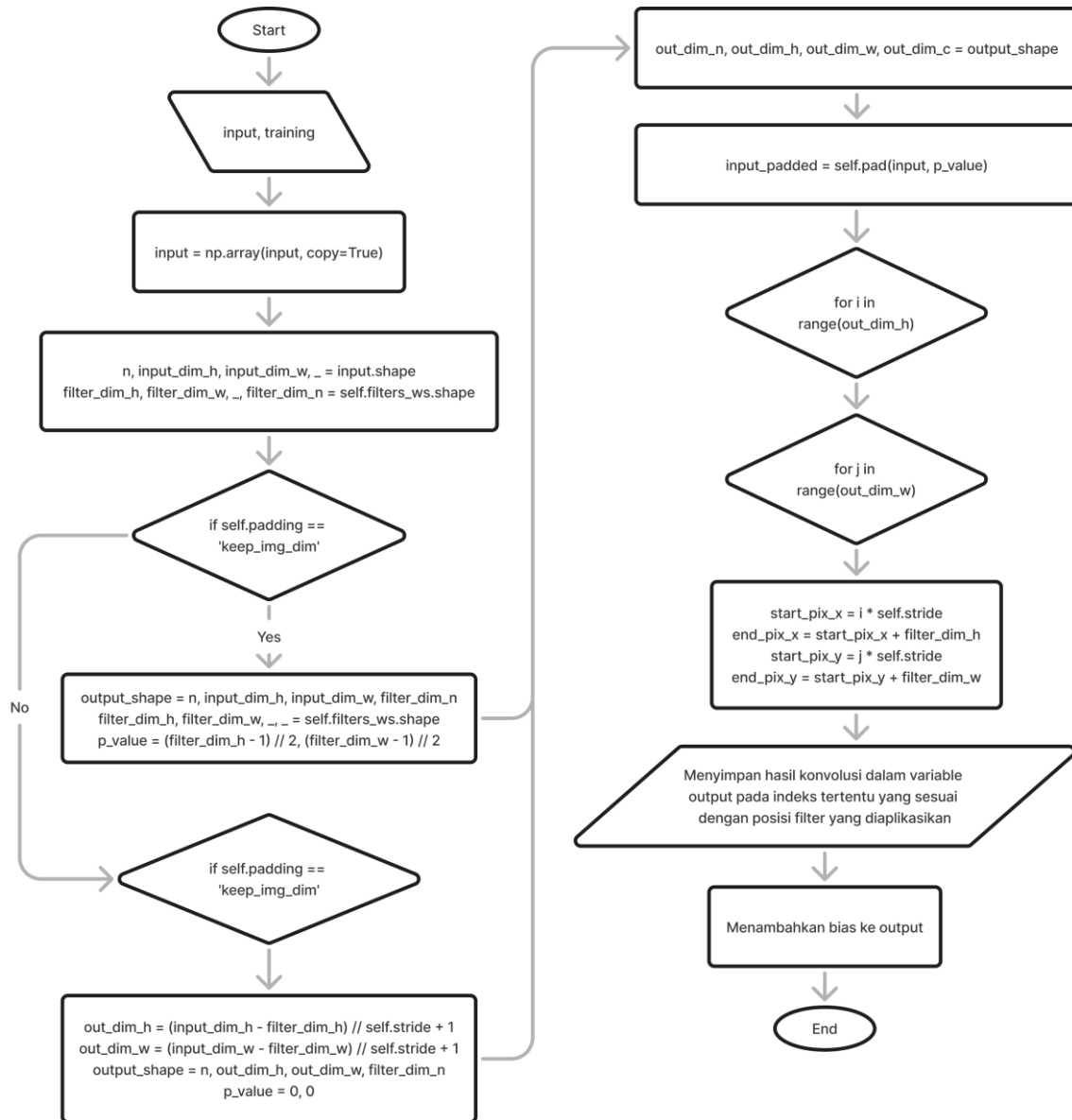
Lampiran 1

Algoritma Convolutional Layer

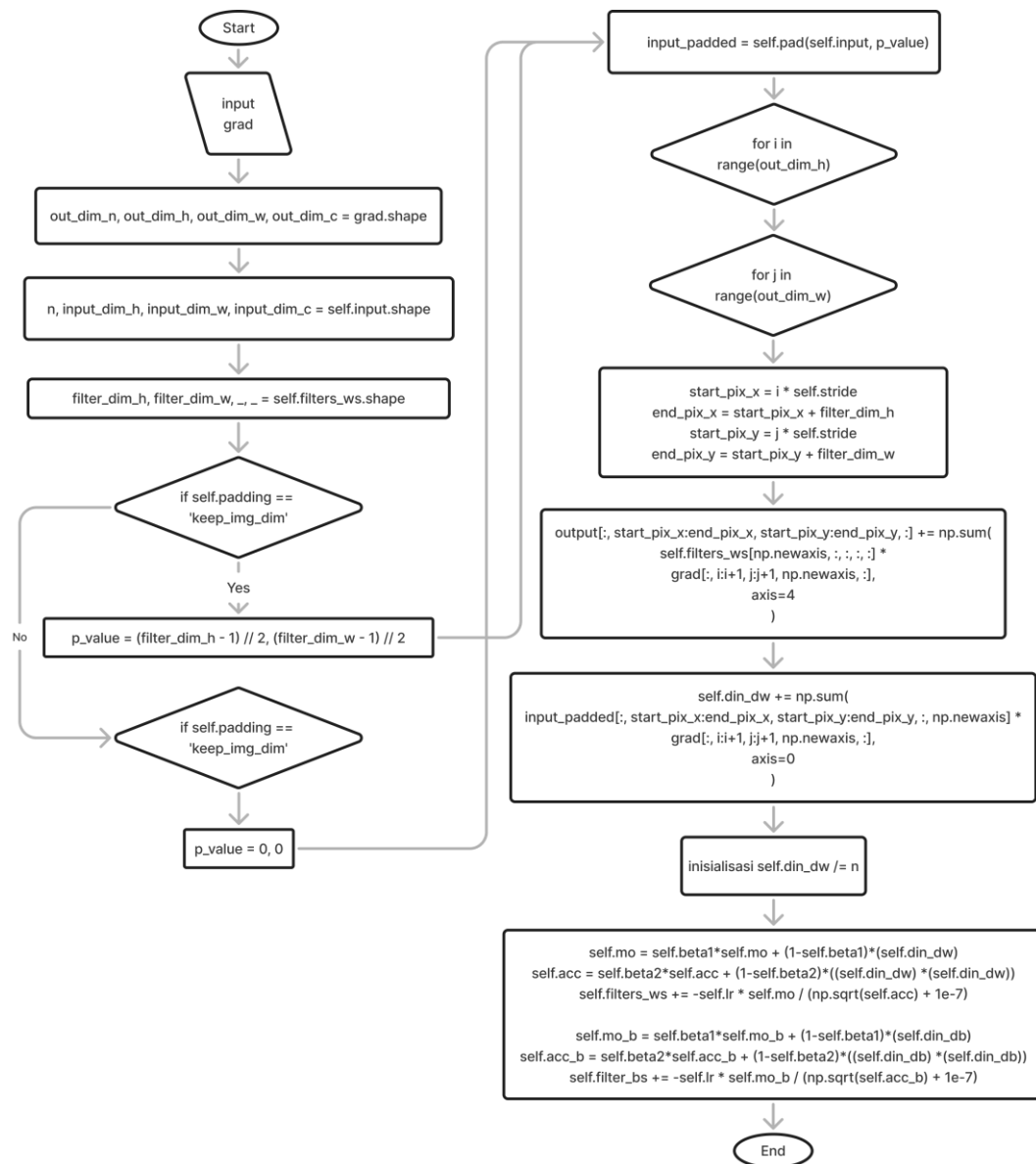
Dalam algoritma *convolutional layer* dibuatkan sebuah kelas dan didalamnya memiliki tiga *function* untuk menangani operasi yang terlibat dalam proses konvolusi yaitu `def _init_`, `def forward` dan `def backward`.



Gambar diatas merupakan *function* `def _init_` bertugas menginisialisasi parameter yang diperlukan.



