KLASIFIKASI HAMA DAN PENYAKIT TANAMAN PADI DENGAN METODE CONVOLUTIONAL NEURAL NETWORK

SKRIPSI



Oleh:

Dzakiyyah Azahra Mooduto

NIM : 531420046

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS NEGERI GORONTALO

2023

# **KATA PENGANTAR**

Alhamdulillahirabbil ‘alamin, puji syukur kepada Allah SWT, yang telah memberikan nikmat kesehatan, waktu, petunjuk kepada penulis sehingga pembuatan laporan dengan judul “Klasifikasi Hama dan Penyakit Tanaman Padi dengan Metode *Convolutional Neural Network*” dapat terselesaikan. Salam semoga selalu dikirimkan kepada baginda Nabi Muhammad SAW. Semoga kita mendapatkan syafaat beliau hingga di hari akhir kelak. Dalam proses pembuatan sampai tersusunnya laporan ini, penulis telah memperoleh bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis mengucapkan terima kasih kepada:

1. Allah Subhanahu Wa Ta’Ala atas segala rahmat dan ridho yang selalu dicurahkan kepada hambaNya.
2. Kedua orang tua saya Ibu Yanti Saleh dan Bapak Andi Fahmi Sorejang dan keluarga tercinta yang telah membimbing dan memotivasi peneliti selama ini.
3. Bapak Dr. Ir. Eduart Wolok, ST., MT selaku Rektor Universitas Negeri Gorontalo, Wakil Rektor I, Wakil Rektor II dan Wakil Rektor III.
4. Bapak Dr. Sardi Salim, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Gorontalo.
5. Ibu Roviana H. Dai, S.Kom., MT selaku Ketua Jurusan Teknik Informatika.
6. Ibu Lillyan Hadjaratie, S.Kom, M.Si selaku dosen pembimbing I dan dosen penasehat akademik yang juga telah membimbing dan banyak membantu peneliti selama ini.
7. Ibu Indhitya R. Padiku, S.Kom, M.Kom, selaku selaku dosen pembimbing II telah membimbing dan banyak membantu peneliti selama ini.
8. Bapak Rahman Takdir, S.Kom, M.Cs selaku penguji 1, Muchlis Polin, S.Kom, M.Com selaku penguji 2, Nikmasari Pakaya, S.Kom., M.T. selaku penguji 3 yang telaah memberikan saran serta arahan untuk proses penyelesain skripsi peneliti.
9. Seluruh Bapak dan Ibu Dosen Jurusan Teknik Informatika Program Studi S1 Sistem Informasi yang telah mendidik, mengajarkan dan memberikan ilmu yang bermanfaat selama peneliti menempuh pendidikan di perkuliahan.
10. Operator Jurusan Teknik Informatikan yang telah membantu dalam pengurusan administrasi.
11. Teman – Teman Kelompok Studi Linux telah memberikan bantuan dan semangat.
12. Teman – Teman Genereasi Baru Indonesia yang telah memberikan bantuan dan semangat.
13. Teman – Teman lain yang tidak sempat saya sebutkan dalam membanti peneliti selama ini.

Peneliti menyadari bahwa masih banyak kekurangan dari laporan ini, baik dari materi maupun Teknik penyajiannya, mengingat kurangnya pengetahuan dan pengalaman peneliti. Oleh karena itu, kritik dan saran yang membangun sangat peneliti harapkan.

Gorontalo, Maret 2024

Peneliti

# **ABSTRAK**

**DZAKIYYAH AZAHRA MOODUTO.** Klasifikasi Hama dan Penyakit Tanaman Padi dengan Metode Convolutional Neural Network (dibimbing oleh Dr. Moh. Hidayat Koniyo, ST., M.Kom, dan Indhitya R. Padiku., S.Kom., M.Kom, MCE)

Padi merupakan tanaman budidaya yang memiliki peran krusial dalam peradaban manusia. Pertumbuhan padi sering kali dihadapi oleh hambatan dan serangan hama serta penyakit tanaman, terutama pada bagian daun. Kendala utama yang sering terjadi adalah keterlambatan dalam diagnosis dan respons manual, yang dapat mengakibatkan kerusakan yang parah dan sulit diatasi. Salah satu permasalahan umum adalah kurangnya pengetahuan petani dan kurangnya tindakan antisipatif. Oleh karena itu, tujuan dari penelitian ini adalah untuk merancang suatu sistem klasifikasi hama dan penyakit pada tanaman padi menggunakan teknologi olah citra digital dengan menerapkan metode Convolutional Neural Network (CNN). Penelitian ini bertujuan agar kerusakan pada tanaman padi akibat penyakit dapat diidentifikasi dengan cepat, mencegah penyebaran lebih lanjut, dan berpotensi meningkatkan produksi tanaman. Penggunaan dataset berupa gambar tanaman padi yang terinfeksi dan tidak terinfeksi menjadi dasar dalam pengembangan model dengan menggunakan library TensorFlow dan bahasa pemrograman Python. Dataset dibagi menjadi dua bagian, yaitu 80% data untuk pelatihan model dan 20% untuk pengujian. Evaluasi kinerja sistem dilakukan dengan mempertimbangkan beberapa parameter pengujian seperti *resize* gambar, jenis *optimizer*, *learning rate*, dan ukuran *batch*. Hasil pengujian menunjukkan bahwa kondisi optimal diperoleh dengan menggunakan parameter berikut: *resize* citra 150x150 piksel, *optimizer* Adam, *learning rate* 0,001, dan ukuran *batch* 32. Sistem ini berhasil mencapai akurasi sebesar 89% dan nilai loss sebesar 0,2972.

Kata kunci : CNN, klasifikasi, hama, penyakit, tanaman padi.

# **DAFTAR ISI**

[**KATA PENGANTAR** 2](#_Toc160500388)

[**ABSTRAK** 4](#_Toc160500389)

[**DAFTAR ISI** 5](#_Toc160500390)

[DAFTAR TABEL 7](#_Toc160500391)

[DAFTAR GAMBAR 8](#_Toc160500392)

[**BAB 1** 9](#_Toc160500393)

[**PENDAHULUAN** 9](#_Toc160500394)

[**1.1.** **Latar Belakang** 9](#_Toc160500395)

[**1.2. Rumusan Masalah** 14](#_Toc160500396)

[**1.3. Ruang Lingkup Penelitian** 14](#_Toc160500397)

[**1.4. Tujuan Penelitian** 14](#_Toc160500398)

[1.5. Manfaat Penelitian 14](#_Toc160500399)

[**BAB 2** 4](#_Toc160500400)

[**TINJAUAN PUSTAKA** 4](#_Toc160500401)

[**2.1. Tanaman Padi** 4](#_Toc160500402)

[**2.2. Hama dan Penyakit Tanaman Padi** 9](#_Toc160500403)

[**2.3. Citra Digital** 11](#_Toc160500404)

[**2.4. *Deep Learning*** 12](#_Toc160500405)

[**2.5. *Convolutional Neural Network*** 13](#_Toc160500406)

[**2.6. Tensorflow** 19](#_Toc160500407)

[**2.7. Keras** 20](#_Toc160500408)

[**2.8. Google Colab** 21](#_Toc160500409)

[**2.9. Penelitian Terkait** 21](#_Toc160500410)

[BAB 3 11](#_Toc160500411)

[METODOLOGI PENELITIAN 11](#_Toc160500412)

[**3.1.** **Analisis Kebutuhan Sistem** 11](#_Toc160500413)

[3.1.1. Spesifikasi Perangkat Keras (*Hardware)* 11](#_Toc160500414)

[3.1.2. Spesifikasi Perangkat Lunak (*Software*) 11](#_Toc160500415)

[**3.2.** **Tahapan Penelitian** 11](#_Toc160500416)

[3.2.1. Desain Sistem 12](#_Toc160500417)

[3.2.1.1. Tahap Preprocessing 14](#_Toc160500418)

[3.2.1.2. Tahap Training 14](#_Toc160500419)

[3.2.1.3. Tahap Testing 15](#_Toc160500420)

[**3.4.** **Data Penelitian** 15](#_Toc160500421)

[**3.5.** **Lokasi dan Waktu Penelitian** 17](#_Toc160500422)

[3.5.1. Lokasi Penelitian 17](#_Toc160500423)

[3.5.2. Waktu Penelitian 17](#_Toc160500424)

[BAB 4 27](#_Toc160500425)

[HASIL DAN PEMBAHASAN 27](#_Toc160500426)

[**3.3.** **Hasil Penelitian** 27](#_Toc160500427)

[**3.4.** **Pembahasan** 43](#_Toc160500428)

[3.4.1. Hasil Pengujian Terbaik 43](#_Toc160500429)

[3.4.2. Model *Deep Learning* Pengujian Akhir 45](#_Toc160500430)

[BAB 5 49](#_Toc160500431)

[PENUTUP 49](#_Toc160500432)

[**5.1. Kesimpulan** 49](#_Toc160500433)

[**5.2. Saran** 49](#_Toc160500434)

# **DAFTAR TABEL**

[Tabel 3. 1. Waktu Penelitian 17](#_Toc160000937)

[Tabel 4. 1. Dataset Penelitian 31](#_Toc160000943)

[Tabel 4. 2. Paramater Model 32](#_Toc160000944)

[Tabel 4. 3. Spesifikasi Google Colab 33](#_Toc160000945)

[Tabel 4. 4. Daftar Package 34](#_Toc160000946)

[Tabel 4. 5. Hasil Skenario 1 36](#_Toc160000947)

[Tabel 4. 6. Hasil Skenario 2 38](#_Toc160000948)

[Tabel 4. 7. Hasil Skenario 3 40](#_Toc160000949)

[Tabel 4. 8. Hasil Skenario 4 43](#_Toc160000950)

[Tabel 4. 9. Hasil Pengujian Terbaik 44](#_Toc160000951)

[Tabel 4. 10. Spesifikasi Arsitektur Model CNN 47](#_Toc160000952)

# **DAFTAR GAMBAR**

[Gambar 1. 1. Luas Hasil Panen Padi Indonesia 2021-2022 2](#_Toc160000853)

[Gambar 1. 2. Luas Panen Padi Indonesia berdasarkan Provinsi 3](#_Toc160000854)

[Gambar 2. 1. Citra Digital 12](#_Toc160000858)

[Gambar 2. 2. Convolutional Neural Network 14](#_Toc160000859)

[Gambar 2. 3. Convolutional Layer 16](#_Toc160000860)

[Gambar 2. 4. Pooling Layer 17](#_Toc160000861)

[Gambar 2. 5. Flatten Layer 18](#_Toc160000862)

[Gambar 2. 6. Fully Connected Layer 19](#_Toc160000863)

[Gambar 3. 1. Alur Penelitian 12](#_Toc160000864)

[Gambar 3. 2. Block Diagram Sistem 13](#_Toc160000865)

[Gambar 3. 3. Flowchart Sistem 13](#_Toc160000866)

[Gambar 3. 4. Alur Prerprocessing 14](#_Toc160000867)

[Gambar 4. 1. Visualisasi Accuracy dan Loss Skenario 1 36](#_Toc160000871)

[Gambar 4. 2. Visualisasi Accuracy dan Loss Skenario 2 38](#_Toc160000872)

[Gambar 4. 3. Visualisasi Accuracy dan Loss Skenario 3 40](#_Toc160000873)

[Gambar 4. 4. Visualisasi Accuracy dan Loss Skenario 4 42](#_Toc160000874)

[Gambar 4. 5. Visualisasi Accuracy dan Loss Hasil Akhir Pengujian 45](#_Toc160000875)

[Gambar 4. 6. Visualisasi Model CNN 46](#_Toc160000876)

