

TESIS

**DETEKSI PENYAKIT TANAMAN PADI BERBASIS DEEP LEARNING
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK**



Disusun oleh:

Nama : Mohammad Nur Fawalq
NIM : 21.52.2116
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023**

TESIS

DETEKSI PENYAKIT TANAMAN PADI BERBASIS DEEP LEARNING MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK

RICE PLANT DISEASE DETECTION BASED ON DEEP LEARNING USING CONVOLUTIONAL NEURAL NETWORK

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Mohammad Nur Fawaiq
NIM : 21.52.2116
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023**

HALAMAN PENGESAHAN

**DETEKSI PENYAKIT TANAMAN PADI BERBASIS DEEP LEARNING
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

**RICE PLANT DISEASE DETECTION BASED ON DEEP LEARNING USING
CONVOLUTIONAL NEURAL NETWORK**

Dipersiapkan dan Disusun oleh

Mohammad Nur Fawalq

21.52.2116

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 6 Desember 2023

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

Yogyakarta, 6 Desember 2023

Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

DETEKSI PENYAKIT TANAMAN PADI BERBASIS DEEP LEARNING MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK

RICE PLANT DISEASE DETECTION BASED ON DEEP LEARNING USING CONVOLUTIONAL NEURAL NETWORK

Dipersiapkan dan Disusun oleh

Mohammad Nur Fawatiq

21.52.2116

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 6 Desember 2023

Pembimbing Utama

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

Anggota Tim Penguji

Hanif Al Fatta, M.Kom., Ph.D.
NIK. 190302096

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Dr. Kumara Ari Yuana, S.T., M.T.
NIK. 190302575

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

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

Yogyakarta, 6 Desember 2023

Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Mohammad Nur Fawaiq

NIM : 21.52.2116

Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

Deteksi Penyakit Tanaman Padi Berbasis Deep Learning Menggunakan Convolutional Neural Network

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

Dosen Pembimbing Pendamping : Dhani Ariatmanto, M.Kom., Ph.D.

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

Yogyakarta, 6 Desember 2023

Yang Menyatakan,



Mohammad Nur Fawaiq

HALAMAN PERSEMBAHAN

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

1. Kedua orang tua (Bapak Ahmad Rifai dan Ibu Wiwik Basyiroh) yang senantiasa memberikan semangat dan doa, semoga selalu dalam lindungan-Nya.
2. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. dan Bapak Dhani Ariatmanto, M.Kom., Ph.D. yang telah memberikan bimbingan dan dorongan selama pelaksanaan penelitian tugas akhir ini.
3. Bapak Ibu dosen Magister Informatika Universitas Amikom Yogyakarta yang telah memberikan ilmu dan arahan yang sangat bermanfaat.
4. Seluruh keluarga, teman-teman MTI Amikom, dan semua pihak yang tidak bisa saya sebutkan satu persatu.

HALAMAN MOTTO

“Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum, sebelum mereka mengubah keadaan diri mereka sendiri.”

– QS Ar-Rad : 11

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan.”

– QS Al-Insyirah : 5

“Barang siapa yang menginginkan dunia, maka hendaklah ia menguasai ilmu. Barang siapa menginginkan akhirat, hendaklah ia menguasai ilmu. Dan barang siapa yang menginginkan keduanya (dunia dan akhirat), hendaklah ia menguasai ilmu.”

– (H.R. Ahmad)

“Menuntut ilmu adalah takwa. Menyampaikan ilmu adalah ibadah. Mengulang-ulang ilmu adalah zikir. Mencari ilmu adalah jihad.”

– Abu Hamid Al Ghazali

“Orang positif saling mendoakan, orang negatif saling menjatuhkan. Orang sukses mengerti pentingnya proses, orang gagal lebih banyak protes.”

“Hidup hanya sebentar dan sekali, maka buatlah berarti.”

KATA PENGANTAR

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

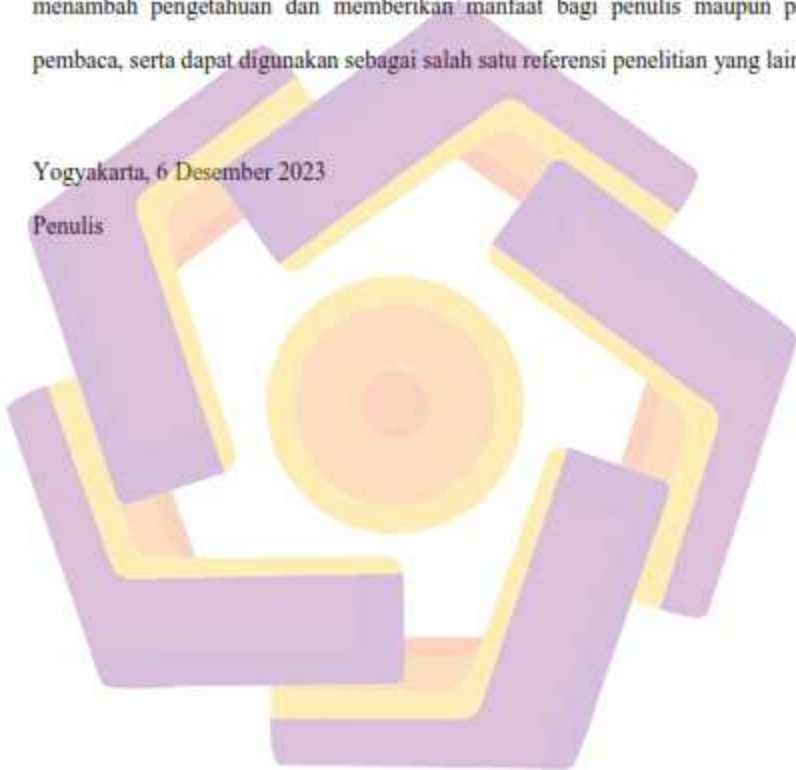
Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. M. Suyanto, M.M. selaku Rektor Universitas Amikom Yogyakarta.
2. Ibu Prof. Dr. Kusriani, M.Kom. selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. dan Bapak Dhani Ariatmanto, M.Kom., Ph.D. selaku dosen pembimbing yang telah sabar memberikan bantuan, masukan, dan saran dalam menyelesaikan tesis ini.
4. Bapak Ibu dosen di Pascasarjana Universitas Amikom Yogyakarta yang telah memberikan banyak ilmu dan pengalaman.
5. Kedua orang tua yang tak pernah lelah dalam memberikan support dan doa.
6. Teman-teman Pascasarjana Magister Informatika angkatan 27 kelas B Universitas Amikom Yogyakarta.

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

Yogyakarta, 6 Desember 2023

Penulis



DAFTAR ISI

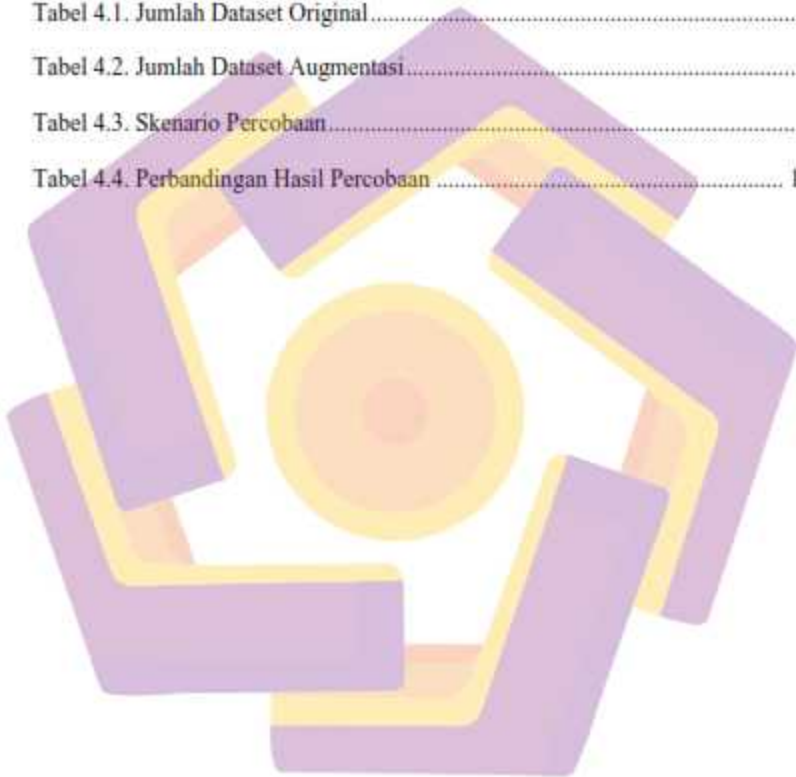
HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
INTISARI.....	xviii
<i>ABSTRACT</i>	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah.....	6
1.3. Batasan Masalah	7
1.4. Tujuan Penelitian	7
1.5. Manfaat Penelitian	8
BAB II TINJAUAN PUSTAKA.....	9
2.1. Tinjauan Pustaka.....	9
2.2. Keaslian Penelitian.....	17

2.3. Landasan Teori.....	27
2.3.1. Penyakit Tanaman.....	27
2.3.2. Citra Digital	30
2.3.2. Pengolahan Citra Digital.....	31
2.3.3. Klasifikasi	33
2.3.4. Convolutional Neural Network.....	34
2.3.5. Augmentasi Data.....	39
2.3.6. Confusion Matrix.....	39
BAB III METODE PENELITIAN.....	41
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	41
3.2. Metode Pengumpulan Data.....	41
3.3. Metode Analisis Data.....	42
3.4. Alur Penelitian	43
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	47
4.1. Pengumpulan Data.....	47
4.2. Analisis Data	48
4.2.1. Augmentasi Data.....	49
4.2.2. Preprocessing Data.....	51
4.2.3. Preprocessing Data Lanjutan (<i>Enhancement</i>).....	52
4.3. Persiapan Skenario CNN	54
4.3.1. Skenario-skenario CNN I	56

4.3.2. Skenario-skenario CNN 2	59
4.3.3. Skenario-skenario CNN 3	60
4.3.4. Skenario-skenario CNN 4	62
4.4. Analisis Hasil Penelitian.....	64
4.4.1 Hasil dan Pengujian CNN 1	66
4.4.2 Hasil dan Pengujian CNN 2	79
4.4.3 Hasil dan Pengujian CNN 3	94
4.4.4 Hasil dan Pengujian CNN 4	108
4.4.5. Perbandingan Hasil Percobaan	121
BAB V PENUTUP	124
5.1. Kesimpulan	124
5.2. Saran	125
DAFTAR PUSTAKA	127

DAFTAR TABEL

Tabel 2.1. Matriks Literatur Review dan Posisi Penelitian.....	17
Tabel 2.2. Confusion Matrix	40
Tabel 4.1. Jumlah Dataset Original	47
Tabel 4.2. Jumlah Dataset Augmentasi.....	49
Tabel 4.3. Skenario Percobaan.....	55
Tabel 4.4. Perbandingan Hasil Percobaan	122



DAFTAR GAMBAR

Gambar 2.1. Penyakit <i>Blast</i> Daun Padi.....	28
Gambar 2.2. Penyakit <i>Blight</i> Daun Padi	29
Gambar 2.3. Penyakit <i>Tungro</i> Daun Padi	30
Gambar 2.4. Matriks Citra Digital	30
Gambar 2.5. Arsitektur Convolutional Neural Network	35
Gambar 2.6. Proses Konvolusi Menggunakan <i>Filter</i>	35
Gambar 2.7. Proses pada <i>Max Pooling</i>	37
Gambar 2.8. Proses <i>Flattening</i>	38
Gambar 2.9. <i>Standard Network</i> (a) dan <i>Network</i> dengan <i>Dropout</i> (b)	38
Gambar 3.1. Kerangka Alur Penelitian	43
Gambar 4.1. Sampel Dataset Original.....	48
Gambar 4.2. Sampel Dataset Augmentasi Original	51
Gambar 4.3. Sampel Dataset Enhancement Original.....	53
Gambar 4.4. Sampel Dataset Augmentasi Enhancement.....	53
Gambar 4.5. Model Arsitektur VGG16.....	57
Gambar 4.6. Arsitektur VGG16 Setelah <i>Fine Tuning</i>	59
Gambar 4.7. Model Arsitektur Xception.....	60
Gambar 4.8. Model Arsitektur NASNetMobile	61
Gambar 4.9. Model Arsitektur Custom X.....	62
Gambar 4.10. Grafik <i>Training & Validation</i> VGG16 (Dataset Original).....	66
Gambar 4.11. <i>Confusion Matrix</i> VGG16 (Dataset Original)	66

Gambar 4.12. <i>Classification Report</i> VGG16 (Dataset Original)	68
Gambar 4.13. Grafik <i>Training & Validation</i> VGG16 (Dataset Augmentasi Original)	69
Gambar 4.14. <i>Confusion Matrix</i> VGG16 (Dataset Augmentasi Original).....	70
Gambar 4.15. <i>Classification Report</i> VGG16 (Dataset Augmentasi Original)	72
Gambar 4.16. Grafik <i>Training & Validation</i> VGG16 (Dataset Enhancement Original).....	72
Gambar 4.17. <i>Confusion Matrix</i> VGG16 (Dataset Enhancement Original)	73
Gambar 4.18. <i>Classification Report</i> VGG16 (Dataset Enhancement Original) ...	75
Gambar 4.19. Grafik <i>Training & Validation</i> VGG16 (Dataset Augmentasi Enhancement)	76
Gambar 4.20. <i>Confusion Matrix</i> VGG16 (Dataset Augmentasi Enhancement) ...	77
Gambar 4.21. <i>Classification Report</i> VGG16 (Dataset Augmentasi Enhancement)	79
Gambar 4.22. Grafik <i>Training & Validation</i> Xception (Dataset Original).....	79
Gambar 4.23. <i>Confusion Matrix</i> Xception (Dataset Original).....	80
Gambar 4.24. <i>Classification Report</i> VGG16 (Dataset Original)	82
Gambar 4.25. Grafik <i>Training & Validation</i> Xception (Dataset Augmentasi Original).....	83
Gambar 4.26. <i>Confusion Matrix</i> Xception (Dataset Augmentasi Original).....	84
Gambar 4.27. <i>Classification Report</i> Xception (Dataset Augmentasi Original)....	86
Gambar 4.28. Grafik <i>Training & Validation</i> Xception (Dataset Enhancement Original).....	86

Gambar 4.29. <i>Confusion Matrix</i> Xception (Dataset Enhancement Original)	87
Gambar 4.30. <i>Classification Report</i> Xception (Dataset Enhancement Original) .	89
Gambar 4.31. Grafik <i>Training & Validation</i> Xception (Dataset Augmentasi Enhancement)	90
Gambar 4.32. <i>Confusion Matrix</i> Xception (Dataset Augmentasi Enhancement) .	91
Gambar 4.33. <i>Classification Report</i> Xception (Dataset Augmentasi Enhancement)	93
Gambar 4.34. Grafik <i>Training & Validation</i> NASNetMobile (Dataset Original) .	94
Gambar 4.35. <i>Confusion Matrix</i> NASNetMobile (Dataset Original)	94
Gambar 4.36. <i>Classification Report</i> NASNetMobile (Dataset Original)	96
Gambar 4.37. Grafik <i>Training & Validation</i> NASNetMobile (Dataset Augmentasi Original).....	97
Gambar 4.38. <i>Confusion Matrix</i> NASNetMobile (Dataset Augmentasi Original)	98
Gambar 4.39. <i>Classification Report</i> Model NASNetMobile (Dataset Augmentasi Original).....	100
Gambar 4.40. Grafik <i>Training & Validation</i> NASNetMobile (Dataset Enhancement Original).....	100
Gambar 4.41. <i>Confusion Matrix</i> NASNetMobile (Dataset Enhancement Original)	101
Gambar 4.42. <i>Classification Report</i> NASNetMobile (Dataset Enhancement Original).....	103
Gambar 4.43. Grafik <i>Training & Validation</i> NASNetMobile (Dataset Augmentasi Enhancement)	104

Gambar 4.44. <i>Confusion Matrix</i> NASNetMobile (Dataset Augmentasi Enhancement)	105
Gambar 4.45. <i>Classification Report</i> NASNetMobile (Dataset Augmentasi Enhancement)	107
Gambar 4.46. Grafik <i>Training & Validation</i> Custom X (Dataset Original)	108
Gambar 4.47. <i>Confusion Matrix</i> Custom X (Dataset Original)	108
Gambar 4.48. <i>Classification Report</i> Custom X (Dataset Original)	110
Gambar 4.49. Grafik <i>Training & Validation</i> Custom X (Dataset Augmentasi Original).....	111
Gambar 4.50. <i>Confusion Matrix</i> Custom X (Dataset Augmentasi Original)	112
Gambar 4.51. <i>Classification Report</i> Custom X (Dataset Augmentasi Original)	114
Gambar 4.52. Grafik <i>Training & Validation</i> Custom X (Dataset Enhancement Original).....	114
Gambar 4.53. <i>Confusion Matrix</i> Custom X (Dataset Enhancement Original) ...	115
Gambar 4.54. <i>Classification Report</i> NASNetMobile (Dataset Enhancement Original).....	117
Gambar 4.55. Grafik <i>Training & Validation</i> Custom X (Dataset Augmentasi Enhancement)	118
Gambar 4.56. <i>Confusion Matrix</i> Custom X (Dataset Augmentasi Enhancement)	119
Gambar 4.57. <i>Classification Report</i> Custom X (Dataset Augmentasi Enhancement)	121

INTISARI

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

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

Dengan latar belakang tersebut maka diusulkan pendekatan sistem pendeteksian penyakit pada daun padi dengan metode *deep learning* menggunakan *Convolutional Neural Network* (CNN) dan beberapa perbandingan arsitektur yaitu VGG16, NASNetMobile, Xception, dan satu arsitektur Custom X. Hasil eksperimen paling baik diperoleh menggunakan arsitektur Xception yang menghasilkan akurasi *training* 99.65%, akurasi validasi 99.31%, dan akurasi *testing* 99.31%.

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

ABSTRACT

Rice is a staple food source for most people around the world, including Indonesia, which is an agrarian country where most of its population grows and consumes rice. However, rice plants also suffer from various diseases, especially on the leaves, such as bacterial leaf blight, rice blast, dan rice tungro. If the infection or disease in rice plants is not identified early on, it will decrease production and harm farmers. To address this problem, information technology can be utilized in identifying diseases using image processing and image classification.

The research process begins with problem identification, literature review, algorithm and method selection. Subsequently, the collection of public datasets commences, followed by dataset augmentation. Then, experimental data training is conducted using various scenarios and architectures. The outcomes of the experiments are evaluated, and, finally, conclusions are drawn, and the report is written. The dataset was taken from public repositories, and data augmentation was also used in this research to increase the dataset's training accuracy.

With this background, a disease detection system approach is proposed using Deep Learning method using Convolutional Neural Network (CNN) and several transfer learning architectures, namely VGG16, NASNetMobile, Xception, and a Custom X architecture for rice leaf disease detection. The best experimental results were obtained using the Xception architecture, where the training accuracy value is 99.65%, validation accuracy is 99.31%, and the testing accuracy is 99.31%.

Keyword: *cnn, deep learning, data augmentation, leaf disease, rice*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Di dunia yang serba modern ini banyak perkembangan yang telah dialami oleh umat manusia dalam memenuhi kebutuhan hidup, baik perkembangan pada bidang sosbudpolhukam atau sosial budaya politik hukum dan ham, ekonomi, kesehatan, pertanian, dan tentunya teknologi yang semakin canggih. Perkembangan yang dirasakan manusia tidak semuanya memiliki dampak positif dikarenakan timbul juga perkembangan yang berdampak negatif salah satunya dalam bidang pertanian. Permasalahan pada perkembangan pertanian sangatlah meresahkan karena pertanian mempunyai peranan penting dalam kehidupan manusia dan berkontribusi penuh dalam ketahanan pangan maupun ekonomi. Beberapa permasalahan yang sering terjadi dalam perkembangan pertanian adalah rusaknya hasil pertanian akibat penyakit yang menyerang pada daun tumbuhan, dimana fungsi daun bagi tumbuhan sangat vital yaitu melakukan proses fotosintesis atau pembuatan makanan pada tumbuhan itu sendiri (Walascha et al., 2021). Penyakit pada daun tanaman menyerang hampir semua jenis tanaman tak terkecuali tanaman komoditas pangan utama seperti padi.

Padi atau dengan nama latin *Oryza sativa* merupakan salah satu tanaman budidaya terpenting dalam peradaban. Meskipun mengacu pada jenis tanaman budidaya, padi juga digunakan untuk mengacu pada beberapa jenis dari marga (genus) yang sama, yang biasa disebut sebagai padi liar. Padi diduga berasal dari

India atau Indocina dan masuk ke Indonesia dibawa oleh nenek moyang yang migrasi dari daratan Asia sekitar 1500 SM. Di negara agraris seperti Cina, India, Bangladesh, dan Indonesia, padi merupakan tanaman utama. Padi jadi penghasil sebagian besar makanan pokok konsumsi masyarakat, diproses menjadi beras dan dimasak menjadi nasi. Maka dapat dirangkum bahwa padi memiliki manfaat yang sangat besar bagi kehidupan manusia karena menjadi sumber makanan pokok masyarakat di dunia, termasuk Indonesia yang merupakan negara agraris dimana sebagian besar penduduknya menanam padi.

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

Magnaporthe grisea dengan gejala bercak runcing dikelilingi warna kuning pucat (Walascha et al., 2021). Jika penyakit pada tanaman dibiarkan saja maka dapat menurunkan produksi, oleh karena itu perlu pendeteksian penyakit tanaman pada waktu yang tepat agar dapat mencegah dan mengendalikan penyakit tanaman secara efektif (Chen, Chen, Zhang, Sun, & Nanehkaran, 2020).

Penanganan masalah pada penyakit daun tanaman dalam mengurangi dampak kerusakan telah banyak dilakukan dalam beberapa penelitian, tidak hanya penelitian dalam bidang pertanian saja tetapi pada bidang teknologi pun turut andil dalam permasalahan ini, salah satunya pemanfaatan bidang informatika dalam mengidentifikasi penyakit yang ada pada daun tanaman dengan menggunakan citra atau bisa kita sebut *image processing*. Pemanfaatan *image processing* dan *image classification* pada deteksi penyakit tanaman dapat membantu para pengelola pertanian dalam mengambil langkah yang efektif dan efisien (Rozaqi, Sunyoto, & Arief, 2021). *Deep learning* telah menjadi salah satu topik hangat dalam dunia *machine learning* karena kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti citra dan suara. Metode *deep learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Metode *Convolutional Neural Network* atau CNN berhasil mengungguli metode *machine learning* lain pada kasus klasifikasi gambar dikarenakan CNN memiliki cara kerja menyerupai fungsi otak pada manusia, dimana komputer akan diberikan data citra untuk dipelajari, dilatih mengenali setiap elemen visual pada citra serta memahami setiap pola citranya, hingga nantinya komputer mampu mengidentifikasi citra tersebut.

Dengan metode *Convolutional Neural Network* (CNN) ini para peneliti secara empiris telah terbukti mampu mengklasifikasikan citra dengan baik seperti yang dilakukan oleh Mohtar Khoiruddin dkk (2022) dengan judul “Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network” mendapatkan hasil untuk epoch 25, 50, 75 dan 100 memiliki akurasi yang beragam. Hasil akurasi training terbaik ada di epoch 100 dengan akurasi 98% dan dilakukan testing menggunakan *confusion matrix* mendapatkan nilai rata-rata akurasi sebesar 98%. Penelitian lain dengan judul “*Classification of Rice Plant Diseases Using the Convolutional Neural Network Method*” yang dilakukan oleh A A JE Veggy Priyanka dan I Made Surya Kumara pada tahun 2021 dengan merancang sebuah aplikasi untuk mengklasifikasikan penyakit padi berdasarkan warna dan tekstur daun menggunakan metode CNN dengan augmentasi data pada dataset. Didapatkan hasil model CNN pada skenario keempat yang dilakukan pada citra penyakit padi mendapatkan akurasi sebesar 95,24%.

