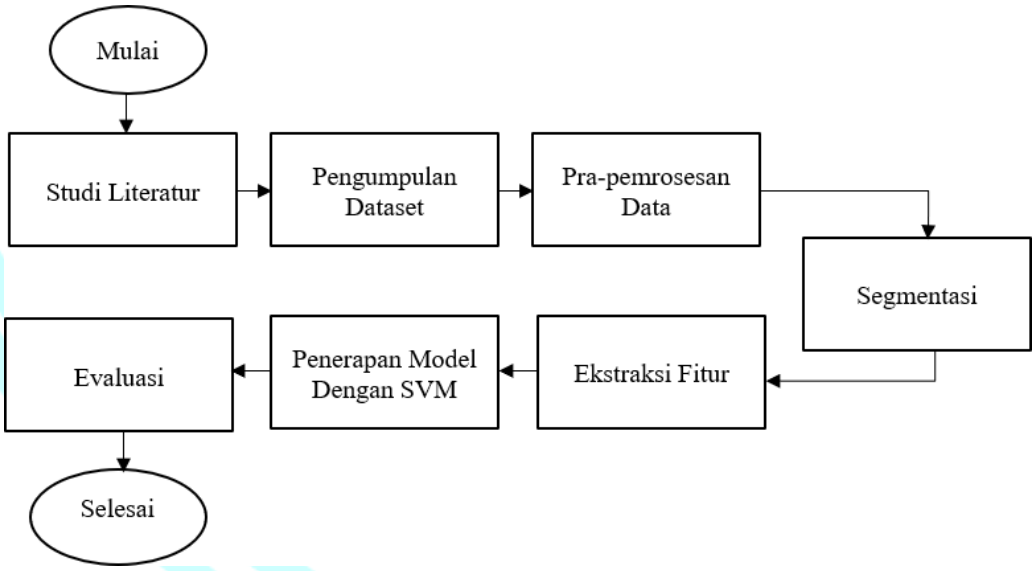
BAB III
METODE PENELITIAN

3.1. Objek Penelitian

Penelitian ini dilakukan dengan dataset berupa data citra penyakit pada daun tanaman padi yang memiliki tiga kelas, yaitu penyakit *blast*, *blight* dan *tungro*. Dalam dataset ini ditemukan ukuran citra yang berbeda-beda sehingga memerlukan tahapan pra-pemrosesan data.

3.2. Prosedur Penelitian

Prosedur pada penelitian ini memiliki alur seperti berikut:



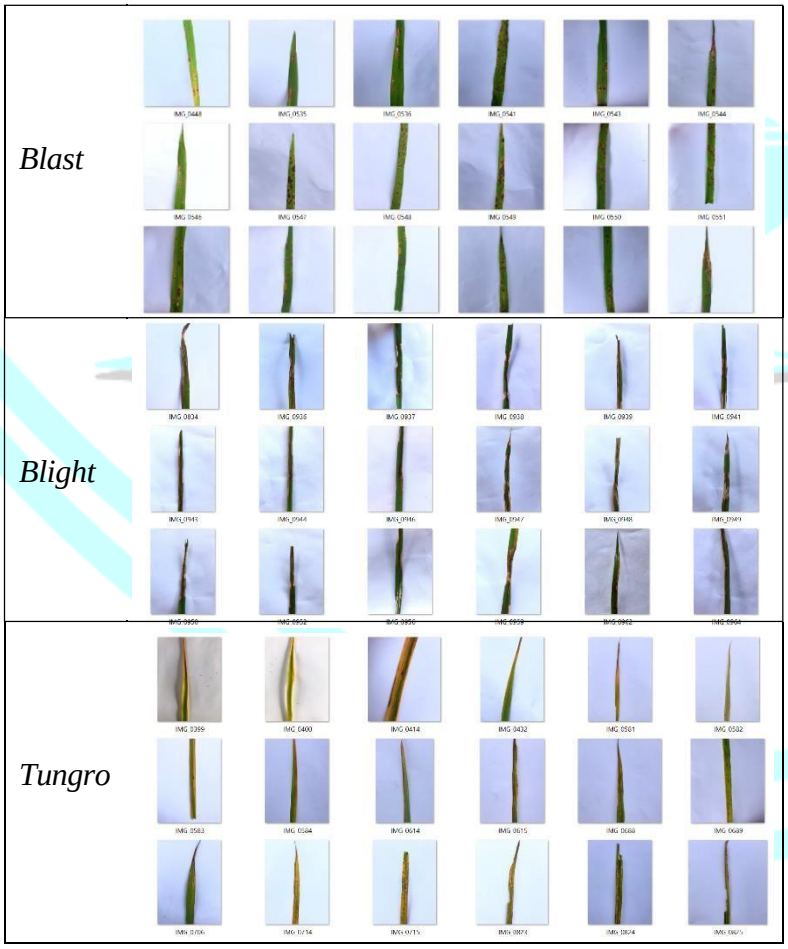
Gambar 3.2 *Flowchart* Alur Prosedur Penelitian

