

BAB 3

METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

3.1.1 DATASET

Penelitian bertujuan untuk mengklasifikasi hama pada daun mangga menggunakan citra digital dan diperoleh metode yang digunakan berupa SVM-RBF. Data yang telah digunakan pada saat proses merancang dan analisis berupa dataset dari *Mendeley* Data tahun 2020 dengan link *website* sebagai berikut <https://data.mendeley.com/datasets/94jf97jzc8/1> [9].

3.1.2 PERANGKAT KERAS (*HARDWARE*)

Perangkat keras yang digunakan saat proses penelitian ini yaitu menggunakan sebuah PC (*Personal Computer*) dengan spesifikasi sebagai berikut:

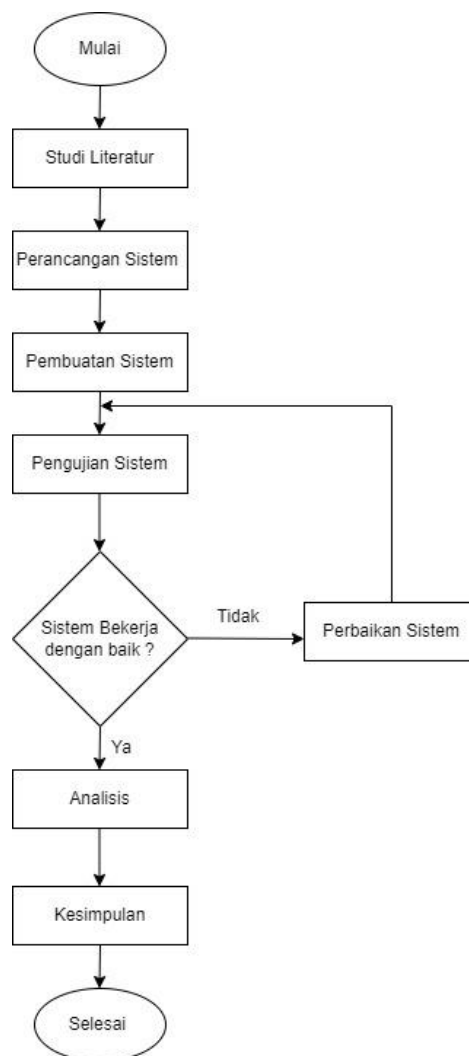
1. *Processor* Intel Core i3.
2. *Windows 10 Professional* (64 Bit).
3. RAM 4GB.

3.1.3 PERANGKAT LUNAK (*SOFTWARE*)

Software yang digunakan pada simulasi dalam menemukan hama yang menyerang didaun mangga menggunakan metode *Support Vector Machine* (SVM) adalah *Phyton*. Maka dari dikarenakan *software* yang akan di *install* bermasalah sehingga untuk uji coba penelitian ini menggunakan *Google Collab* dimana dapat diakses pada situs *website google*.

3.2 ALUR PENELITIAN

Penelitian dapat dilakukan menggunakan simulasi *Phyton* pada *google collab* dengan proses pengolahan citra hama daun mangga. Berikut adalah tahapan-tahapan alur yang dilakukan dalam penelitian, lalu pada Gambar 3.1 berikut dapat dilihat diagram alur penelitian:



Gambar 3.1 Alur Penelitian

Alur penelitian pada Gambar 3.1 diatas menjelaskan bahwa tahapan pertama penelitian yaitu mencari studi literatur guna mengetahui sumber referensi

pada penelitian dengan simulasi pemrograman *Python* untuk penelitian menggunakan metode SVM-RBF. Sistem yang akan dilakukan pada klasifikasi hama pada daun mangga menggunakan *Support Vector Machine* (SVM) dengan kernel *Radial Basis Function* (RBF). Klasifikasi berdasarkan ekstraksi fitur normalisasi. Perancangan sistem berupa pembuatan sistem saat mengolah data *input* pada saat penelitian serta dilakukan juga pengujian agar dapat berjalan dengan baik. Apabila pengujian telah berhasil maka dapat dibuat analisis serta kesimpulan hasil penelitian yang telah dilakukan.