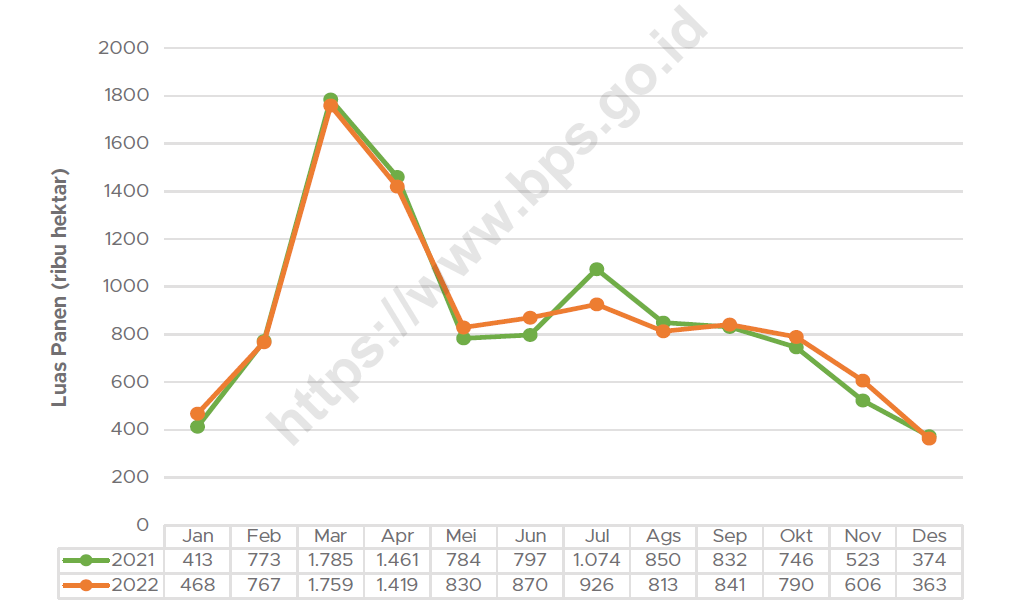
# **DAFTAR LAMPIRAN**

# **BAB 1**

# **PENDAHULUAN**

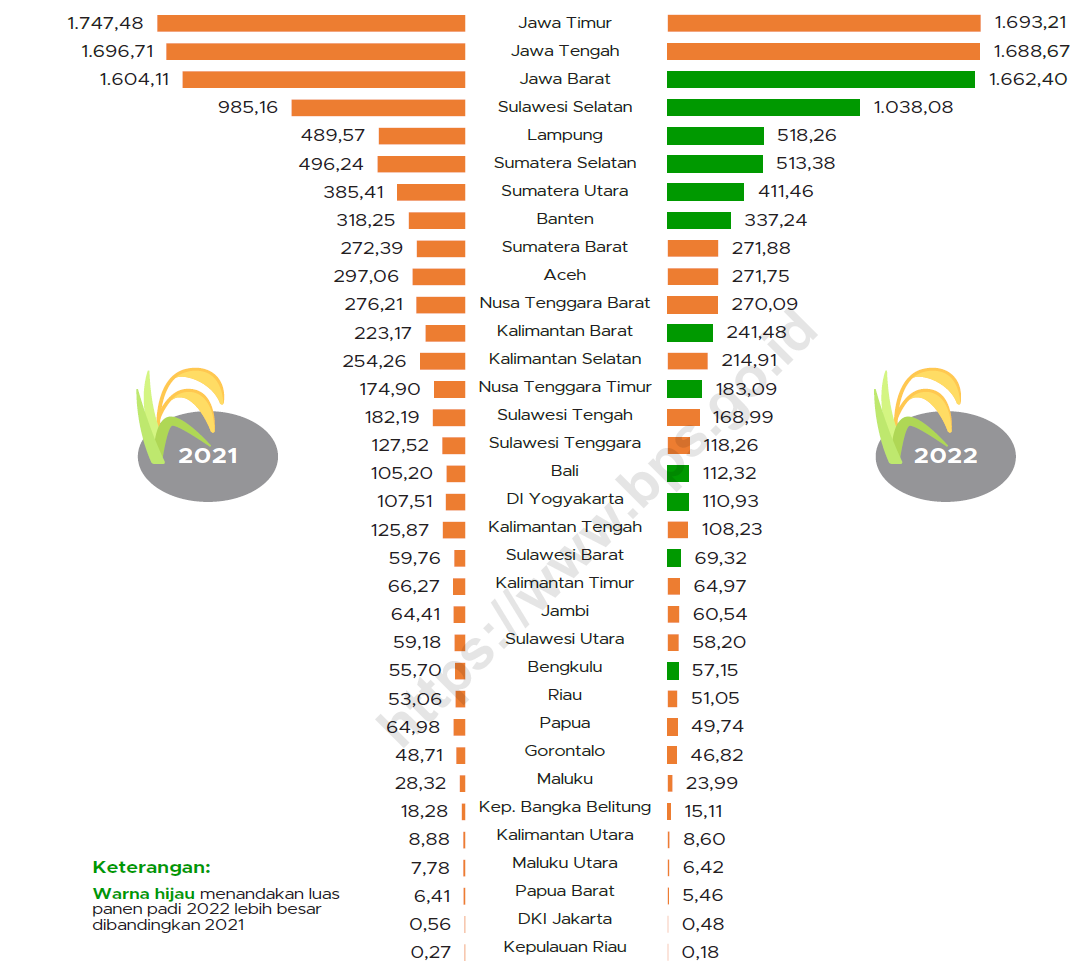
## **Latar Belakang**

Padi (*Oryza sativa*) yang menghasilkan beras merupakan salah satu makanan pokok di Asia, khususnya Indonesia maupun negara tropis lainnya. Produksi padi tiap tahunnya sangat dibutuhkan demi memenuhi kebutuhan pangan di negara Indonesia. Jumlah penduduk yang semakin meningkat menyebabkan tingkat konsumi dan permintaan pasar dari beras akan semakin meningkat. Oleh karena itu, produksi beras harus dipastikan berada dalam angka yang konstan dan tidak mengalami penurunan.



Gambar 1. 1. Luas Hasil Panen Padi Indonesia 2021-2022

Berdasarkan hasil survei Kerangka Sampel Area (KSA), pola panen padi di Indonesia pada 2022 secara umum hampir mirip dengan pola panen padi pada 2021. Jika dibandingkan dengan 2021, luas panen padi 2022 mengalami kenaikan sebesar 40,87 ribu hektar (0,39 persen). Namun, jika dilihat secara lebih detail berdasarkan provinsi, terdapat sebanyak 12 provinsi yang mengalami peningkatan luas panen padi dibandingkan 2021. Sementara itu, 22 provinsi lainnya mengalami penurunan luas panen padi pada 2022 dibandingkan tahun sebelumnya. Berdasarkan gambar di bawah, Provinsi Gorontalo mengalami penurunan dari Angka 48,71 ribu hektar ke 46,82 ribu hektar. Angka tersebut tidak didapatkan secara tiba-tiba tetapi terdapat faktor yang menyebabkan gagal panen di beberapa wilayah (Khasanah & Astuti, n.d.).



Gambar 1. 2. Luas Panen Padi Indonesia berdasarkan Provinsi

Faktor penyebab gagal panen dalam produktivitas tanaman padi sangatlah banyak, salah satunya adalah hama dan penyakit pada daun padi (Saputra dkk., 2021). Serangan dari organisme pengganggu tanaman (OPT) sering menyebabkan rusaknya tanaman padi khususnya di bagian daun (Anam Alidrus dkk., 2021). Selain itu, keterlambatan proses diagnosis dan tindaklanjut secara manual menyebabkan hama dan penyakit pada tanaman padi mencapai pada tahap sangat parah dan tidak tertolong. Hal tersebut terjadi karena minimnya pengetahuan petani serta tidak adanya antisipasi sejak dini, sehingga membutuhkan proses manual dan berakibat pada terjadinya gagal panen.

Hama dan penyakit pada tanaman padi bisa dilihat dari perubahan akar, batang, biji, daun, dan lain-lain. Namun tidak semua manusia dapat mengetahui tentang hama dan penyakit tersebut. Hama dan penyakit akibat serangan OPT tersebut terus berfluktuasi dari satu tempat ke tempat lainnya maupun dari waktu ke waktu. Direktorat Jenderal Tanaman Pangan mengklasifikasi OPT (Organisme Pengganggu Tanaman) utama ke dalam 7 jenis, di antaranya adalah penggerek batang padi putih (PBPP), ulat grayak, wereng batang coklat (wbc), tikus, hama putih palsu (HPP), hawar daun bakteri/kresek. dan *blast.* Tujuh jenis hama dan penyakit tanaman yang beberapa tahun terakhir berpengaruh besar terhadap terjadinya gagal panen di produksi padi Indonesia.

Dinas Pertanian Provinsi Gorontalo menghadapi masalah kurangnya literasi dan antisipasi dari petani terhadap hama dan penyakit tanaman padi dengan beberapa pendekatan, salah satunya adalah dengan menciptakan aplikasi Balintan Smart Mobile. Aplikasi ini digunakan untuk berbagai aktivitas dan kepentingan dalam pelayanan pertanian seperti pemantauan dan pelaporan aktivitas petugas lapang, informasi seputar pertanian, layanan pelaporan petani, dan sebagainya. Akan tetapi, dari berbagai fungsi dan manfaat dari aplikasi tersebut, Dinas Pertanian Provinsi Gorontalo membutuhkan sebuah sistem pendeteksi hama dan penyakit pada tanaman padi untuk petani.

Dewasa ini, sistem klasifikasi citra sering digunakan untuk menjadi solusi sebuah permasalahan. Klasifikasi citra pada umumnya menggunakan algoritma *Convolutional Neural Network* (CNN). CNN adalah sebuah metode jaringan syaraf yang sangat terkenal dan cukup diminati. CNN digunakan untuk mengolah data dengan struktur grid yaitu salah satunya berupa citra dua dimensi dan juga mampu memproses data dengan dimensi tinggi seperti video. CNN bekerja mirip dengan *neural network standard*, hanya saja perbedaannya yaitu menggunakan kernel dua dimensi atau dimensi tinggi di setiap unit pada lapisan CNN yang akan dikonvolusi. Untuk melakukan penggabungan fitur dari spasial dengan bentuk spasial yang serupa media *input* maka digunakan kernel yang ada di dalam CNN. Kemudian pada CNN untuk mengurangi jumlah variabel agar lebih mudah untuk dilakukan pembelajaran yaitu dengan menggunakan beberapa parameter. Berdasarkan dari Namanya yaitu *convolutional neural network* yang menunjukkan bahwa metode jaringan tersebut menggunakan operasi dari matematika yang disebut sebagai konvolusi. CNN juga dilatih untuk mempelajari fitur dari objek untuk dapat memprediksinya (Jinan & Hayadi, 2022).

Penelitian terkait mengenai klasifikasi citra penyakit padi sudah banyak dilakukan sebelumnya, seperti (Swathika dkk., 2021) dalam penelitiannya mengembangkan sebuah sistem identifikasi penyakit pada daun tanaman padi menggunakan CNN *based Deep Learning*. Dataset yang digunakan adalah 3081 gambar padi dengan penyakit dan padi dalam keadaan normal. Dataset tersebut dikumpulkan dari dataset public seperti Kaggle. Dari 3081 gambar, 2772 (90%) gambar dipakai untuk proses *training* dan 10% digunakan untuk *testing* model yang dikembangkan dalam 10 epochs. Setelah tahap *tuning* dan *optimizing*, akurasi yang didapatkan adalah sebesar 70%. Penelitian dari Kawcher Ahmed, Tasmia Rahman Shahidi, Syed Md. Irfanul Alam dan Sifat Momen adalah mendeteksi penyakit pada daun tanaman padi. Penelitian ini menggunakan teknik machine learning untuk mengembangkan model. Metode yang digunakan adalah *supervised classification algorithms* diantaranya *K-Nearest Neighbor* (KNN), *Decision Tree, Naïve Bayes,* dan *Logictic Regression*. Data yang digunakan adalah data yang dibuat secara manual dengan memisahkan daun dengan penyakit menjadi tiga kelas penyakit yang berbeda, yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Peneliti menggunakan teknik *10-fold cross validation* pada dataset yang ada. Akurasi yang didapatkan pada akhirnya bermacam-macam di antaranya *Logistic Regression* sebesar 75.463%, KKN (K=1) 98.8462%, *Decision Tree* 94.9074%dan *Naïve Bayes* 58.7963%.

Berdasarkan masalah klasifikasi hama dan tanaman padi , adanya penelitian untuk membuat sistem klasifikasi hama dan penyakit pada tanaman padi menggunakan *Convolutional Neural Network.* Sistem klasifikasi menggunakan *public dataset* dari Kaggle serta data primer dari Dinas Pertanian Provinsi Gorontalo. Jumlah klasifikasi hama dan penyakit sebanyak 5 kelas yaitu Penggerek Batang Padi Putih (PBPP), Hama Putih Palsu (HPP), Kepinding Tanah, Walang Sangit, Hawar Daun Bakteri/Kresek, dan *Blast*.

## **1.2. Rumusan Masalah**

Bagaimana klasifikasi hama dan penyakit pada tanaman padi menggunakan pengolahan citra digital dengan metode Convolutional Neural Network ?

## **1.3. Ruang Lingkup Penelitian**

1. Metode yang digunakan dalam merancang sistem adalah Convolutional Neural Network (CNN).
2. Klasifikasi digunakan dalam 5 kelas, yaitu normal, penggerek batang (*stem borer*), hama putih palsu (*leaffolder*), hawar daun bakteri/kresek (*bacterial leaf blight*) dan *blast.*
3. Dataset berasal dari *Kaggle.com*, *Public Dataset* dan Dinas Pertanian Provinsi Gorontalo.
4. Format file citra yang digunakan adalah \*.jpeg

## **1.4. Tujuan Penelitian**

Tujuan dari penelitian ini adalah merancang sistem untuk klasifikasi hama dan penyakit pada tanaman padi menggunakan olah citra digital dengan metode CNN.

## 1.5. Manfaat Penelitian

1. Membantu Dinas Pertanian Provinsi Gorontalo untuk mengedukasi petani dan orang awam dalam mendeteksi dini hama dan penyakit tanaman padi secara tepat serta akurat.
2. Melalui pengujian ini, peneliti dapat menentukan seberapa baik akurasi model dalam mengidentifikasi objek yang dikenali.
3. Menggunakan arsitektur terbaik untuk klasifikasi hama dan penyakit pada tanaman padi menggunakan metode *Convolutional Neural Network* (CNN).

# **BAB 2**

# **TINJAUAN PUSTAKA**

## **2.1. Tanaman Padi**

Padi merupakan salah satu tanaman budidaya terpenting dalam peradaban manusia. Masyarakat Indonesia sangat membutuhkan hasil tanaman padi sebagai pangan utama sehari-hari. Ada beberapa tanaman budidaya yang dimanfaatkan masyarakat Indonesia sebagai bahan pangan seperti sagu dan umbi-umbian(Ayunina, 2023).

Pertumbuhan padi tidak dapat dipisahkan dari hambatan dan serangan hama maupun penyakit tanaman padi. Berbagai cara dilakukan oleh pemerintah dan lembaga terkait untuk menurunkan faktor sukses panen dari padi di Indonesia. Akan tetapi, faktor hama dan penyakit masih terus mempengaruhi kehilangan hasil. Hama dan penyakit tersebut di antaranya adalah penggerek batang, kepinding tanah, keong sawah, penyakit blas, kresek, bercak coklat dan pucuk putih (Ayunina, 2023). Faktor abiotik juga berperan seperti curah hujan, irigasi, serta perubahan dan anomali iklim (Yulistiani dkk., 2018).

## **2.2. Hama dan Penyakit Tanaman Padi**

Hama serta penyakit masih menjadi kendala bagi petani karena di setiap musim tanam terjadi ledakan hama pada budidaya padi. Beberapa hama utama yang berpotensi merusak tanaman padi adalah penggerek batang, ulat grayak, tikus, kepinding tanah dan walang sangit. Hal serupa yang sering terjadi, penyakit merupakan salah satu kendala utama produksi berbagai tanaman pertanian. Banyak kerugian yang diakibatkan karena penyakit tanaman yang terlambat untuk didiagnosis dan sudah mencapai tahap yang parah dan menyebabkan gagal panen. Beberapa penyakit yang sangat berpotensi merusak tanaman padi adalah hawar daun, blas, dan wereng coklat (Dypanuryansyah, 2018).

* + 1. Penggerek Batang.

Penggerek batang termasuk hama paling penting pada tanaman padi yang sering menimbulkan kerusakan berat dan kehilangan hasil yang tinggi. Di lapangan, keberadaan hama ini ditandai oleh kehadiran ngengat (kupu-kupu), kematian tunas-tunas padi (sundep, *dead heart*), kematian malai (beluk, *white head*), dan ulat (*larva*) penggerek batang (Puslitbang Tanaman Pangan et al., 2007).

Hama ini dapat merusak tanaman pada semua fase tumbuh, baik pada saat pembibitan, fase anakan, maupun fase berbunga. Gejala tanaman yang terserang hama penggerek batang adalah tanaman muda, daun yang tampak merah kekuningan, pucuk tanaman layu mengering, jika dicabut mudah patah. Jika batang padi dibuka, terlihat Larva di dalam batang. Serangan ini disebut sundep. Pada tanaman generatif Larva memotong tangkai buah (pangkal malai) sehingga gabah menjadi hampa (Murad, Mappa, & Rudin, 2021).