Ada juga penelitian yang dilakukan oleh Syed Md. Minhaz Hossain et al (2020) dengan judul “*Rice Leaf Diseases Recognition Using Convolutional Neural Networks*” dimana peneliti mengusulkan model baru berbasis Convolutional Neural Network untuk mengenali penyakit daun padi dengan mengurangi parameter jaringan serta melakukan augmentasi data citra. Model yang diusulkan mencapai akurasi training tertinggi 99,78% dan akurasi validasi 97,35%. Efektivitas model yang diusulkan dievaluasi pada satu set gambar penyakit daun padi independen dengan akurasi terbaik 97,82% dengan area under curve (AUC) 0,99. Untuk membuat sebuah model algoritma yang mempunyai nilai akurasi yang tinggi

dengan metode *Convolutional Neural Network* maka diperlukan jumlah data yang banyak terlebih citra yang diambil juga harus terklasifikasi terlebih dahulu. Beberapa penelitian menyebutkan bahwa untuk membuat sebuah model mempunyai akurasi tinggi dan mengurangi *overfitting* pada model, salah satunya adalah penelitian yang dilakukan oleh (Enkvetchakul & Surinta, 2021) menyebutkan dalam penelitiannya dengan menggunakan eksperimen pada ribuan dataset penyakit daun pada tanaman dimana pada augmentasi data beberapa teknik kombinasi dilakukan menghasilkan 15 dataset baru dari setiap 1 data asli dan berhasil meningkatkan akurasi menjadi 76,15% dari dataset yang dilatih dengan tanpa melakukan augmentasi data yang nilai akurasinya hanya 63,07%.

Selain itu masih ada penelitian yang berjudul "*An Efficient Disease Detection Technique of Rice Leaf Using AlexNet*" yang ditulis oleh Md. Mafiul Hasan Matin et al (2020). Peneliti menerapkan salah satu teknik CNN arsitektur yaitu AlexNet, untuk mendeteksi tiga kelas penyakit daun padi yang disebut sebagai *bacterial blight*, *brown spot*, dan *leaf smut* dimana dataset berasal dari situs *kaggle* dengan total 120 gambar. Penelitian ini menunjukkan akurasi lebih dari 99% karena penyesuaian teknik yang efisien dan augmentasi gambar menjadi total 1000 gambar. Augmentasi data juga memiliki fungsi untuk menambah keberagaman data sehingga data yang digunakan untuk model klasifikasi menjadi lebih kaya akan informasi. Augmentasi data citra dapat dilakukan dengan cara transisi, transformasi, penambahan noise, rotasi, pembesaran, dan lain sebagainya.

Berdasarkan uraian latar belakang dan studi literatur yang telah dilakukan pada beberapa kasus di atas, masih ada peluang penelitian terkait deteksi penyakit

pada tanaman padi dan juga penggunaan salah satu metode *deep learning* yaitu *Convolutional Neural Network* (CNN), seperti contoh dengan membandingkan beberapa arsitektur yang berbeda dan eksperimen jumlah dataset berbeda. Maka peneliti melakukan analisis dan penelitian dengan judul “Deteksi Penyakit Tanaman Padi Berbasis *Deep Learning* Menggunakan *Convolutional Neural Network*”. Penulis juga akan memakai dataset publik penyakit daun padi dengan melakukan teknik-teknik augmentasi data citra yang ada. Dengan adanya penelitian ini diharapkan dapat diketahui tingkat akurasi model yang diperoleh serta seberapa mampu model CNN mengklasifikasi gambar penyakit daun padi dengan lebih baik dan akurat.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan sebelumnya, maka rumusan permasalahan pada penelitian ini adalah sebagai berikut :

- a. Berapa tingkat akurasi yang dihasilkan oleh metode *Convolutional Neural Network* (CNN) dengan beberapa arsitektur yang berbeda?
- b. Manakah model arsitektur *Convolutional Neural Network* (CNN) yang menghasilkan nilai akurasi tertinggi?
- c. Bagaimana pengaruh augmentasi data pada tingkat akurasi deteksi penyakit tanaman padi menggunakan *Convolutional Neural Network* (CNN)?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini antara lain :

- a. Metode yang digunakan untuk klasifikasi citra adalah *Convolutional Neural Network (CNN)*.
- b. Dataset yang digunakan untuk penelitian ini bersifat publik yang didapatkan dari website www.kaggle.com
- c. Klasifikasi citra daun padi dibagi menjadi 3 kriteria yaitu *blast*, *blight*, dan *tungro*.
- d. Masing-masing kriteria dataset asli sebelum augmentasi berjumlah 80 gambar sehingga total ada 240 citra penyakit daun padi.
- e. Platform penelitian menggunakan *Jupyter Notebook*.
- f. Target penelitian hanya modeling akurasi uji dengan *python* tidak sampai implementasi pembuatan aplikasi *end-user*.

1.4. Tujuan Penelitian

Penelitian ini dilakukan untuk mencapai beberapa tujuan berikut ini :

- a. Mengetahui bagaimana implementasi metode *deep learning* menggunakan CNN untuk mendeteksi penyakit pada tanaman padi dengan teknik-teknik augmentasi yang tepat.
- b. Mencari rancangan arsitektur model klasifikasi dengan nilai akurasi yang terbaik pada identifikasi penyakit daun tanaman padi menggunakan *Convolutional Neural Network (CNN)*.

- c. Model dapat digunakan sebagai bahan penelitian lanjutan yang harapannya dapat membantu para petani dalam mengendalikan penyakit pada tanaman padi secara tepat dan akurat.

1.5. Manfaat Penelitian

Penelitian ini dilakukan agar dapat memberikan manfaat sebagai berikut :

- a. Hasil rancangan model klasifikasi dapat digunakan untuk mempercepat penanganan dalam mengidentifikasi penyakit tanaman padi.
- b. Rancangan model yang diterapkan dapat dijadikan pembelajaran sebagai tambahan ilmu pengetahuan baru.
- c. Memberikan kontribusi penelitian pada penerapan arsitektur *Convolutional Neural Network* (CNN) yang cocok digunakan dan menghasilkan tingkat akurasi yang tinggi.
- d. Hasil dari penelitian sebagai salah satu syarat penyelesaian studi.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Sebelum melaksanakan penelitian terkait identifikasi penyakit pada daun tanaman padi, peneliti terlebih dahulu melakukan peninjauan pustaka terkait penelitian-penelitian yang sudah dilakukan oleh para peneliti sebelumnya sebagai acuan dalam pengembangan penelitian yang akan dilakukan. Dari peninjauan pustaka yang dilakukan peneliti didapatkan beberapa kajian penelitian terdahulu yang berkaitan dan relevan pada penelitian yang dilaksanakan.

Penelitian dengan judul "*Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation*" yang dilakukan oleh (Suseno, Minarno, & Azhar, 2023) mempunyai tujuan untuk mengklasifikasikan penyakit daun padi dan teknik segmentasi citra pada dataset daun padi untuk mengubah citra menjadi bentuk yang lebih mudah untuk dianalisis. Data yang digunakan terdiri dari 3 kelas *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Kemudian dilakukan segmentasi menggunakan dua teknik, yaitu metode *threshold* dan *k-means*. Selanjutnya, dilakukan *augmentasi* data untuk membuat dataset yang digunakan memiliki jumlah yang besar dan bervariasi. Pelatihan menggunakan model VGG16 dengan nilai hiperparameter *optimizer RMSProp* dan tingkat pembelajaran 0,001. Hasil yang diperoleh adalah akurasi sebesar 91,66% pada skenario dengan dataset *kmeans*. Metode klasifikasi menggunakan model yang telah di-pretrained VGG16 dengan teknik segmentasi citra belum dapat

mengungguli beberapa penelitian sebelumnya. Oleh karena itu, diperlukan penelitian lanjutan dengan melakukan augmentasi data dan metode segmentasi lainnya. Penelitian lain dilakukan oleh (Aggarwal et al., 2022) dengan judul "*Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improve AgroBusiness*" dengan tujuan menganalisis tren penelitian dalam domain identifikasi penyakit padi menggunakan kecerdasan buatan menggunakan analisis global, kutipan per tahun dan per negara, dan sebagainya untuk mendukung berbagai peneliti yang bekerja di domain ini. Didapatkan kesimpulan bahwa teknik *artificial intelligence* dan *machine learning* dapat mengimprovisasi agrobisnis dengan spesifik dalam hal deteksi penyakit padi dengan hasil yang berbeda-beda. Terdapat 29 penelitian yang direview dan terakhir tahun 2020 sebanyak 4 paper, sebaiknya ditambah lagi referensi dengan penelitian-penelitian yang lebih baru dan beragam.

Artikel yang ditulis oleh (Jatmika & Saputra, 2022) membahas tentang penerapan *deep learning* untuk mengklasifikasikan atau mengidentifikasi penyakit pada daun padi menggunakan aplikasi *mobile*. Sistem ini akan memudahkan pengguna dalam mendiagnosa penyakit dengan menampilkan hasil diagnosa berupa nama penyakit beserta taksonominya, deskripsi penyakit dan rekomendasi obat untuk solusi penyakitnya. Dari pengujian yang telah dilakukan, disimpulkan bahwa sistem dengan model yang dibuat menggunakan pendekatan transfer learning menghasilkan akurasi yang baik sebesar 90%. Dataset yang digunakan berjumlah 800 gambar serta model lain hanya memperoleh akurasi 62% karena tanpa menggunakan metode *transfer learning* sehingga mengalami *overfitting*. Artikel

lain oleh (Tejaswini, Singh, Ramchandani, Rathore, & Janghel, 2022) mempelajari berbagai pendekatan *machine learning* dan *deep learning* untuk mendeteksi penyakit pada daunnya dengan menghitung *accuracy*, *recall*, dan *precision* untuk mengukur kinerjanya. Penelitian ini diharapkan dapat membantu para petani untuk mendeteksi penyakit pada daun padi agar mendapatkan hasil panen yang sehat. Setelah menganalisis semua model *deep learning* peneliti mendapatkan hasil model konvolusi 5 lapis memiliki akurasi terbaik 78.2%, sementara yang lain seperti VGG19 mendapat akurasi 72.4%, Xception 72.2%, ResNet50 72.2%, dan VGG16 memiliki akurasi paling rendah yaitu 58,4%. Di masa depan penelitian bisa diperlebar cakupannya untuk memasukkan lebih banyak penyakit dan algoritma, serta membuat deteksi penyakit menjadi lebih luas, lebih mudah, dan lebih cepat.

Ada juga penelitian berjudul “Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network” yang dilakukan oleh (Khoiruddin, Junaidi, & Saputra, 2022) dimana penelitian tentang klasifikasi penyakit daun padi menggunakan algoritma CNN bertujuan untuk membantu mengedukasi para petani dalam mengatasi penyakit daun tanaman padi. Dataset berasal dari *kaggle* dengan nama *Leaf Rice Disease*. Hasil yang didapatkan untuk epoch 25, 50, 75 dan 100 memiliki akurasi yang beragam. Hasil akurasi training terbaik ada di epoch 100 dengan akurasi 98% dan dilakukan testing menggunakan *confusion matrix* mendapatkan nilai rata-rata akurasi sebesar 98%. Sayangnya penelitian ini hanya menggunakan 1 arsitektur CNN custom sehingga tidak ada hasil pembandingan lainnya. Penelitian lain oleh (Julianto, Sunyoto, & Wibowo, 2022) dengan judul “Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi

Penyakit Tanaman Padi” dimana arsitektur MobileNetV2 digunakan untuk mengklasifikasi penyakit daun padi dari dataset *Leaf Rice Disease* karena yang memiliki ukuran yang kecil namun dengan performa yang baik. Untuk meningkatkan performa dari model CNN, akan dilakukan optimasi hyperparameter yang terdiri *epoch*, *batch size*, *learning rate* dan *optimizer*. Penelitian ini bertujuan untuk untuk mendapatkan hyperparameter yang optimal sehingga memberikan performa yang baik pada model CNN. Berdasarkan percobaan yang sudah dilakukan, penentuan hyperparameter sangat berpengaruh terhadap performa model. Hyperparameter dengan jumlah epoch 100, batch size 32, learning rate 0,001 dan optimizer RMSProp memberikan hasil yang paling optimal dengan nilai accuracy 97,56%, precision 97,64%, recall 97,57% dan f1-score 97,57%. Pada penelitian ini hanya berfokus pada klasifikasi penyakit yang menyerang tanaman padi saja, diharapkan pada penelitian selanjutnya dapat melakukan klasifikasi pada hama yang menyerang tanaman padi. Perlu melakukan perbandingan dengan arsitektur CNN lainnya seperti DenseNet, Resnet dan Alexnet untuk mendapatkan akurasi yang lebih baik.

Makalah dengan judul “Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network” yang dilakukan oleh (Ibrahim et al., 2022) merancang sebuah sistem klasifikasi tingkat kematangan pucuk daun teh dari jenis sampel varietas Assamica Klon (GMB 7) yang berada pada Pusat Penelitian Teh dan Kina Gambung. Penelitian ini menerapkan metode pengolahan citra digital dengan algoritma Convolutional Neural Network (CNN) menggunakan arsitektur VGGNET19 dan ResNet50. Hasil pengujian sistem

memperoleh nilai akurasi terbaik sebesar 97.5% dengan menggunakan arsitektur VGGNET19, Optimizer RMSprop, learning rate 0.01, batch size 32 dan epoch 100. Penelitian ini dapat dikembangkan lagi dengan mengimplementasikan ke dalam sistem yang mampu mengklasifikasi tingkat kematangan pucuk daun teh secara real-time. Pada makalah lainnya "*Automatic Diagnosis of Rice Diseases Using Deep Learning*" yang dilakukan oleh (Deng et al., 2021) dengan tujuan mengembangkan metode diagnosis otomatis penyakit padi dengan *esemble model* dan diimplementasikan dalam aplikasi *smartphone*, metode dikembangkan menggunakan *deep learning* berdasarkan kumpulan dataset yang berisi 33.026 gambar enam jenis penyakit padi. Akurasi keseluruhan mencapai 91% yang dianggap cukup baik, mengingat kemiripan beberapa jenis penyakit padi. Serta aplikasi *smartphone* memungkinkan klien menggunakan model *ensemble* di server web melalui jaringan yang nyaman dan efisien untuk diagnosis penyakit padi. Keterbatasannya adalah Ensemble Model memiliki banyak parameter yang dapat mempengaruhi kecepatan identifikasi. Penelitian selanjutnya akan dilakukan pada jaringan *pruning* untuk mengurangi jumlah parameter.

(Purbasari, Rahmat, & Putra PN, 2021) juga melakukan penelitian yang bertujuan untuk mendeteksi penyakit pada tanaman padi dengan mengamati daun yang dapat menyebabkan penurunan produksi beras atau dapat mengakibatkan kualitas beras yang buruk, menggunakan pendekatan kecerdasan buatan dengan dataset 2239 gambar penyakit daun padi dari sumber publik. Metode yang digunakan dalam penelitian ini adalah Convolutional Neural Network (CNN). Penelitian ini dianggap telah berhasil mendeteksi penyakit pada citra daun padi

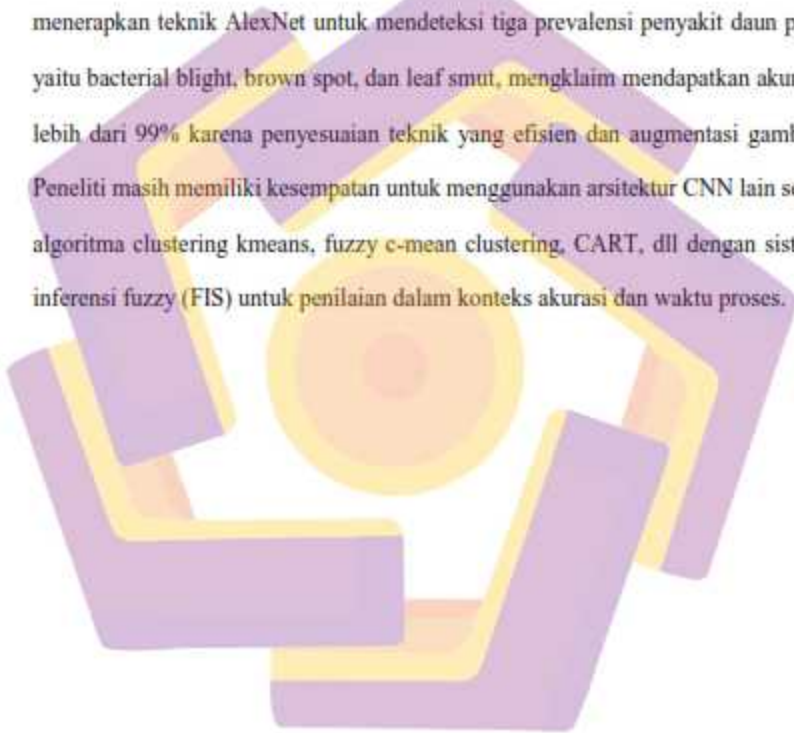
secara otomatis untuk 4 jenis penyakit yaitu blast, blight, brownspot, dan tungro dengan akurasi pelatihan terbaik yang diperoleh sebesar 91%. Hasil eksperimen masih dapat ditingkatkan lagi dengan bereksperimen pada lebih banyak variasi arsitektur CNN dan jumlah lapisan tersembunyi serta node di setiap lapisan. (Priyanka & Kumara, 2021) juga melakukan penelitian guna merancang sebuah aplikasi untuk mengklasifikasikan penyakit padi berdasarkan warna dan tekstur daun menggunakan metode CNN dengan augmentasi data pada dataset. Penelitian ini menggunakan tujuh kelas penyakit tanaman padi yang masing-masing memiliki total 15 gambar. Model CNN pada skenario keempat yang dilakukan pada citra penyakit padi mendapatkan akurasi sebesar 95,24% menggunakan 100 epoch dan menggunakan augmentasi data *zoom range* dan *shear range*. Dataset awal yang digunakan masih sedikit yaitu hanya 105 gambar, ehingga 3 skenario proses pengujian memiliki kinerja buruk pada model yang dilatih. Peneliti menduga bahwa jumlah data pelatihan tidak kompatibel dengan dataset pengujian karena jaringan saraf convolutional membutuhkan data besar berlabel manual untuk proses pelatihan.

Selanjutnya (Oktaviana, Hendrawan, Annas, & Wicaksono, 2021) melakukan penelitian dengan judul "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101" dimana peneliti mengembangkan model yang dapat mengklasifikasikan tiga jenis penyakit tanaman padi dengan pada dataset Rice Leaf Disease sebanyak 120 gambar. Ketiga jenis penyakit tersebut diantaranya *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Smut*. Model dikembangkan dengan menggunakan metode *transfer learning* dengan *pretrained*

model Resnet101 dengan tambahan beberapa lapisan arsitektur. Penggunaan metode deep CNN dengan menggunakan *pretrained* model dari ResNet101 dan tambahan arsitektur pengusul pada bagian *Fully Connected Layer* dalam proses klasifikasi penyakit padi dengan hasil performa accuracy pada *classification report* sebesar 100% dengan data validasi, dengan nilai validation loss rendah yaitu 5,61%. Penelitian kedepannya disarankan untuk menggunakan jumlah kelas dan dataset yang lebih banyak sehingga klasifikasi jenis padi semakin bervariasi sesuai dengan perkembangan penyakit padi kedepannya. Selain hal tersebut, dalam membuat model disarankan mencoba menggunakan metode yang lebih bervariasi seperti menerapkan metode *ensemble learning* yang dapat meningkatkan optimasi model. Penelitian juga dilakukan oleh (Andrianto, Suhardi, Faizal, & Armandika, 2020) dengan mengembangkan sistem deteksi penyakit padi berbasis *deep learning*, yang terdiri dari aplikasi *machine learning* di *server cloud* dan aplikasi di *smartphone* menunjukkan bahwa aplikasi pendeteksi penyakit tanaman padi berbasis *smartphone* berfungsi dengan baik, yaitu mampu mendeteksi penyakit pada tanaman padi dengan kinerja sistem pendeteksi penyakit tanaman padi memiliki nilai akurasi latih 100% dan nilai akurasi uji 60%. Dataset berasal dari kaggle berjumlah 1600 gambar tetapi tidak dijelaskan detailnya, serta nilai akurasi pengujian dapat ditingkatkan dengan kualitas dataset yang baik serta mencoba model yang lain.

Penelitian lain dilakukan oleh (Hossain et al., 2020) dimana peneliti mengusulkan model baru berbasis CNN untuk mengenali penyakit daun padi dengan mengurangi parameter jaringan dengan dataset 4199 gambar penyakit daun

padi menghasilkan nilai akurasi training tertinggi mencapai 99,78% dan akurasi validasi 97,35%. Meskipun hasil akurasi sudah cukup tinggi tetapi dataset yang digunakan adalah data privat berasal dari daerah di Bangladesh saja. (Matin, Khatun, Moazzam, & Uddin, 2020) melakukan penelitian dengan judul "*An Efficient Disease Detection Technique of Rice Leaf Using AlexNet*" dengan menerapkan teknik AlexNet untuk mendeteksi tiga prevalensi penyakit daun padi yaitu bacterial blight, brown spot, dan leaf smut, mengklaim mendapatkan akurasi lebih dari 99% karena penyesuaian teknik yang efisien dan augmentasi gambar. Peneliti masih memiliki kesempatan untuk menggunakan arsitektur CNN lain serta algoritma clustering kmeans, fuzzy c-mean clustering, CART, dll dengan sistem inferensi fuzzy (FIS) untuk penilaian dalam konteks akurasi dan waktu proses.



2.2. Keaslian Penelitian

Tabel 2.1. Matriks Literatur Review dan Posisi Penelitian
Deteksi Penyakit Tanaman Padi Berbasis Deep Learning Menggunakan Convolutional Neural Network

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

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			jumlah yang besar dan bervariasi.			
2	Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improve Agro-Business	Shruti Aggarwal, M. Suchithra, N. Chandramouli, Macha Sarada, Amit Verma, D. Vetrithangam, Bhaskar Pant, and Biruk Ambachew Adugna. Hindawi Scientific Programming 2022	Menganalisis tren penelitian dalam domain identifikasi penyakit padi menggunakan kecerdasan buatan menggunakan analisis global, kutipan per tahun dan per negara, dan sebagainya untuk mendukung berbagai peneliti yang bekerja di domain ini.	Teknik artificial intelligence dan machine learning dapat mengimprovisasi agrobisnis dengan spesifik dalam hal deteksi penyakit padi dengan hasil yang berbeda-beda.	Terdapat 29 penelitian yang direview dan terakhir tahun 2020 sebanyak 4 paper, sebaiknya ditambah lagi referensi dengan penelitian-penelitian yang lebih baru dan beragam.	Penelitian Aggarwal et al, 2022 merupakan literature review tentang deteksi penyakit tanaman padi menggunakan teknik AI dan ML, sedangkan penelitian yang akan dilakukan adalah implementasi CNN untuk deteksi penyakit tanaman padi dengan beberapa arsitektur yang berbeda.
3	Rice Plants Disease Identification Using Deep Learning with Convolutional Neural Network Method	Sunu Jatmika, Danang Eka Saputra. Sinkron : Jurnal dan Penelitian Teknik Informatika. 2022	Penelitian ini membahas tentang penerapan deep learning untuk mengklasifikasikan atau mengidentifikasi penyakit pada daun padi menggunakan aplikasi mobile. Sistem ini akan memudahkan pengguna dalam mendiagnosa penyakit dengan	Dari pengujian yang telah dilakukan, disimpulkan bahwa sistem dengan model yang dibuat menggunakan pendekatan transfer learning menghasilkan akurasi yang baik sebesar 90%. Sedangkan sistem dengan model lainnya memperoleh akurasi sebesar 62%.	Dataset yang digunakan hanya berjumlah 800 gambar. Selain itu model lain hanya memperoleh akurasi 62% karena tanpa menggunakan metode transfer learning sehingga mengalami overfitting.	Penelitian (Jatmika dan Saputra, 2022) membuat implementasi CNN pada aplikasi mobile dengan 2 model. Sedangkan penelitian yang akan dilakukan hanya focus membangun model (tanpa implementasi ke aplikasi mobile) tetapi dengan percobaan

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			menampilkan hasil diagnosa berupa nama penyakit beserta taksonominya, deskripsi penyakit dan rekomendasi obat untuk solusi penyakitnya.			arsitektur yang lebih beragam.
4	Rice Leaf Disease Classification Using CNN	Pallapothala Tejaswini, Priyanshi Singh, Monica Ramchandani, Yogesh Kumar Rathore, Rekh Ram Janghel, IOP Conf. Series: Earth and Environmental Science. 2022	Peneliti mempelajari berbagai pendekatan ML dan deep learning untuk mendeteksi penyakit pada daunnya dengan menghitung accuracy, recall, dan precision untuk mengukur kinerjanya. Penelitian ini diharapkan dapat membantu para petani untuk mendeteksi penyakit pada daun padi agar mendapatkan hasil panen yang sehat.	Setelah menganalisis semua model deep learning peneliti mendapatkan hasil model konvolusi 5 lapis memiliki akurasi terbaik 78,2%, sementara yang lain seperti VGG19 mendapat akurasi 72,4%, Xception 72,2%, ResNet50 72,2%, dan VGG16 memiliki akurasi paling rendah yaitu 58,4%.	Di masa depan peneliti akan memperlebar cakupan untuk memasukkan lebih banyak penyakit dan algoritma, serta membuat deteksi penyakit menjadi lebih luas, lebih mudah, dan lebih cepat.	Penelitian (Tejaswini et al, 2022) menggunakan dataset dengan total 1600 gambar. Sedangkan pada penelitian yang akan dilakukan menggunakan dataset yang berbeda dan arsitektur custom yang berbeda serta transfer learning yang beragam.
5	Klasifikasi Penyakit Daun Padi Menggunakan	Mohtar Khoiruddin, Apri Junaidi, Wahyu Andi Saputra. Journal of Dinda :	Peneliti melakukan penelitian tentang klasifikasi penyakit daun padi menggunakan	Hasil yang didapatkan untuk epoch 25, 50, 75 dan 100 memiliki akurasi yang beragam. Hasil akurasi	Hanya menggunakan 1 arsitektur CNN custom sehingga tidak ada hasil pembandingan. Sebaiknya	Penelitian Khoiruddin et al, 2022 hanya menggunakan 1 arsitektur CNN yang