3.2.1. Studi Literatur

Penelitian ini dibuat berdasarkan hasil belajar dari berbagai sumber pada beberapa jurnal, didapat dari *Google Scholar* dan jurnal Universitas Buana Perjuangan Karawang. Sumber-sumber tersebut tidak hanya memberikan informasi yang mendukung dasar teori, tetapi juga membantu memperluas pemahaman mengenai topik yang dibahas. Kajian literatur ini dilakukan dengan cermat untuk memastikan data yang digunakan akurat dan terpercaya. Semua referensi yang dipakai dicantumkan di bagian daftar pustaka pada akhir dokumen.

3.2.2. Pengumpulan Dataset

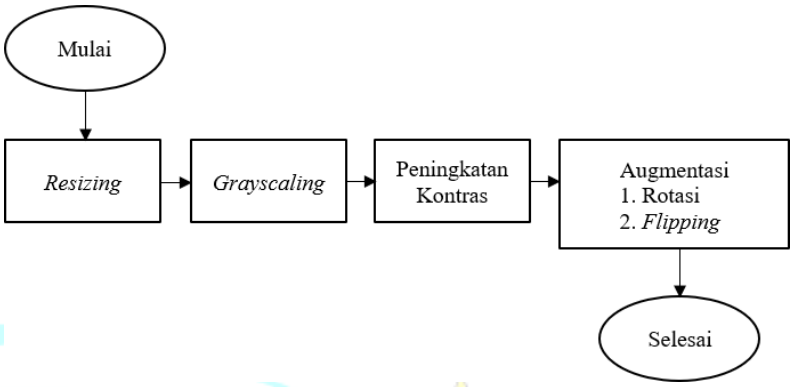
Dataset yang digunakan dalam penelitian ini merupakan kumpulan citra penyakit daun padi yang mencakup tiga jenis utama, yaitu *blast*, *blight*, dan *tungro*. Dataset ini menjadi fondasi utama dalam seluruh proses pengolahan data, mulai dari pra-pemrosesan hingga tahap analisis dan evaluasi model klasifikasi. Citra-citra daun padi tersebut diperoleh dari sumber yang terpercaya, yakni situs *Kaggle*, yang menyediakan data visual berkualitas tinggi untuk keperluan penelitian dan pengembangan model pembelajaran mesin. Secara spesifik, dataset diunduh dari tautan berikut : https://www.kaggle.com/code/stpeteishii/leaf-rice-disease-classify-densenet201?select=blight, dan digunakan untuk mendukung eksperimen klasifikasi penyakit daun secara akurat dan sistematis.



Gambar 3.2. Dataset citra penyakit pada daun padi

3.2.3. Pra-pemrosesan Data

Tahapan-tahapan saat pra-pemrosesan data:



Gambar 3.2 *Flowchart* pra-pemrosesan dataset

Berikut adalah langkah-langkah proses pra-pengolahan data:

1. *Resizing* merupakan langkah awal dalam pra-pemrosesan citra yang bertujuan untuk menyesuaikan ukuran gambar ke dimensi tertentu, agar ukuran gambar seragam. sehingga dapat diproses secara konsisten oleh model atau algoritma. Dimensi yang digunakan 256x256 piksel dipilih berdasarkan kebutuhan analisis. Proses *resizing* menggunakan metode *bicubic interpolation* untuk menjaga detail dengan lebih baik.
2. Selanjutnya proses *grayscaling* citra, yaitu mengubah gambar berwarna (RGB) menjadi gambar *grayscale*. Proses *grayscaling* yang dilakukan dengan tahapan sebagai berikut:
   a. Membaca citra berwarna dalam format BGR (*Blue, Green, Red*) dengan *OpenCV*
   b. Konversi ke *grayscale*, setiap piksel dihitung ulang menjadi satu nilai intensitas keabuan dengan rumus:

$$Gray = R + G + B \tag{3.1}$$

   c. Menampilkan atau menyimpan hasil citra *grayscale.*
3. Peningkatan kontras bertujuan untuk menonjolkan detail penting pada citra, sehingga fitur menjadi lebih mudah dikenali. Metode yang digunakan *Contrast Limited Adaptive Histogram Equalization* (*CLAHE*), yang meningkatkan

kontras lokal sambil membatasi distribusi intensitas untuk mencegah peningkatan *noise* secara berlebihan.

4. Augmentasi digunakan untuk memperbanyak variasi data pelatihan tanpa harus mengumpulkan data baru..

3.2.4. Segmentasi

Segmentasi citra pada proses penelitian ini menggunakan *Thresholding HSV* untuk memisahkan objek utama penyakit daun tanaman padi dari latar belakang yaitu daun padi. Berikut adalah tahapan segmentasi menggunakan metode *Thresholding HSV:*

1. Langkah pertama adalah membaca citra menggunakan pustaka pemrosesan gambar seperti *OpenCV*.
2. Langkah kedua citra dengan format *BGR (Blue, Green, Red)*, harus dikonversi ke HSV karena HSV lebih efektif untuk analisis warna.

   Rumus :

$$H = \frac{atan2\left(\frac{2(R-G-B)}{3(G-B)}\right)\left(\max_{R,G,B}\left(\right)-\min_{R,G,B}\left(\right)\right)}{\max(R,G,B)}\max(R,G,B)) \tag{3.2}$$

   Di mana:

   H (*Hue*) → Nilai warna dalam derajat (0°-360°), dalam *OpenCV* biasanya dinormalisasi ke 0-179.

   S (*Saturation*) → Tingkat kejenuhan warna (0-255).

   V (*Value*) → Intensitas kecerahan warna (0-255).

$$\text{Mask}(x,y) = \begin{cases} 1, & \text{jika } H_{\min} \le H(x,y) \le H_{\max} \text{ dan } S_{\min} \le S(x,y) \le S_{\max} \text{ dan } V_{\min} \le V(x,y) \le V_{\max} \\ 0, & \text{lainnya} \end{cases} \tag{3.3}$$

   Di mana:

   $H_{min}, H_{\max}$ nilai *Hue* yang sesuai dengan warna target.

   S_{min}, S_{max} akurasi untuk membedakan warna yang kuat dan pudar.

   V_{min}, V_{max} intensitas untuk mengabaikan bayangan atau pencahayaan yang berlebihan.

4. Mask yang diperoleh dari *thresholding* digunakan untuk menyaring area tertentu pada citra asli.

3.3.4. Ekstraksi Fitur

Ekstraksi fitur dalam penelitian ini menggunakan tiga metode utama, yaitu *GLCM*, *HOG* , dan *LBP* sebagai perbandingan. Dari matriks *GLCM* yang dihasilkan, diambil enam properti tekstur utama, yaitu *contrast*, *dissimilarity*, *homogeneity*, *energy*, *correlation*, dan *ASM*. Untuk metode *HOG*, citra hasil peningkatan kontras diproses dengan parameter jumlah orientasi *gradien* sebesar 9, ukuran setiap sel piksel 8×8, dan ukuran blok 2×2 sel. Sementara itu, metode *LBP* digunakan untuk mendeskripsikan pola lokal di sekitar setiap piksel. Pola ini dihitung dengan radius 1 piksel dan 8 titik sampel di sekitarnya. Visualisasi dari ketiga metode ini ditampilkan dalam tiga panel, yaitu grafik batang dari nilai fitur *GLCM*, visualisasi arah *gradien HOG*, serta citra *LBP* hasil ekstraksi pola lokal. Terakhir, akurasi ketiga metode divisualisasikan dalam grafik batang menggunakan matplotlib untuk menunjukkan metode ekstraksi fitur terbaik..