* + 1. Hawar daun Bakteri (*bacterial leaf blight* – BLB).

Hawar daun bakteri (HDB) merupakan penyakit bakteri yang tersebar luas dan menurunkan hasil sampai 36%. Penyakit terjadi pada musim hujan atau musim kemarau yang basah, terutama pada lahan sawah yang selalu tergenang, dan dipupuk N tinggi (>= 250 kg urea/ha) (Puslitbang Tanaman Pangan et al., 2007).

Penyakit HDB menghasilkan dua gejala khas, yaitu kresek dan hawar. Kresek adalah gejala yang terjadi pada tanaman berumur <30 hari (pesemaian atau yang baru dipindah). Daun-daun berwarna hijau kelabu, melipat, dan menggulung. Dalam keadaan parah, seluruh daun menggulung, layu dan mati, mirip tanaman yang terserang penggerek batang atau terkena air panas (*lodoh*). Sementara itu, hawar merupakan gejala yang paling umum dijumpai pada pertanaman yang telah mencapai fase tumbuh anakan sampai fase pemasakan (Wijayanto, Kiswanto, & Manurung, 2013).

* + 1. Blas (Blast)

Semula penyakit blas dikenal sebagai salah satu kendala utama pada padi gogo, tetapi sejak akhir 1980-an, penyakit ini juga sudah terdapat pada padi sawah beririgasi. Penyakit yang mampu menurunkan hasil yang sangat besar ini disebabkan oleh jamur pathogen *Pyricularia grisae* (Puslitbang Tanaman Pangan et al., 2007).

Penyakit blas menimbulkan dua gejala khas, yaitu blas daun dan blas leher. Blas daun merupakan bercak coklat kehitaman, berbentuk belah ketupat, dengan pusat bercak berwarna putih. Sementara itu, blas leher berupa bercak coklat kehitaman pada pangkal leher yang dapat mengakibatkan leher malai tidak mampu menopang malai dan patah. Kemampuan patogen membentuk strain dengan cepat menyebabkan pengendalian penyakit ini sangat sulit (Murad et al., 2021).

* + 1. Hama Putih Palsu (*leaffolder)*.

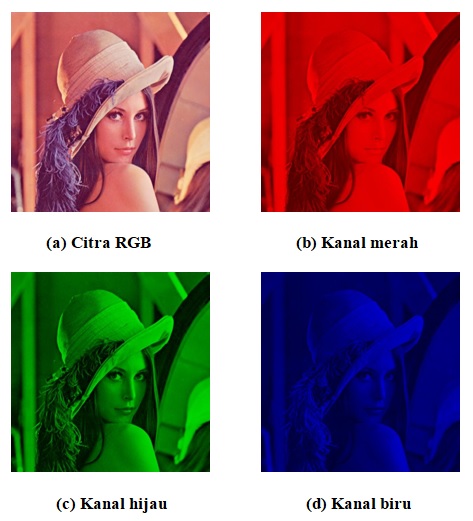
Hama Putih Palsu (HPP) jarang menjadi hama utama padi. Serangannya menjadi berarti bila kerusakan pada daun pada fase anakan maksimum dan fase pematangan mencapai >= 50%. Kerusakan akibat serangan larva HPP terlihat dengan adanya warna putih pada daun di pertanaman (Puslitbang Tanaman Pangan et al., 2007).

Larva memakan jaringan hijau daun dari dalam lipatan daun meninggalkan permukaan bawah daun yang berwarna putih. Siklus hidup hama ini 30-60 hari. Tanda pertama adanya infestasi HPP adalah kehadiran ngengat berwarna kuning coklat yang memiliki 3 buah pita hitam dengan garis lengkap atau terputus pada bagian sayap depan (Murad et al., 2021).

## **2.3. Citra Digital**

Sebuah citra digital dapat didefinisikan oleh sebuah matriks dua dimensi f(x,y) yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel atau elemen terkecil dari sebuah citra. Citra digital merupakan citra yang bisa diproses oleh komputer, dan citra yang dihasilkan oleh perangkat digital bisa secara langsung diproses oleh komputer. Citra analog tak dapat direpresentasikan oleh komputer dan menyebabkan tidak diprosesnya citra analog tersebut. Untuk dapat diproses oleh komputer citra analog harus dikonversikan ke dalam citra digital (Dewi, 2018).

Pada ruang warna RGB ini masing-masing piksel memiliki warna tertentu yaitu merah (*Red*), hijau(*Green*), dan biru(*Blue)*. Ruang warna RGB ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau, dan biru setiap pikselnya.



Gambar 2. 1. Citra Digital

Pada Gambar di atas, terdapat citra warna 8-bit. Lalu sebagai contoh, sebuah vector r = (x,y,z). Kemudian untuk warna, variabel-variabel tersebut diganti oleh komponen R (*Red*), G(*Green*), B(*Blue*). Setiap warna dasar memiliki nilai minimum nol (0) dan nilai maksimum nol (0) dan nilai maksimum sebesar 255 bit (Rizky, 2023).

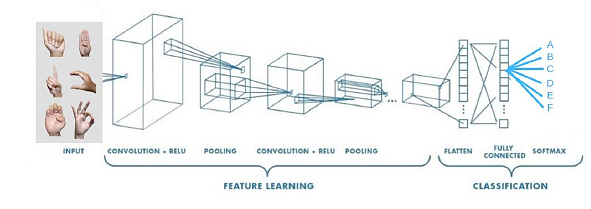
## **2.4. *Deep Learning***

*Deep learning* adalah bagian dari *Machine Learning* yang berusaha untuk memprediksi beberapa lapisan implementasi menjadi *output* gambar. Kinerja saraf dalam otak manusia menginspirasi sistematika kerja dari *Deep learning*. Teknik tingkat tinggi dari *Deep learning* memungkinkan sistem mengenali dan mengklasifikasikan gambar, animasi, serta suara(Rao Moparthi dkk., 2021). *Deep learning* biasanya digunakan untuk mengatasi masalah mulai dari kumpulan data berdimensi rendah hingga tinggi. Secara teknis *Deep learning* dapat diartikan teknik dalam *neural network* atau jaringan saraf tiruan (JST) dengan percepatan proses pembelajaran JST tersebut menggunakan banyak lapisan jaringan. Dengan menggunakan *Deep learning*, waktu yang dibutuhkan untuk training akan semakin sedikit karena masalah hilangnya gradien pada propagasi balik akan semakin rendah. *Convolutional Neural Network* (CNN) adalah arsitektur pembelajaran mendalam yang paling umum digunakan dalam banyak metode pola.

## **2.5. *Convolutional Neural Network***

*Convolutional Neural Network* (CNN) merupakan variasi dari *Multilayer Perceptron* (MLP) yang digunakan untuk menganalisis data dua dimensi berupa gambar. Karena kedalaman jaringannya yang besar dan aplikasi yang luas untuk menggambarkan data, CNN diklasifikasikan sebagai *Deep Neural Network*. Konvolusi adalah matriks dengan fungsi memfilter citra. CNN adalah arsitektur yang dapat dilatih dalam beberapa tahap.

Input dan output dari setiap tahapan adalah beberapa array yang disebut peta fitur. Keluaran setiap tahapan berupa peta fitur hasil olahan dari semua lokasi masukan. Proses tersebut adalah proses *training*. Proses pembentukannya terdiri dari 3 lapisan yaitu, *convolutional layer*, *activation layer*, dan *pooling layer*. Lapisan konvolusi (*convolutional layer*) adalah tingkat pertama dalam arsitektur CNN. Tahap ini diakhiri dengan memanfaatkan kernel dalam ukuran tertentu. Jumlah kernel yang digunakan ditentukan oleh jumlah fitur yang dibuat. Fungsi aktivasi (*activation layer*) umumnya merupakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), mengikuti tahap kedua. Lapisan penyatuan (*pooling layer*) digunakan pada langkah ketiga dan prosedur ini diulang sampai peta fitur yang memadai diperoleh untuk melanjutkan ke jaringan saraf yang sepenuhnya terhubung. *Fully connected network* merupakan *output class*(Khoiruddin dkk., 2022).



Gambar 2. 2. Convolutional Neural Network

Berdasarkan Gambar 2.2.di atas, pada dasarnya terdapat dua tahap dalam arsitektur CNN, yaitu tahapan *feature learning* dan *classification*. Tahapan *Feature Learning* meliputi *Convolutional Layer*, *Rectified Linear Unit* (ReLU) *Activation Layer,* dan *Pooling Layer*. Sementara itu, pada tahap *classification* memiliki lapisan berupa *Flatten Layer*, *Fullly Connected Layer*, dan *Softmax*.

* + 1. Input Citra

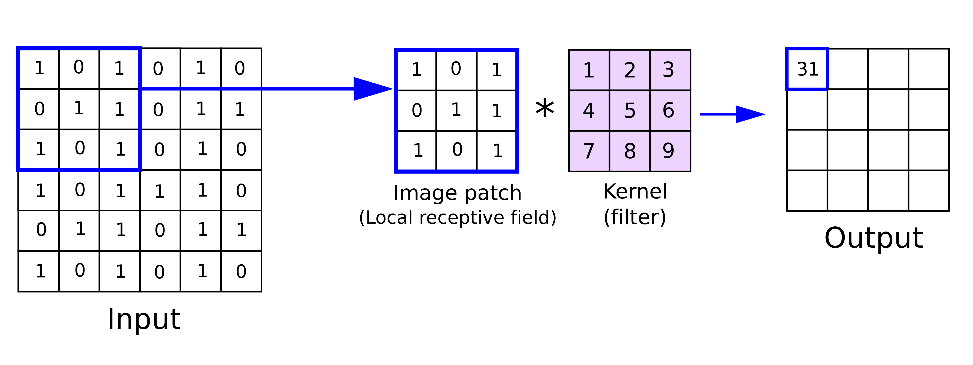
Pada lapisan ini, nilai piksel dari citra yang akan diolah dimasukkan untuk menjadi nilai *input* (masukkan). Setiap ukuran citra dengan channel warna yang berbeda akan mempengaruhi proses dari konvolusi setelahnya. Contohnya, apabila citra mempunyai ukuran 64x64 dengan tiga channel warna RGB (*Red, Green, Blue*) maka input akan berupa *pixel array* berukuran 64x64x3 (Panahatan, 2023). Input dari Citra akan digunakan untuk melakukan serangkaian operasi konvolusi, *pooling* dan lapisan lainnya untuk mengekstraksi fitur dari citra.

* + 1. *Convolutional Layer.*

*Convolutional layer* adalah lapisan pertama yang menerima input gambar langsung ke arsitektur. Operasi dalam lapisan Convolutional Layer identik dengan operasi konvolusi, yang terdiri dari operasi filter kombinasi linier pada wilayah lokal gambar. Filter mewakili bidang reseptif neuron yang terkait dengan wilayah lokal dari gambar input. Sebuah lapisan konvolusi melakukan operasi konvolusi pada output dari lapisan sebelumnya. Lapisan ini merupakan proses utama yang mendasari CNN. Tujuan dari citra yang berbelit-belit adalah untuk mengekstrak fitur dari citra masukan. Secara umum, operasi konvolusi dapat didefinisikan dengan menggunakan rumus:

………………….(1)

Pada fungsi s(t) menghasilkan output tunggal yaitu *feature map*, ketika argumen pertama berupa input yang merupakan x dan argument kedua yang merupakan w sebagai kernel atau filter. Jika melihat input sebagai citra dua dimensi, maka (t) bisa diasumsikan sebagai sebuah piksel(Sheila dkk., 2023).



Gambar 2. 3. Convolutional Layer

* + 1. Aktivasi Rectified Linear Units (ReLU).

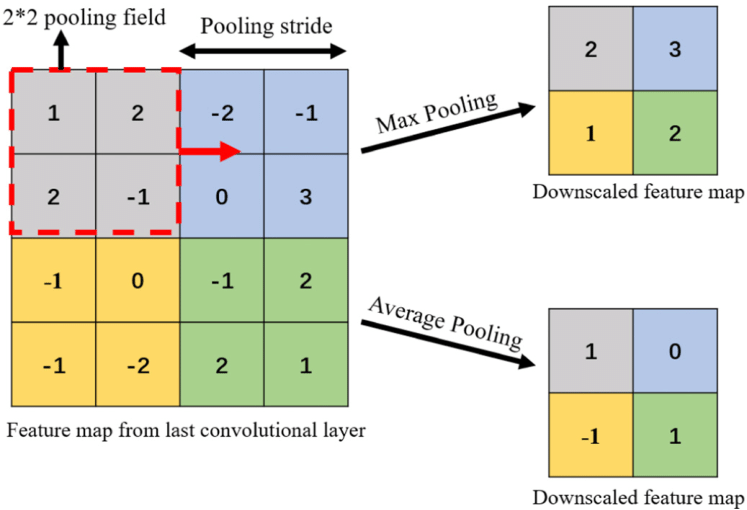
Aktivasi *Rectified Linear Units* (ReLU) adalah prosedur untuk memperkenalkan nonlinear dan meningkatkan representasi model. ReLU mengubah semua nilai piksel negative menjadi nol, memungkinkan *feature map* digambarkan lebih akurat dalam kaitannya dengan gambar. Persamaan dibawah ini merupakan aktivasi untuk ReLU :

……………………(2)

Dengan input sama dengan output ketika *x* positif and *output* sama dengan nol ketika x negatif (Rizky et al., 2023).