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Convolutional Neural Network	Data Science, Information Technology, and Data Analytics. 2022	algoritma CNN bertujuan untuk membantu edukasi para petani dalam mengatasi penyakit daun tanaman padi. Dataset berasal dari kaggle dengan nama Leaf Rice Disease Indonesia.	training terbaik ada di epoch 100 dengan akurasi 98% dan dilakukan testing menggunakan confusion matrix mendapatkan nilai rata-rata akurasi sebesar 98%.	juga menggunakan arsitektur yang sudah ada seperti VGG16, ResNet, dan lain sebagainya.	dibuat penulis sendiri, sedangkan penelitian yang akan dilakukan juga menggunakan arsitektur-arsitektur populer yang ada sebagai pembandingan.
6	Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi	Afis Julianto, Andi Sunyoto, dan Ferry Wahyu Wibowo. Jurnal TEKNI MEDIA: Teknologi Informasi dan Multimedia. 2022	Arsitektur MobileNet-V2 digunakan untuk mengklasifikasi penyakit daun padi dari dataset Leaf Rice Disease Indonesia karena yang memiliki ukuran yang kecil namun dengan performa yang baik. Untuk meningkatkan performa dari model CNN, akan dilakukan optimasi hyperparameter yang terdiri epoch, batch size, learning rate dan optimizer. Penelitian ini bertujuan untuk untuk	Berdasarkan percobaan yang sudah dilakukan, penentuan hyperparameter sangat berpengaruh terhadap performa model. Hyperparameter dengan jumlah epoch 100, batch size 32, learning rate 0,001 dan optimizer RMSProp memberikan hasil yang paling optimal dengan nilai accuracy 97,56%, precision 97,64%, recall 97,57% dan f1-score 97,57%.	Pada penelitian ini hanya berfokus pada klasifikasi penyakit yang menyerang tanaman padi saja, diharapkan pada penelitian selanjutnya dapat melakukan klasifikasi pada hama yang menyerang tanaman padi. Perlu melakukan perbandingan dengan arsitektur CNN lainnya seperti DenseNet, Resnet dan Alexnet untuk mendapatkan akurasi yang lebih baik.	Penelitian Julianto et al, 2022 hanya menggunakan 1 arsitektur CNN yaitu MobileNet-V2, sedangkan penelitian penulis akan menggunakan beberapa arsitektur CNN yang berbeda sebagai pembandingan.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			mendapatkan hyperparameter yang optimal sehingga memberikan performa yang baik pada model CNN.			
7	Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network	Nur Ibrahim, Gita Ayu Lestary, Faniesia Saufana Hanafi, Khaerudin Saleh, Nor Kumalasari Caecar Pratiwi, Muthia Syafika Haq, Adhi Irianto Mastur. ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika. 2022	Pada penelitian ini dirancang sebuah sistem klasifikasi tingkat kematangan pucuk daun teh dari jenis sampel varietas Assamica Klon (GMB 7) yang berada pada Pusat Penelitian Teh dan Kina Gambung. Penelitian ini menerapkan metode pengolahan citra digital dengan algoritma Convolutional Neural Network (CNN) menggunakan Arsitektur VGGNET19 dan ResNet50.	Hasil pengujian sistem memperoleh nilai akurasi terbaik sebesar 97.5% dengan menggunakan arsitektur VGGNET19, Optimizer RMSprop, learning rate 0.01, batch size 32 dan epoch 100.	Penelitian ini dapat dikembangkan lagi dengan mengimplementasikan ke dalam sistem yang mampu mengklasifikasi tingkat kematangan pucuk daun teh secara real-time.	Penelitian yang dilakukan (Ibrahim et al., 2022) menggunakan dataset untuk objek gambar pucuk daun teh, sedangkan penelitian yang akan dilakukan untuk objek berbeda yaitu gambar penyakit daun tanaman padi.
8	Automatic Diagnosis of Rice	Ruoling Deng, Ming Tao, Hang Xing,	Dalam penelitian ini metode diagnosis	Akurasi keseluruhan mencapai 91% yang	Keterbatasannya adalah Ensemble Model	Penelitian (Deng et al., 2021) menggunakan

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Diseases Using Deep Learning	Xiuli Yang, Chuang Liu, Kaifeng Liao and Long Qi. Frontiers in Plant Science. 2021	otomatis dikembangkan dan diimplementasikan pada aplikasi smartphone. Metode ini dikembangkan menggunakan deep learning berdasarkan kumpulan dataset yang berisi 33.026 gambar enam jenis penyakit padi.	dianggap cukup baik, mengingat kemiripan beberapa jenis penyakit padi. Serta aplikasi smartphone memungkinkan klien menggunakan model ensemble di server web melalui jaringan yang nyaman dan efisien untuk diagnosis penyakit padi.	memiliki banyak parameter yang dapat mempengaruhi kecepatan identifikasi. Penelitian selanjutnya akan dilakukan pada jaringan pruning untuk mengurangi jumlah parameter.	pendekatan ensemble model sedangkan penelitian yang akan dilakukan menggunakan pendekatan arsitektur transfer learning dan custom model.
9	Detection of Rice Plant Diseases using Convolutional Neural Network	I Y Purbasari, B Rahmat, and C S Putra PN. IOP Conference Series: Materials Science and Engineering. 2021	Penelitian ini bertujuan untuk mendeteksi penyakit pada tanaman padi dengan mengamati daun yang dapat menyebabkan penurunan produksi beras atau dapat mengakibatkan kualitas beras yang buruk, menggunakan pendekatan kecerdasan buatan. Metode yang digunakan dalam penelitian ini adalah Convolutional Neural Network (CNN).	Penelitian ini dianggap telah berhasil mendeteksi penyakit pada citra daun padi secara otomatis untuk 4 jenis penyakit yaitu blast, blight, brownspot, dan tungro dengan akurasi pelatihan terbaik yang diperoleh sebesar 91%.	Hasil eksperimen ini dapat ditungkatkan lagi dengan bereksperimen pada lebih banyak variasi arsitektur CNN dan jumlah lapisan tersembunyi serta node di setiap lapisan.	Penelitian (Purbasari et al., 2021) menggunakan beberapa arsitektur custom sendiri sehingga hasil akurasi kurang tinggi. Pada penelitian yang akan dilakukan penulis tidak hanya mencoba menggunakan arsitektur custom sendiri tetapi juga menggunakan arsitektur CNN yang sudah populer dan teruji sebelumnya.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
10	Classification of Rice Plant Diseases Using the Convolutional Neural Network Method	A A JE Veggy Priyanka, I Made Surya Kumara. Lontar Komputer : Jurnal Ilmiah Teknologi Informasi. 2021	Penelitian ini merancang sebuah aplikasi untuk mengklasifikasi penyakit padi berdasarkan warna dan tekstur daun menggunakan metode CNN dengan augmentasi data pada dataset. Penelitian ini menggunakan tujuh kelas penyakit tanaman padi yang masing-masing memiliki total 15 gambar.	Model CNN pada skenario keempat yang dilakukan pada citra penyakit padi mendapatkan akurasi sebesar 95,24% menggunakan 100 epoch dan menggunakan augmentasi data zoom range dan shear range.	Dataset awal terlalu sedikit yaitu hanya 105 gambar. Sehingga 3 skenario proses pengujian memiliki kinerja buruk pada model yang dilatih. Peneliti menduga bahwa jumlah data pelatihan tidak kompatibel dengan dataset pengujian karena jaringan saraf convolutional membutuhkan data besar berlabel manual untuk proses pelatihan.	Penelitian (Priyanka dan Kumara, 2021) menggunakan arsitektur CNN VGG-16, sedangkan penelitian yang akan dilakukan menggunakan beberapa perbandingan arsitektur dan jumlah dataset yang lebih besar.
11	Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101	Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, Galih Wasis Wicaksono. Rumah Jurnal Elektronik Ikatan Ahli Informatika Indonesia (IAII). 2021	Penelitian ini mengembangkan model yang dapat mengklasifikasi tiga jenis penyakit tanaman padi dengan pada dataset Rice Leaf Disease. Ketiga jenis penyakit tersebut diantaranya: Bacterial Leaf Blight, Brown Spot, dan Leaf	Penggunaan metode Deep CNN dengan menggunakan pretrained model dari ResNet101 dan tambahan arsitektur pengusul pada bagian Fully Connected Layer dalam proses klasifikasi penyakit padi dengan hasil performa accuracy pada classification report sebesar 100% dengan	Penelitian kedepannya disarankan untuk menggunakan jumlah kelas dan dataset yang lebih banyak sehingga klasifikasi jenis padi semakin bervariasi sesuai dengan perkembangan penyakit padi kedepannya. Selain hal tersebut, dalam membuat	Penelitian yang dilakukan (Oktaviana dkk, 2021) hanya menggunakan arsitektur CNN Resnet101, sedangkan penelitian yang akan dilakukan menggunakan beberapa arsitektur berbeda. Dataset juga akan berbeda dan lebih

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			Smut. Model dikembangkan dengan menggunakan metode transfer learning dengan pretrained model Resnet101 dengan tambahan beberapa lapisan arsitektur.	data validasi, dengan nilai validation loss rendah yaitu 5,61%.	model disarankan mencoba menggunakan metode yang lebih bervariasi seperti menerapkan metode ensemble learning yang dapat meningkatkan optimasi model.	banyak, karena penelitian Oktaviana dkk 2021 hanya menggunakan total 120 gambar untuk 3 kelas yang diambil dari UCI Machine Learning Repository dengan judul data Rice Leaf Diseases Data Set.
12	Smartphone Application for Deep Learning-Based Rice Plant Disease Detection	Heri Andrianto, Suhardi, Ahmad Faizal, Fladio Armandika. International Conference on Information Technology Systems and Innovation (ICTISI). 2020	Mengembangkan sistem deteksi penyakit padi berbasis deep learning yang terdiri dari aplikasi machine learning di server cloud dan aplikasi di smartphone. Aplikasi smartphone berfungsi untuk menangkap gambar daun tanaman padi, mengirimkannya ke aplikasi di cloud server, dan menerima hasil klasifikasi berupa informasi jenis penyakit tanaman.	Hasil penelitian menunjukkan bahwa aplikasi pendeteksi penyakit tanaman padi berbasis smartphone berfungsi dengan baik sehingga dianggap mampu mendeteksi penyakit pada tanaman padi. Kinerja sistem pendeteksi penyakit tanaman padi dengan arsitektur VGG16 memiliki nilai akurasi latih 100% dan nilai akurasi uji 60%.	Dataset berasal dari kaggle berjumlah 1600 gambar tetapi tidak dijelaskan detail link ataupun namanya. Nilai akurasi pengujian dapat ditingkatkan dengan menambah jumlah dan kualitas dataset.	Penelitian (Andrianto et al, 2020) menggunakan arsitektur CNN VGG-16, sedangkan penelitian yang akan dilakukan menggunakan beberapa perbandingan arsitektur dan menambah sumber dataset yang berbeda.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
13	Rice Leaf Diseases Recognition Using Convolutional Neural Networks	Syed Md. Minhaz Hossain, Md. Monjur Morhsed Tanjil, Mohammed Abser Bin Ali, Mohammad Zihadul Islam, Md. Saiful Islam, Sabrina Mobassirin, Iqbal H. Sarker & S. M. Riazul Islam, International Conference on Advanced Data Mining and Applications, 2020	Dalam makalah ini, peneliti mengusulkan model baru berbasis Convolutional Neural Network untuk mengenali penyakit daun padi dengan mengurangi parameter jaringan serta melakukan augmentasi data citra.	Model yang diusulkan mencapai akurasi training tertinggi 99,78% dan akurasi validasi 97,35%. Efektivitas model yang diusulkan dievaluasi pada satu set gambar penyakit daun padi independen dengan akurasi terbaik 97,82% dengan area under curve (AUC) 0,99. Selain itu, eksperimen klasifikasi biner telah dilakukan dan model yang diusulkan mencapai tingkat pengenalan masing-masing sebesar 97%, 96%, 96%, 93%, dan 95% untuk kelas Blast, Brownspot, Bakterial Daun, Sheath Blight dan Tungro. Hasil ini menunjukkan efektivitas dan keunggulan model peneliti.	Meskipun memiliki akurasi yang cukup baik, peneliti bertujuan untuk meningkatkan keandalan dan ketahanan model pada dataset yang berbeda dari wilayah lain atau untuk kasus penyakit lain.	Penelitian (Hossain et al, 2020) menggunakan model CNN custom dan dataset privat 5 penyakit padi di Bangladesh. Sedangkan penelitian yang akan dilakukan menggunakan beberapa arsitektur CNN yang lain dan dataset publik.
14	An Efficient Disease Detection Technique of Rice	Md. Mafizul Hasan Matin, Amina Khatun, Md. Golam	Dalam makalah ini peneliti menerapkan salah satu teknik CNN	Penelitian ini menunjukkan akurasi lebih dari 99% karena penyesuaian teknik	Peneliti masih memiliki kesempatan untuk menggunakan arsitektur	Penelitian (Matin et al 2020) hanya menggunakan arsitektur

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Leaf Using AlexNet	Moazzam, Mohammad Shorif Uddin. Journal of Computer and Communications. 2020	arsitektur yaitu AlexNet, untuk mendeteksi tiga kelas penyakit daun padi yang disebut sebagai bacterial blight, brown spot, dan leaf smut. Dataset dari kaggle dengan total 120 gambar.	yang efisien dan augmentasi gambar menjadi total 1000 gambar.	CNN lain serta algoritma clustering k-means, fuzzy c-mean clustering, CART, dll dengan sistem inferensi fuzzy (FIS) untuk penilaian dalam konteks akurasi dan waktu proses.	AlexNet. Pada penelitian yang akan dilakukan menerapkan uji coba untuk arsitektur-arsitektur CNN lainnya seperti VGG16, Xception, dan lain-lain.

2.3. Landasan Teori

2.3.1. Penyakit Tanaman

Salah satu faktor yang mempengaruhi tinggi rendahnya produksi padi adalah adanya serangan organisme pengganggu tanaman. Organisme pengganggu tanaman (OPT) adalah semua organisme yang dapat menyebabkan dan menimbulkan kerusakan fisik, gangguan fisiologi dan biokimia, atau kompetisi hara terhadap tanaman budidaya. Organisme yang menjadi penyebab penyakit pada tanaman adalah patogen. Patogen yang dapat menyebabkan penyakit pada tanaman antara lain golongan cendawan, bakteri molikut (bakteri tanpa dinding sel), nematoda, protozoa dan virus (Walascha et al., 2021). Patogen tersebut akan mengganggu tanaman secara terus menerus, akibatnya aktivitas sel tanaman menjadi abnormal sehingga menimbulkan gejala pada tanaman. Salah satu bagian tanaman padi yang mendapat serangan patogen adalah daun.

Dalam penelitian ini terdapat 3 kelas penyakit daun paling populer dari 5 penyakit daun yang biasanya menyerang tanaman padi (Walascha et al., 2021), diantaranya :

1. *Blast*

Penyakit *blast* daun disebabkan oleh jamur *Magnaporthe grisea* dengan gejala adanya bercak-bercak yang runcing pada bagiannya ujung bercak, tepi bercak berwarna coklat dan berwarna kelabu atau keputih-putihan pada bagian pusat bercak. Bercak akan terus berkembang dan dikelilingi oleh area yang berwarna kuning pucat (Walascha et al., 2021).

Contoh gambar penyakit *blast* pada daun padi dapat dilihat pada Gambar 2.1.



Gambar 2.1. Penyakit *Blast* Daun Padi

2. Blight

Penyakit *blight* atau juga disebut hawar daun disebabkan oleh bakteri *Xanthomonas campestris* dengan gejala warna daun berubah menjadi kelabu atau abu-abu. Perkembangan penyakit ini menyebabkan helaian daun tanaman mengering dan melipat sepanjang ibu tulang daun (Walascha et al., 2021). Contoh gambar penyakit *blight* pada daun padi dapat dilihat pada Gambar 2.2.



Gambar 2.2. Penyakit *Blight* Daun Padi

3. *Tungro*

Tungro adalah penyakit virus pada padi yang biasanya menyerang pada fase pertumbuhan vegetatif dan menyebabkan tanaman tumbuh kerdil dan berkurangnya jumlah anakan. Pelepah dan helaian daun memendek dan daun yang terserang sering berwarna kuning sampai kuning-oranye. Daun muda sering berlurik atau strip berwarna hijau pucat sampai putih dengan panjang berbeda sejajar dengan tulang daun. Gejala mulai dari ujung daun yang lebih tua. Daun menguning berkurang bila daun yang lebih tua terinfeksi. Biasanya beberapa bidang lahan terserang sepanjang sawah. Dua spesies wereng hijau *Nephotettix malayanus* dan *N. virescens* adalah serangga utama yang menyebarkan virus *tungro*. Contoh gambar penyakit *tungro* pada daun padi dapat dilihat pada Gambar 2.3.

Gambar 2.3. Penyakit *Tungro* Daun Padi

2.3.2. Citra Digital

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

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

Gambar 2.4. Matriks Citra Digital

Pada Gambar 2.4 merupakan contoh penulisan citra digital ke dalam suatu matriks. Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$ di

mana x dan y adalah koordinat spasial (bidang), dan amplitudo dari f pada setiap pasangan koordinat (x, y) disebut intensitas atau skala keabuan dari bayangan pada titik tersebut. Jika nilai x , y , dan intensitas f semuanya berhingga, besaran diskrit, kita menyebut bayangan itu sebagai bayangan digital (Gonzalez & Woods, 2008).

2.3.2. Pengolahan Citra Digital

Pengolahan citra digital atau bisa juga disebut *image processing* adalah proses pengolahan pada suatu citra hasil keluaran berupa citra juga (Nafi'iyah & Mujilawati, 2018). Awal mula dari pengolahan citra adalah perbaikan untuk citra, kemudian dengan perkembangan zaman dibarengi dengan perkembangan dunia komputer maka *image processing* semakin berkembang dan banyak digunakan untuk berbagai kebutuhan. Seiring perkembangannya, pengolahan citra memiliki tujuan utama sebagai berikut :

- a. Perbaikan kualitas yang dimiliki citra, dimana keluaran dari hasil pengolahan citra ini adalah dapat menampilkan informasi pada citra secara lebih jelas.
- b. Ekstraksi ciri dari suatu citra, yaitu hasil keluaran dari proses pengolahan ini adalah berupa informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik, atau dengan kata lain komputer melakukan interpretasi terhadap informasi yang ada pada citra melalui besaran numerik data yang dibedakan secara jelas.

Pengolahan citra digital dari tujuan utamanya telah banyak dikembangkan pada saat ini untuk diterapkan ke dalam berbagai bidang untuk membantu manusia dalam melakukan analisis dan pekerjaan (Rozaqi, Sunyoto, & Arief, 2021). Bidang-bidang pengembangan pengolahan citra digital pada kehidupan sehari-hari diantaranya :

1. Bidang Pertanian

Pengolahan citra digital pada bidang pertanian dapat dimanfaatkan seperti untuk mengklasifikasikan buah yang matang secara otomatis atau mendeteksi penyakit pada tanaman.

2. Bidang Medis / Kesehatan

Pada bidang ini pengolahan citra digital terletak pada pekerjaan yang membutuhkan suatu ketelitian dalam menghitung dan menganalisis. Contohnya dokter dapat dipermudah pekerjaannya dalam mendiagnosis penyakit yang diderita pasien dari olah gambar rontgen atau pada diagnosis darah manusia.

3. Bidang Industri

Bidang industri sangat membutuhkan kecepatan dan ketelitian yang lebih cepat daripada apabila ditangani manual oleh manusia, contohnya untuk sortir suatu barang yang didasarkan pada perbedaan warna, ukuran, ataupun kode tertentu yang tertera pada suatu barang.

4. Bidang Transportasi

Bidang ini bisa sangat terbantu dengan pengolahan citra contohnya kamera CCTV yang terpasang di jalanan dalam memantau lalu lintas,

menghitung jumlah kendaraan yang melintas, analisis kemacetan, deteksi plat kendaraan di gerbang tol, dan sebagainya.

5. Bidang Pemetaan Lahan

Citra yang diambil oleh satelit dapat diolah untuk mendapatkan informasi seperti menghitung luas suatu hutan di daerah tertentu dan deteksi dini kebakaran hutan.

6. Bidang Fotografi & Film

Fotografi & film adalah bidang yang banyak memanfaatkan pengolahan citra contohnya gambar atau video iklan yang ditampilkan begitu menarik dan enak dipandang mata karena hasil rekayasa citra menggunakan software editing seperti Photoshop.

7. Bidang Pencarian Cerdas

Penerapan pencarian gambar berbasis kecerdasan buatan berdasarkan kandungan citra dapat ditemukan pada *search engine* milik Google, dengan mengunggah suatu citra maka google dapat menampilkan citra yang sejenis dari berbagai sumber.

2.3.3. Klasifikasi

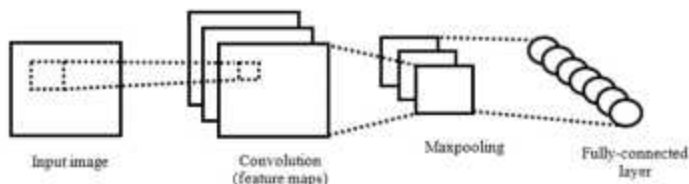
Klasifikasi banyak digunakan dalam berbagai hal, diantaranya untuk deteksi kecurangan, pengolahan pelanggan, deteksi kematangan buah, identifikasi penyakit dan masih banyak lainnya. Klasifikasi sendiri memiliki pengertian yaitu suatu teknik yang digunakan untuk mempelajari sekumpulan data yang mana menghasilkan suatu aturan dan ketentuan tertentu yang bisa digunakan untuk

mengenali data-data yang baru dan belum pernah ditemui sebelumnya (Suyanto, 2017). Klasifikasi dalam *machine learning* dikategorikan ke dalam *supervised learning*, atau pembelajaran yang diawasi. *Supervised learning* memiliki artian bahwa data yang digunakan sudah dikelompokkan pada kelas-kelas yang sudah diberi label.

2.3.4. Convolutional Neural Network

Convolutional Neural Network atau disingkat CNN suatu pengembangan dari Multilayer Perceptron (MLP) yang dapat digunakan untuk mengolah data citra khususnya gambar dua dimensi. *Convolutional Neural Network* (CNN) sering digunakan untuk mengenali benda atau mendeteksi suatu objek (Arrofiqoh & Harintaka, 2018).

Convolutional Neural Network merupakan bagian dari *deep neural network* yang umumnya digunakan untuk pengenalan dan pemrosesan gambar karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada arsitektur *Convolutional Neural Network* (CNN) dapat menangkap informasi kontekstual yang terkadang di dalam data seperti pixel yang saling berdekatan pada sebuah citra, yang berarti model CNN dapat melakukan ekstraksi ciri pada sebuah citra secara mandiri. Adapun arsitektur umum pada CNN dapat dilihat pada Gambar 2.5.



Gambar 2.5. Arsitektur Convolutional Neural Network

Sebuah model *Convolutional Neural Network* pada dasarnya terdiri dari beberapa layer untuk proses penyelesaian klasifikasi citra sebagai berikut :

a. *Input Layer*

Layer atau lapisan ini adalah dimana citra yang akan digunakan sebagai dataset untuk proses klasifikasi dimasukkan ke dalam model klasifikasi.

b. *Convolutional Layer*

Layer atau lapisan ini berfungsi untuk operasi konvolusi (*convolution*) terhadap sejumlah node pada citra menggunakan beberapa *filter*. Pada operasi ini membantu *neural network* (jaringan saraf) untuk mengenali objek seperti citra daun berdasarkan atribut-atributnya.