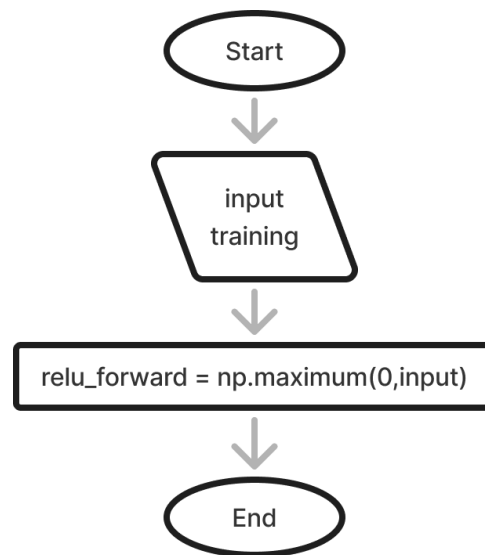
Gambar diatas merupakan *function def forward* untuk menghitung output *convolutional layer* dengan menerapkan filter ke gambar input atau peta fitur. Bergantung pada jenis *padding*, input dapat diberi *padding* untuk mempertahankan atau mengurangi dimensinya. Metode ini mengulangi dimensi input, menerapkan operasi konvolusi untuk menghasilkan peta fitur keluaran, yang kemudian dikembalikan dengan bias tambahan



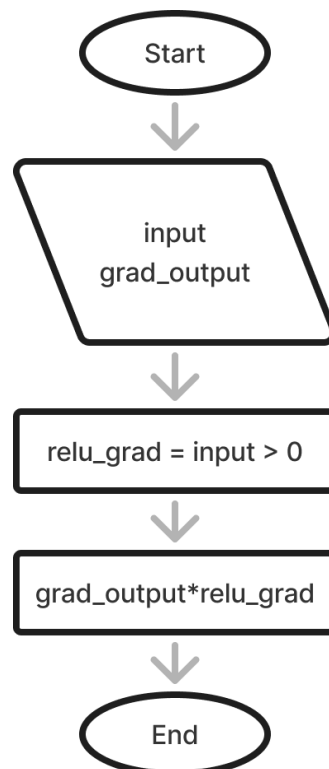
Gambar diatas merupakan *function def backward* bertanggung jawab untuk menghitung gradien yang diperlukan untuk memperbarui bobot dan bias lapisan konvolusional selama *backpropagation*. Gradien ini kemudian digunakan untuk menyesuaikan bobot dan bias, dengan menggunakan pengoptimal Adam.

Lampiran 2

Algoritma ReLU



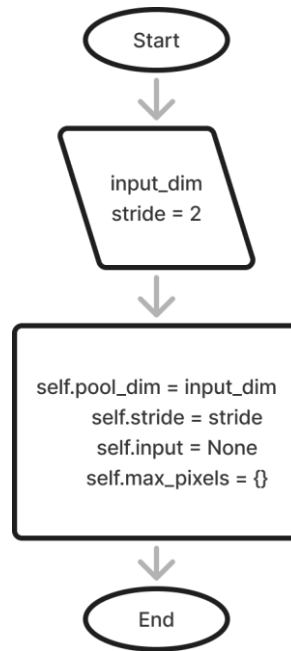
Gambar diatas merupakan *function forward* pada ReLU, disini menetapkan nilai negatif ke nol untuk memastikan output tidak negatif.