* + 1. *Pooling layer.*

*Pooling layer* mencakup filter dengan ukuran dan *stride* tertentu. Tiap pergeseran ditentukan oleh jumlah *stride* yang digeser pada seluruh bagian *feature map* atau *activation map*. *Pooling layer* yang umum digunakan merupakan *max pooling* dan *average pooling*. Fungsi dari *pooling layer* adalah untuk mereduksi *input* secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*(Anam Alidrus et al., 2021). . Gambar di bawah merupakan ilustrasi dari *pooling layer.*

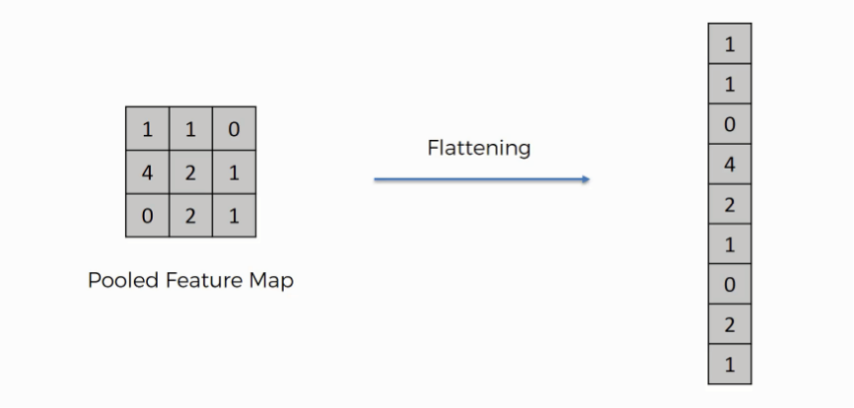


Gambar 2. 4. Pooling Layer

*Pooling layer* memiliki dua jenis *pooling* yang umum digunakan yaitu *Max Pooling* dan *Average Pooling*. *Max Pooling* bekerja dengan cara membagi citra (*output* dari lapisan sebelumnya) menjadi beberapa area kemudian setiap pergeseran filter akan diambil nilai *pixel* terbesar pada setiap area dan selanjutnya disusun menjadi matriks baru dengan ukuran yang lebih kecil. *Average Pooling* bekerja dengan cara yang sama namun nilai yang diambil adalah nilai rata-rata pada setiap area.

* + 1. *Flatten Layer*

Sebelum mencapai *flatten layer*, *output* dari lapisan-lapisan sebelumnya berbentuk multidimensi. Pada *flatten layer*, nilai matriks dari *feature map* yang awalnya berbentuk matriks multidimensi kemudian diubah menjadi vektor satu dimensi (KUTIPAN, JAMUR). Setelah itu, *output* dari *flatten layer* akan menjadi nilai *input* pada *fully connected layer* untuk mendapatkan hasil klasifikasi.



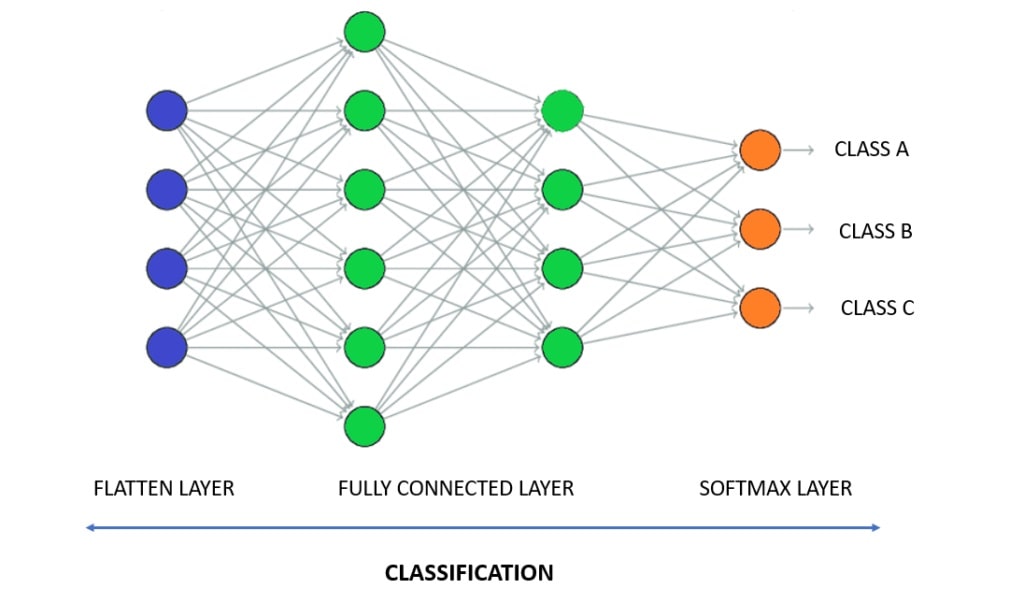
Gambar 2. 5. Flatten Layer

* + 1. *Fully-connected layer*

Semua neuron aktivasi dari lapisan sebelumnya terhubung ke neuron di lapisan berikutnya disebut dengan *fully-connected layer*. Setelah *convolutional layer, pooling layer, dan flatten layer, fully connected Layer* akan diterapkan. *Output* dari prosedur *Convolution Layer* dan *Pooling Layer* adalah *feature map*, yang kemudian diratakan melalui *flatten layer* dan dimasukkan ke dalam *fully connected layer*. Lapisan Linear termasuk dalam lapisan ini untuk menentukan nilai setiap kelas. Lapisan ini juga memiliki fungsi aktivasi, yaitu *LogSoftmax*, yang digunakan untuk menentukan probabilitas setiap kelas, dengan kemungkinan kerugian *negative log-likelihood* yang digunakan dalam perhitungan *loss*. Persamaan *softmax* dapat dilihat pada persamaan berikut.

……………………………..(3)

Dengan 𝑓𝑗 adalah probabilitas setiap kelas, 𝑧𝑗 adalah nilai vector ke-𝑗, dan 𝑘 adalah adalah jumlah kelas (Anam Alidrus et al., 2021).



Gambar 2. 6. Fully Connected Layer

## **2.6. Tensorflow**

Tensorflow adalah *library* perangkat lunak yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan pembelajaran mesin dan penelitian jaringan syaraf dalam. Tensorflow memiliki perkakas, pustaka, dan sumber daya komunitas yang komprehensif serta fleksibel yang dapat memudahkan jejaring serta peneliti dalam mengembangkan *machine learning* serta tim pengembang dapat dengan mudah membangun dan menerapkan aplikasi yang didukung algoritma *machine learning*. Tensorflow menggabungkan aljabar komputasi dan teknik pengoptimalan kompilasi, mempermudah penghitungan dengan mempersingkat waktu dalam melakukan perhitungan (Dewi, 2018).

Berikut adalah fitur utama dari Tensorflow.

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan *array multidimension* (tensors)
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin.
3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. Tensorflow bisa menulis kode yang sama dan menjalankannya dengan baik di CPU, GPU, atau pun TPU.
4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

Tensorflow menyediakan *Object Detection API* untuk mempermudah pengembangan aplikasi *deep learning* untuk membuat perangkat lunak pendeteksian objek. TensorFlow *Object Detection API* adalah *open source framework* yang dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek. Sistem tersebut sudah banyak diterapkan pada berbagai produk Google seperti pencarian gambar, deteksi wajah dan plat nomor kendaraan pada Google Streetview, Google Assistant, dan lainnya (Sheila et al., 2023).

## **2.7. Keras**

Keras merupakan salah satu *package* yang digunakan untuk menyelesaikan permasalahan mengenai *neural network*. Keras dikembangkan dan fokusnya adalah mempercepatn eksperimen pada proses konvolusi dan *recurrent* pada *neural networks*, maupun kombinasi antar keduanya. Pembuatan model jaringan syaraf menggunakan Keras tidak perlu menuliskan kode untuk mengekspresikan perhitungan matematisnya satu per satu. Hal ini dikarenakan Keras sudah menyediakan beberapa model dasar untuk CNN dan dioptimasi untuk mempermudah penelitian tentang *Deep Learning*. Proses komputasi menggunakan Keras berjalan dengan lancar dan baik menggunakan GPU maupun CPU (Shafira, 2018).

## **2.8. Google Colab**

Google telah mengembangkan Google Colaboratory, yaitu suatu kerangka kerja *online* yang memungkinkan pengguna untuk menulis dan menjalankan kode *Deep Learning* dan *Machine Learning*. Kerangka kerja ini menyediakan berbagai versi python dan lingkungan *runtime* yang berbeda. Selain itu, pengguna dapat mengunduh kumpulan data yang lebih besar langsung dari server ke Google Drive dengan kecepatan tinggi. Google Colab juga menyediakan akses ke sumber daya komputasi dan memori yang cepat, yang memungkinkan pengguna untuk melakukan komputasi yang intensif dan menyimpan model pembelajaran yang akan digunakan untuk prediksi (Kanani & Padole, 2019).

## **2.9. Penelitian Terkait**

2.9.1. Hasil Penelitian Rizal Amegia Saputra, dkk (2021)

“Penerapan Algoritma Convolutional Neural Network dan Arsitektur MobileNet Pada Aplikasi Deteksi Penyakit Daun Padi” oleh Rizal Amegia Saputra, Sri Wasyianti, Adi Supriyatna, Dede Firmansyah Saefudin dari Universitas Bina Sarana Informatika dalam Jurnal Swabumi. Penelitian ini bertujuan untuk melakukan ekstrak fitur penyakit daun padi dengan algoritma Neural Network, penelitian ini menunjukkan bahwa algoritma Neural Network dapat digunakan dalam klasifikasi penyakit daun padi melalui aplikasi berbasis Android.

Dalam penelitian ini, peneliti mengembangkan sebuah model klasifikasi penyakit daun padi dengan algoritma CNN dan arsitektur MobileNet dengan *feature extraction* dan *feature map*. Peneliti menggunakan dataset yang diambil dari UCI repository dengan objek data terdiri dari tiga jenis penyakit yaitu *bacterial leaf blight, brown spot,* serta *leaf smut*. Proses *training* dan *tuning* dari model yang dikembangkan yaitu sebanyak 100 epochs.

Hasil dari penelitian ini menunjukkan aplikasi yang dikembangkan mampu mengklasifikasi gambar jenis penyakit daun padi menggunakan algoritma CNN dengan arsitektur MobileNetV1 dan *Feature Extraction* memiliki akurasi yang tinggi yaitu sebesar 92% namun dengan jumlah data cukup kecil dan penggunaan *epochs* sebanyak 100, mengakibatkan nilai validasi kesalahan lebih tinggi dibandingkan nilai kesalahan pada proses pelatihan, sehingga akan terjadi *overfitting*.

Adapun perbedaannya terdapat pada objek klasifikasi. Penelitian ini terfokus pada tiga penyakit daun padi yaitu *bacterial leaf blight, brown spot,* serta *leaf smut.* Selain itu, peneliti menggunakan arsitektur model MobileNetV1 dan mengambil data dari *public dataset*.

2.9.2. Hasil Penelitian Annida Purnamawati, dkk (2020)

“Deteksi Penyakit Daun Padi pada Tanaman Padi menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN” oleh Annida Purnamawati, Wawan Nugroho, Destiana Putri, dan Wahyutama Fitri Hidayat dari STMIK Nusa Mandiri dan Universitas Bina Sarana Informatika, Jurusan Ilmu Komputer dalam Jurnal Nasional Informatika dan Teknologi Jaringan. Penelitian ini bertujuan untuk melakukan *Machine learning classifier*. Untuk dapat melihat klasifikasi dari jenis penyakit maka peneliti mencari akurasi dengan menggunakan perbandingan berbagai macam metode algoritma yaitu dengan Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN.

Dalam penelitian ini, peneliti mengusulkan perbandingan metode klasifikasi dalam dataset Rice Leaf Diseases yang terbagi atas tiga kelas yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Peneliti menggunakan analisis dengan data kualitatif dimana merupakan suatu Analisa yang digunakan apabila angka-angka dan perhitungan menggunakan rumus yang ada hubungannya dengan penelitian.

Hasil dari penelitian ini menunjukkan bahwa pada pengembangan *machine learning classifier* dengan metode algoritma Random Forest, Decision Tree, dan Naïve Bayes terjadi *overfitting*. Sementara itu, untuk pengembangan dengan metode algoritma SVM terjadi *underfitting*. *Good models* didapatkan dengan menggunakan algoritma KNN dengan nilai akurasi sebesar 87%.

Adapun perbedaannya terdapat pada metode penelitian. Penelitian ini menggunakan lima metode sekaligus untuk dilakukan perbandingan yaitu Random Forest, Decision Tree, Naïve Bayes, SVM serta KNN.

2.9.3. Hasil Penelitian Kawcher Ahmed, dkk. (2021)

Penelitian dari Kawcher Ahmed, Tasmia Rahman Shahidi, Syed Md. Irfanul Alam dan Sifat Momen adalah mendeteksi penyakit pada daun tanaman padi. Penelitian ini menggunakan teknik machine learning untuk mengembangkan model. Metode yang digunakan adalah *supervised classification algorithms* diantaranya *K-Nearest Neighbor* (KNN), *Decision Tree, Naïve Bayes,* dan *Logictic Regression*. Data yang digunakan adalah data yang dibuat secara manual dengan memisahkan daun dengan penyakit menjadi tiga kelas penyakit yang berbeda, yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Peneliti menggunakan teknik *10-fold cross validation* pada dataset yang ada. Akurasi yang didapatkan pada akhirnya bermacam-macam di antaranya *Logistic Regression* sebesar 75.463%, KKN (K=1) 98.8462%, *Decision Tree* 94.9074% dan *Naïve Bayes* mendapatkan hasil akhir sebesar 58.7963%.

# **BAB 3**

# **METODOLOGI PENELITIAN**

## **Analisis Kebutuhan Sistem**

### Spesifikasi Perangkat Keras (*Hardware)*

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dan kamera yang ada pada *smartphone*, dengan spesifikasi sebagai berikut :

1. Spesifikasi Laptop
   1. Tipe Laptop : DELL Inspiron 5402
   2. Prosessor : 11th Gen Intel(R) Core(TM) i5-1135G7
   3. RAM : 8,00 GB
   4. Tipe Sistem : 64-bit
2. Spesifikasi Kamera
   1. Iphone 11

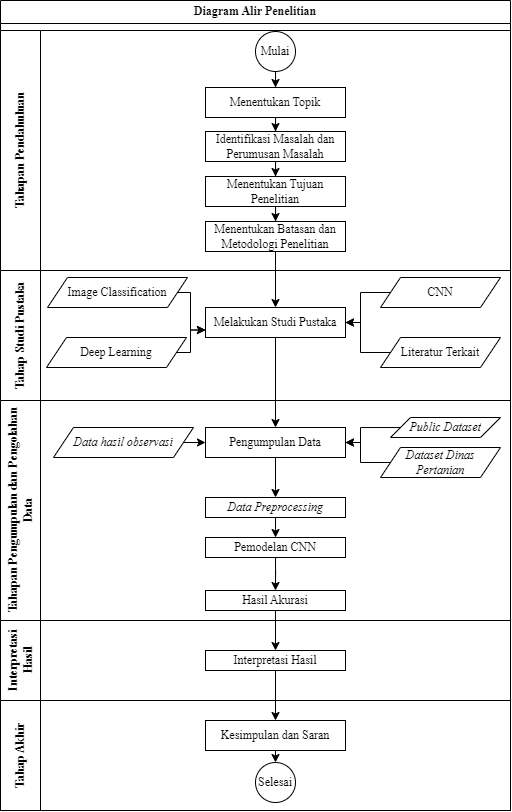
### Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini dengan spesifikasi sebagai berikut.