2	4	9	1	4
2	1	4	4	6
1	1	2	9	2
7	3	5	1	3
2	3	4	8	5

Image

×

1	2	3
-4	7	4
2	-5	1

Filter /
Kernel

=

51	66	20
31	49	101
15	53	-2

Feature

Gambar 2.6. Proses Konvolusi Menggunakan *Filter*

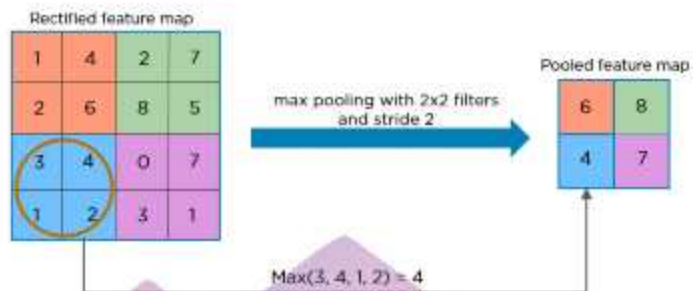
Lapisan konvolusi terdiri dari susunan sedemikian rupa *neuron* yang membentuk sebuah *filter*. Pada Gambar 2.6 adalah contoh proses konvolusi dengan menggunakan filter 3x3, pada proses konvolusi ini piksel pada citra dikalikan dengan piksel *filter* dan menghasilkan keluaran yang biasa disebut *feature map*.

c. *ReLU Layer*

Layer ini merupakan fungsi aktivasi dari output layer sebelumnya. Pada arsitektur *Convolutional Neural Network*, fungsi aktivasi terletak pada perhitungan akhir keluaran *feature map* atau sesudah proses perhitungan konvolusi atau *pooling* untuk menghasilkan suatu pola fitur. Pada fungsi ReLU atau *Rectified Linear Unit* melakukan *threshold* dari 0 hingga *infinity*, dalam artian fungsi ini masukan dari neuron-neuron berupa bilangan negatif, maka fungsi ini akan menerjemahkan nilai tersebut ke dalam nilai 0, dan jika masukan bernilai positif maka *output* dari *neuron* adalah nilai aktivasi itu sendiri.

d. *Pooling Layer*

Pooling layer merupakan lapisan yang mengurangi dimensi dari *feature map*. Pada proses ini digunakan untuk mengurangi resolusi gambar dengan tetap mempertahankan informasi pada gambar, sehingga akan memberikan kecepatan yang lebih dalam melakukan komputasi untuk klasifikasi. Lapisan ini biasa diterapkan setelah lapisan konvolusi.

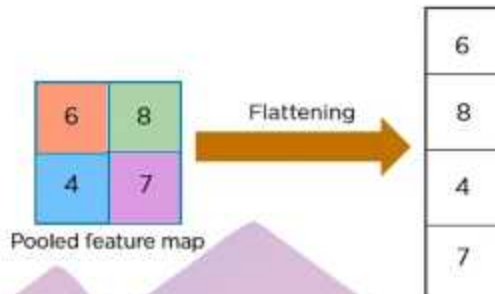


Gambar 2.7. Proses pada *Max Pooling*

Lapisan pooling memiliki beberapa jenis diantaranya adalah *max pooling* dan *average pooling*. Lapisan pooling yang digunakan terdiri dari sebuah *filter* yang memiliki ukuran tertentu dan juga *stride* tertentu, dimana *stride* adalah jumlah pergeseran dari *filter* pada saat proses *pooling*. Gambar 2.7 merupakan contoh proses *max pooling* pada citra dengan ukuran 2×2 dan *stride* 2, dari beberapa nilai diambil nilai yang tertinggi dan menyimpannya di *output*, kemudian *filter* akan bergeser ke kanan sebanyak 2 *stride* atau sebanyak 2 kotak.

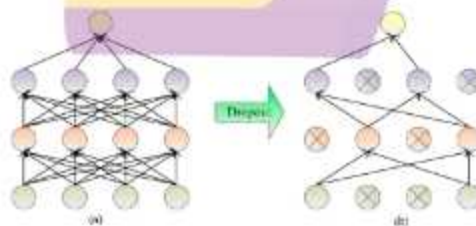
e. *Flattening*

Tahapan *flattening* adalah tahapan merubah matriks yang ada di *pooling layer* menjadi satu kolom atau menjadi sebuah vektor tunggal. Vektor ini akan menjadi bagian dari input layer di *fully-connected layer*. Kita cukup mengambil baris demi baris dan menggabungkan menjadi satu baris satu kolom. Contoh proses *flattening* terdapat pada Gambar 2.8.

Gambar 2.8. Proses *Flattening*

f. *Dropout Regularization*

Dropout Regularization adalah teknik regulasi *neural network* dimana beberapa *neuron* akan dipilih secara acak dan tidak dipakai selama proses pelatihan data. Dengan menghilangkan suatu *neuron* berarti menghilangkannya sementara dari *network* yang tidak menggunakan teknik *dropout*, dengan *neural network* yang menggunakan teknik *dropout*. *Dropout* mengacu pada menghilangkan *neuron* yang berupa *hidden layer* maupun *visible layer* di dalam *neuron*. Setiap *neuron* akan diberikan probabilitas yang bernilai antara 0 dan 1. Berikut contoh penerapan *dropout regularization* ada pada Gambar 2.9.

Gambar 2.9. *Standard Network* (a) dan *Network dengan Dropout* (b)

g. *Fully Connected Layer*

Lapisan ini berfungsi untuk menghitung hasil *output* dari layer konvolusi dan *pooling*. Pada lapisan konvolusi keluaran yang dihasilkan masih dalam bentuk *multi-dimensional array*, maka diperlukan proses bernama *flatten* dimana maksud *flatten* disini adalah mengubah keluaran hasil konvolusi (*feature map*) menjadi bentuk vektor.

2.3.5. Augmentasi Data

Augmentasi data adalah sebuah teknik memanipulasi sebuah data tanpa kehilangan inti atau esensi dari data tersebut. Pada penjelasan sederhananya proses augmentasi adalah proses dimana kita menggandakan data yang kita punya, pada penelitian ini penulis menggunakan data citra penyakit daun padi dengan berbagai macam teknik yang ada seperti memutar gambar agar terlihat berbeda. Fungsi augmentasi digunakan karena dataset yang masih dalam jumlah yang sedikit dan ingin menambahkan ragam dataset menjadi lebih banyak lagi untuk meningkatkan akurasi dan menghindari terjadinya *overfitting*.

2.3.6. Confusion Matrix

Metode *confusion matrix* adalah salah satu metode pengujian untuk kinerja suatu model klasifikasi. *Confusion Matrix* adalah table untuk mengukur kinerja algoritma klasifikasi atau model klasifikasi atau *classifier* (Faisal & Nugrahadi, 2019).

Tabel 2.2. Confusion Matrix

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Pada Tabel 2.2 dapat dijelaskan keterangan sebagai berikut :

- TP (True Positive) merupakan banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas positif.
- FN (False Negative) merupakan banyaknya data yang kelas aktualnya adalah kelas positif dengan kelas prediksinya merupakan kelas negatif.
- FP (False Positive) merupakan banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas positif.
- TN (True Negative) merupakan banyaknya data yang kelas aktualnya adalah kelas negatif dengan kelas prediksinya merupakan kelas negatif.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah eksperimental, yaitu melakukan sebuah percobaan atau skenario untuk membuktikan suatu konsep. Eksperimen yang dilakukan menggunakan beberapa rancangan arsitektur yang berbeda berdasarkan lapisan algoritma *Convolutional Neural Network* (CNN) untuk mencari akurasi terbaik. Dari segi sifatnya penelitian yang dilakukan adalah deskriptif yaitu membuat gambaran atau deskripsi dari percobaan yang dilakukan secara sistematis. Penelitian ini menggunakan pendekatan kuantitatif karena hasil penelitian bersifat objektif dengan skala numerik.

3.2. Metode Pengumpulan Data

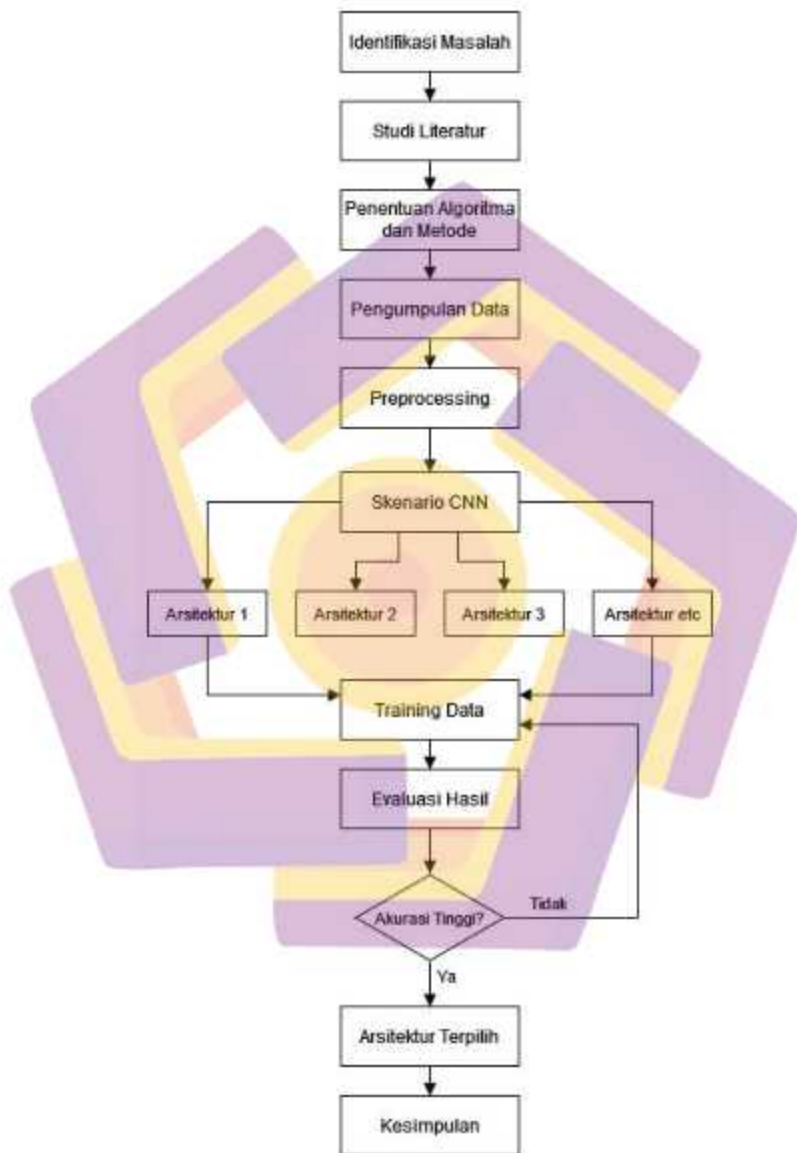
Data yang digunakan pada penelitian ini memanfaatkan data publik yang dapat diakses semua orang. Penyedia data publik yang digunakan adalah website www.kaggle.com. *Kaggle* menjadi tempat bagi para peneliti atau siapapun yang ingin berkontribusi dalam menyediakan data demi kemajuan penelitian di dunia, selain itu website ini juga dapat digunakan para peneliti untuk membagikan hasil percobaan seperti uji coba dengan metode menggunakan bahasa pemrograman *python*. Penelitian ini menggunakan data yang didapatkan dari website *kaggle* dengan nama *Leaf Rice Disease*. Data yang digunakan adalah data jenis gambar

atau citra daun padi. Citra daun padi yang didapatkan terbagi menjadi 3 kelas dengan rincian *blast*, *blight*, dan *tungro*.

3.3. Metode Analisis Data

Metode analisis data merupakan tahapan pengelolaan data yang digunakan pada penelitian ini untuk mendapatkan hasil yang diharapkan dari permasalahan yang telah dirumuskan. Pada penelitian ini analisis yang digunakan berupa analisis kuantitatif dengan mengolah data yang telah dikumpulkan menggunakan metode *Convolutional Neural Network* (CNN). Data daun tanaman padi yang telah dikumpulkan akan dilakukan proses *preprocessing* dengan augmentasi data citra sebelum digunakan proses klasifikasi daun berpenyakit di masing-masing kategori. Setelah dataset siap maka dilakukan skenario training untuk membangun model dengan arsitektur-arsitektur CNN yang telah dipilih. Pembagian data training sebesar 80%, data validation 10%, dan data testing 10%. Setelah tahap pembuatan model maka langkah selanjutnya adalah melakukan pengetesan model yang telah dibangun, dimana tahap ini kita akan menguji model-model yang telah dibuat mana yang menghasilkan akurasi tertinggi. Penjelasan lebih detail dalam melakukan klasifikasi daun padi akan dipaparkan pada alur penelitian.

3.4. Alur Penelitian



Gambar 3.1. Kerangka Alur Penelitian

Penyelesaian penelitian dalam mengidentifikasi penyakit pada daun tanaman padi dilakukan dalam beberapa kegiatan yang dilakukan secara bertahap. Kerangka alur penelitian yang akan dilaksanakan ditunjukkan pada Gambar 3.1. Terdapat beberapa tahapan untuk menyelesaikan penelitian tentang identifikasi penyakit pada daun tanaman padi. Tahapan-tahapan penelitian ini dapat dijabarkan lebih lanjut sebagai berikut :

a. Identifikasi Masalah

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

b. Studi Literatur

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

c. Penentuan Algoritma dan Metode

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

d. Pengumpulan Data

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

e. Preprocessing

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

f. Skenario CNN

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

g. Training Data

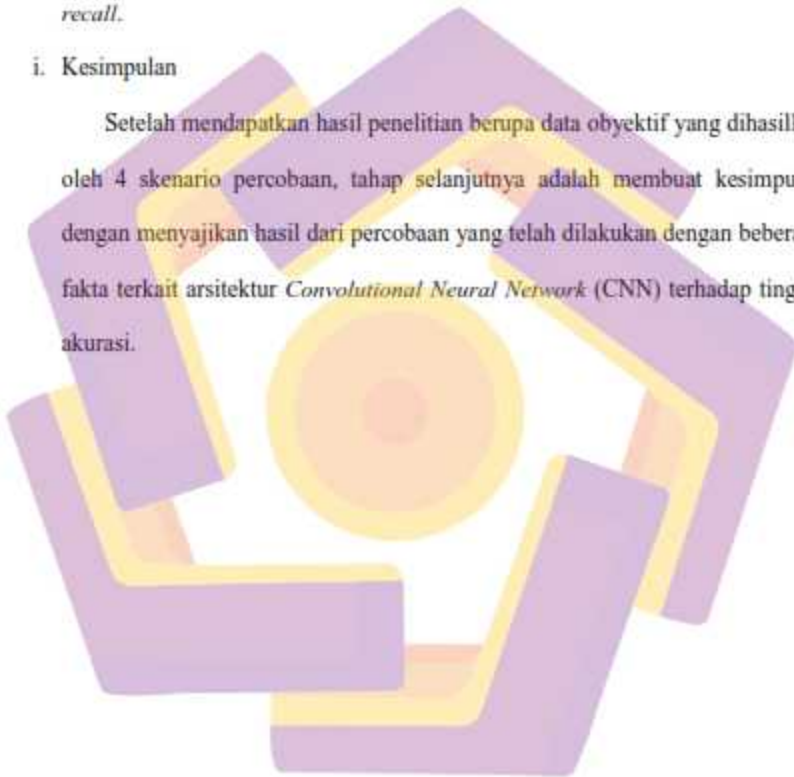
Setelah melakukan skenario yang ada maka mulai dilakukan percobaan menggunakan dataset citra daun padi yang telah disiapkan sebelumnya. Training data bertujuan untuk melatih keempat skenario yang ada agar saat tahap evaluasi dapat menghasilkan nilai akurasi yang tinggi.

h. Evaluasi Hasil

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

i. Kesimpulan

Setelah mendapatkan hasil penelitian berupa data obyektif yang dihasilkan oleh 4 skenario percobaan, tahap selanjutnya adalah membuat kesimpulan dengan menyajikan hasil dari percobaan yang telah dilakukan dengan beberapa fakta terkait arsitektur *Convolutional Neural Network* (CNN) terhadap tingkat akurasi.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengumpulan Data

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

Tabel 4.1. Jumlah Dataset Original

Sampel Data	Jumlah Data	Sumber Data
<i>Blast</i>	80	Kaggle (<i>Leaf Rice Disease</i>)
<i>Blight</i>	80	
<i>Tungro</i>	80	
Total	240	

Detail jumlah dari dataset awal yang digunakan pada penelitian ini ditampilkan pada Tabel 4.1. Total dari keseluruhan data yang digunakan sebelum

dilakukan proses *preprocessing* selanjutnya adalah 240 citra, dimana sebarannya dibagi ke dalam 3 kelas dengan perincian daun *blast*, *blight*, dan *tungro* masing-masing berjumlah 80 buah. Contoh dataset original per kelas dari website *kaggle* sebelum dilakukan *preprocessing* dapat dilihat pada Gambar 4.1.



Gambar 4.1. Sampel Dataset Original

4.2. Analisis Data

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

4.2.1. Augmentasi Data

Augmentasi data merupakan teknik untuk menambahkan jumlah citra yang digunakan dengan cara merubah atau memodifikasi citra. Augmentasi data juga memiliki fungsi untuk menambah keberagaman data sehingga data yang digunakan untuk model klasifikasi menjadi lebih kaya akan informasi. Augmentasi data citra dapat dilakukan dengan cara transisi, transformasi, penambahan noise, rotasi, pembesaran, dan lain sebagainya. Pada penelitian ini proses augmentasi data dilakukan pada seluruh kelas menggunakan beberapa teknik seperti *rotate clockwise*, *vertical flip*, *warp shift*, *blurring image*, dan lain sebagainya. Implementasi augmentasi data citra dilakukan menggunakan tools *jupyter notebook* dengan *library keras*.

Tabel 4.2. Jumlah Dataset Augmentasi

Sampel Data	Jumlah Data	Sumber Data
<i>Blast</i>	480	Kaggle (<i>Leaf Rice Disease</i>)
<i>Blight</i>	480	
<i>Tungro</i>	480	
Total	1440	

Jumlah dataset dari kaggle yang telah dilakukan augmentasi data dapat dilihat pada Tabel 4.2, dimana masing-masing kelas dilakukan augmentasi hingga berjumlah 480 gambar sehingga total menjadi 1440 gambar. Berikut adalah potongan kode augmentasi data pada daun padi untuk menambah datanya.

```

...
def anticlockwise_rotation(image):
    angle= random.randint(0,180)
    return rotate(image, angle)

def clockwise_rotation(image):
    angle= random.randint(0,180)
    return rotate(image, -angle)

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

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

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

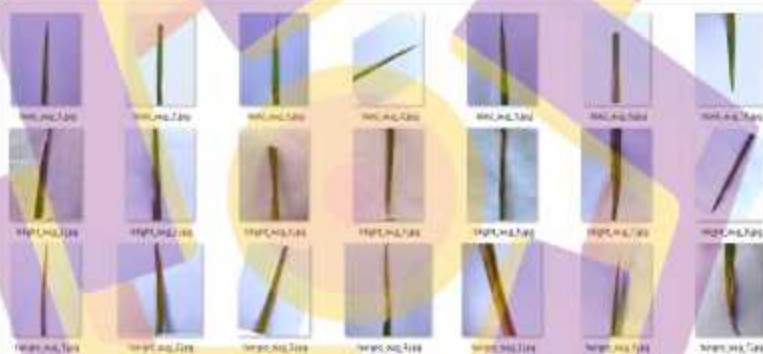
def add_noise(image):
    return random_noise(image)

def blur_image(image):
    return cv2.GaussianBlur(image, (9,9),0)
...
transformations =
{
    'rotate anticlockwise': anticlockwise_rotation,
    'rotate clockwise': clockwise_rotation,
    'horizontal flip': h_flip,
    'vertical flip': v_flip,
    'warp shift': warp_shift,
    'adding noise': add_noise,
    'blurring image': blur_image
}
...

```

Pada potongan kode di atas dapat kita lihat ada sekitar 7 pendefinisian *function* untuk melakukan augmentasi data antara lain *anticlockwise_rotation* untuk memutar gambar berlawanan arah jarum jam, *clockwise_rotation* untuk memutar gambar searah jarum jam, *h_flip* atau singkatan dari *horizontal flip* untuk membalikkan gambar secara horizontal, *v_flip* atau singkatan dari *vertical flip*

untuk membalikkan gambar secara vertikal atau tegak lurus dari atas ke bawah, *warp_shift* untuk membelokkan dan menggeser gambar, *add_noise* untuk menambahkan *noise* atau derau acak ke dalam gambar, dan terakhir *blur_image* untuk mengaburkan atau menghaluskan gambar dengan cara mengurangi detail tajam dan meredam tingkat kontras dalam gambar. Kemudian baris terakhir adalah pemanggilan masing-masing function yang disimpan pada variable *transformations* untuk digunakan pada *script* lanjutan. Contoh hasil augmentasi data citra pada masing-masing kelas dapat kita lihat pada Gambar 4.2.



Gambar 4.2. Sampel Dataset Augmentasi Original

4.2.2. Preprocessing Data

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

selanjutnya adalah pembagian data untuk digunakan pada model klasifikasi. Pembagian data disini adalah melakukan pembagian data menjadi *data training*, *data validation*, dan *data testing*. *Data training* adalah yang digunakan untuk melatih model klasifikasi. *Data validation* merupakan data yang digunakan untuk melihat apakah hasil pelatihan model klasifikasi sudah baik atau memiliki kendala *overfitting* / *underfitting*. *Overfitting* terjadi ketika model pembelajaran mesin terlalu kompleks dalam data pelatihan, bukan hanya pola yang sebenarnya, model ini "menghafal" data pelatihan dari pada "mengerti" pola di dalamnya. Sedangkan *underfitting* terjadi ketika model pembelajaran mesin terlalu sederhana untuk memahami kompleksitas data pelatihan, model ini tidak dapat menangkap pola yang ada dalam data dengan baik. Terakhir adalah *data testing* yang berarti data uji, data ini digunakan untuk menguji model klasifikasi yang dilatih dan nantinya dapat digunakan untuk evaluasi model klasifikasi yang digunakan. Pembagiannya adalah *data training* memiliki jumlah 80% data, kemudian *data validation* 10%, dan terakhir *data testing* 10%.

4.2.3. Preprocessing Data Lanjutan (*Enhancement*)

Pada tahap ini penulis melakukan *preprocessing* lanjutan atau *enhancement* data gambar dengan membuang *background* bawaan menjadi full warna putih dan juga melakukan zoom sehingga proporsi gambar daun menjadi mayoritas dibanding *background* sehingga diharapkan meningkatkan hasil akurasi. Hasil proses lanjutan ini pada dataset original yang berjumlah 240 gambar dari *kaggle* dapat dilihat pada Gambar 4.3 di bawah.



Gambar 4.3. Sampel Dataset Enhancement Original

Setelah dataset original di-*enhancement* lebih lanjut kemudian dilakukan proses augmentasi kembali seperti proses augmentasi sebelumnya untuk menambah jumlah dataset menjadi total 1440 gambar. Hasil proses ini dapat dilihat pada Gambar 4.4 di bawah ini.



Gambar 4.4. Sampel Dataset Augmentasi Enhancement

4.3. Persiapan Skenario CNN

Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dalam melakukan klasifikasi penyakit pada citra daun padi. CNN memiliki beberapa lapisan arsitektur yang dapat disesuaikan untuk melakukan klasifikasi. Kelebihan yang dimiliki oleh CNN adalah lapisan *convolutional*-nya yang dapat digunakan untuk ekstraksi ciri atau ekstraksi tekstur dari citra secara mendalam. Lapisan yang tidak kalah berguna adalah lapisan *pooling* yang digunakan untuk mengurangi dimensi yang dimiliki oleh citra dengan tetap mempertahankan informasi penting yang dimiliki citra. Pembuatan skenario dilakukan sesuai dengan jenis penelitian yang bersifat eksperimental dimana penulis akan mencoba bereksperimen menggunakan algoritma dan metode yang telah ditetapkan sebelumnya dengan beberapa arsitektur yang berbeda.

