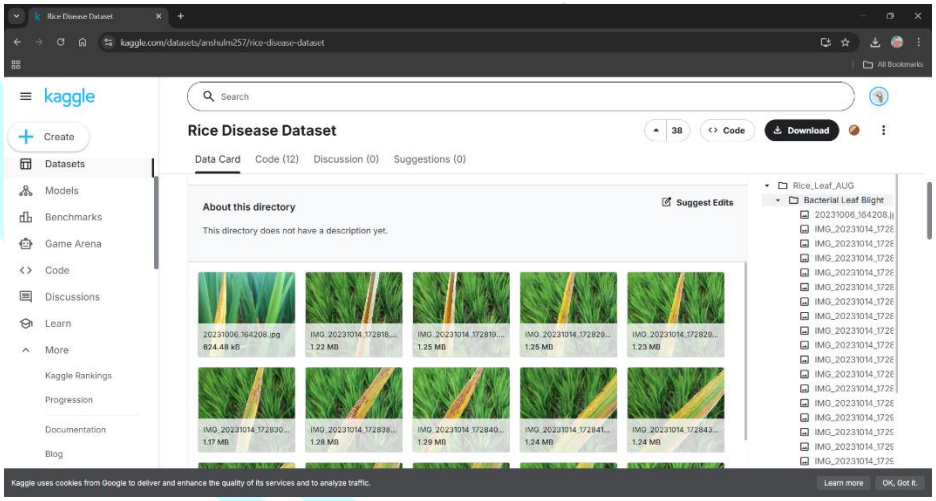
BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah sistem diagnosis penyakit tanaman padi berbasis *Computer Vision* yang mengintegrasikan model *Deep Learning* untuk klasifikasi jenis penyakit dengan teknik pengolahan citra digital untuk estimasi tingkat keparahan (*severity*). Sistem ini dirancang untuk melakukan analisis komparasi antara dua arsitektur *Convolutional Neural Network* (CNN), yaitu ResNet50 dan EfficientNetV2-S, dalam mengenali empat kategori kondisi daun: *Rice Blast* (Blas), *Brown Spot* (Bercak Coklat), *Bacterial Leaf Blight* (Hawar Daun), dan *Healthy* (Sehat) berdasarkan fitur visual lesi yang diekstraksi dari citra.



Gambar 3. 1 Tampilan Dataset Penyakit Daun Padi pada Platform Kaggle

Sumber: (https://www.kaggle.com/datasets/anshulm257/rice-disease-dataset)

Data yang digunakan dalam penelitian ini merupakan data sekunder yang bersumber dari repositori publik Kaggle dengan judul "Rice Leaf Diseases Detection" yang dipublikasikan oleh Anshul (2024). Secara keseluruhan, dataset asli memiliki total 3.829 citra yang terbagi ke dalam 6 kelas penyakit dengan distribusi sekitar 600 citra per kelas. Namun, disesuaikan dengan batasan masalah penelitian ini yang berfokus pada penyakit utama padi di Indonesia, dilakukan penyaringan (filtering) data. Penelitian ini hanya menggunakan 4 kelas dari dataset tersebut total 654 citra, yaitu:

1. *Bacterial Leaf Blight* (Hawar Daun Bakteri)
2. *Brown Spot* (Bercak Coklat)

3. *Leaf Blast* (Blas Daun)
4. *Healthy Rice Leaf* (Daun Sehat)

Dua kelas lainnya (Leaf Scald dan Sheath Blight) tidak diikutsertakan dalam eksperimen ini. Penghapusan dua kelas minor, seperti *Leaf Scald* dan *Sheath Blight*, didasarkan pada dua pertimbangan utama. Pertama, langkah ini bertujuan untuk mengurangi ketidakseimbangan data (*imbalance class*) karena distribusi citra pada penyakit minor tersebut sangat sedikit dibandingkan kelas penyakit utama. Jika dipaksakan untuk dilatih bersamaan, hal ini berisiko memicu bias pada model (*majority class bias*) dan menurunkan performa metrik evaluasi secara keseluruhan. Kedua, penghapusan ini dilakukan untuk menghindari *overlapping* fitur visual. Beberapa penyakit minor memiliki karakteristik bercak (*lesion*) yang secara visual sangat mirip dengan penyakit utama pada tahap awal infeksi. Dengan mencegah kebingungan model akibat kemiripan fitur tersebut, model dapat difokuskan untuk mencapai tingkat presisi dan *recall* yang lebih tinggi dalam mendeteksi tiga penyakit yang paling krusial secara ekonomi bagi petani.