1. Google Colaboratory
2. Python

## **Tahapan Penelitian**

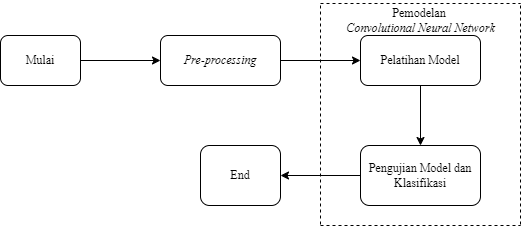
Langkah atau tahapan penelitian yang dilakukan dalam penelitian ini digambarkan melalui Gambar 3.1.berikut.



Gambar 3. 1. Alur Penelitian

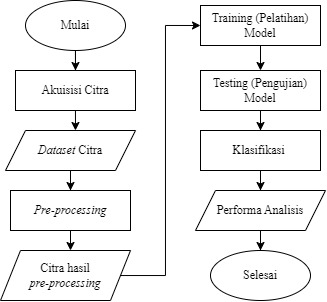
### 3.2.1. Desain Sistem

Pada penelitian ini dibuat sebuah rancangan desain sistem yang digunakan untuk klasifikasi hama dan penyakit tanaman padi menggunakan metode *Convolutional Neural Network.* Gambar 3.2.menjelaskan metode *Convolutional Neural Network* menggunakan blok diagram sistem.



Gambar 3. 2. Block Diagram Sistem

Berdasarkan Gambar 3.2.tahapan awal input citra akan berupa 5 jenis kelas hama dan penyakit yang berbeda yaitu normal, penggerek batang, hama putih palsu, hawar daun dan blas. Selanjutnya sistem yang dirancang akan melalui tahapan *preprocessing* yang diawali dengan akuisisi citra. Akuisisi citra digunakan untuk memperoleh citra dari sumber yang kemudian akan digunakan pada tahap *preprocessing* untuk dijadikan sebagai *input* pada alur selanjutnya.

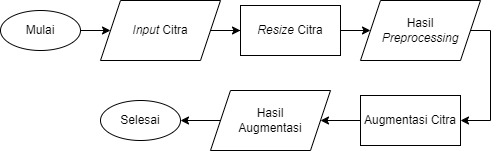


Gambar 3. 3. Flowchart Sistem

Pada *flowchart* di Gambar 3.3.menjelaskan alur dari sistem yang lebih mendetail dengan metode *convolutional neural network*. Setelah melewati tahap klasifikasi hasil tersebut akan dianalisis menggunakan parameter performa yang akan diukur. Hasil yang akan didapat nantinya akan berbentuk nilai-nilai dari masing-masing parameter pengujian.

### Tahap Preprocessing

Pada *preprocessing* ini merupakan tahap pertama dari pengolahan citra sebelum citra dikirim ke tahap pelatihan. *Preprocessing* digunakan untuk meningkatkan kualitas data yang dikumpulkan. Tahapan pada *preprocessing* dapat dilihat pada Gambar 3.4.



Gambar 3. 4. Alur Prerprocessing

Berdasarkan Gambar 3.4., pertama akan dilakukan *input* data yang sudah diakuisisi. Kemudian data tersebut akan melewati proses *resize* yang bertujuan untuk mengubah ukuran citra menjadi ukuran yang diinginkan. Setelah dilakukan *resize*, data akan diaugmentasi agar citra yang akan dilatih dalam model dapat lebih beragam dan banyak.

### Tahap Training

Pada tahapan ini, sistem akan melakukan pengenalan terhadap citra hama dan penyakit tanaman padi untuk diklasifikasikan sesuai dengan kelasnya. Proses klasifikasi menggunakan TensorFlow melalui dua tahap yaitu *training* dan *validation*. Tahap *training* dilakukan untuk mencari parameter terbaik pada citra yang kemudian dijadikan acuan untuk proses klasifikasi.

Pada tahapan ini akan dilakukan pengujian skenario terhadap parameter-paramater yang dapat mempengaruhi hasil simulasi sistem. Pada penelitian ini terdapat 4 (empat) variasi parameter yang digunakan, antara lain :

1. Paramater *resize* citra dengan ukuran yang berbeda 64x64 *pixel*, 128x128 *pixel*, dan 150x150 *pixel*.
2. Paramater jenis *optimizer* yang terdiri dari Adam, Nadam dan RMSprop.
3. Parameter *learning rate* dengan nilai yang bervariasi dari 0,1; 0,001; 0,0001.
4. Parameter *batch size* dengan ukuran 32, 64, 128.

### Tahap Testing

Pada tahap ini akan dilakukan pengujian oleh model sistem yang telah dibuat dan dilatih sebelumnya. Performasi sistem akan dinilai dari parameter nilai akurasi, *precision*, *recall*, *fl-score*, dan *loss* dari model-model yang telah dihasilkan oleh data pengujian. Peneliti akan melakukan tiga tahapan pengujian performansi sistem terhadap jumlah *hidden layer*, *optimizer*, dan *learning rate*.

## **Data Penelitian**

* + 1. Sumber Data

Sumber data dari penelitian ini diambil dari *Public Dataset* dan Dinas Pertanian Provinsi Gorontalo.

* + 1. Jenis Data
       1. Data Primer

Data primer dalam penelitian ini diperoleh dari hasil koordinasi dan komunikasi dengan Dinas Pertanian Provinsi Gorontalo.

* + - 1. Data Sekunder

Data sekunder dalam penelitian ini didapatkan dari penelitian-peneltian ini yang memiliki topik permasalahan yang sama dengan penelitian ini. Penelitian tersebut menjadi literatur sekaligus bahan bagi peneliti untuk mempelajari proses pembangunan sistem model machine learning untuk klasifikasi hama dan penyakit tanaman padi.

* + 1. Teknik Pengumpulan Data

Adapun teknik pengumpulan data yaitu sebagai berikut:

* + - 1. Wawancara

Pada tahap ini peneliti melakukan wawancara dengan pihak terkait yaitu Pengendali Organisme Pengganggu Tanaman (POPT), tenaga pertanian dari Dinas Pertanian Provinsi Gorontalo, dan petani padi.

* + - 1. Studi Pustaka

Peneliti mempelajari dan mengaitkan literatur yang berhubungan dengan permasalahan yang dihadapi yaitu literatur tentang mendeteksi dan klasifikasi hama dan penyakit tanaman padi. Langkah ini dilakukan sebagai landasan teori serta pedoman dalam menganalisa masalah.

* + - 1. Observasi

Teknik ini dilakukan dengan mengumpulkan data dan informasi mengenai hama dan penyakit tanaman padi. Observasi dilakukan dengan melakukan peninjauan langsung pada lahan yang terdampak OPT padi.

* + - 1. Dokumentasi

Peneliti mengambil sejumlah besar fakta dan data tersimpan dalam bentuk dokumentasi, yaitu gambar. Tujuan dari pengambilan gambar adalah untuk mendapatkan dataset yang kemudian akan dilatih dalam pengembangan model.

## **Lokasi dan Waktu Penelitian**

### Lokasi Penelitian

Penelitian ini berlokasi di Provinsi Gorontalo, khususnya di lahan tanaman padi dibawah pembinaan Dinas Provinsi Gorontalo.

### Waktu Penelitian

Waktu Penelitian ini dilakukan selama 6 bulan, terhitung dari bulan November 2023 sampai dengan Maret 2024.

Tabel 3. 1. Waktu Penelitian

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Tahap Penelitian | Tahun 2023-2024 | | | | | |
| Bulan | | | | | |
| Nov | Okt | Des | Jan | Feb | Mar |
| 1 | Observasi, Pengumpulan Data Tahap Awal dan Studi Literatur |  |  |  |  |  |  |
| 2 | Pengumpulan Dataset Citra |  |  |  |  |  |  |
| 3 | Pengembangan Sistem : Tahap *Preprocessing* dan Tahap *Training* |  |  |  |  |  |  |
| 4 | Tahap *Testing* |  |  |  |  |  |  |
| 5 | Pembuatan Laporan |  |  |  |  |  |  |

# **BAB 4**

# **HASIL DAN PEMBAHASAN**

## **Hasil Penelitian**

Bab ini berisi tentang hasil data dari proses penelitian klasifikasi hama dan penyakit tanaman padi dengan menggunakan metode *convolutional neural network* dengan arsitektur

* + 1. Pengumpulan Data

Data yang digunakan dalam membangun model CNN merupakan data gambar tanaman padi mulai dari batang dan daun. Data ini diperoleh dari 4 dataset yaitu *Paddy Doctor Dataset*, *Public Dataset* dari RoboFlow, Sistem Informasi Data OPT & DPT Direktorat Perlindungan Tanaman Pangan, dan dataset hasil observasi di lapangan. Mekanisme pengumpulan Dataset pertama yaitu mengunduh dataset dari situs Kaggle. Dataset yang kedua diunduh dari situs Roboflow. Sementara itu dataset yang ketiga di-*scraping* dari situs atau sistem informasi resmi Direktorat Perlindungan Tanaman Pangan melalui akun Dinas Pertanian Provinsi Gorontalo. Data piksel dari kumpulan gambar tersebut akan digunakan sebagai *input* pada model CNN. Ketiga dataset yang telah terkumpul tersebut digabungkan menjadi sebuah dataset baru. Terdapat beberapa masalah yang ditemukan saat proses penggabungan ini dilakukan seperti format data gambar yang berbeda, gambar yang terduplikasi, data gambar yang rusak dan tidak bisa dibaca, serta data gambar yang belum dilabeli.

Pada dataset *Paddy Doctor Dataset*, terdapat sekitar 13.900 gambar, yang tentunya tidak hanya terdiri dari gambar beberapa kelas hama dan penyakit tanaman padi. Data pada dataset ini memiliki format JPG. Dataset yang diperlukan adalah 5 kelas yaitu normal*,* penggerek batang, hama putih palsu, hawar daun dan blas. Oleh karena itu, data yang diambil dari dataset ini adalah data gambar tanaman padi normal, hama putih palsu, dan blas. Sementara itu, untuk dataset yang diunduh dari situs Roboflow, terdapat sekitar 26.661 gambar yang terdiri dari beberapa label/kelas. Data pada dataset ini memiliki format JPG. Kemudian data yang diambil dari dataset ini adalah data dengan kelas hama putih palsu. Dataset yang diambil dari Sistem Informasi Data OPT & DPT ada sekitar 1.388 data gambar dengan kelas yang sudah sesuai dengan kriteria. Format data gambar dalam dataset ini adalah JPG dan PNG. Akan tetapi, setelah difilter data PNG tergolong sangat sedikit dan diputuskan untuk dikeluarkan dari dataset. Sumber data terakhir yaitu dengan pengumpulan secara langsung di daerah terserang hama/penyakit. Data yang terkumpul yaitu sekitar 500 data gambar.

Seluruh gambar dari keempat dataset tersebut terdiri dari 3 (tiga) kanal warna (*color channel*) yaitu RGB atau *Red*, *Green*, *Blue* (Merah, Hijau, Biru). Di samping itu, gambar-gambar di seluruh dataset tersebut memiliki orientasi gambar yang beragam (vertikal, horizontal, diagonal) dengan berbagai ukuran gambar. Keseluruhan dataset tersebut difilter menjadi 5 kelas yang menjadi fokus utama yaitu normal, penggerek batang (*stem borer*), hama putih palsu (*leaffolder*), hawar daun (*bacterial leaf blight*), dan blas (*blast*). Berikut merupakan contoh gambar dari masing-masing kelas tersebut.



Gambar 1. Normal



Gambar 2 Penggerek Batang (Stem Borer)



Gambar 3 Hama Putih Palsu (Leaffolder)



Gambar 4 Hawar Daun (Bacterial Leaf Blight)



Gambar 5 Blas (Blast)

* + 1. Tahap *Preprocessing*

Setelah dataset dikumpulkan dan terdiri dari beberapa kelas yang tidak diperlukan, maka data difilter dan diambil yang sesuai dengan 5 kelas utama. Gambar yang diambil adalah data gambar normal, penggerek batang, hama putih palsu, hawar daun dan blas. Keempat dataset ini kemudian digabungkan berdasarkan kelasnya yang kemudian menghasilkan dataset gabungan baru. Dataset yang duplikat difilter dan dihapus, gambar yang *error* dan tidak sesuai dengan ekstensi dihapus. Adapun dataset gabungan baru memiliki proporsi sebagai berikut.

Tabel 4. 1. Dataset Penelitian

|  |  |
| --- | --- |
| Nama Kelas | Jumlah Gambar |
| Normal | 1.764 |
| Stem\_borer | 678 |
| Leaffolder | 500 |
| Bacterial\_leaf\_blight | 732 |
| Blas | 1.775 |

Pada tahap *preprocessing* ini dilakukan standarisasi dan normalisasi sebelum data digunakan sebagai masukan (*input*) pada model CNN. *Library* utama yang digunakan pada tahapan ini yaitu TensorFlow dan Keras. Pada penelitian ini, *preprocessing* yang dilakukan yaitu melakukan *resize* dan *rescale* pada setiap gambar agar memiliki ukuran yang sama. *Resize* atau pengubahan ukuran dilakukan saat menyiapkan *input pipeline* sebelum pembangunan model sehingga gambar menjadi berukuran sesuai nilai yang ditetapkan. Sedangkan *rescale* atau penskalaan ulang dilakukan saat mneyusun arsitektur lapisan model (tergabung dalam arsitektur lapisan model).