Arsitektur dari skenario CNN yang akan digunakan pada penelitian kali ini adalah beberapa arsitektur lapisan yang telah ada sebelumnya yaitu VGG16, Xception, dan NASNetMobile serta arsitektur yang penulis rancang sendiri. Alasan penulis juga memasukkan rancangan arsitektur sendiri yaitu penulis ingin mencoba dan mengetahui proses dan penentuan dari setiap lapisan arsitektur yang dibuat. Untuk menciptakan hasil yang baik, penulis tidak ingin hanya menggunakan arsitektur yang dirancang sendiri namun akan menggunakan arsitektur yang telah teruji hasil dan kinerjanya. Pemilihan ini dilakukan juga sebagai perbandingan dan tolok ukur arsitektur yang dibuat penulis. VGG16 sendiri dipilih karena cukup populer dan dari penelitian Priyanka & Kumara tahun 2021 dengan judul *Classification Of Rice Plant Diseases Using the Convolutional Neural Network*

Method mendapatkan tes akurasi lumayan tinggi sebesar 95,24%. Selanjutnya Xception dan NASNetMobile dipilih karena Tejaswini dkk (2022) melalui penelitiannya yang berjudul *Rice Leaf Disease Classification Using CNN* juga menggunakan Xception yang mendapatkan hasil cukup baik sebesar 72,2%, selain itu dari web Keras dengan alamat <https://keras.io/api/applications> yang memiliki banyak pilihan arsitektur, keduanya mendapatkan akurasi yang cukup tinggi dan size yang cukup ringan.

Tabel 4.3. Skenario Percobaan

No	Arsitektur	Skenario	Dataset	Jumlah Data
1	VGG16 (CNN 1)	1	Original	80
2		2	Augmentasi Original	1440
3		3	Enhancement Original	80
4		4	Augmentasi Enhancement	1440
5	Xception (CNN 2)	1	Original	80
6		2	Augmentasi Original	1440
7		3	Enhancement Original	80
8		4	Augmentasi Enhancement	1440
9	NASNetMobile (CNN 3)	1	Original	80
10		2	Augmentasi Original	1440
11		3	Enhancement Original	80
12		4	Augmentasi Enhancement	1440
13	Custom X (CNN 4)	1	Original	80
14		2	Augmentasi Original	1440
15		3	Enhancement Original	80
16		4	Augmentasi Enhancement	1440

Terdapat total 16 skenario percobaan dari 4 arsitektur CNN yang telah dibuat, dimana masing-masing arsitektur dibagi menjadi 4 skenario dengan olah dataset yang berbeda dari sumber yang sama, detailnya dapat kita lihat pada Tabel 4.3. Implementasi skenario CNN dilakukan dengan bahasa *python* di *jupyter*

notebook menggunakan *library-library* yang dibutuhkan seperti *keras*, *tensorflow*, *numpy*, *pandas*, *sklearn*, *matplotlib*, dan lain-lain.

4.3.1. Skenario-skenario CNN 1

Pada skenario CNN 1 ini penulis menggunakan arsitektur yang telah tersedia sebelumnya yaitu VGG16 (*Visual Geometry Group 16*) dengan beberapa olah dataset yang berbeda dari sumber yang sama yaitu dataset original, dataset augmentasi original, dataset enhancement original, dan dataset augmentasi enhancement. VGG16 adalah sebuah arsitektur *deep convolutional neural network* yang terdiri atas 16 layer dan merupakan salah satu arsitektur paling awal yang digunakan dalam *transfer learning*. VGG16 dikembangkan oleh Karen Simonyan dan Andrew Zisserman dan dipublikasikan pada tahun 2014 dalam salah satu paper mereka (Simonyan & Zisserman, 2014). Model ini kemudian memenangkan kompetisi 2014 ILSVRC Challenge atas perolehan top-5 test accuracy mencapai 92,7% pada klasifikasi 14 juta gambar dalam 1000 kelas. Alasan lain penggunaan VGG16 karena salah satu arsitektur terbaik dengan telah memenangkan kompetisi ILSVR (*imagenet*) pada tahun 2014. Arsitektur VGG16 berisikan *feature extraction* yang berisikan *convolutional layer* dan *pooling layer* yang saling berhubungan. Setiap layer tersusun terdiri dari 16 blok layer sesuai dengan namanya.

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1100160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359008
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359008
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359008
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359008
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359008
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0

Total params: 14714688 (56.13 MB)
 Trainable params: 14714688 (56.13 MB)
 Non-trainable params: 0 (0.00 Byte)

Gambar 4.5. Model Arsitektur VGG16

Susunan model dari arsitektur VGG16 dapat dilihat pada Gambar 4.5 di atas, dimana *input* citra sudah di-set otomatis menjadi 256x246 pixel, untuk ukuran *filter* berukuran 3x3 dengan *stride* 1 yang berada pada setiap layer yang ada pada *convolutional layer*. *Pooling layer* menggunakan *max pooling* dengan jumlah *filter* 2x2. Tetapi lapisan di atas belum bisa dijadikan arsitektur untuk proses selanjutnya karena hanya lapisan bawaan dan masih belum mempunyai lapisan sesuai dengan percobaan yang dibutuhkan, maka langkah selanjutnya yang dilakukan setelah mendapatkan layer arsitektur VGG16 adalah *fine tuning*. *Fine tuning* adalah proses pengambilan bobot dari arsitektur yang telah dilatih sebelumnya menggunakan *imagenet*. Teknik ini juga dapat digunakan untuk menyesuaikan jumlah kelas pada

convolutional layer terhadap objek yang akan dilatih, berikut potongan kode dari proses *fine tuning*.

```
...
model_name='vgg16-d4aug'
print("Building model with", base_model)
model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.Conv2D(filters=32,
padding='same', kernel_size=3, activation='relu', strides=1),
    tf.keras.layers.MaxPool2D(pool_size=2,
strides=2),
    tf.keras.layers.Dropout(rate=0.5),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(3, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001), loss='categorical_crossentropy', metrics='accuracy')
...
```

Tahap ini peneliti akan menyisipkan *convolutional layer* dengan *activation relu*, *dropout* untuk mengurangi *overfitting*, fungsi *flatten* untuk membuat citra yang berbentuk array menjadi bentuk vektor, serta terakhir yaitu fungsi *dense* yang menambahkan layer pada *fully connected layer* dengan *activation softmax* untuk menghubungkan fitur-fitur yang telah diekstraksi oleh lapisan-lapisan sebelumnya agar menghasilkan *output* akhir seperti kelas prediksi, selanjutnya model tersebut di-*compile*. Hasil setelah *fine tuning* dapat dilihat pada Gambar 4.6.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 8, 8, 512)	14714688
conv2d (Conv2D)	(None, 8, 8, 32)	147488
max_pooling2d (MaxPooling2D)	(None, 4, 4, 32)	0
dropout (Dropout)	(None, 4, 4, 32)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 3)	1539

Total params: 14862176 (56.70 MB)
 Trainable params: 149027 (582.14 KB)
 Non-trainable params: 14714688 (56.13 MB)

Gambar 4.6. Arsitektur VGG16 Setelah *Fine Tuning*

4.3.2. Skenario-skenario CNN 2

Pada skenario CNN 2 penulis menggunakan arsitektur yang telah tersedia lainnya yaitu Xception dengan beberapa olah dataset yang berbeda dari sumber yang sama yaitu dataset original, dataset augmentasi original, dataset enhancement original, dan dataset augmentasi enhancement. Arsitektur Xception adalah salah satu inovasi dalam dunia *Convolutional Neural Networks (CNN)* yang diperkenalkan oleh *Google Research*. Ini adalah jenis arsitektur yang mengambil ide dari arsitektur *Inception*, tetapi dengan beberapa peningkatan yang signifikan. Xception lebih terfokus pada pemisahan fitur-fitur spasial dan fitur-fitur kanal dalam jaringan. Susunan arsitektur Xception yang pertama berisi *input layer* dengan input citra ukuran tertentu, kemudian *entry flow* yang menjadi bagian awal jaringan dengan bagian *convolution layer*, *batch normalization* untuk menstabilkan dan mempercepat pelatihan, *activation layer* seperti ReLU. Selanjutnya ada *separable*

4.3.4. Skenario-skenario CNN 4

Pada skenario CNN 4 penulis menggunakan arsitektur custom yang penulis rancang sendiri yang memanfaatkan berbagai referensi dari arsitektur custom yang pernah dibuat sebelumnya dengan beberapa olah dataset yang berbeda dari sumber yang sama yaitu dataset original, dataset augmentasi original, dataset enhancement original, dan dataset augmentasi enhancement. Susunan *learning architecture* model yang akan digunakan pada skenario-skenario CNN ke-4 dapat dilihat pada Gambar 4.9.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 226, 256, 12)	888
max_pooling2d (MaxPooling2D)	(None, 113, 128, 12)	0
dropout (Dropout)	(None, 113, 128, 12)	0
flatten (Flatten)	(None, 354208)	0
dense (Dense)	(None, 5)	1573767

 Total params: 1573767 (6.08 MB)
 Trainable params: 1573767 (6.08 MB)
 Non-trainable params: 0 (0.00 Byte)

Gambar 4.9. Model Arsitektur Custom X

Penulis menggunakan layer sederhana dan tidak tebal berlapis-lapis karena hasil model yang baik untuk klasifikasi citra tidak mutlak bergantung pada hal tersebut. Dasar referensi untuk penentuan setting ini berasal dari literatur ilmiah dan hasil eksperimen sebelumnya, pembuat model biasanya akan melakukan serangkaian eksperimen yang disebut pencarian *hyperparameter* untuk menemukan kombinasi terbaik yang sesuai dengan data dan masalah spesifik yang dihadapi (Khoiruddin, Junaidi, & Saputra, 2022). Penulis menggunakan 1 *convolution layer*

yaitu conv2d (Conv2D) (256, 256, 32) yang berarti pada lapisan ini citra memiliki ukuran 256 x 156 pixel dan memiliki 32 jumlah *channel* atau *filter* yang digunakan untuk ekstraksi ciri atau tekstur dari citra. Setelah proses konvolusi akan dilanjutkan proses *pooling* pada baris max_pooling2d (MaxPooling2D) dengan pool size 2,2. Seperti yang dijelaskan sebelumnya bahwa *pooling* merupakan proses untuk mereduksi atau mengurangi resolusi yang dimiliki citra, dan *max pooling* merupakan filter untuk mereduksi resolusi citra dengan mengambil nilai piksel tertinggi dari filter yang melewati citra. Hasil dari tahap konvolusi dan pooling dinamakan *feature map* dan ini adalah hasil ekstraksi ciri yang bisa digunakan untuk klasifikasi. Fungsi *dropout* dengan nilai 0.4 mengindikasikan bahwa selama pelatihan, 40% dari neuron dalam lapisan tertentu akan dinonaktifkan secara acak pada setiap iterasi, hal ini membantu mencegah *overfitting* dengan memaksa jaringan untuk tidak bergantung terlalu kuat pada koneksi tertentu, sehingga meningkatkan generalisasi pada data yang tidak terlihat. Tahap selanjutnya adalah *flatten* yang merupakan proses merubah *feature map* hasil pooling menjadi bentuk vector, sehingga dapat dilakukan klasifikasi dengan *neural network*. Pada baris terakhir yang merupakan lapisan *dense* terdapat angka 3 yang menunjukkan jumlah kelas yang digunakan yaitu kelas *blast*, kelas *blight*, dan kelas *tungro*. Susunan *learning architecture* Custom X ini juga dapat digunakan untuk mengolah dataset lain selain citra penyakit tanaman padi, dan juga bisa dioprek serta ditingkatkan lebih lanjut pada penelitian selanjutnya.

Setelah dibuat model-model di atas, setiap skenario yang digunakan mulai dari skenario 1 sampai 16 akan diujikan dalam *training data* dengan menggunakan

25 *epochs*. Maksud epoch disini adalah ketika seluruh data telah melalui proses pembelajaran data pada *neural network*. Setiap skenario yang digunakan akan diujikan menggunakan rasio pembagian 80% untuk data training, 10% data validation, dan 10% sisanya sebagai data testing. Sehingga jika per kelas original data memiliki 80 gambar dengan total 240 gambar, maka setelah pembagian rasio menjadi 192 gambar untuk data training, 24 gambar untuk data validation, dan sisanya 24 gambar untuk data testing. Kemudian jika per kelas setelah augmentasi data memiliki 480 gambar dengan total 1440 gambar, maka setelah pembagian rasio menjadi 1152 gambar untuk data training, 144 gambar untuk data validation, dan sisanya 144 gambar untuk data testing.

4.4. Analisis Hasil Penelitian

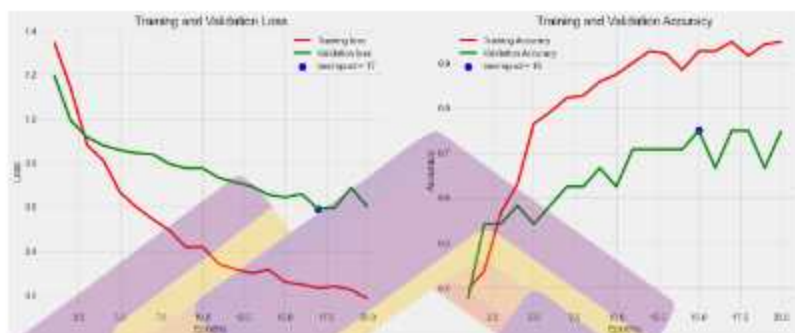
Proses klasifikasi pada daun tanaman padi dilakukan dengan menggunakan skenario-skenario percobaan yang telah ditentukan pada penjelasan sebelumnya. Hasil dari percobaan yang telah dilakukan akan dianalisis dan dijelaskan lebih rinci. Metode pengujian yang digunakan pada penelitian ini adalah *Confusion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

metrik tunggal. Serta *support* memberikan informasi tentang seberapa banyak data yang termasuk dalam kelas tertentu.

Training loss dan *validation loss*, serta *training accuracy* dan *validation accuracy* adalah metrik-metrik yang digunakan untuk mengukur kinerja model pada saat pelatihan dan evaluasi pada *machine learning* atau *deep learning*. *Training loss* adalah metrik yang mengukur sejauh mana model telah belajar dari data pelatihan. Ini adalah ukuran seberapa baik model memprediksi data pelatihan. Tujuan utama selama pelatihan adalah mengurangi *training loss* sekecil mungkin. *Validation loss* adalah metrik yang digunakan untuk mengukur sejauh mana model dapat menggeneralisasi dari data pelatihan ke data yang belum pernah dilihat sebelumnya (data validasi). Ini adalah ukuran seberapa baik model dapat melakukan prediksi pada data baru. *Training accuracy* adalah metrik yang mengukur sejauh mana model benar-benar memprediksi kelas atau label yang benar pada data pelatihan. *Validation accuracy* adalah metrik yang digunakan untuk mengukur sejauh mana model benar-benar memprediksi kelas atau label yang benar pada data validasi, yang belum pernah dilihat selama pelatihan.

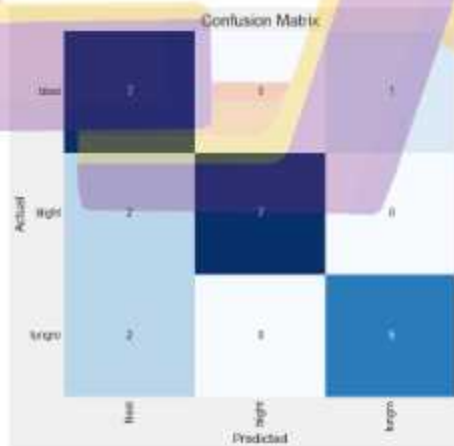
4.4.1 Hasil dan Pengujian CNN 1

1. Skenario CNN 1 pada Dataset Original



Gambar 4.10. Grafik *Training & Validation* VGG16 (Dataset Original)

Pada Gambar 4.10 terlihat bahwa grafik *training and validation loss* model arsitektur VGG16 pada dataset original dengan *epoch* terbaik terdapat pada urutan 17. Sedangkan grafik *training and validation accuracy* menunjukkan *epoch* terbaik pada urutan 15 dengan nilai *train_loss* 0.2589, *train_accuracy* 0.9271, *val_loss* 0.6433, dan *val_accuracy* 0.7500.



Gambar 4.11. *Confusion Matrix* VGG16 (Dataset Original)

Gambar 4.11 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 1 pada dataset original dengan model arsitektur VGG16. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.11 jumlah TP adalah $7 + 7 + 5 = 19$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(19/24) * 100 = 79.16\%$ atau jika dibulatkan menjadi 79%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 7 / (7 + 2 + 2) = 7 / 11 = 0.6363 * 100 = 64\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 7 / (7 + 0 + 0) = 7 / 7 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 5 / (5 + 0 + 1) = 5 / 6 = 0.8333 * 100 = 83\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang

kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(0.6363 + 1 + 0.8333) / 3 = 2.4696 / 3 = 0.8232 * 100 = 82\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 7 / (7 + 0 + 1) = 7 / 8 = 0.875 * 100 = 88\%$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 7 / (7 + 2 + 0) = 7 / 9 = 0.7777 * 100 = 78\%$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 5 / (5 + 0 + 2) = 5 / 7 = 0.7142 * 100 = 71\%$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.875 + 0.7777 + 0.7142) / 3 = 2.3669 / 3 = 0.7889 * 100 = 79\%$.

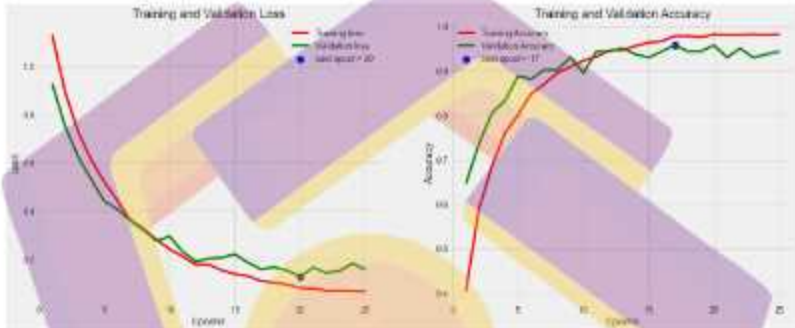
Classification Report:

	precision	recall	f1-score	support
blast	0.84	0.88	0.74	8
blight	1.00	0.78	0.88	9
tungro	0.83	0.71	0.77	7
accuracy			0.79	24
macro avg	0.82	0.79	0.79	24
weighted avg	0.83	0.79	0.80	24

Gambar 4.12. *Classification Report* VGG16 (Dataset Original)

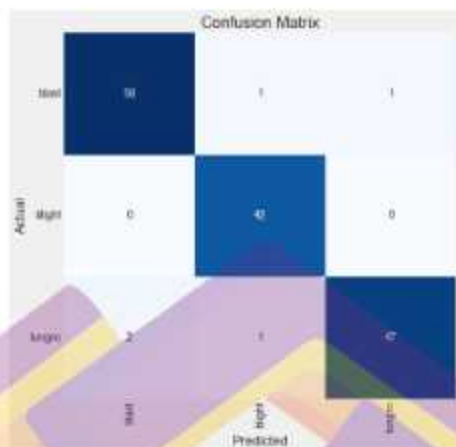
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.12.

2. Skenario CNN 1 pada Dataset Augmentasi Original



Gambar 4.13. Grafik *Training & Validation* VGG16 (Dataset Augmentasi Original)

Pada Gambar 4.13 terlihat bahwa grafik *training and validation loss* model arsitektur VGG16 pada dataset augmentasi original semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 20. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, dimana *epoch* terbaik pada urutan 17 dengan nilai *train_loss* 0.1101, *train_accuracy* 0.9792, *val_loss* 0.1586, dan *val_accuracy* 0.9583.



Gambar 4.14. *Confusion Matrix* VGG16 (Dataset Augmentasi Original)

Gambar 4.14 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 1 pada dataset augmentasi original dengan model arsitektur VGG16. Pada *confusion matrix* terdapat 4 istilah yang digunakan untuk menghitung nilai *accuracy*, *precision*, dan *recall* yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.14 jumlah TP adalah $50 + 42 + 47 = 139$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(139/144) * 100 = 96.52\%$ atau jika dibulatkan menjadi 97%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 50 / (50 + 0 + 2) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 42 / (42 + 1 + 1) = 42 / 44 = 0.9545 * 100 = 95\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 47 / (47 + 0 + 1) = 47 / 48 = 0.9791 * 100 = 98\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.9615 + 0.9545 + 0.9791) / 3 = 2.8951 / 3 = 0.9650 * 100 = 97\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 50 / (50 + 1 + 1) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 42 / (42 + 0 + 0) = 42 / 42 = 1 * 100 = 100\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 47 / (47 + 1 + 2) = 47 / 50 = 0.94 * 100 = 94\%$$

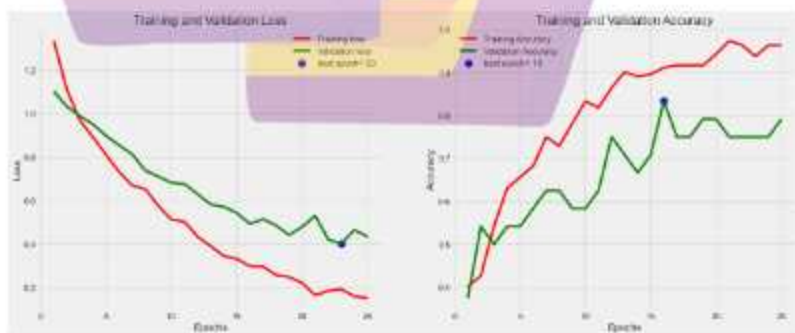
Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9615 + 1 + 0.94) / 3 = 2.9015 / 3 = 0.9671 * 100 = 97\%$.

Classification Report:				
	precision	recall	f1-score	support
blast	0.96	0.96	0.96	52
blight	0.95	1.00	0.98	42
tungro	0.98	0.94	0.96	56
accuracy			0.97	144
macro avg	0.97	0.97	0.97	144
weighted avg	0.97	0.97	0.97	144

Gambar 4.15. Classification Report VGG16 (Dataset Augmentasi Original)

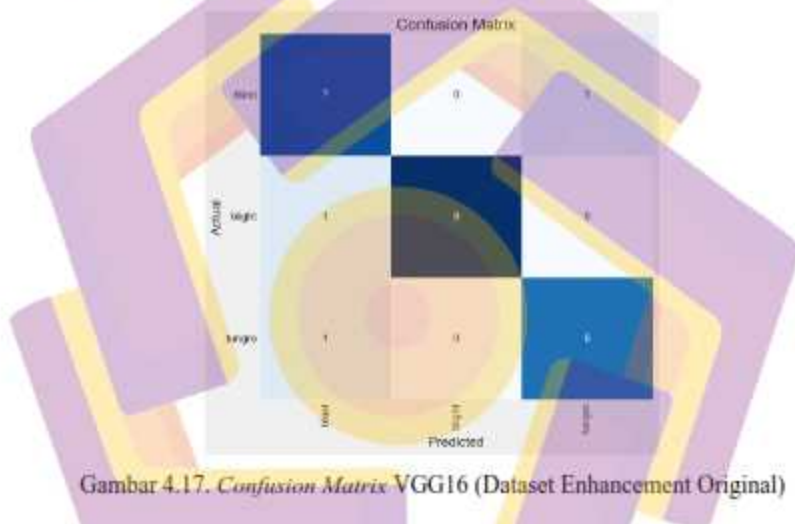
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.15.

3. Skenario CNN 1 pada Dataset Enhancement Original



Gambar 4.16. Grafik Training & Validation VGG16 (Dataset Enhancement Original)

Pada Gambar 4.16 terlihat bahwa grafik *training and validation loss* model arsitektur VGG16 pada dataset enhancement original semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 23. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 16 dengan nilai *train_loss* 0.2984, *train_accuracy* 0.9115, *val_loss* 0.4931, dan *val_accuracy* 0.8333.