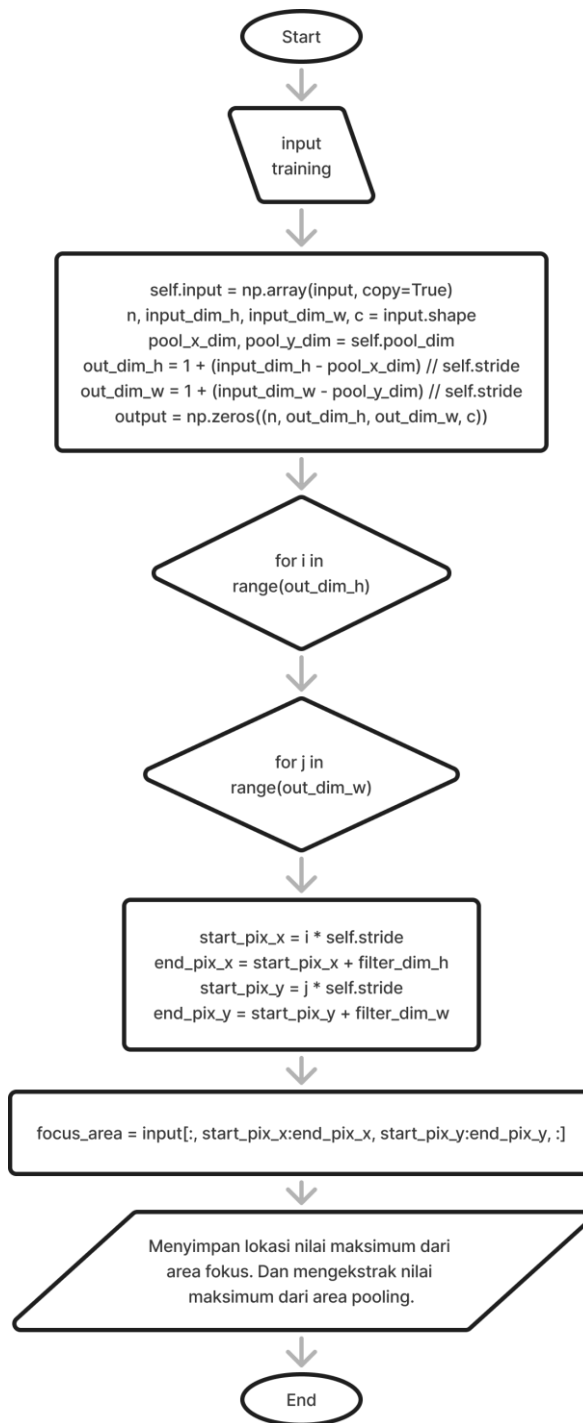
Gambar diatas merupakan *function backward*, dimana gradien akan bernilai 1 jika input positif dan 0 jika input negatif. Lalu mengembalikan gradien output yang dikalikan dengan mask ReLU.