Untuk meningkatkan diversitas pada data gambar yang digunakan untuk *training*, terdapa sebuah teknik yang digunakan yaitu augmentasi data (*data augmentation*). Teknik augmentasi data bekerja dengan cara menerapkan berbagai jenis transformasi pada data gambar. Teknik ini dapat meningkatkan kinerja dan hasil dari model *machine learning* yang dibangun. Augmentasi yang dilakukan pada dataset yaitu berupa *zoom, rotation*, *shear*, dan *horizontal flip*. Nilai dari setiap parameter tersebut adalah sebagai berikut

Tabel 4. 2. Paramater Model

|  |  |
| --- | --- |
| Parameter | Nilai |
| Zoom\_range | 0.2 |
| Rotation\_range | 30 |
| Shear\_range | 0.2 |
| Horizontal\_flip | True |

* + 1. Tahap *Training* dan *Testing*

Pada tahapan *training* serta *testing* atau pemodelan mencakup dua proses yaitu pembentukan model dan menentukan parameter yang terbaik dalam melatih model. Namun, sebelum dilakukan pembangunan model, dataset yang sudah digabungkan dibagi menjadi 2 (dua) bagian yaitu *training* dan *validation*. Untuk memudahkan pembagian ini, dataset gabungan diunggah ke dalam Google Drive dengan folder kelas yang berbeda. Hal ini dilakukan karena Google Drive dapat dipasangkan/dihubungkan dengan Google Colab yang akan memudahkan proses akses data untuk pengembangan model. Adapun versi Python yang digunakan pada Google Colab merupakan Python 3 dan jenis *runtime* yang digunakan bukan merupakan GPU (*Graphics Processing Unit*) atau TPU (*Tensor Processing Unit*). Google Colab yang digunakan adalah versi gratis dan bukan versi yang berbayar. Google Colab dijalankan pada aplikasi Google Chrome.

Berikut spesifikasi Google Colab yang digunakan.

Tabel 4. 3. Spesifikasi Google Colab

|  |  |
| --- | --- |
| Spesifikasi Google Colab | |
| Arsitektur Sistem | CPU |
| Sistem Operasi | Windows 11 |
| RAM | 12.7 GB |
| Kapasitas Penyimpanan | 107.7 GB |

Setelah *file* baru di Google Colab dibuat, dilakukan pemeriksaan apakah dataset yang diunggah ke Google Drive dapat dibaca atau tidak. Kemudian data tersebut dibagi menjadi dua bagian yaitu *train* dan *validation*. Pembagian ini menggunakan parameter dalam package TensorFlow Keras yaitu *validation split*. Dataset dibagi dengan proporsi 20% *validation* dan 80% *train*.

Model CNN dibangun menggunakan bantuan beberapa *library* dalam bahasa pemograman Python. Adapun *library* yang digunakan pada penelitian kali ini yaitu:

Tabel 4. 4. Daftar Package

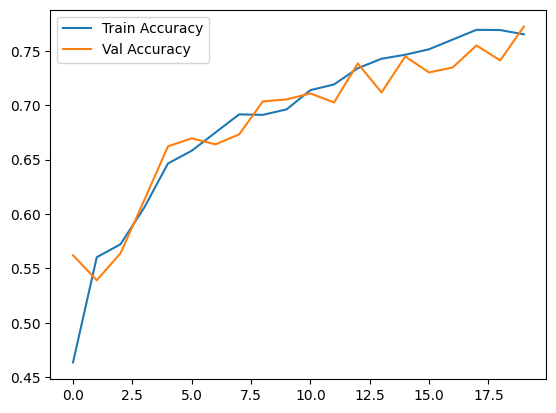
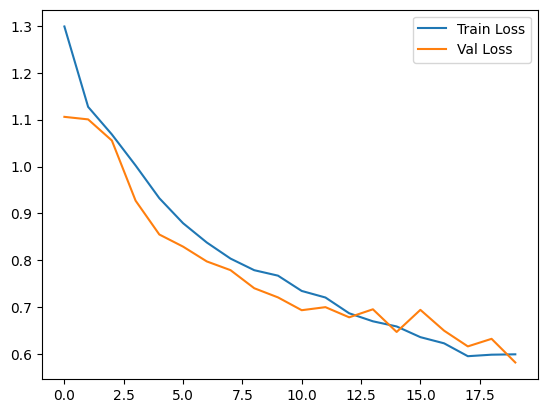
|  |  |
| --- | --- |
| Nama *Library* | Versi |
| Python | 3.10.12 |
| Matplotlib | 3.7.1 |
| Numpy | 1.25.2 |
| TensorFlow | 2.15.0 |
| Keras | 2.15.0 |
| Pathlib | 1.0.1 |

Setelah menyiapkan *library* dan dataset gambar, selanjutnya adalah pembagunan model dengan melakukan beberapa skenario di dalamnya untuk mendapatkan hasil yang paling optimal. Analisis untuk membangun model tersebut ditinjau dari beberapa parameter yaitu *resize*, jenis *optimizer*, nilai *learning rate*, dan *batch size*. Hasil pengujian dari masing-masing parameter divisualisasikan dalam bentuk tabel dan grafik *accuracy* dan *loss*. Kemudian analisis pada keberhasilan sistem diukur menggunakan parameter akurasi, presisi, *recall*, dan *F1-Score*. Dalam pengujian sistem ini memiliki 4 skenario. Berikut adalah hasil dari masing-masing skenario tersebut.

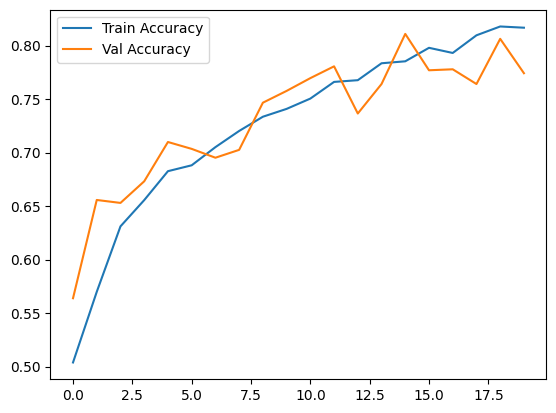
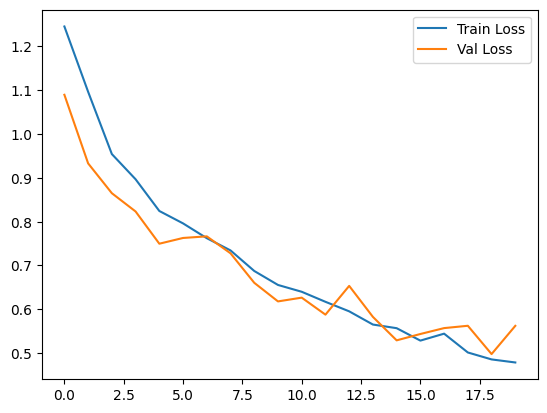
1. Pengujian Skenario 1 : *Resize* Citra

Pada skenario pertama dilakukan proses *resize* citra yang akan digunakan pada tahapan ini. Ada 3 ukuran citra yang akan digunakan yaitu 64x64 piksel, 128x128 piksel, dan 150x150 piksel. Dalam skenario ini parameter tetap yang digunakan adalah *optimizer* Adam dengan *epoch* 20 kemudian *learning rate* dengan nilai 0.0001 dan *batch size* dengan nilai 64. Hasil dari pengujian ini ditujukan untuk mengetahui kualitas terbaik pada sistem berdasarkan *image size*.

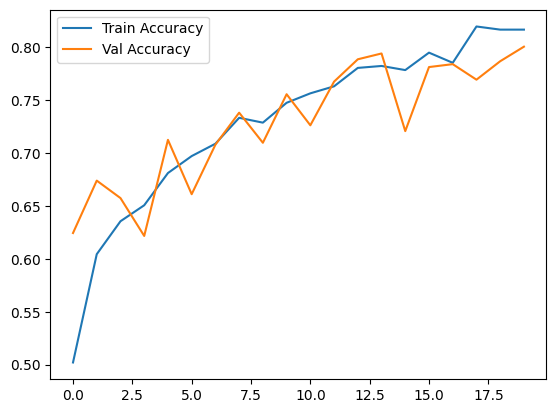
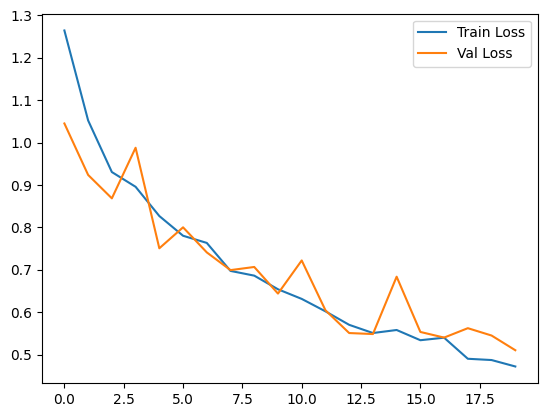
*Input Size 64x64*

*Input Size 128x128*

*Input Size 150x150*

Gambar 4. 1. Visualisasi Accuracy dan Loss Skenario 1

Tabel 4. 5. Hasil Skenario 1

|  |  |  |
| --- | --- | --- |
| *Resize* | Akurasi | Loss |
| 64x64 | 76.51% | 0.5998 |
| 128x128 | 81.66% | 0.4787 |
| 150x150 | 81.69% | 0.4722 |

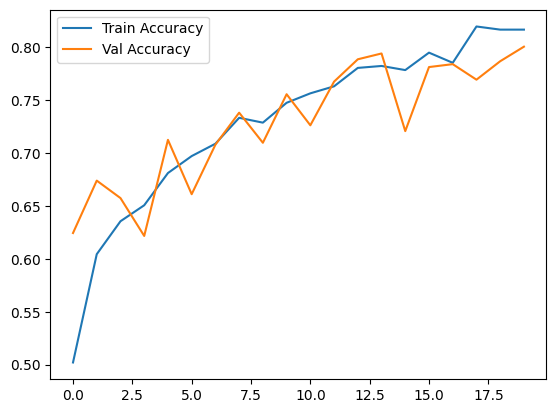
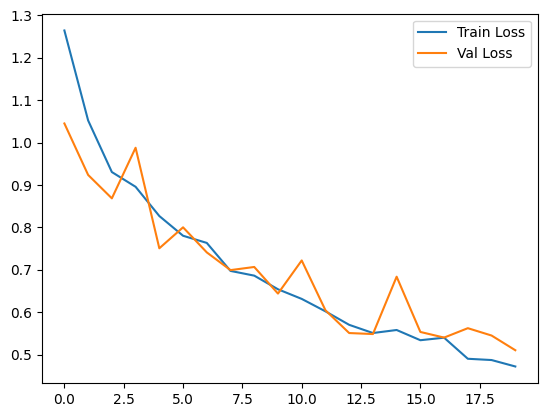
Dari Tabel 4.5.didapatkan hasil *resize* citra dengan akurasi terbaik pada sistem tersebut yaitu citra dengan ukuran 150x150piksel. Hal itu dinyatakan dengan nilai akurasi yang diperoleh sebesar 81.69% dengan nilai *loss* sebesar0.4722.

Selanjutnya pada Gambar 4.1.menunjukkan grafik yang telah diperoleh dari hasil *training* dan *validation* model dengan *input size* yang berbeda. Berdasarkan hasil pengujian tersebut, menunjukkan hasil stabil dengan akurasi yang paling baik adalah ukuran 150x150 *pixel* terutama di grafik *accuracy* yang mendapatkan angka yang lebih baik dari yang lain. Pada *input size* 128x128 cukup stabil, akan tetapi angka akurasi dengan parameter yang sama tidak lebih baik dari ukuran 150x150. Sementara itu, ukuran 64x64 memiliki angka akurasi paling rendah dengan parameter yang sama.

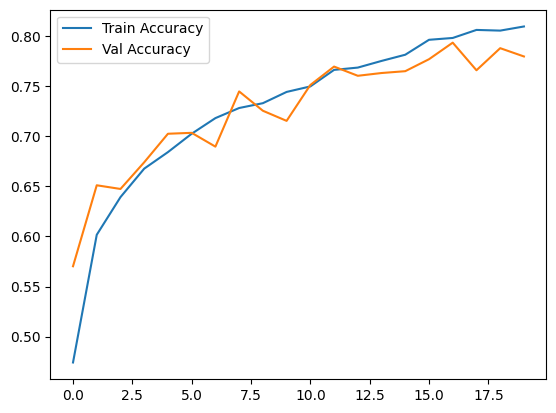
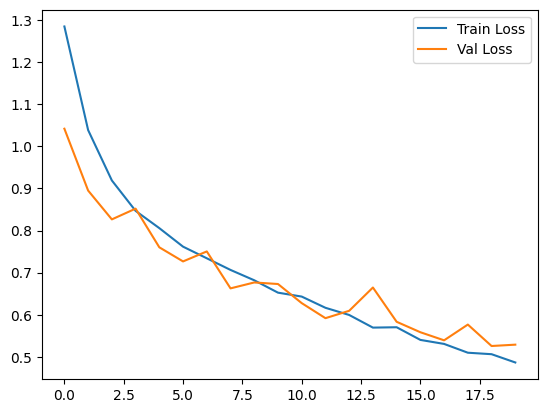
1. Pengujian Skenario 2 : Perubahan Jenis *Optimizer*

Pada skenario pertama telah dilakukan pengujian dengan melakukan perubahan pada input citra. Dalam skenario ini dilakukan tahapan perubahan jenis *optimizer* yaitu Adam, Nadam, dan RMSprop. Skenario ini parameter yang tetap adalah *image size*, *learning rate*, *epoch*, dan *batch size*. Seperti pengujian sebelumnya hasil pengujian ini juga dilihat melalui nilai akurasi dan *loss* untuk mengetahui kualitas dari masing-masing *optimizer* guna menentukan parameter tetap selanjutnya.