Gambar 4.17. *Confusion Matrix* VGG16 (Dataset Enhancement Original)

Gambar 4.17 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 1 pada dataset enhancement original dengan model arsitektur VGG16. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.17 jumlah TP adalah $7 + 8 + 6 = 21$,

kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(21/24) * 100 = 87.5\%$ atau jika dibulatkan menjadi 88%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 7 / (7 + 1 + 1) = 7 / 9 = 0.7777 * 100 = 78\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 6 / (6 + 0 + 1) = 6 / 7 = 0.8571 * 100 = 86\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.7777 + 1 + 0.8571) / 3 = 2.6348 / 3 = 0.8782 * 100 = 88\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 7 / (7 + 0 + 1) = 7 / 8 = 0.875 * 100 = 88\%.$$

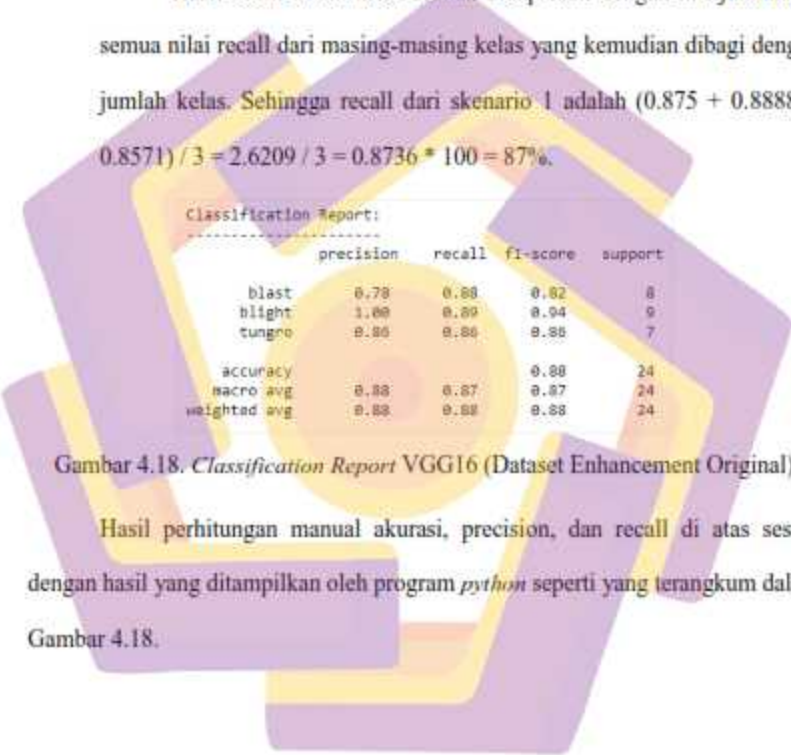
b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 8 / (8 + 1 + 0) = 8 / 9 = 0.8888 * 100 = 89\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 6 / (6 + 0 + 1) = 6 / 7 = 0.8571 * 100 = 86\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.875 + 0.8888 + 0.8571) / 3 = 2.6209 / 3 = 0.8736 * 100 = 87\%$.

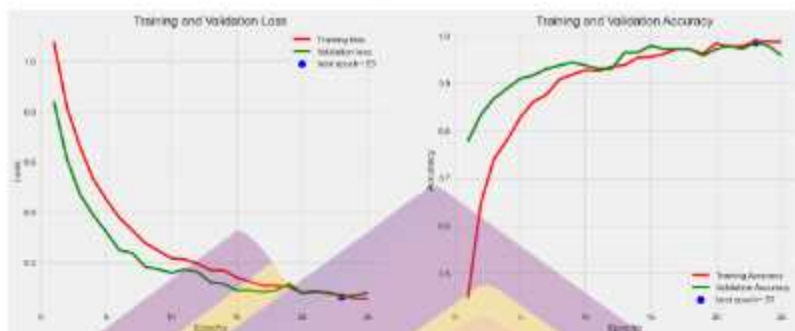


Classification Report:				
	precision	recall	f1-score	support
blast	0.78	0.88	0.82	8
blight	1.00	0.89	0.94	9
tungro	0.86	0.86	0.86	7
accuracy			0.88	24
macro avg	0.88	0.87	0.87	24
weighted avg	0.88	0.88	0.88	24

Gambar 4.18. *Classification Report* VGG16 (Dataset Enhancement Original)

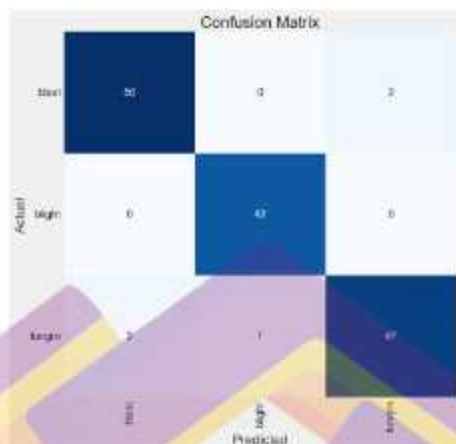
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.18.

4. Skenario CNN 1 pada Dataset Augmentasi Enhancement



Gambar 4.19. Grafik *Training & Validation* VGG16 (Dataset Augmentasi Enhancement)

Pada Gambar 4.19 terlihat bahwa grafik *training and validation loss* model arsitektur VGG16 pada dataset augmentasi enhancement semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 23. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 23 dengan nilai *train_loss* 0.0693, *train_accuracy* 0.9887, *val_loss* 0.0622, dan *val_accuracy* 0.9861.



Gambar 4.20. *Confusion Matrix* VGG16 (Dataset Augmentasi Enhancement)

Gambar 4.20 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 1 pada dataset augmentasi enhancement dengan model arsitektur VGG16. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.20 jumlah TP adalah $50 + 42 + 47 = 139$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(139/144) * 100 = 96.52\%$ atau jika dibulatkan menjadi 97%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 50 / (50 + 0 + 2) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 42 / (42 + 0 + 1) = 42 / 43 = 0.9767 * 100 = 98\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 47 / (47 + 0 + 2) = 47 / 49 = 0.9591 * 100 = 96\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.9615 + 0.9767 + 0.9591) / 3 = 2.8973 / 3 = 0.9657 * 100 = 97\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 50 / (50 + 0 + 2) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 42 / (42 + 0 + 0) = 42 / 42 = 1 * 100 = 100\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 47 / (47 + 1 + 2) = 47 / 50 = 0.94 * 100 = 94\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9615 + 1 + 0.94) / 3 = 2.9015 / 3 = 0.9671 * 100 = 97\%$.

Classification Report:

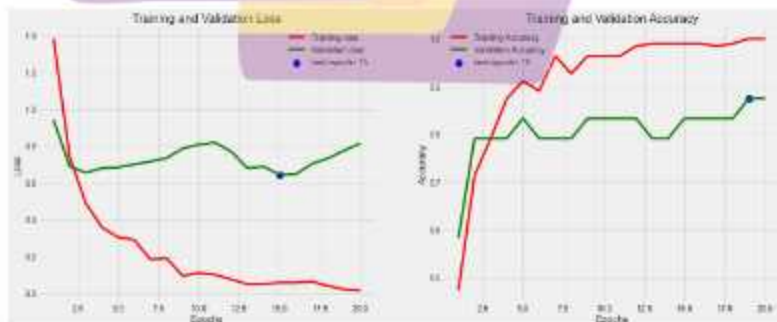
	precision	recall	f1-score	support
blast	0.96	0.96	0.96	52
blight	0.98	1.00	0.99	42
tungro	0.96	0.94	0.95	50
accuracy			0.97	144
macro avg	0.97	0.97	0.97	144
weighted avg	0.97	0.97	0.97	144

Gambar 4.21. Classification Report VGG16 (Dataset Augmentasi Enhancement)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.21.

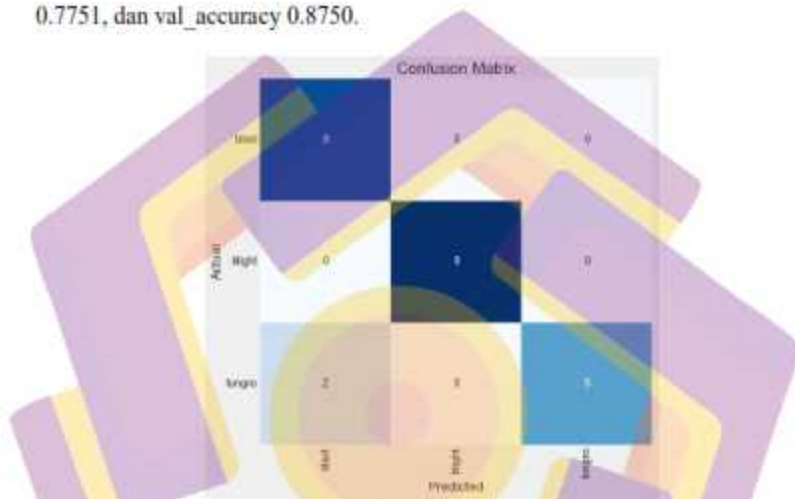
4.4.2 Hasil dan Pengujian CNN 2

1. Skenario CNN 2 pada Dataset Original



Gambar 4.22. Grafik Training & Validation Xception (Dataset Original)

Pada Gambar 4.22 terlihat bahwa grafik *training and validation loss* model arsitektur Xception pada dataset original dengan *epoch* terbaik terdapat pada urutan 15. Sedangkan grafik *training and validation accuracy* menunjukkan *epoch* terbaik pada urutan 19 dengan nilai *train_loss* 0.0207, *train_accuracy* 1.0000, *val_loss* 0.7751, dan *val_accuracy* 0.8750.



Gambar 4.23. *Confusion Matrix* Xception (Dataset Original)

Gambar 4.23 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 2 pada dataset original dengan model arsitektur Xception. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.23 jumlah TP adalah $8 + 9 + 5 = 22$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai

akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(22/24) * 100 = 91.66\%$ atau jika dibulatkan menjadi 92%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 8 / (8 + 0 + 2) = 8 / 10 = 0.8 * 100 = 80\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 9 / (9 + 0 + 0) = 9 / 9 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 5 / (5 + 0 + 0) = 5 / 5 = 1 * 100 = 100\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.8 + 1 + 1) / 3 = 2.8 / 3 = 0.9333 * 100 = 93\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

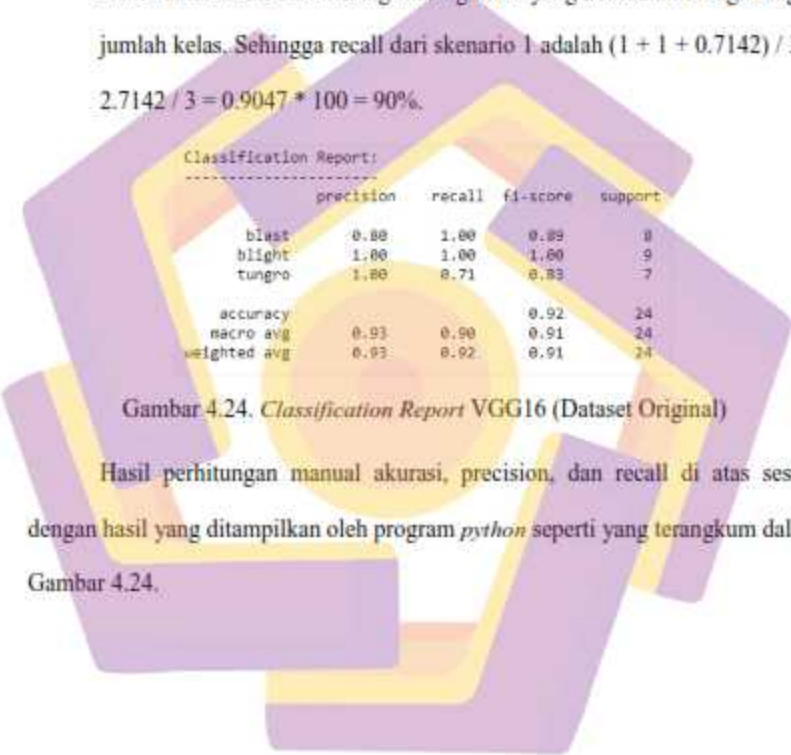
b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 9 / (9 + 0 + 0) = 9 / 9 = 1 * 100 = 100\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 5 / (5 + 0 + 2) = 5 / 7 = 0.7142 * 100 = 71\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(1 + 1 + 0.7142) / 3 = 2.7142 / 3 = 0.9047 * 100 = 90\%$.



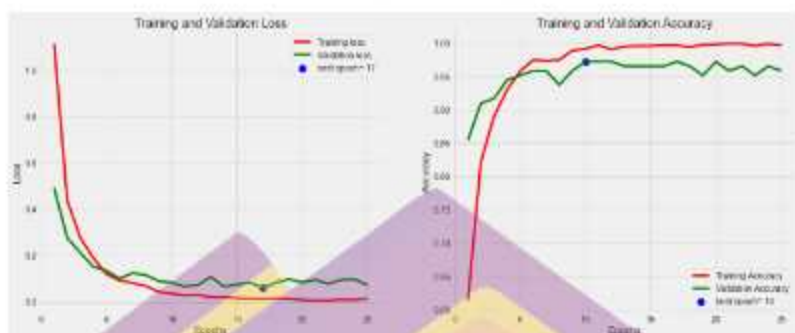
Classification Report:

	precision	recall	f1-score	support
blast	0.88	1.00	0.89	8
blight	1.00	1.00	1.00	9
tungro	1.00	0.71	0.83	7
accuracy			0.92	24
macro avg	0.93	0.90	0.91	24
weighted avg	0.93	0.92	0.91	24

Gambar 4.24. *Classification Report* VGG16 (Dataset Original)

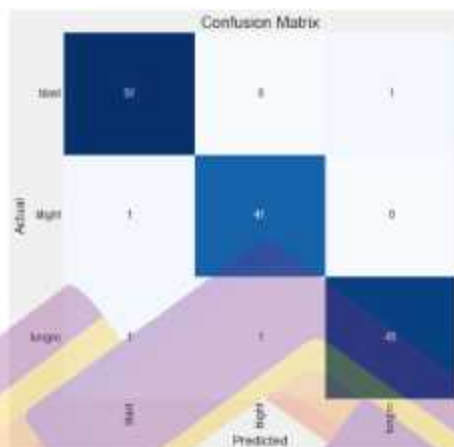
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.24.

2. Skenario CNN 2 pada Dataset Augmentasi Original



Gambar 4.25. Grafik *Training & Validation Xception* (Dataset Augmentasi Original)

Pada Gambar 4.25 terlihat bahwa grafik *training and validation loss* model arsitektur Xception pada dataset augmentasi original semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 17. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 10 dengan nilai *train_loss* 0.0368, *train_accuracy* 0.9913, *val_loss* 0.0837, dan *val_accuracy* 0.9722.



Gambar 4.26. *Confusion Matrix* Xception (Dataset Augmentasi Original)

Gambar 4.26 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 2 pada dataset augmentasi original dengan model arsitektur Xception. Nilai akurasi, *precision*, dan *recall* akan dihitung sebagai berikut.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.26 jumlah TP adalah $51 + 41 + 48 = 140$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(140/144) * 100 = 97.22\%$ atau jika dibulatkan menjadi 97%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 51 / (51 + 1 + 1) = 51 / 53 = 0.9622 * 100 = 96\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 41 / (41 + 0 + 1) = 41 / 42 = 0.9761 * 100 = 98\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 48 / (48 + 0 + 1) = 48 / 49 = 0.9795 * 100 = 98\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(0.9622 + 0.9761 + 0.9795) / 3 = 2.9178 / 3 = 0.9726 * 100 = 97\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 51 / (51 + 0 + 1) = 51 / 52 = 0.9807 * 100 = 98\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 41 / (41 + 1 + 0) = 41 / 42 = 0.9761 * 100 = 98\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 48 / (48 + 1 + 1) = 48 / 50 = 0.96 * 100 = 96\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan

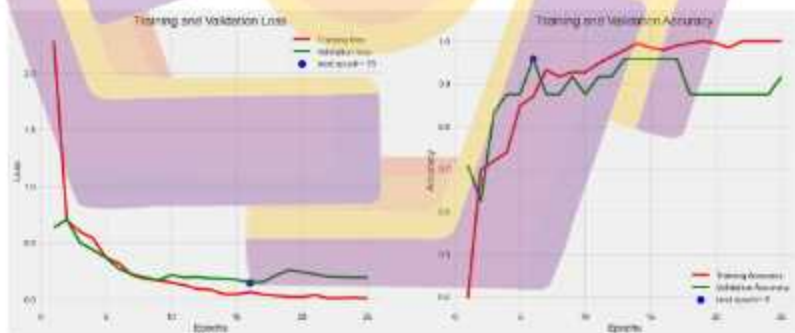
jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9807 + 0.9761 + 0.96) / 3 = 2.9162 / 3 = 0.9720 * 100 = 97\%$.

Classification Report:				
	precision	recall	f1-score	support
blast	0.96	0.98	0.97	52
blight	0.98	0.96	0.96	42
tungro	0.98	0.96	0.97	50
accuracy			0.97	144
macro avg	0.97	0.97	0.97	144
weighted avg	0.97	0.97	0.97	144

Gambar 4.27. *Classification Report* Xception (Dataset Augmentasi Original)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.27.

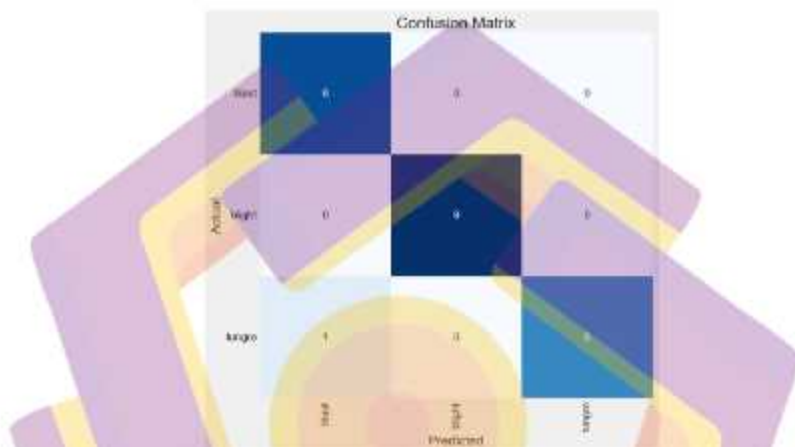
3. Skenario CNN 2 pada Dataset Enhancement Original



Gambar 4.28. Grafik *Training & Validation* Xception (Dataset Enhancement Original)

Pada Gambar 4.28 terlihat bahwa grafik *training and validation loss* model arsitektur Xception pada dataset enhancement original semakin rendah artinya

semakin baik, dan *epoch* terbaik terdapat pada urutan 16. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 6 dengan nilai *train_loss* 0.3168, *train_accuracy* 0.8698, *val_loss* 0.2687, dan *val_accuracy* 0.9583.



Gambar 4.29. *Confusion Matrix* Xception (Dataset Enhancement Original)

Gambar 4.29 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 2 pada dataset enhancement original dengan model arsitektur VGG16. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.29 jumlah TP adalah $8 + 9 + 6 = 23$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai

akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(23/24) * 100 = 95.83\%$ atau jika dibulatkan menjadi 96%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 8 / (8 + 0 + 1) = 8 / 9 = 0.8888 * 100 = 89\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 9 / (9 + 0 + 0) = 9 / 9 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 6 / (6 + 0 + 0) = 6 / 6 = 1 * 100 = 100\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.8888 + 1 + 1) / 3 = 2.8888 / 3 = 0.9626 * 100 = 96\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 9 / (9 + 0 + 0) = 9 / 9 = 1 * 100 = 100\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 6 / (6 + 0 + 1) = 6 / 7 = 0.8571 * 100 = 86\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(1 + 1 + 0.8571) / 3 = 2.8571 / 3 = 0.9523 * 100 = 95\%$.

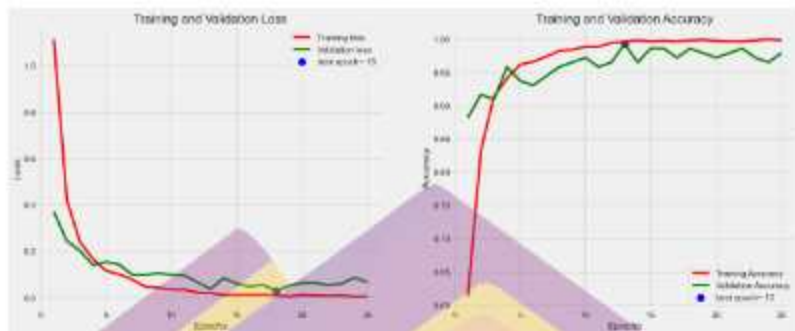
Classification Report:

	precision	recall	f1-score	support
blast	0.80	1.00	0.94	8
blight	1.00	1.00	1.00	9
tungro	1.00	0.86	0.92	7
accuracy			0.96	24
macro avg	0.96	0.95	0.95	24
weighted avg	0.96	0.96	0.96	24

Gambar 4.30. *Classification Report* Xception (Dataset Enhancement Original)

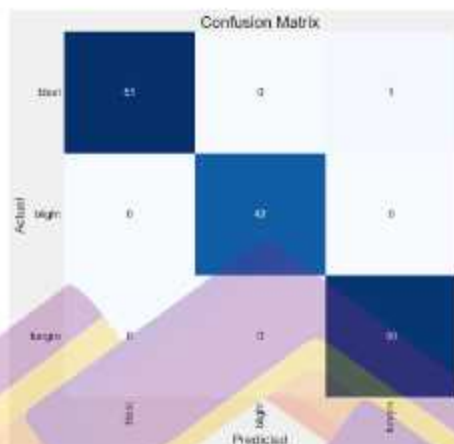
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.30.

4. Skenario CNN 2 pada Dataset Augmentasi Enhancement



Gambar 4.31. Grafik *Training & Validation Xception (Dataset Augmentasi Enhancement)*

Pada Gambar 4.31 terlihat bahwa grafik *training and validation loss* model arsitektur Xception pada dataset augmentasi enhancement semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 18. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 13 dengan nilai *train_loss* 0.0197, *train_accuracy* 0.9965, *val_loss* 0.0371, *val_accuracy* 0.9931.



Gambar 4.32. *Confusion Matrix* Xception (Dataset Augmentasi Enhancement)

Gambar 4.32 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 2 pada dataset augmentasi enhancement dengan model arsitektur Xception. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.32 jumlah TP adalah $51 + 42 + 50 = 143$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(143/144) * 100 = 99.30\%$ atau jika dibulatkan menjadi 99%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 51 / (51 + 0 + 0) = 51 / 51 = 1 * 100 = 100\%$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 42 / (42 + 0 + 0) = 42 / 42 = 1 * 100 = 100\%$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 50 / (50 + 0 + 1) = 50 / 51 = 0.9803 * 100 = 98\%$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(1 + 1 + 0.9803) / 3 = 2.9803 / 3 = 0.9934 * 100 = 99\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 51 / (51 + 0 + 1) = 51 / 52 = 0.9807 * 100 = 98\%$$

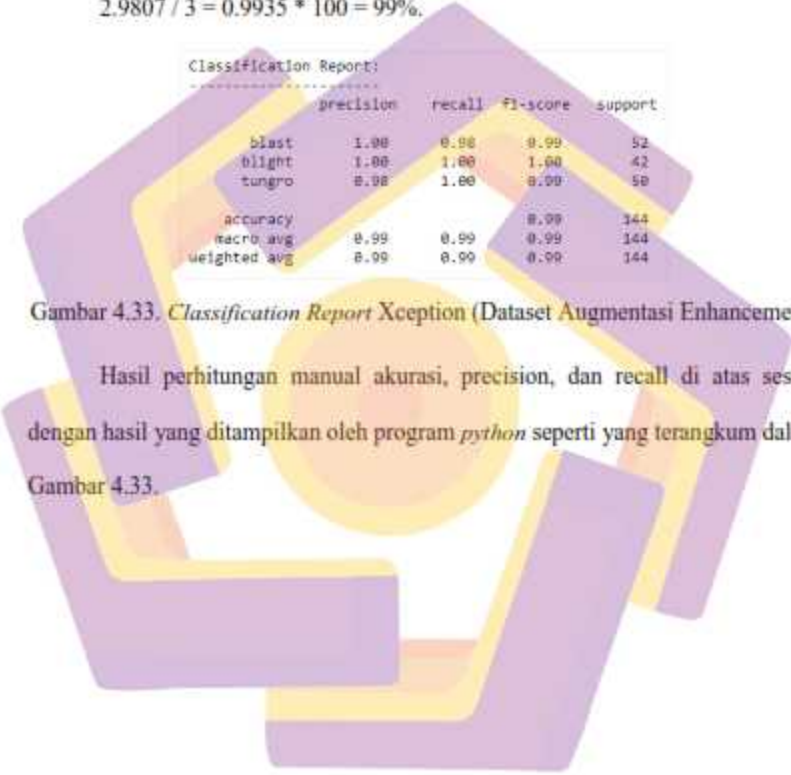
b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 42 / (42 + 0 + 0) = 42 / 42 = 1 * 100 = 100\%$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 50 / (50 + 0 + 0) = 50 / 50 = 1 * 100 = 100\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9807 + 1 + 1) / 3 = 2.9807 / 3 = 0.9935 * 100 = 99\%$.



Classification Report:

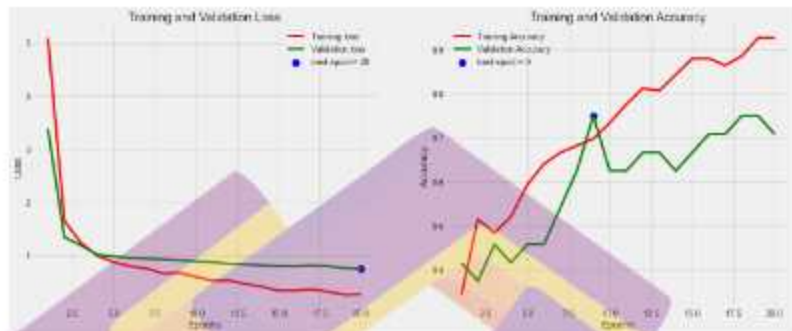
	precision	recall	f1-score	support
blast	1.00	0.98	0.99	52
blight	1.00	1.00	1.00	42
tungro	0.98	1.00	0.99	50
accuracy			0.99	144
macro avg	0.99	0.99	0.99	144
weighted avg	0.99	0.99	0.99	144

Gambar 4.33. *Classification Report* Xception (Dataset Augmentasi Enhancement)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.33.