3.2.5. Penerapan Model SVM

Algoritma *modelling* SVM

ALGORITMA *modelling* SVM klasifikasi penyakit pada citra daun padi

---

INPUT:

- Dataset citra (berisi gambar dan label)
- *Split Dataset*
- Pilih Kernel (RBF)

OUTPUT:

- Model SVM terlatih

LANGKAH-LANGKAH:

1. *Split dataset* menjadi:
   - *Training set* (untuk melatih model)
   - *Testing set* (untuk evaluasi model)
2. Pilih kernel RBF
3. Latih model SVM menggunakan *training* set

Proses penerapan dilakukan setelah pra-pemrosesan dataset di mana citra penyakit daun tanaman padi telah melalui langkah penting untuk meningkatkan kualitas data dan mempermudah analisis selanjutnya. Berikut adalah penjelasannya:

1. *Split Dataset*.

Setelah proses pra-pemrosesan selesai, dataset dibagi menjadi dua *subset*, yaitu dataset pelatihan (*training*) dan dataset pengujian (*testing*). Pembagian dataset dilakukan secara hati-hati untuk memastikan distribusi kelas penyakit pada daun tanaman padi tetap seimbang di kedua *subset*. Dataset pelatihan dipakai untuk melatih moder supaya bisa mengenali pola dari data yang diberikan, sedangkan dataset pengujian digunakan untuk mengevaluasi kemampuan model dalam menggeneralisasi pada data baru.

Umumnya, dataset dibagi dengan 80% data *training* dan 20% data *testing*, meskipun proporsi ini dapat disesuaikan tergantung pada ukuran dataset. Pembagian dilakukan menggunakan teknik seperti *train-test split* yang tersedia dalam pustaka *scikit-learn*. Parameter seperti *stratify* digunakan untuk menjaga keseimbangan distribusi kelas penyakit dalam dataset. Dengan menjaga keseimbangan ini, model dapat dilatih dengan data yang representatif, sehingga menghindari bias terhadap salah satu kelas penyakit. Pembagian yang baik memungkinkan proses pelatihan dan evaluasi berjalan optimal, sehingga kinerja model dapat diukur secara objektif pada tahap berikutnya.

2. Training Model

Pada tahap ini, model *Support Vector Machine* (SVM) dilatih menggunakan dataset pelatihan yang telah disiapkan. Model SVM bertujuan untuk menemukan *hyperplane* optimal yang memisahkan tiga kelas (*blast*, *bliht*, dan *tungro*) dengan margin maksimal. Model dilatih untuk mengenali hubungan antara fitur-fitur yang telah diekstraksi dari citra dengan label kelasnya. Proses pelatihan dilakukan dengan memilih *Kernel RBF*, yang merupakan pilihan tepat ketika data tidak dapat dipisahkan secara *linier*. *Kernel RBF* bekerja dengan cara memetakan data ke ruang berdimensi lebih tinggi, sehingga pola yang kompleks dalam data dapat dipisahkan secara lebih efektif. Dalam ruang baru ini, pemisahan antara kelas dilakukan menggunakan kurva

atau fungsi yang sesuai dengan distribusi data. Kernel seperti *Radial Basis Function (RBF)* sering digunakan karena fleksibilitasnya dalam menangani pola data yang tidak *linier*.

3.2.6. Evaluasi

Setelah model SVM dilatih, langkah selanjutnya adalah mengevaluasi kinerjanya menggunakan *confusion matrix*. *Confusion matrix* memberikan gambaran visual tentang bagaimana model mengklasifikasikan data, dengan menunjukkan jumlah prediksi yang benar dan salah dalam masing-masing kelas (misalnya, blas, hawar daun, dan tungro). Dalam matriks ini terdapat empat elemen utama: *True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN)*.

Confusion matrix membantu memahami kesalahan klasifikasi yang dilakukan oleh model. Dari matriks ini, kita juga dapat menghitung metrik-metrik evaluasi penting lainnya, seperti *precision, recall, F1-score* dan *Accuravy*. *Precision* mengukur seberapa tepat prediksi model untuk suatu kelas positif (misalnya, *blast*), sedangkan untuk mengukur seberapa baik model dalam mengidentifikasi semua kasus *true positive* menggunakan *recall*. Rata-rata harmonis antara *precision* dan *recall* adalah *F1-score*, memberikan gambaran umum tentang keseimbangan kinerja model dalam menangani kedua metrik tersebut. *Accuracy* untuk mengukur kemampuan hasil prediksi atau klasifikasi yang benar dari seluruh data yang diuji.