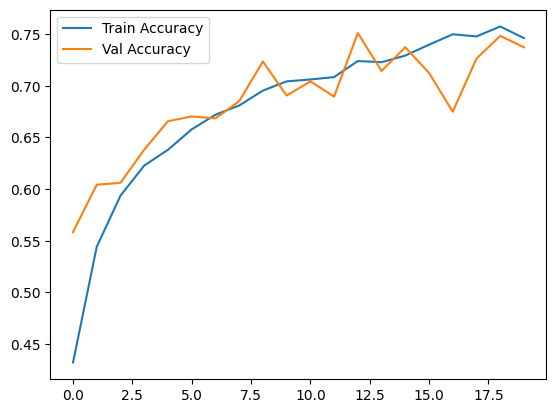
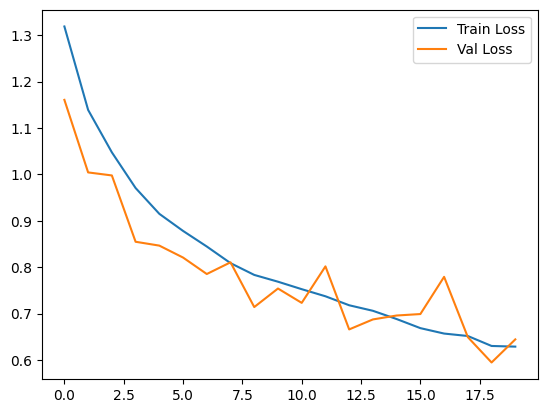
*Optimizer Adam*

*Optimizer Nadam*

*Optimizer RMSprop*

Gambar 4. 2. Visualisasi Accuracy dan Loss Skenario 2

Tabel 4. 6. Hasil Skenario 2

|  |  |  |
| --- | --- | --- |
| *Optimizer* | Akurasi | *Loss* |
| Adam | 81.69% | 0.4722 |
| Nadam | 80.95% | 0.4864 |
| RMSprop | 75.27% | 0.6513 |

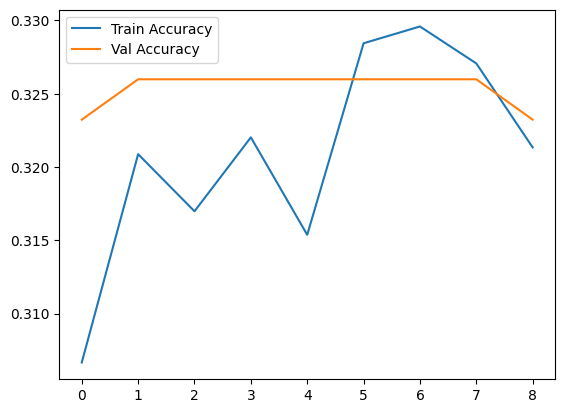
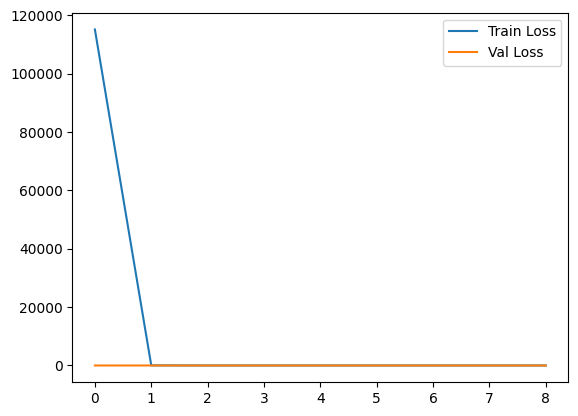
Pada skenario 1, parameter tetapnya adalah *optimizer* Adam. Setelah diuji, *optimizer* Adam masih lebih unggul dari kedua parameter *optimizer* lainnya. Hal ini dapat dilihat dari hasil *training* model berdasarkan Tabel 4.6. di atas. Angka yang ditunjukkan dari *optimizer* Adam unggul dari Nadam dan RMSprop. *Optimizer* RMSprop memiliki akurasi paling rendah dengan nilai *loss* paling tinggi.

Berdasarkan Gambar 4.2, ditunjukkan bahwa grafik dari keseluruhan pengujian *optimizer* tidak terindikasi *overfitting* atau pun *underfitting*. Ketiga *optimizer* cukup stabil dengan beragam hasil akurasi akhir. Berdasarkan hasil analisis, *fixed* parameter selanjutnya akan menggunakan *optimizer* Adam.

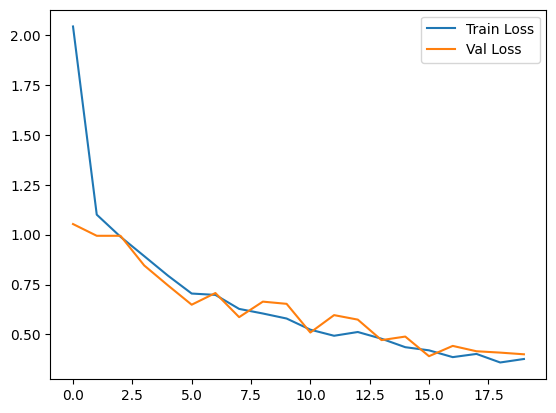
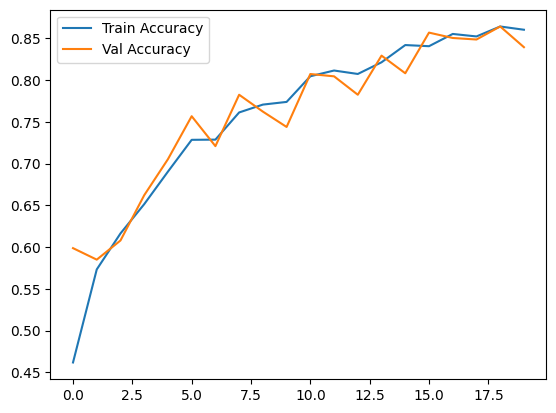
1. Pengujian Skenario 3 : Perubahan Nilai *Learning Rate*

Pada skenario ketiga melakukan tahapan perubahan nilai *learning rate*. Adapun *learning rate* yang digunakan adalah 0,01;0,001;0,0001. Dalam skenario ini parameter tetap yang digunakan yaitu *image size*, *optimizer*, *epoch*, dan *batch size*. Hasil pengujian skenario ini akan digunakan untuk mengetahui kualitas terbaik sistem klasifikasi dari nilai *learning rate* untuk menentukan parameter selanjutnya.

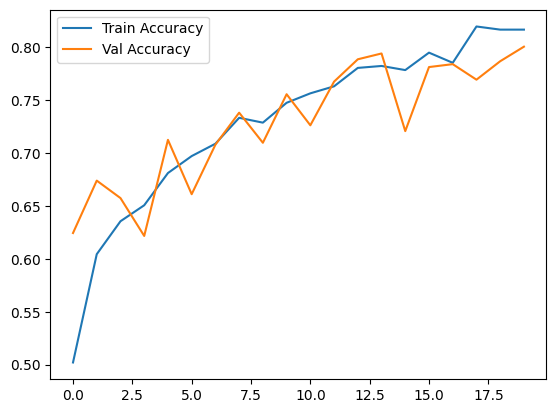
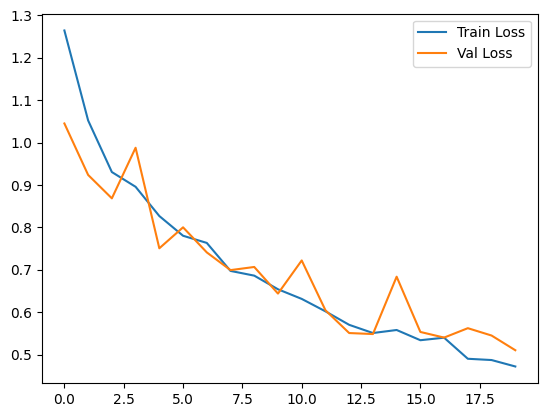
*Learning Rate 0.1*

*Learning Rate 0.001*



*Learning Rate 0.0001*

Gambar 4. 3. Visualisasi Accuracy dan Loss Skenario 3

Tabel 4. 7. Hasil Skenario 3

|  |  |  |
| --- | --- | --- |
| *Learning Rate* | Akurasi | *Loss* |
| 0.1 | 32.13% | 1.501 |
| 0.001 | 86.02% | 0.3767 |
| 0.0001 | 81.69% | 0.4722 |

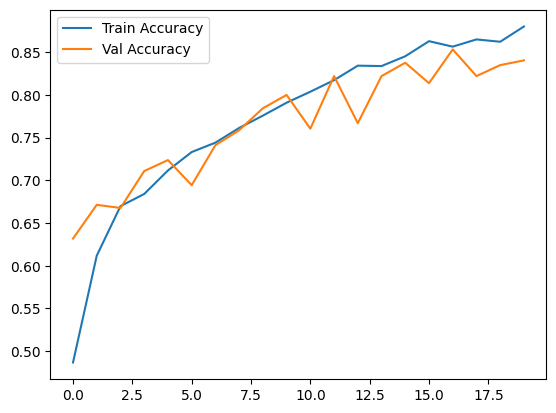
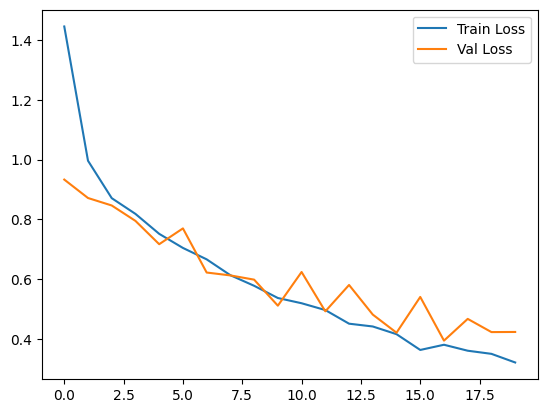
Berdasarkan Tabel 4.7., diperoleh akurasi terbaik pada pengujian *learning rate* 0,001. Akurasi dan *loss* yang dihasilkan adalah sebesar 86,02% dan 0,3767. Pengujian model dengan parameter 0,1 mengalami *early stop* karena proses training yang melewati syarat maksimal waktu dalam 5 *epochs*, oleh karena itu grafik yang ditunjukkan adalah garis datar setelah iterasi ke-1 dan e*poch* terhenti di angka 8. Sementara itu, *learning rate* 0,0001 cukup stabil namun memiliki waktu yang sangat lama dan angka akurasi yang masih kecil dan *loss* cukup besar.

Pada skenario 2, *learning rate* yang dipakai adalah 0,0001. Namun, setelah dilakukan pengujian dan hasil analisis, penulis menentukan *learning rate* paling baik adalah 0,001. Hal tersebut didukung dengan Gambar 4.3.yang menunjukkan akurasi terbaik ada di *learning rate* 0,001. Skenario selanjutnya akan menggunakan *fixed* parameter dengan *learning rate* 0,001.

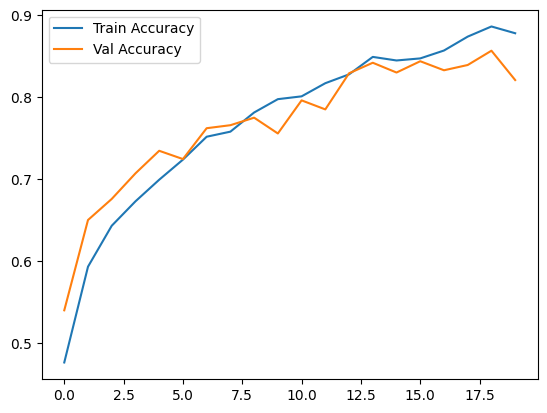
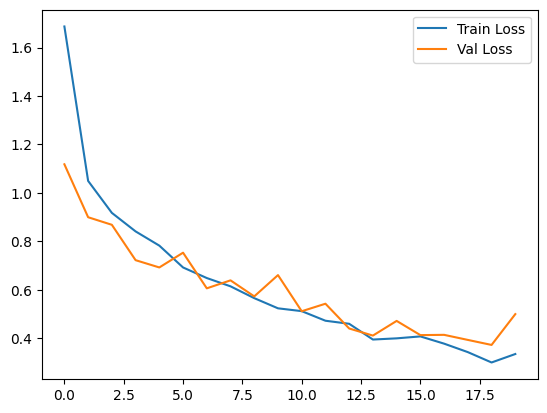
1. Pengujian Skenario 4 : Perubahan Nilai *Batch*

Pada skenario keempat melakukan tahapan perubahan ukuran *batch*. Adapun *batch* yang digunakan adalah 32,64,128. Dalam skenario ini parameter tetap yang digunakan yaitu *image size*, *optimizer*, *epoch*, dan *learning rate*. Hasil pengujian skenario ini akan digunakan untuk mengetahui kualitas terbaik sistem klasifikasi dari ukuran *batch.*

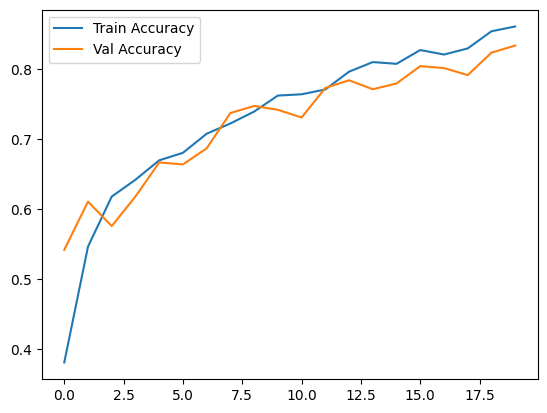
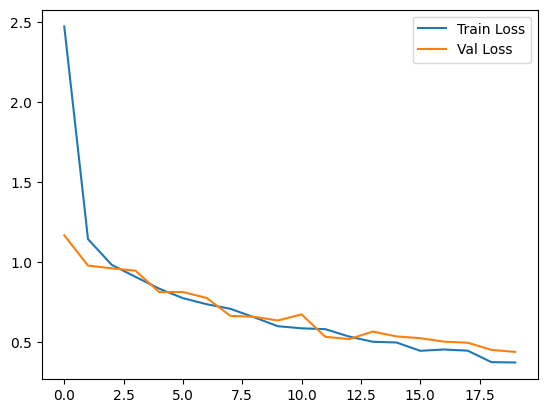
*Ukuran Batch 32*

*Ukuran Batch 64*

*Ukuran Batch 128*

Gambar 4. 4. Visualisasi Accuracy dan Loss Skenario 4

Tabel 4. 8. Hasil Skenario 4

|  |  |  |
| --- | --- | --- |
| *Batch* | Akurasi | *Loss* |
| 32 | 87.99% | 0.3216 |
| 64 | 87.81% | 0.3357 |
| 128 | 86.11% | 0.3750 |

Berdasarkan hasil pada Tabel 4.8.diperoleh akurasi terbaik pada pengujian *batch size* 32 dengan nilai akurasi 87,99% dan *loss* 0,7963. Selain itu, dari tabel di atas, dapat disimpulkan bahwa nilai *batch size* yang semakin besar belum tentu menghasilkan hasil pengujian sistem yang semakin baik.

Kemudian pada Gambar 4.4.memperlihatkan perbandingan grafik hasil pengujian *batch size*. Pada masing-masing grafik memiliki hasil grafik yang baik dengan tidak adanya *overfitting* ataupun *underfitting*. Pada pengujian ini hasil grafik terbaik dengan akurasi paling baik adalah hasil pengujian nilai *batch size* 32.

