BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian yang digunakan berupa dataset *image* yang diperoleh melalui platform Kaggle yang terdiri dari kumpulan gambar daun mangga yang menunjukkan berbagai hama. Dataset yang disediakan berisi 4000 gambar yang mencakup delapan kategori daun mangga yang terkena hama. Dataset *image* ini akan digunakan sebagai data utama untuk mengklasifikasikan apakah suatu *image* menunjukkan adanya daun yang terkena hama atau tidak.

3.1.1 Waktu Penelitian

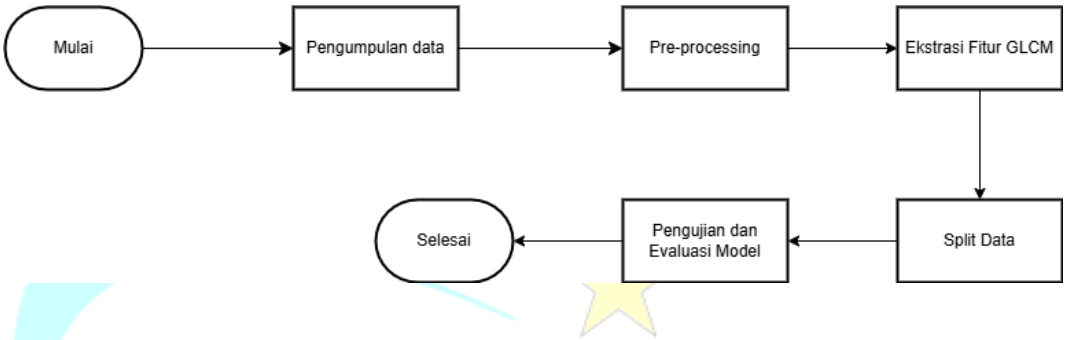
Penelitian ini akan dilaksanakan di Dinas Pertanian dan Ketahanan Pangan Kabupaten Karawang. Penelitian direncanakan berlangsung selama 3 bulan, dimulai pada bulan November 2024 hingga selesai. Proses penelitian akan melalui beberapa tahapan, mulai dari pengumpulan data hingga penyusunan laporan. Dibawah ini adalah rincian tabel waktu penelitian dapat dilihat pada Tabel 3.1

Tabel 3. 1 Waktu pelaksanaan

Keterangan	Waktu pelaksanaan											
	November				Desember				Januari			
	1	2	3	4	1	2	3	4	1	2	3	4
Pengumpulan Data												
Pre-processing Data												
Ekstrasi Fitur GLCM												
Split Data												
Pengujian dan Evaluasi model												
Laporan Penelitian												

3.1. Rancangan Penelitian

Penelitian ini akan dilaksanakan melalui beberapa tahapan yang sistematis dan berurutan, dimulai dari pengumpulan data, *pre-processing*, Ekstrasi Fitur, Split Data, Pengujian dan Evaluasi. Berikut ini adalah tahapan utama dari prosedur penelitian



Gambar 3.1 Rancangan Penelitian

3.1.1. Pengumpulan Data

Pengumpulan data merupakan langkah pertama yang sangat penting. Data yang digunakan dalam penelitian ini adalah citra daun mangga yang terkena berbagai jenis hama. Dataset yang digunakan terdiri dari gambar-gambar yang dilabeli secara jelas yaitu *Anthracnose*, *Bacterial Canker*, *Cutting Weevil*, *Die Back*, *Gall Midge*, *Healthy*, *Powdery Mildew*, dan *Sooty Mould*. Gambar-gambar ini berasal dari *platform Kaggle* yang telah dipilih untuk kualitas dan keberagaman gambar yang dimiliki. Gambar ini berisi 4000 gambar, link dataset dapat dibuka pada *platform kaggle* https://www.kaggle.com/datasets/aryashah2k/mango-leaf-disease-dataset/data . Gambar daun mangga yang terkena hama dapat ditunjukkan pada gambar 3.2.

(a) (b) (c) (d)

(e) (f) (g) (h)

Gambar 3.2 (a).*Antchracnose*, (b). *Bacterial Canker*, (c).*Cutting Weevil*, (d).*Die Back*, (e).*Gall Midge*, (f).*Healthy*,(g).*Powdery Mildew*, *dan* (h).*Sooty Mould*

Sumber: *Kaggle*

3.1.2. *Pre-Processing* Data

Data yang telah dikumpulkan akan melalui tahap *pre-processing* yang meliputi beberapa langkah penting untuk memastikan kualitas data yang digunakan dalam pelatihan model. Proses *pre-processing* ini bertujuan untuk mempersiapkan data sehingga model dapat memprosesnya dengan baik. Berikut adalah langkah-langkah yang akan dilakukan:

1. *Resize*

   Gambar-gambar dalam dataset akan diubah ukurannya menjadi seragam 100x100 piksel. Lalu, gambar ditampilkan gambar sebelum di *resize* dan setelah di *resize*. Berikut merupakan contoh *pseudocode* untuk *Resize* (mengubah ukuran gambar).