Lampiran 3

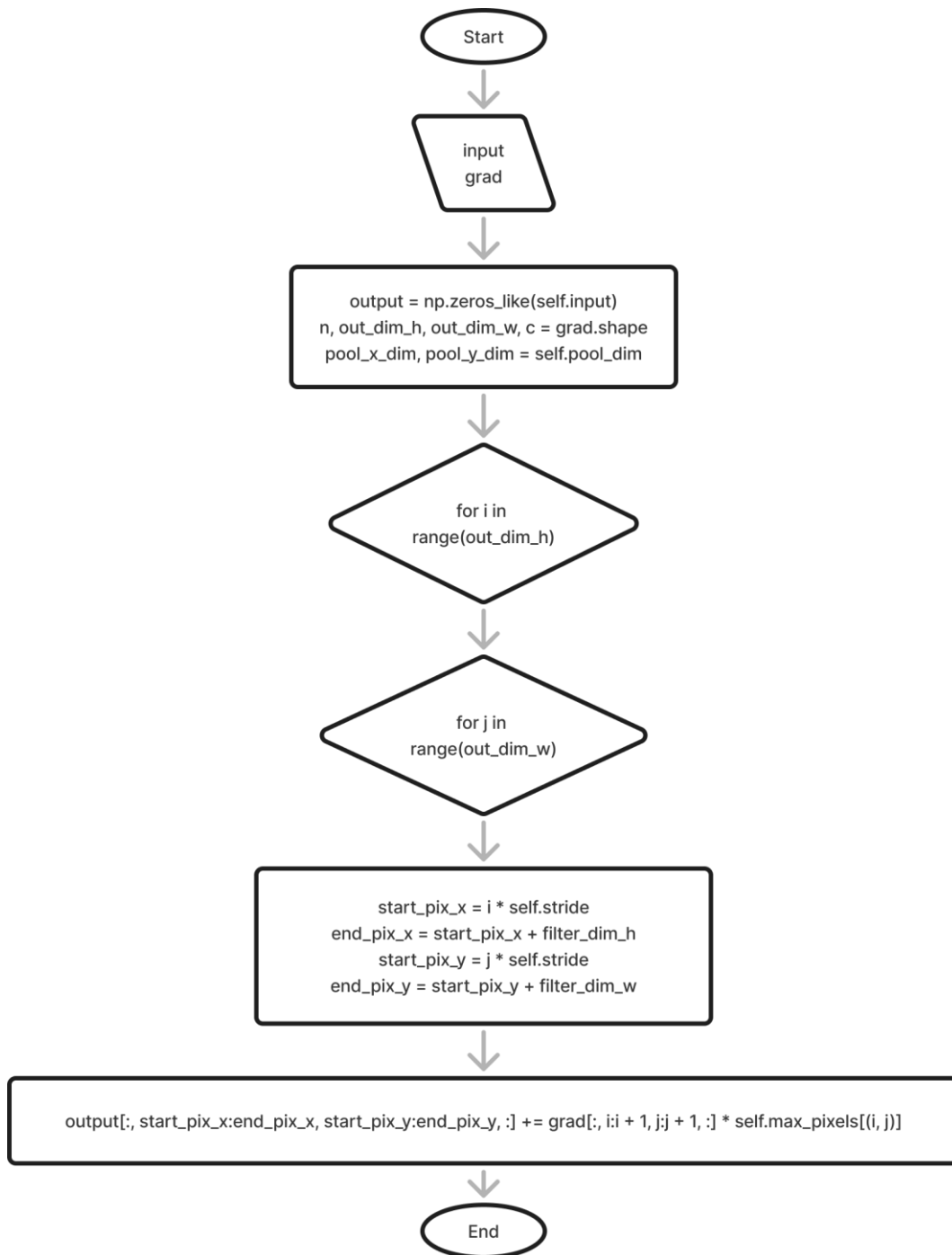
Algoritma Pooling Layer



Gambar diatas merupakan *function def _init_* yang bertugas menginisialisasi parameter yang diperlukan.



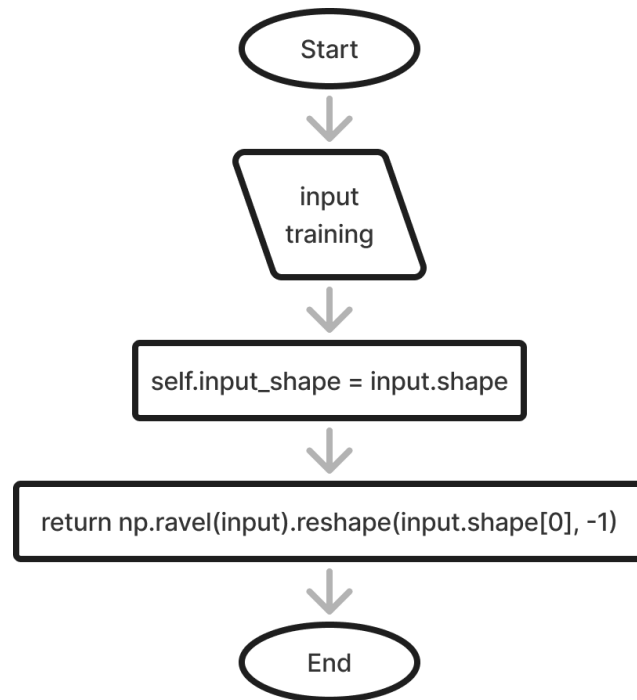
Gambar diatas merupakan *function def forward* yang bertugas melakukan operasi *pooling* dengan cara mengidentifikasi nilai maksimum dan sesuai dengan lokasinya kemudian disimpan.