Data mentah dari keempat kelas terpilih tersebut selanjutnya melalui tahapan pra-pemrosesan (*preprocessing*) (seperti *resizing* dan normalisasi) serta teknik augmentasi data (*data augmentation*) untuk memperbanyak variasi sampel dan mencegah *overfitting* pada saat pelatihan model *Deep Learning*. Setiap kelas memiliki keragaman visual yang mencakup variasi tahap infeksi, perbedaan latar belakang (*background*), serta kondisi pencahayaan yang bervariasi. Keragaman ini sangat krusial untuk mengevaluasi kemampuan generalisasi kedua model arsitektur dalam menghadapi kondisi lingkungan pertanian yang tidak ideal.

Seluruh citra dalam dataset diproses melalui tahapan pra-pemrosesan (*preprocessing*) dan augmentasi data sebelum digunakan untuk melatih model CNN guna mendapatkan akurasi klasifikasi yang optimal, setelah di lakukan tahap pra-pemrosesan menjadi 2.569 citra. Selain itu, sistem ini menerapkan pendekatan hibrida menggunakan algoritma segmentasi warna berbasis ruang warna HSV (*Hue, Saturation, Value*) pada pustaka OpenCV untuk memisahkan area lesi penyakit dari jaringan daun sehat. Rasio piksel hasil segmentasi tersebut kemudian dikalkulasi untuk menentukan persentase kerusakan fisik dan mengkategorikan tingkat keparahan penyakit secara kuantitatif.

3.1.1 Alat dan Bahan Penelitian

Pelaksanaan penelitian ini menggunakan perangkat keras dan perangkat lunak yang mendukung proses pengembangan model *Deep Learning*, pelatihan (*training*), hingga implementasi aplikasi sistem pakar berbasis web. Adapun spesifikasi alat dan bahan yang digunakan adalah sebagai berikut:

1) Perangkat Keras (*Hardware*)
   a) Laptop: Ideapad Gaming 315IAH7
   b) CPU: Intel i5-12500H
   c) RAM: 16 GB
   d) GPU: RTX 3050 Laptop
   e) Penyimpanan: 512 GB
2) Perangkat Lunak (*Software*)
   a) Sistem Operasi: Windows 11 (64-bit)
   b) Code Editor: Visual Studio Code atau Jupyter Notebook
   c) Lingkungan Pengembangan: Anaconda (Conda Environment padi_gpu)
   d) Bahasa Pemrograman: Python 3.9.25
   e) Framework & Library Utama:
      1. *Deep Learning*: TensorFlow-GPU 2.10.1 & Keras 2.10.0 (Pembangunan dan pelatihan model ResNet50 & EfficientNetV2-S)
      2. *Computer Vision*: OpenCV (opencv-python 4.9.0.80) (Pra-pemrosesan citra dan segmentasi warna HSV untuk kalkulasi *severity*)
      3. Web Framework (*Backend*): Flask 3.0 (Pembuatan antarmuka pengguna/Dashboard sistem)
      4. *Frontend*: HTML5, CSS3, & JavaScript (Vanilla JS) (Untuk antarmuka pengguna yang responsif).
      5. Manipulasi Data: NumPy 1.26.4 (Operasi matriks citra), Pandas 2.3.3 (Analisis data tabular)
      6. Evaluasi Model: Scikit-Learn 1.6.1 (Perhitungan *Confusion Matrix*, Akurasi, Presisi, *Recall*, *F1-Score*)
      7. Visualisasi: Matplotlib 3.9.4, Seaborn 0.13.2 (Grafik performa pelatihan dan visualisasi hasil prediksi)

f) Akselerator GPU: CUDA Toolkit 11.2.2 & cuDNN 8.1.0 (Driver untuk optimalisasi komputasi GPU pada TensorFlow)