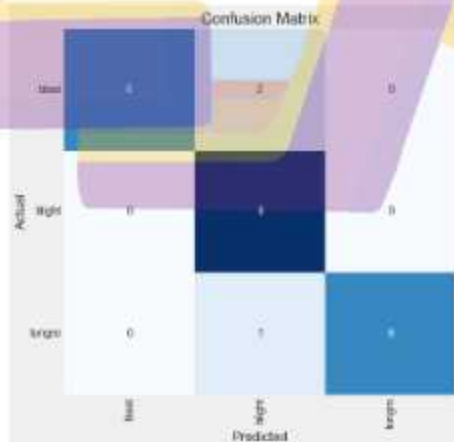
4.4.3 Hasil dan Pengujian CNN 3

1. Skenario CNN 3 pada Dataset Original



Gambar 4.34. Grafik *Training & Validation* NASNetMobile (Dataset Original)

Pada Gambar 4.34 terlihat bahwa grafik *training and validation loss* model arsitektur NASNetMobile pada dataset original dengan *epoch* terbaik terdapat pada urutan 20. Sedangkan grafik *training and validation accuracy* menunjukkan *epoch* terbaik pada urutan 9 dengan nilai *train_loss* 0.6763, *train_accuracy* 0.6979, *val_loss* 0.9081, dan *val_accuracy* 0.7500.



Gambar 4.35. *Confusion Matrix* NASNetMobile (Dataset Original)

Gambar 4.35 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 3 pada dataset original dengan model arsitektur NASNetMobile. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.35 jumlah TP adalah $6 + 9 + 6 = 21$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(21/24) * 100 = 87.5\%$ atau jika dibulatkan menjadi 88%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 6 / (6 + 0 + 0) = 6 / 6 = 1 * 100 = 100\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 9 / (9 + 2 + 1) = 9 / 12 = 0.75 * 100 = 75\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 6 / (6 + 0 + 0) = 6 / 6 = 1 * 100 = 100\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang

kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(1 + 0.75 + 1) / 3 = 2.75 / 3 = 0.9166 * 100 = 92\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 6 / (6 + 2 + 0) = 6 / 8 = 0.75 * 100 = 75\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 9 / (9 + 0 + 0) = 9 / 9 = 1 * 100 = 100\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 6 / (6 + 1 + 0) = 6 / 7 = 0.8571 * 100 = 86\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.75 + 1 + 0.8571) / 3 = 2.6071 / 3 = 0.8690 * 100 = 87\%$.

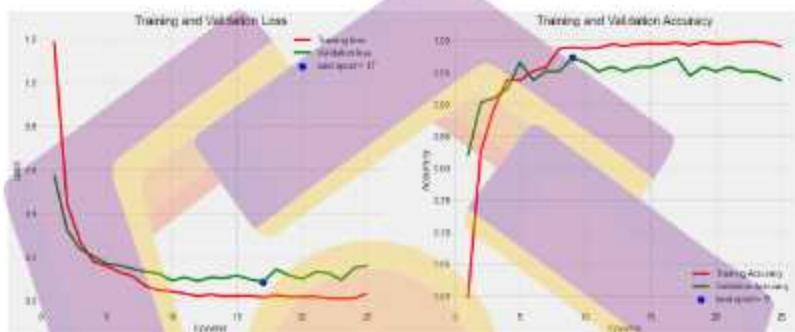
Classification Report:

	precision	recall	f1-score	support
blast	1.00	0.75	0.86	8
blight	0.75	1.00	0.86	9
tungro	1.00	0.86	0.92	7
accuracy			0.88	24
macro avg	0.92	0.87	0.88	24
weighted avg	0.91	0.88	0.88	24

Gambar 4.36. *Classification Report* NASNetMobile (Dataset Original)

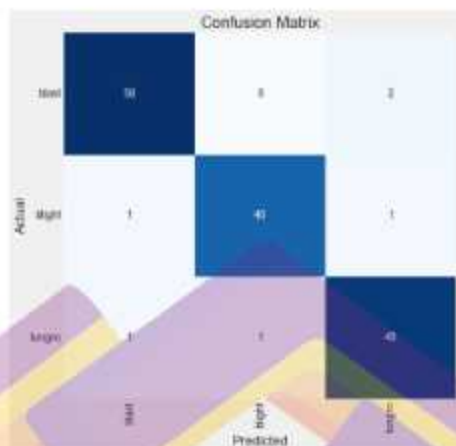
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.36.

2. Skenario CNN 3 pada Dataset Augmentasi Original



Gambar 4.37. Grafik *Training & Validation* NASNetMobile (Dataset Augmentasi Original)

Pada Gambar 4.37 terlihat bahwa grafik *training and validation loss* model arsitektur NASNetMobile pada dataset augmentasi original semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 17. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 9 dengan nilai *train_loss* 0.0471, *train_accuracy* 0.9887, *val_loss* 0.1250, dan *val_accuracy* 0.9722.



Gambar 4.38. *Confusion Matrix* NASNetMobile (Dataset Augmentasi Original)

Gambar 4.38 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 3 pada dataset augmentasi original dengan model arsitektur NASNetMobile. Nilai akurasi, *precision*, dan *recall* akan dihitung sebagai berikut.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.38 jumlah TP adalah $50 + 40 + 48 = 138$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(138/144) * 100 = 95.83\%$ atau jika dibulatkan menjadi 96%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 50 / (50 + 1 + 1) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 40 / (40 + 0 + 1) = 40 / 41 = 0.9756 * 100 = 98\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 48 / (48 + 1 + 2) = 48 / 51 = 0.9411 * 100 = 94\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(0.9615 + 0.9756 + 0.9411) / 3 = 2.8782 / 3 = 0.9584 * 100 = 96\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 50 / (50 + 0 + 2) = 50 / 52 = 0.9615 * 100 = 96\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 40 / (40 + 1 + 1) = 40 / 42 = 0.9523 * 100 = 95\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 48 / (48 + 1 + 1) = 48 / 50 = 0.96 * 100 = 96\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan

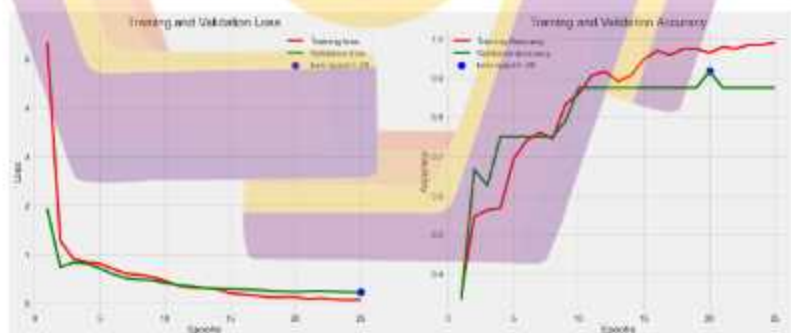
jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9615 + 0.9523 + 0.96) / 3 = 2.8738 / 3 = 0.9579 * 100 = 96\%$.

Classification Report:				
	precision	recall	f1-score	support
blest	0.96	0.96	0.96	52
blight	0.98	0.95	0.96	42
tungro	0.94	0.96	0.95	58
accuracy			0.96	144
macro avg	0.96	0.96	0.96	144
weighted avg	0.96	0.96	0.96	144

Gambar 4.39. *Classification Report Model NASNetMobile (Dataset Augmentasi Original)*

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.39.

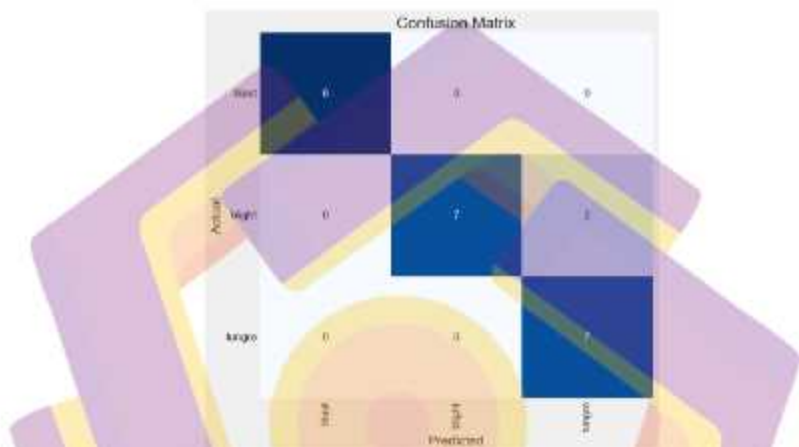
3. Skenario CNN 3 pada Dataset Enhancement Original



Gambar 4.40. Grafik *Training & Validation NASNetMobile (Dataset Enhancement Original)*

Pada Gambar 4.40 terlihat bahwa grafik *training and validation loss* model arsitektur NASNetMobile pada dataset enhancement original semakin rendah

artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 25. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 20 dengan nilai *train_loss* 0.1298, *accuracy* 0.9635, *val_loss* 0.2361, dan *val_accuracy* 0.9167.



Gambar 4.41. *Confusion Matrix* NASNetMobile (Dataset Enhancement Original)

Gambar 4.41 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 3 pada dataset enhancement original dengan model arsitektur NASNetMobile. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.41 jumlah TP adalah $8 + 7 + 7 = 22$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai

akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(22/24) * 100 = 91.66\%$ atau jika dibulatkan menjadi 92%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 7 / (7 + 0 + 0) = 7 / 7 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 7 / (7 + 2 + 0) = 7 / 9 = 0.7777 * 100 = 78\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(1 + 1 + 0.7777) / 3 = 2.7777 / 3 = 0.9259 * 100 = 93\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 7 / (7 + 0 + 2) = 7 / 9 = 0.7777 * 100 = 78\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 7 / (7 + 0 + 7) = 7 / 14 = 0.5 * 100 = 50\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(1 + 0.7777 + 1) / 3 = 2.7777 / 3 = 0.9259 * 100 = 93\%$.

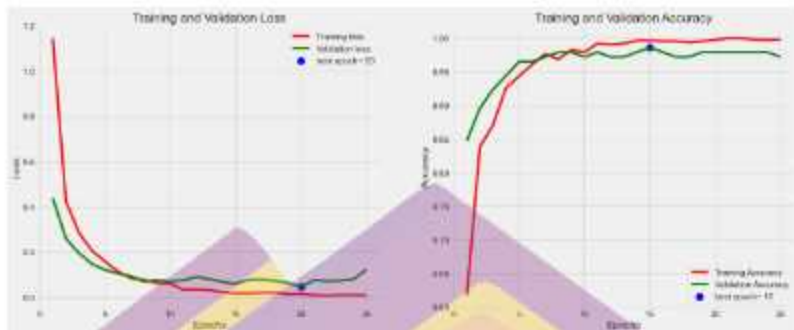
Classification Report:

	precision	recall	f1-score	support
blast	1.00	1.00	1.00	8
blight	1.00	0.75	0.88	9
tungro	0.78	1.00	0.88	7
accuracy			0.92	24
macro avg	0.93	0.93	0.92	24
weighted avg	0.94	0.92	0.92	24

Gambar 4.42. *Classification Report* NASNetMobile (Dataset Enhancement Original)

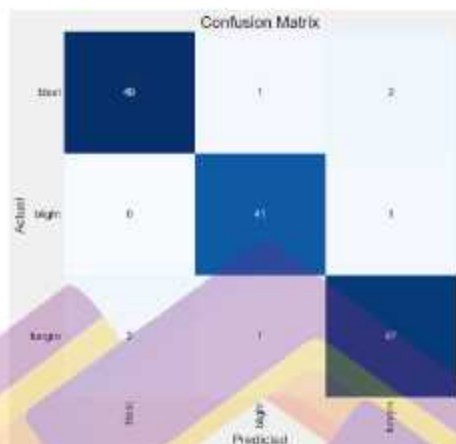
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.42.

4. Skenario CNN 3 pada Dataset Augmentasi Enhancement



Gambar 4.43. Grafik *Training & Validation* NASNetMobile (Dataset Augmentasi Enhancement)

Pada Gambar 4.43 terlihat bahwa grafik *training and validation loss* model arsitektur NASNetMobile pada dataset augmentasi enhancement semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 20. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 15 dengan nilai *train_loss* 0.0182, *train_accuracy* 0.9965, *val_loss* 0.0592, dan *val_accuracy*: 0.9861.



Gambar 4.44. *Confusion Matrix* NASNetMobile (Dataset Augmentasi Enhancement)

Gambar 4.44 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 3 pada dataset augmentasi enhancement dengan model arsitektur NASNetMobile. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.44 jumlah TP adalah $49 + 41 + 47 = 137$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(137/144) * 100 = 95.13\%$ atau jika dibulatkan menjadi 95%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 49 / (49 + 0 + 2) = 49 / 51 = 0.9607 * 100 = 96\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 41 / (41 + 1 + 1) = 41 / 43 = 0.9534 * 100 = 95\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 47 / (47 + 1 + 2) = 47 / 50 = 0.94 * 100 = 94\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.9607 + 0.9534 + 0.94) / 3 = 2.8541 / 3 = 0.9513 * 100 = 95\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 49 / (49 + 1 + 2) = 49 / 52 = 0.9423 * 100 = 94\%.$$

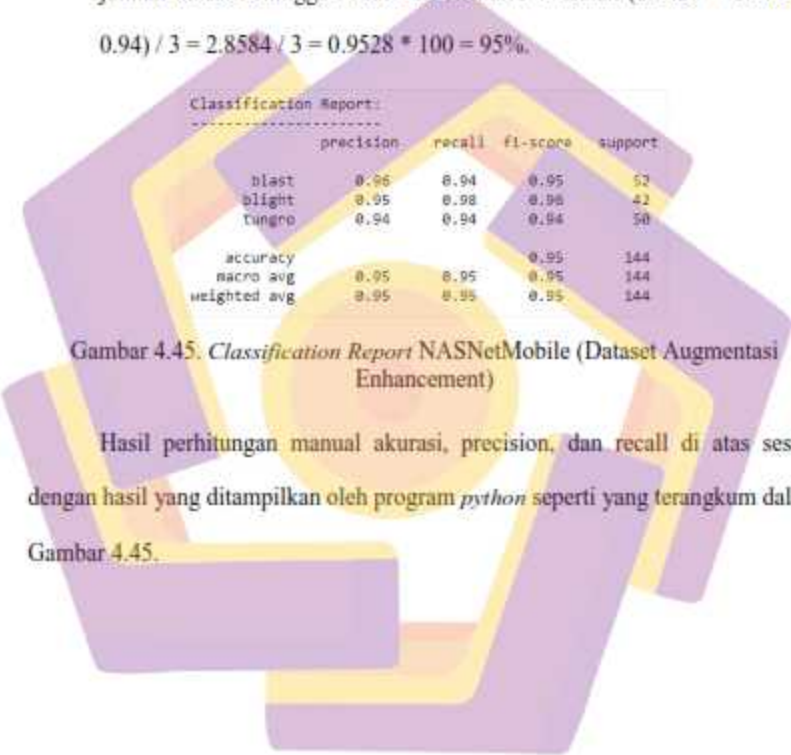
b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 41 / (41 + 0 + 1) = 41 / 42 = 0.9761 * 100 = 98\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 47 / (47 + 1 + 2) = 47 / 50 = 0.94 * 100 = 94\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9423 + 0.9761 + 0.94) / 3 = 2.8584 / 3 = 0.9528 * 100 = 95\%$.



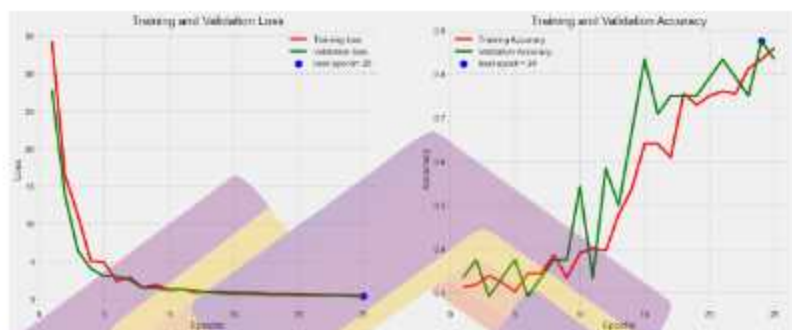
Classification Report:				
	precision	recall	f1-score	support
blast	0.96	0.94	0.95	52
blight	0.95	0.98	0.96	42
tungro	0.94	0.94	0.94	50
accuracy			0.95	144
macro avg	0.95	0.95	0.95	144
weighted avg	0.95	0.95	0.95	144

Gambar 4.45. *Classification Report* NASNetMobile (Dataset Augmentasi Enhancement)

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.45.

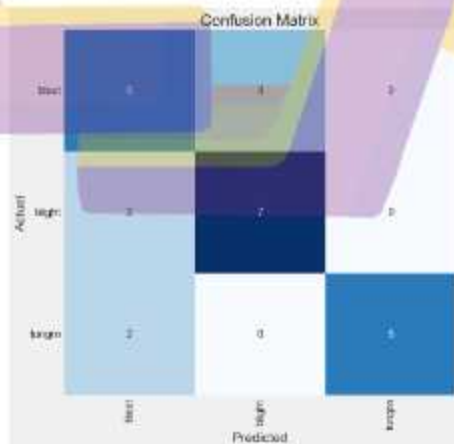
4.4.4 Hasil dan Pengujian CNN 4

1. Skenario CNN 4 pada Dataset Original



Gambar 4.46. Grafik *Training & Validation Custom X (Dataset Original)*

Pada Gambar 4.46 terlihat bahwa grafik *training and validation loss* model arsitektur Custom X pada dataset original dengan *epoch* terbaik terdapat pada urutan 25. Sedangkan grafik *training and validation accuracy* menunjukkan *epoch* terbaik pada urutan 24 dengan nilai *train_loss* 0.5327, *accuracy* 0.8333, *val_loss* 0.4316, dan *val_accuracy* 0.8750.



Gambar 4.47. *Confusion Matrix Custom X (Dataset Original)*

Gambar 4.47 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 4 pada dataset original dengan model arsitektur Custom X. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada Gambar 4.47 jumlah TP adalah $5 + 7 + 5 = 17$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(17/24) * 100 = 70.83\%$ atau jika dibulatkan menjadi 71%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 5 / (5 + 2 + 2) = 5 / 9 = 0.5555 * 100 = 56\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 7 / (7 + 3 + 0) = 7 / 10 = 0.7 * 100 = 70\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 5 / (5 + 0 + 0) = 5 / 5 = 1 * 100 = 100\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang

kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(0.5555 + 0.7 + 1) / 3 = 2.2555 / 3 = 0.7518 * 100 = 75\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 5 / (5 + 3 + 0) = 5 / 8 = 0.625 * 100 = 62\%$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 7 / (7 + 2 + 0) = 7 / 9 = 0.7777 * 100 = 78\%$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 5 / (5 + 0 + 2) = 5 / 7 = 0.7142 * 100 = 71\%$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.625 + 0.7777 + 0.7142) / 3 = 2.1169 / 3 = 0.7056 * 100 = 71\%$.

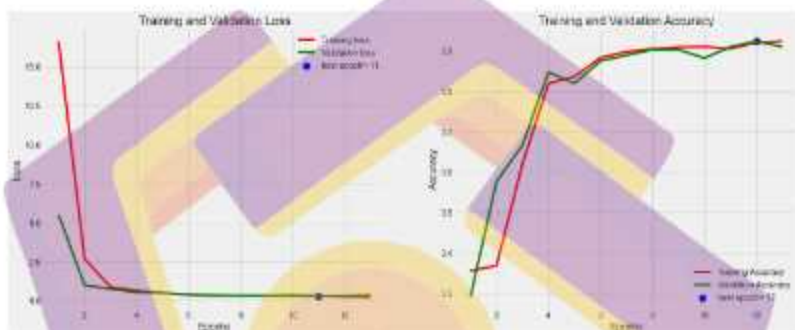
Classification Report:

	precision	recall	f1-score	support
blast	0.56	0.62	0.59	8
blight	0.70	0.78	0.74	9
tungro	1.00	0.71	0.83	7
accuracy			0.71	24
macro avg	0.75	0.71	0.72	24
weighted avg	0.74	0.71	0.72	24

Gambar 4.48. *Classification Report Custom X (Dataset Original)*

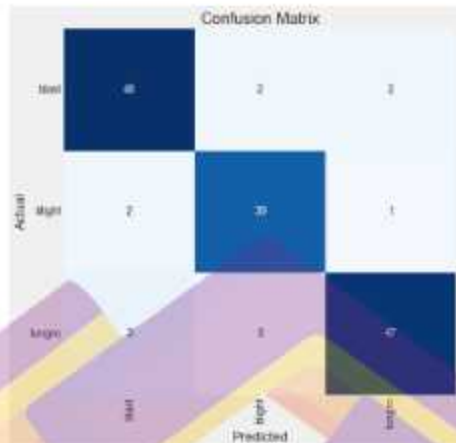
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.48.

2. Skenario CNN 3 pada Dataset Augmentasi Original



Gambar 4.49. Grafik *Training & Validation Custom X (Dataset Augmentasi Original)*

Pada Gambar 4.49 terlihat bahwa grafik *training and validation loss* model arsitektur Custom X dari penulis pribadi pada dataset augmentasi original. Jika grafik semakin rendah artinya semakin baik, dan *epoch 11* menjadi terbaik. Sedangkan grafik *training and validation accuracy* menunjukkan jika semakin tinggi artinya pelatihan semakin baik, serta *epoch 12* menjadi terbaik dengan nilai *train_loss* 0.2589, *train_accuracy* 0.9201, *val_loss* 0.3054, dan *val_accuracy* 0.9.



Gambar 4.50. *Confusion Matrix* Custom X (Dataset Augmentasi Original)

Gambar 4.50 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 4 pada dataset augmentasi original dengan model arsitektur Custom X dari penulis. Nilai akurasi, *precision*, dan *recall* akan dihitung sebagai berikut.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada gambar di atas jumlah TP adalah $48 + 39 + 47 = 134$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(134/144) * 100 = 93.06\%$ atau jika dibulatkan menjadi 93%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 48 / (48 + 2 + 3) = 48 / 53 = 0.9056 * 100 = 91\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 39 / (39 + 2 + 0) = 39 / 41 = 0.9512 * 100 = 95\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 47 / (47 + 1 + 2) = 47 / 50 = 0.94 * 100 = 94\%.$$

Hasil precision secara keseluruhan didapatkan dengan menjumlahkan semua nilai precision dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga precision dari skenario 1 adalah $(0.9056 + 0.9512 + 0.94) / 3 = 2.7968 / 3 = 0.9322 * 100 = 93\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 48 / (48 + 2 + 2) = 48 / 52 = 0.9230 * 100 = 92\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 39 / (39 + 2 + 1) = 39 / 42 = 0.9285 * 100 = 93\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 47 / (47 + 0 + 3) = 47 / 50 = 0.94 * 100 = 94\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan

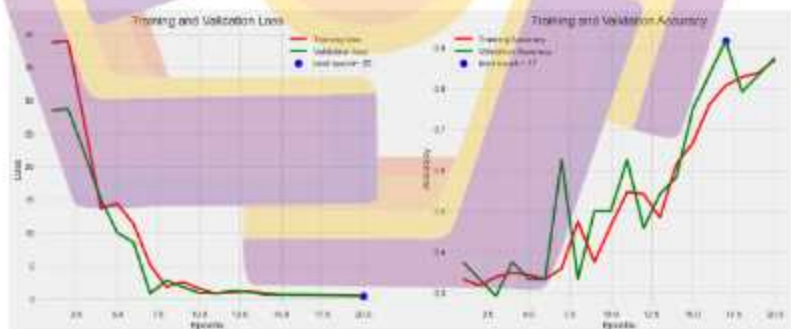
jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9230 + 0.9285 + 0.94) / 3 = 2.7815 / 3 = 0.9305 * 100 = 93\%$.

Classification Report:				
	precision	recall	f1-score	support
blast	0.91	0.92	0.91	52
blight	0.95	0.93	0.94	42
tungro	0.94	0.94	0.94	50
accuracy			0.93	144
macro avg	0.93	0.93	0.93	144
weighted avg	0.93	0.93	0.93	144

Gambar 4.51. *Classification Report Custom X (Dataset Augmentasi Original)*

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.51.