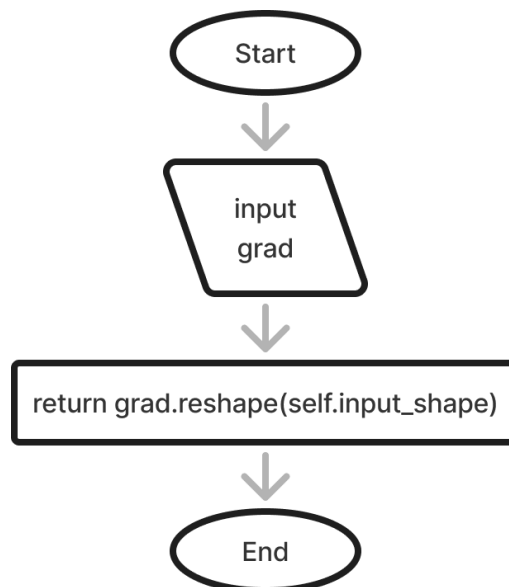
Gambar diatas merupakan *function def backward* bertugas menyebarkan gradien ke posisi nilai maksimum yang direkam selama *forward pass*.

Lampiran 4

Algoritma Flatten Layer



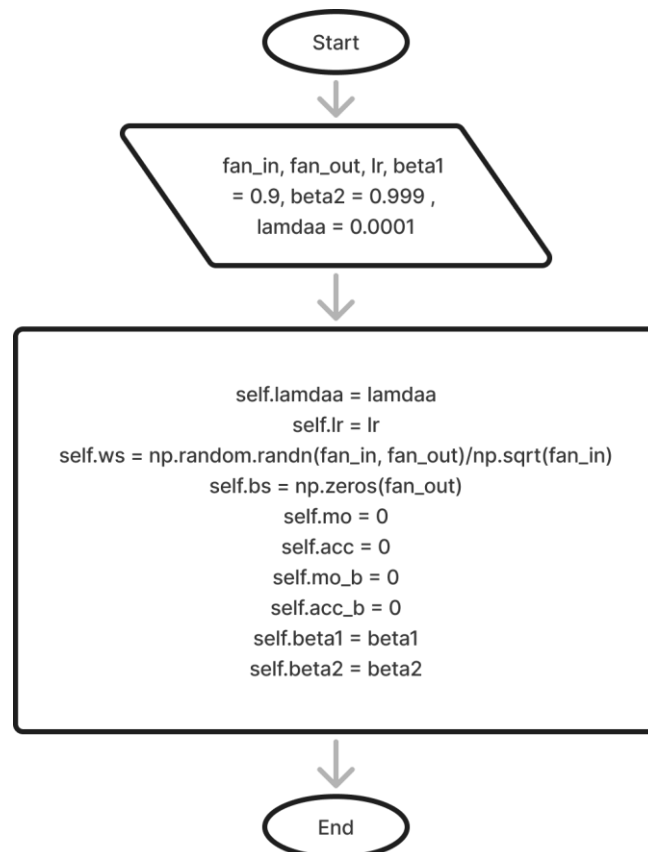
Gambar diatas merupakan *function def forward* untuk melakukan perubahan dari multidimensi menjadi satu dimensi.



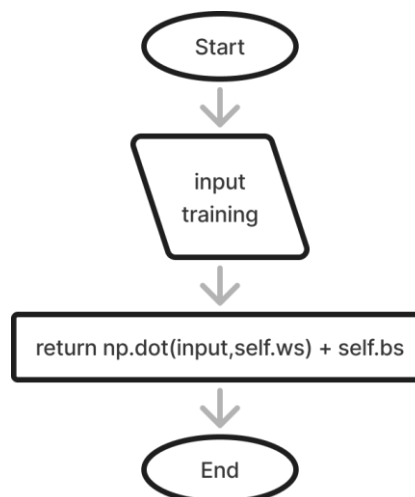
Gambar diatas merupakan *function def backward* untuk melakukan perubahan ulang bentuk gradien ke bentuk asli.