Resize

Mulai

#Langkah 1 : Ambil dataset yang akan dirubah pikselnya

Baca dataset yang berisi gambar dari file atau sumber lainnya

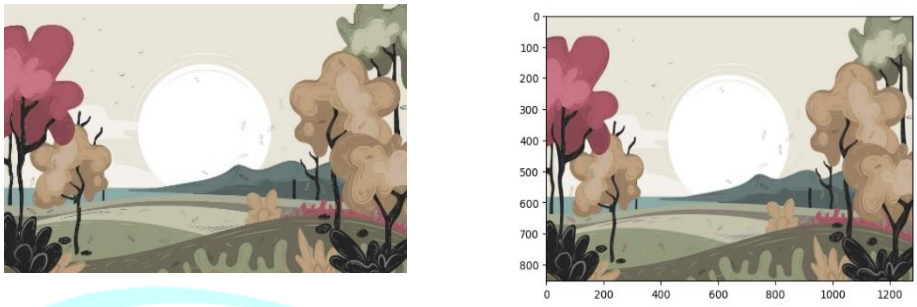
#Langkah 2: Merubah ukuran semua gambar menjadi 100 piksel

#Langkah 3: Menampilkan hasil gambar *resize*

Menampilkan beberapa gambar sebelum dilakukan *resize* dan sesudah dilakukan *resize*

Selesai

Hasil:



(a) (b)

Gambar 3.3 (a).Asli, (b).Hasil *resize*

2. Augmentasi Data

Teknik *augmentasi* data menggunakan *flipping.* Gambar ini akan dilakukan dengan teknik augmentasi citra dengan cara membalik gambar secara horizontal. Setelah dilakukan. Gambar asli, dan gambar horizontal akan ditampilkan. Berikut merupakan contoh *pseudocode* untuk *flipping* (membalikkan gambar).

Flipping

Mulai

#Langkah 1 : Ambil dataset yang akan dirubah

Baca dataset yang berisi gambar dari file atau sumber lainnya

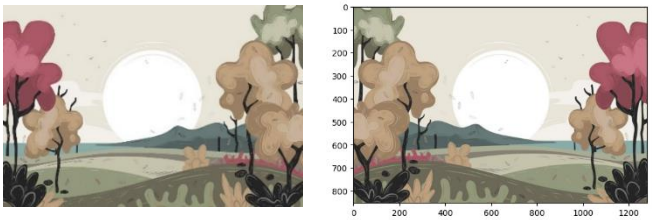
#Langkah 2: Merubah ukuran horizontal semua gambar dirubah dengan nilai 1

#Langkah 4: Menampilkan hasil gambar *flipping*

Menampilkan beberapa gambar sebelum dilakukan *flipping* dan sesudah dilakukan *flipping*

Selesai

Hasil:



(a) (b)

Gambar 3.4 (a).Asli, (b).Hasil horizontal

3.1.3. Ekstrasi Fitur *Gray Level Co-Occurrence Matrix*

Pada tahap ini gambar akan ditentukan arahnya horizontal jaraknya 1 piksel dan arahnya adalah 0° dengan level ke abu-abuannya 256. Berikut merupakan contoh *pseudocode* untuk tahapan GLCM

Gray Level Co-Occurrence Matrix

Mulai

#Langkah 1 : Ambil dataset yang akan dirubah

Baca dataset yang berisi gambar dari file atau sumber lainnya

#Langkah 2: Merubah ukuran horizontal semua gambar dirubah menjadi 0° dengan jarak 1 piksel

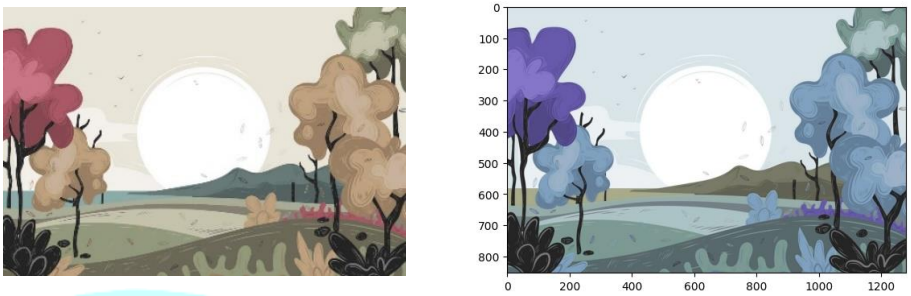
#Langkah 3: Merubah warna horizontal dengan nilai level ke abu-abu an 256 dengan jarak 1 piksel untuk semua gambar

#Langkah 4: Menampilkan hasil gambar dari ekstrasi fitur GLCM

Menampilkan beberapa gambar sebelum dilakukan ekstrasi fitur GLCM dan sesudah dilakukan ekstrasi fitur GLCM

Selesai

Hasil:



(a) (b)

Gambar 3.5 (a).Asli, (b).Hasil GLCM

3.1.4. Split Data

Tahap pembagian data dalam penelitian ini dilakukan dengan membagi dataset menjadi dua bagian utama, yaitu 80% untuk data latih (*training* data) dan 20% untuk data uji (*testing* data). Pembagian ini bertujuan untuk memastikan bahwa model dapat dilatih menggunakan sebagian data (data latih) dan diuji kinerjanya pada data yang tidak digunakan selama pelatihan (data uji). Proses pembagian dilakukan secara sistematis untuk menjaga pembagian data yang seimbang di setiap kelas dalam dataset.