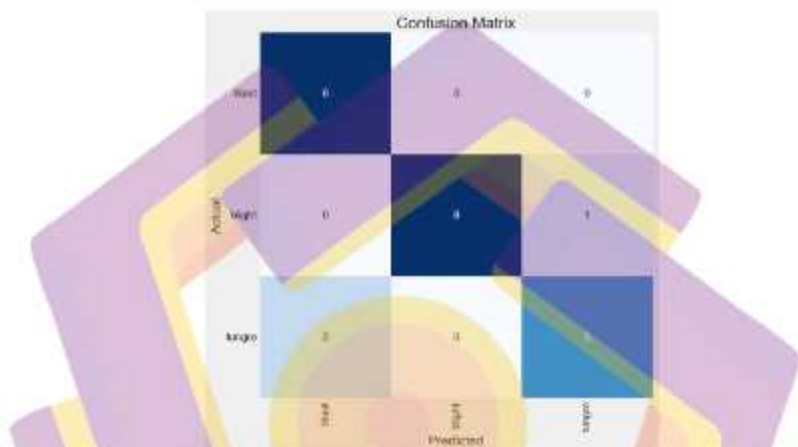
3. Skenario CNN 4 pada Dataset Enhancement Original



Gambar 4.52. Grafik *Training & Validation Custom X (Dataset Enhancement Original)*

Pada Gambar 4.52 terlihat bahwa grafik *training and validation loss* model arsitektur Custom X pada dataset enhancement original semakin rendah artinya

semakin baik, dan *epoch* terbaik terdapat pada urutan 20. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 17 dengan nilai *train_loss* 0.6450, *train_accuracy* 0.8073, *val_loss* 0.5870, dan *val_accuracy* 0.9167.



Gambar 4.53. *Confusion Matrix Custom X (Dataset Enhancement Original)*

Gambar 4.53 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 4 pada dataset enhancement original dengan model arsitektur Custom X. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai *precision* dan *recall* dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada gambar di atas jumlah TP adalah $8 + 8 + 5 = 21$, kemudian jumlah data testing secara keseluruhan adalah 24 data. Nilai

akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(21/24) * 100 = 87.5\%$ atau jika dibulatkan menjadi 88%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 8 / (8 + 0 + 2) = 8 / 10 = 0.8 * 100 = 80\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 5 / (5 + 1 + 0) = 5 / 6 = 0.8333 * 100 = 83\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.8 + 1 + 0.8333) / 3 = 2.6333 / 3 = 0.8777 * 100 = 88\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 8 / (8 + 0 + 0) = 8 / 8 = 1 * 100 = 100\%.$$

b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 8 / (8 + 0 + 1) = 8 / 9 = 0.8888 * 100 = 89\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 5 / (5 + 0 + 2) = 5 / 7 = 0.7142 * 100 = 71\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(1 + 0.8888 + 0.7142) / 3 = 2.603 / 3 = 0.8676 * 100 = 87\%$.

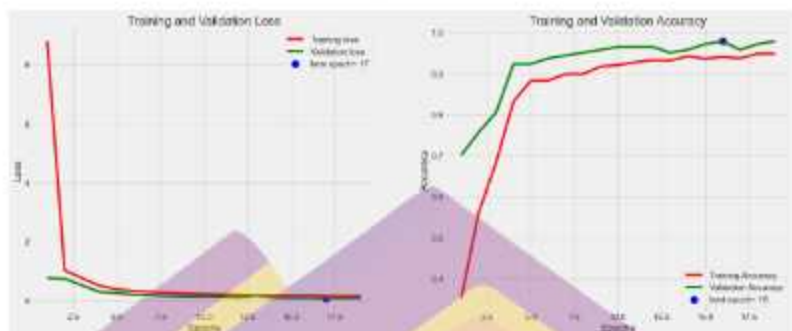
Classification Report:

	precision	recall	f1-score	support
blast	0.88	1.00	0.89	8
blight	1.00	0.89	0.94	9
tungro	0.83	0.71	0.77	7
accuracy			0.88	24
macro avg	0.88	0.87	0.87	24
weighted avg	0.88	0.88	0.87	24

Gambar 4.54. *Classification Report* NASNetMobile (Dataset Enhancement Original)

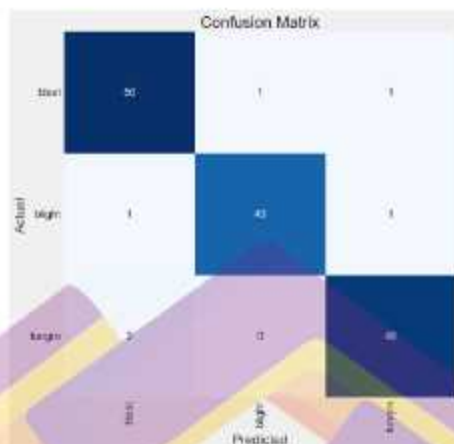
Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.54.

4. Skenario CNN 4 pada Dataset Augmentasi Enhancement



Gambar 4.55. Grafik *Training & Validation Custom X (Dataset Augmentasi Enhancement)*

Pada Gambar 4.55 terlihat bahwa grafik *training and validation loss* model arsitektur NASNetMobile pada dataset augmentasi enhancement semakin rendah artinya semakin baik, dan *epoch* terbaik terdapat pada urutan 17. Sedangkan grafik *training and validation accuracy* menunjukkan semakin tinggi artinya pelatihan semakin baik, serta *epoch* terbaik pada urutan 16 dengan nilai *train_loss* 0.1889, *train_accuracy* 0.9410, *val_loss* 0.0914, dan *val_accuracy*: 0.9792.



Gambar 4.56. *Confusion Matrix* Custom X (Dataset Augmentasi Enhancement)

Gambar 4.56 adalah representasi hasil pengujian *confusion matrix* dari skenario percobaan CNN 4 pada dataset augmentasi enhancement dengan model arsitektur Custom X. Klasifikasi penyakit daun padi memiliki 3 kelas, maka pengujian yang digunakan adalah *confusion matrix for multi-class* dimana untuk mendapatkan nilai precision dan recall dihitung masing-masing kelas.

1. Akurasi

Nilai akurasi didapatkan dari keseluruhan jumlah *True Positive* dari masing-masing kelas. Pada gambar di atas jumlah TP adalah $50 + 40 + 48 = 138$, kemudian jumlah data testing secara keseluruhan adalah 144 data. Nilai akurasi adalah total TP dibagi total data kemudian dikalikan 100%, maka $(137/144) * 100 = 95.83\%$ atau jika dibulatkan menjadi 96%.

2. Precision

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

a. Nilai Precision Kelas *Blast*

$$\text{Precision} = 50 / (50 + 1 + 2) = 50 / 53 = 0.9433 * 100 = 94\%.$$

b. Nilai Precision Kelas *Blight*

$$\text{Precision} = 40 / (40 + 1 + 0) = 40 / 41 = 0.9756 * 100 = 98\%.$$

c. Nilai Precision Kelas *Tungro*

$$\text{Precision} = 48 / (48 + 1 + 1) = 48 / 50 = 0.96 * 100 = 96\%.$$

Hasil *precision* secara keseluruhan didapatkan dengan menjumlahkan semua nilai *precision* dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga *precision* dari skenario 1 adalah $(0.9433 + 0.9756 + 0.96) / 3 = 2.8789 / 3 = 0.9596 * 100 = 96\%$.

3. Recall

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

a. Nilai Recall Kelas *Blast*

$$\text{Recall} = 50 / (50 + 1 + 1) = 50 / 52 = 0.9615 * 100 = 96\%.$$

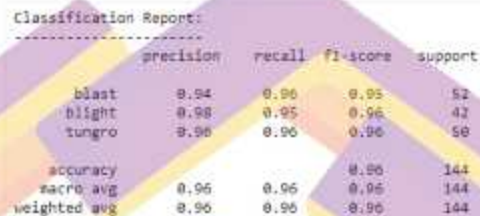
b. Nilai Recall Kelas *Blight*

$$\text{Recall} = 40 / (40 + 1 + 1) = 40 / 42 = 0.9523 * 100 = 95\%.$$

c. Nilai Recall Kelas *Tungro*

$$\text{Recall} = 48 / (48 + 0 + 2) = 48 / 50 = 0.96 * 100 = 96\%.$$

Hasil recall secara keseluruhan didapatkan dengan menjumlahkan semua nilai recall dari masing-masing kelas yang kemudian dibagi dengan jumlah kelas. Sehingga recall dari skenario 1 adalah $(0.9615 + 0.9523 + 0.96) / 3 = 2.8738 / 3 = 0.9579 * 100 = 96\%$.



	precision	recall	f1-score	support
blast	0.94	0.96	0.95	52
blight	0.98	0.95	0.96	42
tungro	0.98	0.96	0.96	50
accuracy			0.96	144
macro avg	0.96	0.96	0.96	144
weighted avg	0.96	0.96	0.96	144

Gambar 4.57. *Classification Report Custom X (Dataset Augmentasi Enhancement)*

Hasil perhitungan manual akurasi, precision, dan recall di atas sesuai dengan hasil yang ditampilkan oleh program *python* seperti yang terangkum dalam Gambar 4.57.

4.4.5. Perbandingan Hasil Percobaan

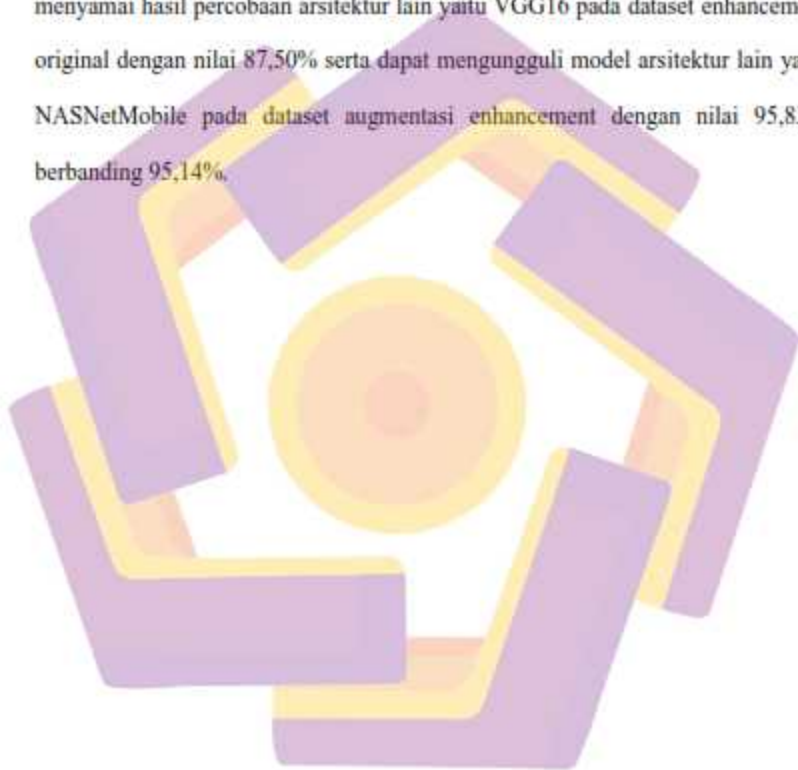
Sebagai bahan perbandingan hasil percobaan pada masing-masing model arsitektur *Convolutional Neural Network (CNN)* dengan olah dataset berbeda yang telah dilakukan, berikut penulis sajikan Tabel 4.4 sebagai detail nilai yang didapatkan pada masing-masing skenario CNN.

Tabel 4.4. Perbandingan Hasil Percobaan

No	Arsitektur	Dataset	Training Accuracy	Validation Accuracy	Testing Accuracy	Precision	Recall
1	VGG16	Original	92,71%	75%	79,17%	82,32%	78,89%
2		Augmentasi Original	97,92%	95,83%	96,53%	96,50%	96,71%
3		Enhancement Original	91,15%	83,35%	87,50%	87,82%	87,36%
4		Augmentasi Enhancement	98,87%	98,61%	96,53%	96,57%	96,71%
5	Xception	Original	100%	87,50%	91,67%	93,33%	90,47%
6		Augmentasi Original	99,13%	97,22%	97,22%	97,26%	97,20%
7		Enhancement Original	86,98%	95,83%	95,83%	96,26%	95,23%
8		Augmentasi Enhancement	99,63%	99,31%	99,31%	99,34%	99,35%
9	NASNet Mobile	Original	69,79%	75%	87,50%	91,66%	86,90%
10		Augmentasi Original	98,87%	97,22%	95,83%	95,84%	95,79%
11		Enhancement Original	96,35%	91,67%	91,67%	92,59%	92,59%
12		Augmentasi Enhancement	99,65%	98,61%	95,14%	95,13%	95,28%
13	Custom X	Original	83,33%	87,50%	70,83%	75,18%	70,56%
14		Augmentasi Original	92,01%	90,00%	93,06%	93,22%	93,05%
15		Enhancement Original	80,73%	91,67%	87,50%	87,77%	86,76%
16		Augmentasi Enhancement	94,10%	97,92%	95,83%	95,96%	95,79%

Dapat dilihat pada Tabel 4.4 bahwa hasil percobaan dengan memanfaatkan augmentasi data original untuk memperbanyak dataset pada masing-masing arsitektur (1440 gambar) mendapatkan nilai akurasi testing yang lebih baik dibanding data mentah original (80 gambar), yaitu VGG16 dari 79,17% meningkat menjadi 96,53%, Xception dari 91,67% meningkat menjadi 97,22%, NASNetMobile dari 87,50% menjadi 95,83%, dan Custom X dari 70,83% meningkat menjadi 93,06%. Kemudian proses *preprocessing* lanjutan atau *enhancement* data gambar dengan menghapus background agar tidak mengganggu proses deteksi dan juga zoom agar lebih terlihat jelas fokus objeknya pada masing-masing arsitektur juga meningkatkan nilai akurasi pada data original (jumlah sama 80 gambar), yaitu VGG16 dari 79,17% menjadi 87,50%, Xception dari 91,67% menjadi 95,83%, NASNetMobile dari 87,50% menjadi 91,67%, dan Custom X dari

70,83% menjadi 87,50%. Selanjutnya untuk *enhancement* data setelah augmentasi meningkat pada arsitektur Xception yaitu dari 97,22% menjadi 99,31% dan pada arsitektur Custom X yaitu dari 93,06% menjadi 95,83%. Terdapat juga temuan menarik bahwa model arsitektur rancangan penulis sendiri yaitu Custom X dapat menyamai hasil percobaan arsitektur lain yaitu VGG16 pada dataset *enhancement* original dengan nilai 87,50% serta dapat mengungguli model arsitektur lain yaitu NASNetMobile pada dataset augmentasi *enhancement* dengan nilai 95,83% berbanding 95,14%.



BAB V

PENUTUP

5.1. Kesimpulan

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

1. Setelah dilakukan percobaan pada masing-masing skenario CNN didapatkan hasil arsitektur VGG16 mendapatkan nilai akurasi tertinggi 96,53% pada dataset setelah augmentasi original dan pada dataset setelah augmentasi enhancement. Arsitektur Xception mendapatkan nilai akurasi tertinggi 99,31% pada dataset setelah augmentasi enhancement. Arsitektur NASNetMobile mendapatkan nilai akurasi tertinggi 95,83% pada dataset setelah augmentasi original. Terakhir arsitektur rancangan sendiri Custom X mendapatkan nilai akurasi tertinggi 95,83% pada dataset setelah augmentasi enhancement.
2. Rancangan arsitektur terbaik pada penelitian ini untuk mendeteksi dan mengklasifikasi citra penyakit pada daun padi adalah Xception karena mengungguli hasil dari arsitektur-arsitektur lain di semua skenario. Hasil terbaik diperoleh pada skenario ke 8 pada CNN 2 menggunakan arsitektur Xception dengan nilai akurasi testing tertinggi sebesar 99.31% setelah dilakukan augmentasi dan *enhancement* data lanjutan pada dataset.
3. Augmentasi dilakukan untuk menambah keberagaman data sehingga dataset yang digunakan untuk model klasifikasi menjadi lebih kaya akan

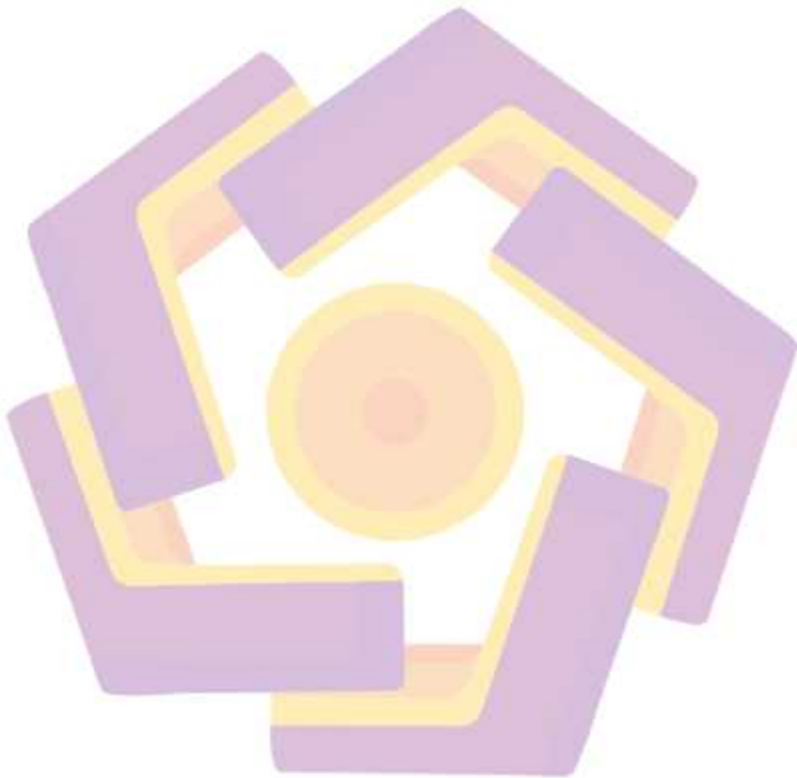
informasi. Hal ini terbukti dari nilai akurasi setelah augmentasi data lebih tinggi dari data akurasi pada dataset original, contohnya pada skenario CNN 2 dengan arsitektur Custom X dimana pada saat sebelum augmentasi mendapatkan nilai akurasi 70,83% kemudian setelah augmentasi naik menjadi 93,06% dan setelah preprocessing lanjutan atau enhancement naik lagi menjadi 95,83%.

5.2. Saran

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

1. Jumlah dan varian dataset sebaiknya diperkaya atau diperbanyak lagi agar hasil yang didapatkan lebih optimal, sekaligus semakin meyakinkan bahwa model yang dibangun terbukti baik dengan variasi dataset lain.
2. Mencoba menggunakan metode *transfer learning* dengan arsitektur lain seperti ResNet, AlexNet, DenseNet, MobileNet, Inception, dan lain sebagainya.
3. Mencoba membangun custom arsitektur sendiri dengan beberapa layer tambahan atau susunan yang berbeda serta epoch yang lebih banyak guna mendapatkan hasil yang lebih maksimal.
4. Mencoba menggunakan cara *preprocessing* atau *enhancement* data dan augmentasi yang lain guna meningkatkan akurasi pada arsitektur-arsitektur yang belum optimal.

5. Mengembangkan penentuan penanganan apa yang akan dilakukan ketika penyakit pada daun tanaman padi sudah diidentifikasi, penanganan yang dilakukan bisa berdasarkan jenis dan level penyakit pada setiap pohonnya.



DAFTAR PUSTAKA

PUSTAKA BUKU

- Faisal, M. R., & Nugrahad, D. T. (2019). *Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R*. Banjarbaru: Scripta Cendekia.
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing, 3rd edition*. New Jersey, US: Prentice Hall.
- Nafi'iyah, N., & Mujilawati, S. (2018). *Buku Ajar Citra Binarisasi Dan Enhancement*. Sleman: Deepublish.
- Siahaan, V., & Sianipar, R. H. (2020). *Panduan Praktis dan Komplet Pemrosesan Citra Digital Dengan MatLab*. BALIGE PUBLISHING.
- Suyanto. (2017). *Data Mining untuk Klasifikasi dan Klusterisasi Data*. Bandung: Informatika

PUSTAKA JURNAL ILMIAH ATAU PROSIDING

- Aggarwal, S., Suchithra, M., Chandramouli, N., Sarada, M., Verma, A., Vettrithangam, D., ... Ambachew Adugna, B. (2022). Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improve Agro-Business. *Scientific Programming*, 2022, 1–13. <https://doi.org/10.1155/2022/1757888>
- Andrianto, H., Suhardi, Faizal, A., & Armandika, F. (2020). Smartphone Application for Deep Learning-Based Rice Plant Disease Detection. *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, 387–392. IEEE. <https://doi.org/10.1109/ICITSI50517.2020.9264942>
- Arrofiqoh, E. N., & Harintaka, H. (2018). Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi. *GEOMATIKA*, 24(2), 61. <https://doi.org/10.24895/JIG.2018.24-2.810>
- Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173, 105393. <https://doi.org/10.1016/j.compag.2020.105393>
- Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., & Qi, L. (2021). Automatic Diagnosis of Rice Diseases Using Deep Learning. *Frontiers in Plant Science*, 12. <https://doi.org/10.3389/fpls.2021.701038>

- Enkvetchakul, P., & Surinta, O. (2021). Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition. *Applied Science and Engineering Progress*. <https://doi.org/10.14416/j.asep.2021.01.003>
- Hossain, S. Md. M., Tanjil, Md. M. M., Ali, M. A. Bin, Islam, M. Z., Islam, Md. S., Mobassirin, S., ... Islam, S. M. R. (2020). *Rice Leaf Diseases Recognition Using Convolutional Neural Networks*. https://doi.org/10.1007/978-3-030-65390-3_23
- Ibrahim, N., Lestary, G. A., Hanafi, F. S., Saleh, K., Pratiwi, N. K. C., Haq, M. S., & Mastur, A. I. (2022). Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 162. <https://doi.org/10.26760/elkomika.v10i1.162>
- Jatmika, S., & Saputra, D. E. (2022). Rice Plants Disease Identification Using Deep Learning with Convolutional Neural Network Method. *Sinkron*, 7(3), 2008–2016. <https://doi.org/10.33395/sinkron.v7i3.11540>
- Julianto, A., Sunyoto, A., & Wibowo, F. W. (2022). Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, 3(2), 98–105. <https://doi.org/10.46764/teknimedia.v3i2.77>
- Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, 2(1), 37–45. <https://doi.org/10.20895/dinda.v2i1.341>
- Matin, Md. M. H., Khatun, A., Moazzam, Md. G., & Uddin, M. S. (2020). An Efficient Disease Detection Technique of Rice Leaf Using AlexNet. *Journal of Computer and Communications*, 08(12), 49–57. <https://doi.org/10.4236/jcc.2020.812005>
- Oktaviana, U. N., Hendrawan, R., Annas, A. D. K., & Wicaksono, G. W. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(6), 1216–1222. <https://doi.org/10.29207/resti.v5i6.3607>
- 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>

- Purbasari, I. Y., Rahmat, B., & Putra PN, C. S. (2021). Detection of Rice Plant Diseases using Convolutional Neural Network. *IOP Conference Series: Materials Science and Engineering*, 1125(1), 012021. <https://doi.org/10.1088/1757-899X/1125/1/012021>
- Rozaqi, A. J., Sunyoto, A., & Arief, M. rudyanto. (2021). Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network. *Creative Information Technology Journal*, 8(1), 22. <https://doi.org/10.24076/citec.2021v8i1.263>
- Simonyan, K., & Zisserman, A. (2014, September 4). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556S*
- Suseno, J. R. K., Minarno, A. E., & Azhar, Y. (2023). Implementation of Pretrained VGG16 Model for Rice Leaf Disease Classification using Image Segmentation. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. <https://doi.org/10.22219/kinetik.v8i1.1592>
- Tejaswini, P., Singh, P., Ramchandani, M., Rathore, Y. K., & Janghel, R. R. (2022). Rice Leaf Disease Classification Using Cnn. *IOP Conference Series: Earth and Environmental Science*, 1032(1), 012017. <https://doi.org/10.1088/1755-1315/1032/1/012017>
- Walascha, A., Febriana, A., Saputri, D., Haryanti, D. S. N., Tsania, R., Sanjaya, Y., & Priyanti. (2021). Review Artikel: Inventarisasi Jenis Penyakit yang Menyerang Daun Tanaman Padi (*Oryza sativa* L.). *Prosiding Seminar Nasional Biologi*, 1(2), 471–478. Padang. <https://doi.org/10.24036/proseminasbio/vol1/150>