Lampiran 5

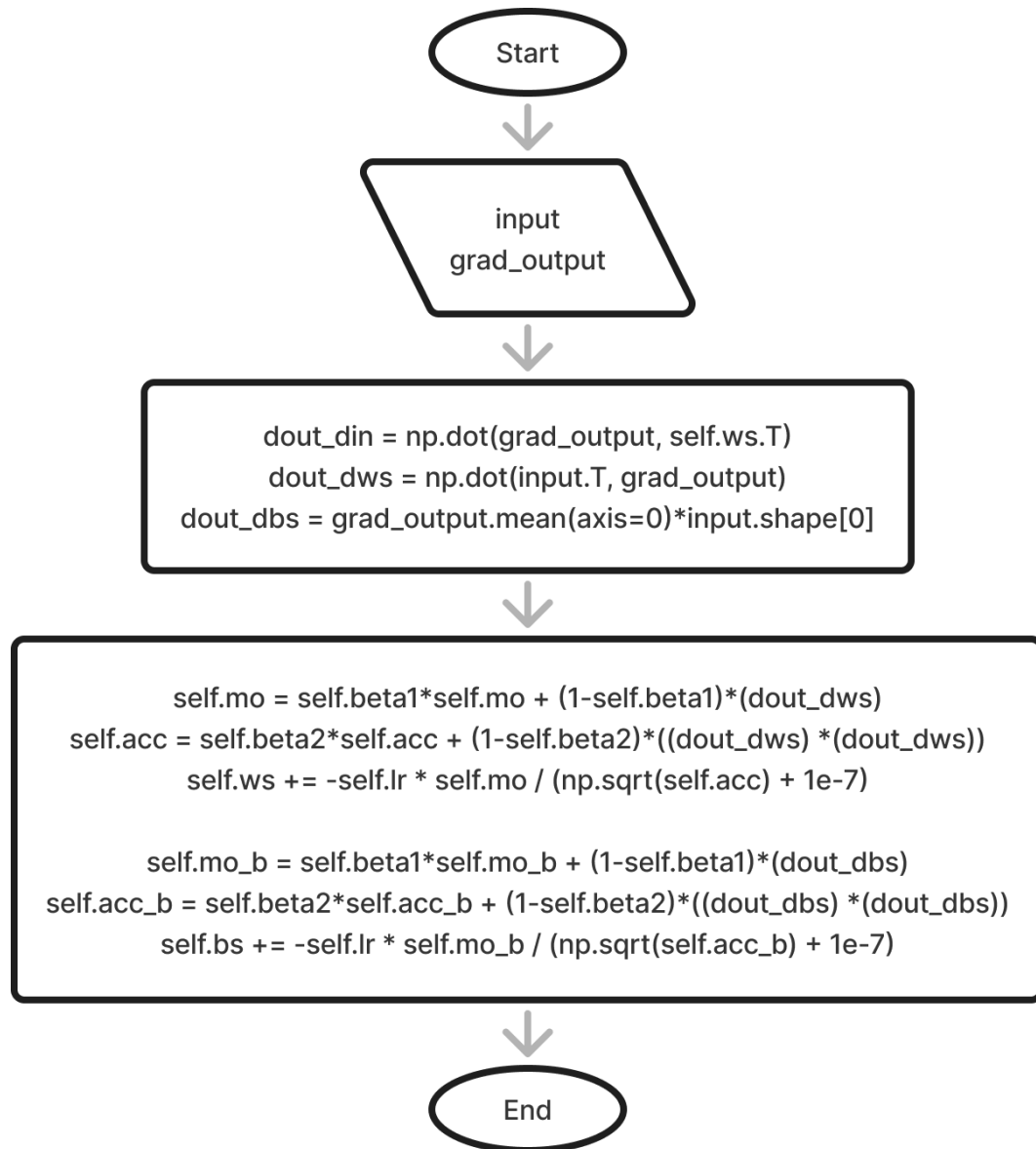
Algoritma Fully Connected Layer



Gambar diatas merupakan *function def _init_* digunakan untuk menginisialisasi parameter yang diperlukan.