## **Pembahasan**

Setelah melakukan tahapan *training* dan *testing*, hasil pengujian 4 paramater berbeda menghasilkan *hyperparameter* yang terbaik untuk klasifikasi hama dan penyakit tanaman padi dengan data yang sudah difilterisasi. Berikut adalah pembahasan dari hyperparameter terbaik dari hasil uji.

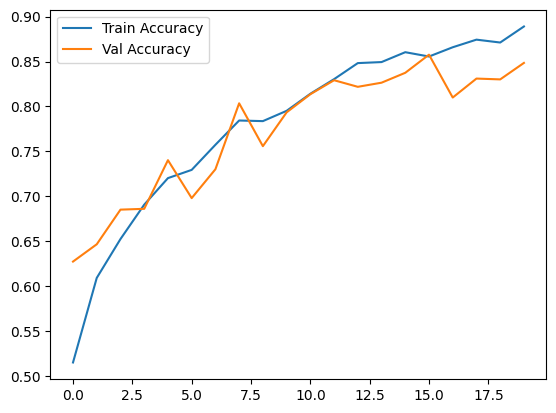
### Hasil Pengujian Terbaik

Pengujian sistem klasifikasi hama dan penyakit tanaman padi dengan metode convolutional neural network menggunakan 4 skenario yang berbeda dan dipilih parameter dengan hasil terbaik di setiap skenarionya. Parameter yang paling sesuai dengan sistem adalah ukuran citra 150x150 *pixel*, *optimizer* Adam, *learning rate* 0,001, dan *batch size* 32. Pada Tabel 4.9.ditunjukkan hasil pengujian menggunakan parameter yang sudah ditentukan.

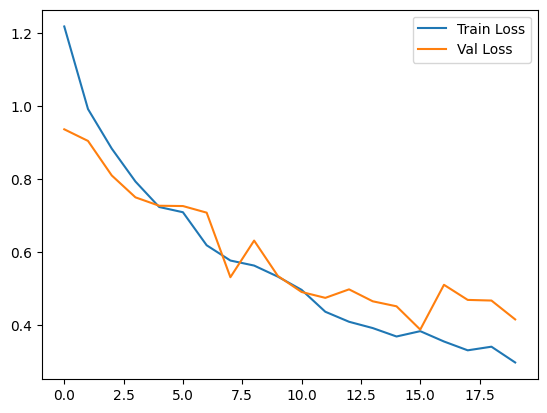
Tabel 4. 9. Hasil Pengujian Terbaik

|  |  |  |
| --- | --- | --- |
| **Hyperparameter** | **Akurasi** | **Loss** |
| *Input Size :* 150x150 *pixel* | 88,91% | 0.4154 |
| *Optimizer :* Adam |
| *Learning Rate :* 0,001 |
| *Batch Size* : 32 |

*Model Accuracy*



*Model Loss*

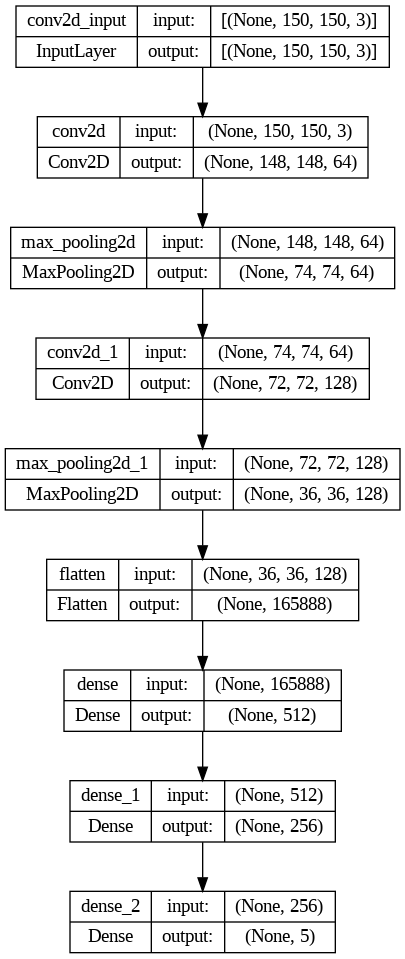


Gambar 4. 5. Visualisasi Accuracy dan Loss Hasil Akhir Pengujian

Berdasarkan Gambar 4.5.grafik di atas, dapat dilihat akurasi dari model dan *loss* stabil dan tidak terdeteksi *overfitting* ataupun *underfitting*. Hasil dari grafik tersebut didapatkan dari hasil penyusunan model dengan mempertimbangkan banyaknya dataset gambar serta kemampuan perangkat dalam melakukan training. *Training Accuracy* dan *Validation* stabil dengan angka yang terus meningkat hingga mencapai angka 88,91%. Sementara itu, nilai *loss* tidak melonjak tinggi dan tetap stabil dengan mencapai angka 0,4154.

### Model *Deep Learning* Pengujian Akhir

Proses pengujian hingga akhirnya menemukan parameter terbaik dalam memaksimalkan nilai akurasi dari *training* data untuk model, maka memerlukan arsitektur model CNN yang optimal dan tidak berlebihan. Dalam penelitian ini, tahap pengujian akhir dari data menggunakan arsitektur yang disusun tanpa melakukan proses *transfer learning*. Adapun arsitektur model CNN dari penelitian dapat divisualisasikan sebagai berikut.



Gambar 4. 6. Visualisasi Model CNN

Model CNN di atas memiliki 2 (dua) pasang lapisan konvolusi Conv2D (*Convolutional Layer*) dan *pooling* MaxPool2D yang disusun secara berurutan untuk mengekstrak fitur dari gambar dan kemudian mengurangi dimensi *feature map* yang dihasilkan lapisan konvolusi Conv2D. Dimensi yang berkurang ini dapat dilihat dari ukuran panjang dan lebar gambar sepanjang prosesnya melalui lapisan konvolusi dan *pooling* dari 150x150 (ukuran gambar input) sehingga menjadi 36x36 pada lapisan MaxPool2D sebelum diproses oleh lapisan Flatten. Fungsi aktivasi yang digunakan pada seluruh lapisan konvolusi Conv2D adalah RelU, yang melibatkan nonlinier pada model, mendorong model untuk berfokus pada fitur yang paling penting dan relevan sehingga proses pelatihan menjadi lebih efektif dan efisien (komputasi program lebih cepat).

Setelah diproses oleh lapisan konvolusi dan *pooling*, *feature map* diproses lapisan Flatten sehingga berubah menjadi bentuk vektor 1 (satu) dimensi. Vektor ini kemudian dimasukkan kepada lapisan Dense yang bertindak sebagai *fully-connected layer* untuk melakukan klasifikasi multi kelas. Algoritma optimasi Adam digunakan untuk meminimalisasi *loss function* saat proses pelatihan, dan fungsi kerugian *categorical cross-entropy* digunakan untuk menghitung selisih data yang diprediksi dengan data sebenarnya. Algoritma optimasi Adam digunakan dengan tingkat pembelajaran 0,001 yang memperbaharui bobot selama proses pelatihan model yang dilakukan dalam 20 *epoch*. Metrik yang digunakan untuk mengevaluasi hasil kinerja model merupakan *accuracy*.

Arsitektur model CNN yang dibangun adalah sebagai berikut.

Tabel 4. 10. Spesifikasi Arsitektur Model CNN

|  |  |  |
| --- | --- | --- |
| Lapisan Model | Konfigurasi Kompilasi Model | Jumlah Epoch |
| * Conv2D dengan 64 filter, kernel 3x3, fungsi aktivasi ‘RelU’, serta menerima bentuk masukan sesuai dengan *image size*, dan memiliki 3 kanal warna (RGB) * MaxPool2D dengan kernel berukuran 2x2 * Conv2D dengan 128 filter, kernel berukuran 3x3, fungsi aktivasi ‘RelU’ * MaxPool2D dengan kernel berukuran 2x2 * Flatten * Dense dengan 512 unit, fungsi aktivasi ‘RelU’ * Dense dengan 256 unit, fungsi aktivasi ‘RelU’ * Dense dengan 5 unit, fungsi aktivasi ‘Softmax’ | * Algoritma optimasi ‘Adam’ dengan tingkat pembelajaran 0,001. * Fungsi kerugian ‘Categorical Cross-Entropy’ * Metrik yang dievaluasi yaitu ‘Accuracy’ | 20 |

# BAB 5

# PENUTUP

## **5.1. Kesimpulan**

Berdasarkan penelitian yang dilakukan, dapat dicetuskan sejumlah kesimpulan sebagai berikut.

1. Untuk menghasilkan model dengan akurasi yang optimal, penelitian ini dilakukan 4 skenario yaitu *image size*, jenis *optimizer*, *learning rate*, dan ukuran *batch size*. Akurasi tertinggi yang diperoleh saat sistem menggunakan kombinasi parameter *resize* citra 150x150 *pixel*, *optimizer* Adam, *learning rate* 0,001, dan *batch size* 32.
2. Pada akhir pengujian model, penelitian ini memperoleh hasil akurasi sistem sebesar 88,91%. Sementara itu, nilai *loss* tidak melonjak tinggi dan tetap stabil dengan mencapai angka 0,4154.

## **5.2. Saran**

Berdasarkan penelitian yang sudah dilakukan, sebagai upaya dalam pengembangan penelitian ini di masa yang akan datang nanti, maka terdapat beberapa saran yaitu sebagai berikut.

1. Menambahkan jumlah kelas hama dan penyakit tanaman padi agar lebih maksimal penggunaan model CNN dalam kehidupan.
2. Melakukan tahapan *deploy* model yaitu menintegrasikannya dengan aplikasi dalam bentuk *website* ataupun *mobile*.
3. Melakukan uji dengan *object detection* agar pendeteksian hama dan penyakit oleh mesin dapat lebih akurat dan valid.

DAFTAR PUSTAKA

Anam Alidrus, S., Aziz, M., & Virgantara Putra, O. (2021). Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*.

Ayunina, R. (2023). *TINGKAT SERANGAN DAN KEHILANGAN HASIL TANAMAN PADI OLEH HAMA WALANG SANGIT DAN PENYAKIT BLAS DI KECAMATAN X KOTO SINGKARAK, KABUPATEN SOLOK, SUMATERA BARAT*. Institut Pertanian Bogor, Bogor.

Dewi, S. R. (2018). *DEEP LEARNING OBJECT DETECTION PADA VIDEO MENGGUNAKAN TENSORFLOW DAN CONVOLUTIONAL NEURAL NETWORK (Studi Kasus: Klasifikasi Gambar Meja dan Kursi Motif Ukiran Jepara)*. Universitas Islam Indonesia, Yogyakarta.

Dypanuryansyah, M. R. (2018). *IDENTIFIKASI SAWAH TERSERANG HAMA/PENYAKIT MENGGUNAKAN CITRA LANDSAT 8 (OLI) DI KABUPATEN INDRAMAYU*. Institut Pertanian Bogor, Bogor.

Jinan, A., & Hayadi, B. H. (2022). Klasifikasi Penyakit Tanaman Padi Mengunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). *Journal of Computer and Engineering Science*, *1*(2), 37–44.

Kanani, P., & Padole, M. (2019). Deep learning to detect skin cancer using google colab. *International Journal of Engineering and Advanced Technology*, *8*(6), 2176–2183. https://doi.org/10.35940/ijeat.F8587.088619

Khasanah, I. N., & Astuti, K. (n.d.). *LUAS PANEN DAN PRODUKSI PADI DI INDONESIA 2022*. Badan Pusat Statistik.

Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda*, *2*(1), 37–45. Retrieved from https://www.kaggle.com/tedisetiady/leaf-rice-disease-

Murad, N., Mappa, H., & Rudin, S. (2021). *PENGENALAN HAMA DAN PENYAKIT PADA TANAMAN PADI* (1st ed.; Dr. Ir. M. D. Mario & S. P. , Y. Ismail, Eds.). Gorontalo: Dinas Pertanian Provinsi Gorontalo.

Panahatan, A. J. (2023). *IDENTIFIKASI JAMUR YANG LAYAK DAN TIDAK LAYAK KONSUMSI MELALUI CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN ARSITEKTUR EFFICIENTNET*. Universitas Telkom, Bandung.

Puslitbang Tanaman Pangan, BPTP SUMUT, BPTP Riau, BPTP Lampung, BPTP DKI, BPTP DIY, … IRRI. (2007). *Hama dan Penyakit Tanaman Padi* (3rd ed.).

Rao Moparthi, N., Bhattacharyya, D., Balakrishna, G., & Prashanth, J. S. (2021). Paddy Leaf Disease Detection using CNN. *Volatiles & Essent. Oils*, *8*(4), 800–806.

Rizky, A. T., Fua’adah, R. Y. N., & Rizal, S. (2023). *DETEKSI PENYAKIT PADA TANAMAN JAGUNG BERBASIS PENGOLAHAN CITRA DIGITAL DENGAN METODE CONVOLUTIONAL NEURAL NETWORK*. Bandung.

Saputra, R. A., Wasyianti, S., Supriyatna, A., & Saefudin, D. F. (2021). Penerapan Algoritma Convolutional Neural Network Dan Arsitektur MobileNet Pada Aplikasi Deteksi Penyakit Daun Padi. *JURNAL SWABUMI*, *9*(2). Retrieved from https://archive.ics.uci.edu/ml/datasets/Rice

Shafira, T. (2018). *IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORKS UNTUK KLASIFIKASI CITRA TOMAT MENGGUNAKAN KERAS*.

Sheila, S., Kharil Anwar, M., Saputra, A. B., Pujianto, R., & Sari, I. P. (2023). Deteksi Penyakit pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Multinetics*, *9*(1), 27–34. Retrieved from https://www.kaggle.com/datasets/tedisetiady/leaf-

Swathika, R., Srinidhi, S., Radha, N., & Sowmya, K. (2021). Disease identification in paddy leaves using CNN based deep learning. *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, 1004–1008. https://doi.org/10.1109/ICICV50876.2021.9388557

Wijayanto, B., Kiswanto, & Manurung, G. O. (2013). *Hama dan Penyakit Utama Tanaman Padi* (J. Hadipurwanta, Ed.). Bandar Lampung: Balai Pengkajian Teknologi Pertanian (BPTP) Lampung.

Yulistiani, R., Ramadhan, I., Said, Q., & Santoni, M. M. (2018). Metode Regresi Linier Berganda dan SVR dalam Menentukan Tingkat Pengaruh Cuaca Terhadap Produktivitas Padi di Indonesia. *Sistem Informasi Dan Keamanan Siber (SEINASI-KESI) Jakarta-Indonesia*. Retrieved from http://www.bps.go.id