3.1.2 Jadwal Penelitian

Tabel 3. 1 Jadwal Penelitian

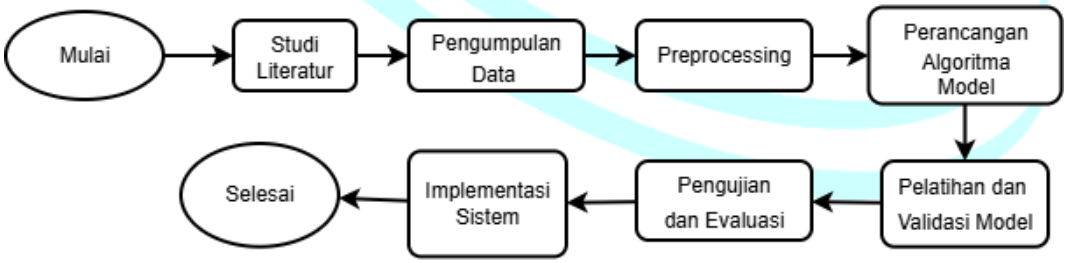
No.	Kegiatan Penelitian	Bulan 1	Bulan 2	Bulan 3	Bulan 4
1.	Studi Literatur & Perumusan Masalah				
2.	Pengumpulan Data (*Data Collection*)				
3.	Pra-pemrosesan Data (*Preprocessing*)				
4.	Perancangan & Pelatihan Model				
5.	Pengembangan Modul Severitas				
6.	Evaluasi & Analisis Komparasi				
7.	Implementasi Sistem (Web App)				
8.	Penyusunan Laporan Tugas Akhir				

3.2 Prosedur Penelitian

Prosedur penelitian ini terdiri dari beberapa tahapan yang disusun secara sistematis dan saling berkaitan untuk mencapai tujuan diagnosa yang presisi. Tahap awal dimulai dengan pengumpulan dataset *"Rice Leaf Disease Image"* yang dipublikasikan oleh Anshul melalui platform Kaggle, yang kemudian dianalisis untuk memastikan keterwakilan visual dari empat kelas kondisi daun (Blas, Bercak Coklat, Hawar Daun, dan Sehat). Setelah data terkumpul, dilakukan proses pra-pemrosesan (*preprocessing*) yang meliputi penyeragaman resolusi citra dan augmentasi data geometris. Tahap ini bertujuan menghasilkan *input* data yang konsisten dan bervariasi sebelum digunakan dalam proses pelatihan model *Deep Learning*.

Pada tahap berikutnya, dua arsitektur *Convolutional Neural Network* (CNN), yaitu ResNet50 dan EfficientNetV2-**S**, dibangun dan dilatih secara terpisah menggunakan data yang telah diproses. Proses pelatihan dilakukan secara bertahap menggunakan teknik *Transfer Learning*, disertai pemantauan metrik akurasi, *loss*, presisi, *recall*, dan *F1-score* untuk memastikan model memenuhi kriteria performa yang diharapkan tanpa mengalami *overfitting* atau *underfitting*. Jika model belum mencapai performa optimal, dilakukan penyesuaian *hyperparameter* (seperti *learning rate*) menggunakan mekanisme *callbacks* hingga hasil yang diperoleh stabil.

Paralel dengan pelatihan model klasifikasi, dikembangkan pula modul estimasi tingkat keparahan (*severity*) menggunakan teknik *Computer Vision* berbasis pustaka OpenCV. Modul ini menghitung persentase kerusakan fisik pada daun melalui segmentasi warna di ruang HSV. Setelah model klasifikasi mencapai performa terbaik dan modul severitas teruji, langkah selanjutnya adalah mengintegrasikan keduanya ke dalam sistem aplikasi berbasis web menggunakan *framework* Flask. Sistem ini dirancang untuk bekerja secara interaktif dalam mendiagnosis jenis penyakit sekaligus memberikan *grading* tingkat keparahan berdasarkan *input* citra pengguna. Tahap terakhir adalah pengujian sistem secara langsung untuk memastikan kehandalan model dalam mendeteksi penyakit dengan akurat pada berbagai variasi visual daun padi. Alur lengkap prosedur penelitian ditunjukkan pada Gambar 3.1.