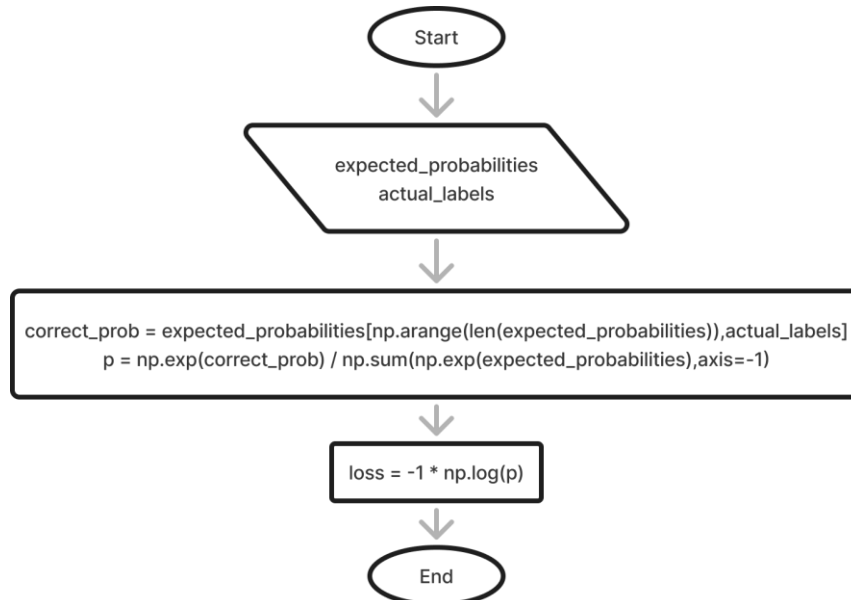
Gambar diatas merupakan *function def forward* pada dasarnya menggabungkan fitur yang diekstrak dari lapisan sebelumnya menjadi keluaran mentah.



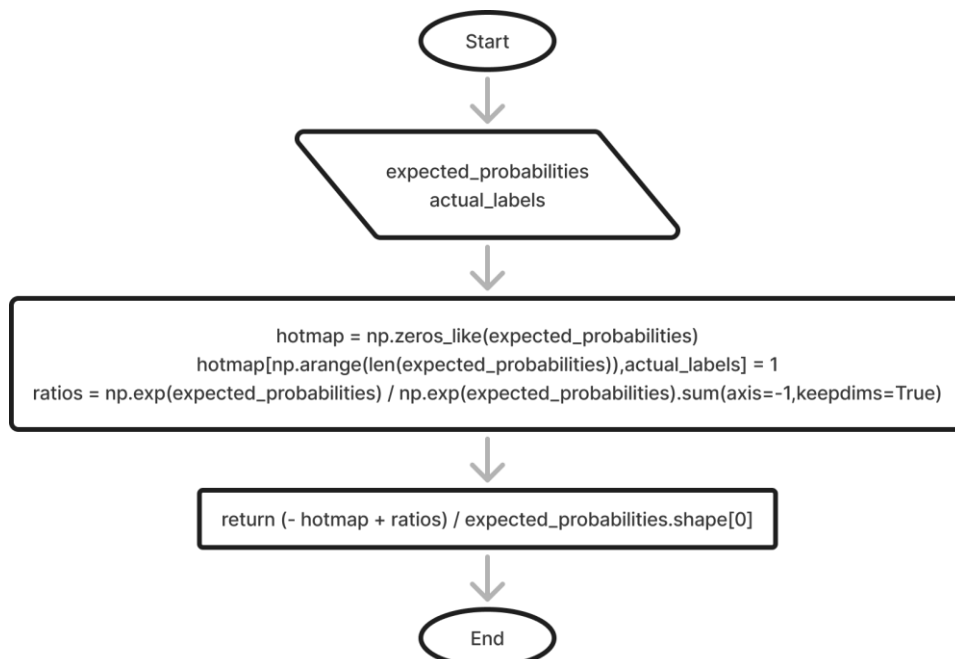
Gambar diatas merupakan *function def backward* disini menghitung gradien yang dibutuhkan untuk proses *backpropagation*. Yang mana gradien tersebut akan digunakan untuk pembaruan bobot dan bias pada optimasi ADAM.

Lampiran 6

Algoritma Negative Log-Likelihood Loss



Gambar diatas merupakan *function def NLL*, digunakan untuk mengetahui/mengukur seberapa besar perbedaan antara prediksi model dan label yang benar, dan digunakan untuk mengevaluasi kinerja model.



Gambar diatas merupakan *function def backNLL*, digunakan untuk menghitung gradien dari *loss* terhadap probabilitas yang diprediksi. Gradien ini memberikan informasi tentang bagaimana menyesuaikan bobot dan bias dalam model untuk

mengurangi *loss*. Gradien ini kemudian digunakan dalam proses *backpropagation* untuk memperbarui parameter model.