3.1.5. Pengujian dan Evaluasi

Implementasi model pada penelitian ini menggunakan algoritma *Support Vector Machine* dan *K-Nearest Neighbor* untuk mengklasifikasikan daun mangga yang terkena hama berdasarkan tekstur daun

a. *Support Vector Machine*

Pada tahap ini SVM mencari *hyperplane* (garis pemisah) terbaik yang memisahkan kelas-kelas dalam data. *SVM* berusaha memaksimalkan margin antara kelas yang berbeda untuk memperoleh model yang generalisasi lebih baik. Berikut merupakan contoh *pseudocode* untuk tahapan SVM mencari *hyperplane*

Support Vector Machine

Mulai

#Langkah 1 : Ambil dataset baik data latih maupun data uji

Baca dataset baik data latih maupun data uji untuk melatih sebuah model,

misalnya untuk data latih (X_*train,* Y,*train*) dan untuk data uji (X_*test*, Y_*test*)

#Langkah 2: Menentukan jenis *kernel* yang akan digunakan oleh SVM:

Gambar akan ditentukan oleh *kernel RBF* untuk pemisah fungsi *Radial Basis Function*

#Langkah 3: Melatih model SVM menggunakan data latih (X_*train*, Y_*train*)

Model dilatih dengan mencari kurva terbaik (*hyperplane*), model menentukan data penting membantu membuat pemisah tersebut (*support vectors*)

#Langkah 4: Melatih model SVM menggunakan data uji (X_*test*, Y_*test*)

Menghitung posisi data terhadap garis pemisah (*hyperplane*), memprediksikan sebuah kelas

#Langkah 5: Mengevaluasi performa model

Membandingkan hasil prediksi dengan label sebenarmya (Y_*test*), menghitung *matrix* evaluasi (akurasi, presisi, *recall*, dan *f1-score* untuk setiap kelas

Selesai

b. *K-Nearest Neighbor*

Algoritma ini akan bekerja dengan cara mencari K data terdekat dengan data yang ingin diprediksi. Berdasarkan kelas mayoritas dari data tetangga terdekat tersebut, prediksi akan dibuat. Berikut merupakan contoh *pseudocode* untuk tahapan *K-Nearest Neighbor*

K-Nearest Neighbor

Mulai

#Langkah 1 : Ambil dataset baik data latih maupun data uji

Baca dataset baik data latih maupun data uji untuk label misalnya untuk data latih berlabel (X_*train,* Y,*train*), untuk fitur (X, *test)*

#Langkah 2: Memilih jumlah tetangga terdekat (K) misalnya K = 3

#Langkah 3: Menyiapkan data X_*test* dan X_*train*

Menggunakan data X_*test* dan X_*train* untuk dihitung jarak *euclidean*

#Langkah 4: Menentukan label untuk data uji

Melihat label dari K tetangga terdekat, memilih label yang paling sering muncul (mayoritas) sebagai prediksi

#Langkah 5: Menyimpan hasil prediksi dalam (X_*test*)

#Langkah 6: Mengevaluasi performa model

Membandingkan hasil prediksi dengan label sebenarmya (X_*test*), menghitung *matrix* evaluasi (akurasi, presisi, *recall*, dan *f1-score* untuk setiap kelas

Selesai

Setelah selesai proses pengujian dilanjutkan dengan tahap evaluasi yaitu menggunakan *Confusion Matrix.* Evaluasi ini akan digunakan untuk menghitung performa model yang diterapkan setelah tahap *modelling* selesai. *Confusion Matrix* menentukan performa model dengan nilai-nilai yang terkandung didalamnya. Yaitu,akan menggunakan akurasi presisi, *recall*, dan *f1-score.* Data yang telah di pisahkan kemudian di testing untuk menghitung akurasi, presisi, *recall*, dan *f1-score*. Untuk menghitung nilai tersebut menggunakan *Confusion Matrix* yang terdiri dari empat elemen utama yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN). Berikut untuk menghitung evaluasi:

a) **Akurasi (*Accuracy*):** Mengukur proporsi keseluruhan prediksi yang benar, baik untuk kelas positif maupun negatif.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

b) **Presisi (*Precision*):** Mengukur seberapa tepat model dalam memprediksi kasus positif.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

c) **Recall (*Sensitivity*):** Mengukur sejauh mana model mampu mengidentifikasi kasus positif.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

d) **F1-Score:** Merupakan rata-rata harmonis antara presisi dan recall, yang memberikan gambaran keseimbangan antara keduanya.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Setelah melewati rancangan tahapan pengumpulan data, *preprocessing*, ekstrasi fitur, split data, pengujian dan evaluasi, maka hasil pengujian klasifikasi daun mangga yang terkena hama akan dijelaskan dan dibahas lebih lanjut di bab selanjutnya.