Gambar 3. 2 Diagram Alir (*Flowchart*) Prosedur Penelitian

Sumber: (Dokumen Pribadi,2025)

3.2.1 Studi Literatur

Tahap awal penelitian dimulai dengan studi literatur yang mendalam untuk memahami konsep dasar patologi penyakit padi, karakteristik visual dari tiga penyakit utama (Blas, Bercak Coklat, dan Hawar Daun), serta metode *Deep*

Learning mutakhir yang relevan. Referensi utama diperoleh dari jurnal nasional dan internasional bereputasi yang membahas implementasi arsitektur CNN seperti ResNet50 dan EfficientNetV2 dalam domain pertanian. Fokus utama pada tahap ini adalah mengidentifikasi kesenjangan (*research gap*) dari penelitian terdahulu, khususnya mengenai kurangnya integrasi antara klasifikasi jenis penyakit dengan estimasi tingkat keparahan yang akurat dan efisien secara komputasi.

Masalah utama yang diangkat adalah bagaimana membandingkan kinerja dua arsitektur dalam menangani dataset penyakit padi yang memiliki variasi visual tinggi, serta bagaimana merancang metode perhitungan severitas yang objektif menggunakan teknik pengolahan citra digital. Hasil dari tahap ini adalah kerangka konseptual penelitian yang menjadi landasan bagi tahapan eksperimental selanjutnya.

3.2.2 Pengumpulan Data (*Data Collection*)

Data citra dikumpulkan dari dataset sekunder yang tersedia secara publik di platform Kaggle dengan nama "*Rice Leaf Disease Image*". Dataset ini dipilih karena memiliki jumlah sampel yang memadai dan variasi visual yang representatif untuk kondisi lapangan. Setelah melalui tahap penyaringan (filtering) dan augmentasi, total citra yang digunakan dalam penelitian ini berjumlah 2.569 citra. Dataset tersebut kemudian dibagi secara proporsional menjadi tiga himpunan bagian: Data Latih (70%) sebanyak 1.797 citra, Data Validasi (20%) sebanyak 512 citra, dan Data Uji (10%) sebanyak 260 citra.. Struktur direktori data disusun berdasarkan empat kategori kelas utama: *Blast* (berisi citra daun terinfeksi jamur *Pyricularia oryzae*), Blight (berisi citra daun terinfeksi bakteri *Xanthomonas oryzae*), *Brown Spot* (berisi citra daun terinfeksi jamur *Bipolaris oryzae*), dan Healthy (berisi citra daun padi sehat tanpa gejala visual), gambar 3.2 menampilkan contoh sampel citra asli dari dataset untuk setiap kelas penyakit.



Gambar 3. 3 Contoh Dataset Asli Citra Daun Padi

Sumber: (Kaggle/Anshul)

Terlihat bahwa citra asli memiliki latar belakang yang bervariasi (tanah, langit, atau daun lain) dan kondisi pencahayaan yang berbeda, yang menjadi tantangan bagi model klasifikasi. Setelah itu, dilakukan proses kurasi data secara manual untuk membuang citra yang rusak, buram, atau tidak relevan yang dapat mengganggu proses pelatihan model. Setiap citra diperiksa labelnya untuk memastikan tidak ada kesalahan pelabelan (*mislabeling*) dari sumber aslinya. Proses verifikasi ini penting untuk menjamin bahwa data yang masuk ke dalam model adalah data berkualitas tinggi (*clean data*), karena kualitas data sangat menentukan performa akhir model *Deep Learning*.

3.2.3 Pra-pemrosesan Data (*Preprocessing*)

Sebelum masuk ke tahap pelatihan, data mentah diproses terlebih dahulu melalui serangkaian teknik pra-pemrosesan untuk meningkatkan kualitas dan keseragaman *input* model. Langkah pertama adalah *Resizing*, di mana seluruh citra diubah resolusinya menjadi dimensi tetap 224 × 224 piksel. Ukuran ini dipilih karena merupakan standar *input* untuk arsitektur ResNet50 dan EfficientNetV2 yang menggunakan bobot *pre-trained* ImageNet. Langkah kedua adalah Normalisasi, yaitu mengubah nilai intensitas piksel dari rentang 0-255 menjadi rentang 0-1 atau menstandarisasinya berdasarkan rata-rata (mean) dan deviasi standar (std) ImageNet. Normalisasi bertujuan mempercepat konvergensi algoritma optimasi selama pelatihan.

Selanjutnya, diterapkan teknik Augmentasi Data menggunakan fungsi *ImageDataGenerator* dari pustaka Keras. Augmentasi bertujuan untuk memperbanyak variasi data latih secara artifisial dan mencegah model menghafal data (*overfitting*). Transformasi geometris yang diterapkan secara acak meliputi rotasi dalam rentang tertentu, pembalikan *horizontal* (*horizontal flip*), pergeseran posisi (*shift*), dan pembesaran (*zoom*). Terakhir, dataset dibagi (*Data Splitting*) menjadi tiga himpunan bagian: Data Latih (70%) untuk melatih bobot model, Data Validasi (20%) untuk evaluasi performa per-*epoch*, dan Data Uji (10%) untuk pengujian akhir pada data yang belum pernah dilihat (*unseen data*).



Gambar 3. 4 Visualisasi Hasil Augmentasi Data

Sumber: (Dokumen Pribadi,2025)

3.2.4 Perancangan Model

Sistem yang dirancang menerapkan pendekatan *hybrid* yang mengintegrasikan model *Deep Learning* berbasis *Convolutional Neural Network* (CNN) untuk tugas klasifikasi dengan algoritma *Computer Vision* konvensional untuk estimasi tingkat keparahan (*severity*). Secara teknis, rancangan model dalam penelitian ini adalah sebagai berikut:

1. Modul Validasi Objek (*Pre-check*): Sistem melakukan pemindaian awal terhadap citra *input* dengan memeriksa rasio warna hijau. Jika rasio warna hijau berada di bawah ambang batas (*threshold*) yang ditentukan, maka sistem akan menolak citra tersebut dan menganggapnya sebagai objek non-daun padi. Hal ini bertujuan untuk meminimalkan kesalahan prediksi (*false prediction*) pada tahap selanjutnya

2. Modul Klasifikasi Penyakit (*Deep Learning*):
   a) *Input*: Citra RGB yang telah melalui proses validasi diubah ukurannya menjadi dimensi 224 × 224 piksel untuk disesuaikan dengan arsitektur model.
   b) Proses: Citra diproses menggunakan proses inferensi dari salah satu model yang dikomparasikan, yaitu EfficientNetV2-S atau ResNet50.
   c) Logika *Output*: Jika hasil prediksi menunjukkan kategori *Healthy Rice Leaf* (Daun Sehat), maka sistem secara otomatis akan menetapkan nilai severitas sebesar 0% tanpa melanjutkan ke proses segmentasi.
3. Modul Estimasi Severitas (*Computer Vision*): Jika citra teridentifikasi memiliki penyakit, maka sistem akan menjalankan algoritma berbasis pustaka OpenCV. Sebagai langkah optimasi dan pencegahan kesalahan prediksi (*False Positive*), sistem dirancang dengan aturan khusus: apabila *input* citra terdeteksi sebagai kelas *Healthy Rice Leaf*, maka algoritma segmentasi HSV akan dilewati (di-*bypass*) dan sistem akan langsung menetapkan tingkat keparahan pada *Grade* 0 (0%). Jika selain kelas sehat, algoritma segmentasi akan memisahkan area daun dan lesi menggunakan masker biner dengan langkah-langkah:
   a) Segmentasi Citra: Memisahkan area daun sehat dengan area lesi penyakit melalui transformasi ruang warna dari BGR ke HSV (*Hue*, *Saturation*, *Value*) dan penerapan teknik *thresholding* untuk menghasilkan masker biner.
   b) Kalkulasi Rasio: Menghitung persentase tingkat kerusakan secara matematis dengan membagi total luas piksel area lesi dengan total luas piksel area keseluruhan daun.
   c) Grading: Hasil persentase tersebut kemudian dikonversikan ke dalam kategori tingkat keparahan (Ringan, Sedang, Berat) berdasarkan standar IRRI.
4. Standarisasi Penentuan kategori tingkat keparahan dalam penelitian ini mengacu pada standar *Standard Evaluation System for Rice* (SES) yang diterbitkan oleh International Rice Research Institute (IRRI, 2025), skala

asli 0-9 telah disederhanakan menjadi skala 0-3 untuk menyesuaikan dengan logika sistem, seperti terlihat pada Tabel 3.2.

Tabel 3. 2 Standarisasi Penentuan Tingkat Keparahan (IRRI, 2025)

Grade	Tingkat Keparahan	Rentang Kerusakan (Rasio)
0	Sehat (*Healthy*)	0%
1	Ringan (*Mild*)	0.1% – 10%
2	Sedang (*Moderate*)	10.1% – 25%
3	Berat (*Severe*)	> 25%

3.2.5 Pelatihan dan Validasi Model

Proses pelatihan model dilakukan menggunakan algoritma optimasi Adam (*Adaptive Moment Estimation*) dengan *learning rate* awal yang disesuaikan. Fungsi kerugian (*loss function*) yang digunakan adalah *Categorical Cross-Entropy*, yang sesuai untuk masalah klasifikasi multi-kelas. Selama pelatihan, model belajar mengenali pola fitur penyakit dari data latih dan dievaluasi kinerjanya menggunakan data validasi pada setiap akhir *epoch*.

Untuk mengontrol jalannya pelatihan dan mencegah *overfitting*, diterapkan mekanisme *Callbacks*. *ReduceLROnPlateau* digunakan untuk menurunkan *learning rate* secara otomatis jika akurasi validasi tidak mengalami peningkatan dalam beberapa *epoch*, memungkinkan model untuk mencapai minimum global yang lebih presisi. Selain itu, *EarlyStopping* diaktifkan untuk menghentikan proses pelatihan secara dini jika *validation loss* tidak membaik setelah sejumlah *epoch* tertentu (*patience*), sehingga menghemat waktu komputasi dan mencegah model menghafal *noise* pada data latih. *ModelCheckpoint* juga digunakan untuk menyimpan bobot model terbaik yang memiliki akurasi validasi tertinggi.

3.2.6 Pengujian dan Evaluasi

Setelah proses pelatihan selesai, model terbaik dari masing-masing arsitektur dievaluasi menggunakan Data Uji (*Test Set*) yang benar-benar baru dan belum pernah dilihat oleh model sebelumnya. Evaluasi kinerja dilakukan secara komprehensif menggunakan *Confusion Matrix* untuk memetakan prediksi benar dan salah pada setiap kelas penyakit. Dari matriks ini, dihitung metrik kuantitatif utama yaitu Akurasi (tingkat kebenaran keseluruhan), Presisi (ketepatan prediksi

positif), *Recall* (kemampuan mendeteksi penyakit), dan *F1-Score* (rata-rata harmonik presisi dan *recall*).

Selain metrik performa klasifikasi, dilakukan juga analisis komparasi dari aspek efisiensi komputasi. Parameter yang dibandingkan meliputi waktu yang dibutuhkan untuk satu epoch pelatihan (*training time*) dan ukuran *file* model akhir (*model size*). Analisis komparasi ini bertujuan untuk menentukan model mana yang memberikan *trade-off* terbaik antara akurasi deteksi dan efisiensi sumber daya, sehingga paling layak untuk diimplementasikan pada sistem nyata dengan keterbatasan perangkat keras.

3.2.7 Implementasi Sistem

Tahap akhir penelitian adalah mengimplementasikan model terpilih dan modul severitas ke dalam sebuah aplikasi sistem pakar berbasis web. Aplikasi ini dibangun menggunakan *framework Flask* yang memungkinkan pengembangan antarmuka pengguna (*User Interface*) interaktif dengan bahasa Python. Sistem dirancang agar mudah digunakan (*user-friendly*), di mana pengguna cukup mengunggah citra daun padi melalui antarmuka web.

Di sisi *backend*, sistem akan memproses citra unggahan tersebut melalui dua jalur: jalur pertama menggunakan model CNN untuk memprediksi jenis penyakit dan tingkat kepercayaan (*confidence score*), sedangkan jalur kedua menggunakan algoritma OpenCV untuk menghitung persentase kerusakan dan menentukan grade keparahan. Hasil dari kedua proses ini kemudian ditampilkan kembali ke pengguna dalam bentuk visualisasi yang informatif, lengkap dengan gambar hasil segmentasi area sakit. Pengujian fungsionalitas sistem dilakukan untuk memastikan seluruh fitur berjalan dengan baik dan responsif.