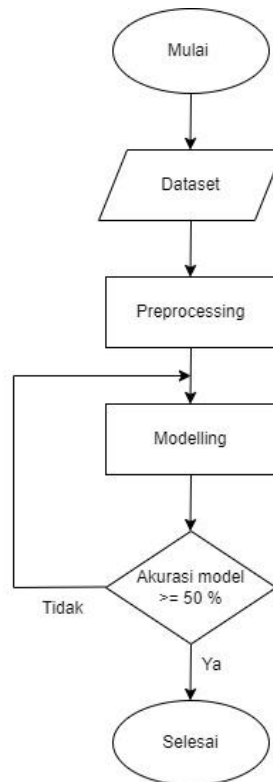
3.2.1 STUDI LITERATUR

Studi literatur dilakukan untuk mencari referensi yang akan dipakai dalam penulisan. Studi literatur ini didapatkan dari beberapa penelitian sebelumnya yang berasal dari beberapa sumber jurnal akurat sebagai pendukung penulisan penelitian ini. Setelah data secara keseluruhan sudah tersedia, maka penulis akan menganalisa data tersebut sehingga dapat ditarik dari suatu kesimpulan.

3.2.2 PERANCANGAN SISTEM

Penelitian ini dirancang dengan blok diagram agar dapat merancang suatu program dengan baik. Kemudian penelitian merupakan rangkuman dari ide serta permasalahan yang akan diselesaikan dengan tujuan merancang programs secara bertahap. Klasifikasi hama daun mangga ini cukup susah digolongkan. Selanjutnya hama ini dapat mengakibatkan kerusakan pada daun maupun penularan penyakit pada tanaman lainnya. Oleh karena itu dibutuhkan suatu sistem yang dapat untuk membantu manusia dalam mengklasifikasi hama daun mangga. Sistem ini nantinya akan mempertimbangkan beberapa kriteria agar dapat menghasilkan dengan baik. Klasifikasi hama pada daun mangga menggunakan *Support Vector Machine* (SVM) untuk 3 kelas serta 1 daun sehat dan 1 kernel berupa *Radial Basis Function* (RBF). Ekstraksi fitur yang digunakan

denga metode normalisasi. Selanjutnya pada Gambar 3.2 berikut dapat dilihat blok diagram perancangan sistem:



Gambar 3.2 Diagram Perancangan Sistem

Gambar 3.2 diatas ialah diagram perancangan sistem yang menggunakan dataset yang di dalamnya terdapat *training*, valid dan *testing*. Maka dari itu gambar tidak bisa diubah secara otomatis sehingga perlu diubah ke angka terlebih dahulu. Data *training* disini artinya untuk melakukan pembelajaran pada model, data valid untuk memprediksi data *training* dan data *testing* untuk melakukan *testing* dari model. Selanjutnya akan masuk ke dalam proses *processing*, dimana ada gambar yang harus diubah ke angka atau *array*. Setelah gambar tersebut diubah ke angka atau *array* akan dilakukan normalisasi artinya teknik untuk memperpendek jarak angka. Setelah itu, akan masuk kedalam modeling yang nantinya ada k-fold dan parameter. Parameter yang digunakan yaitu parameter C atau *Cost*, dengan menggunakan *confusion matrix*. Parameter dihasilkan tingkat akurasi, presisi, *recall* dan *f1-score* yang baik apabila model tersebut bagus. Sedangkan ketika tingkat akurasi yang dihasilkan kurang baik maka data tersebut

akan masuk ke modelling dan akan diolah lagi hingga akurasi tersebut bagus sekitar lebih dari 50%. Selanjutnya apabila akurasi sudah dinyatakan bagus maka program tersebut dinyatakan selesai.

3.2.3 PEMBUATAN SISTEM

Pembuatan sistem dilakukan dengan Bahasa Pemrograman Python yaitu suatu bahasa pemrograman tinggi sehingga bersifat *interactive*, *interpreter*, *object-oriented* dimana mampu beroperasi semua platform seperti *windows*, *linux* dan *mac*. Bahasa Pemrograman *Python* tercantum bahasa yang mudah dipelajari serta dipahami sebab memiliki sintaks cukup jelas. *Python* menggunakan *library* berupa *pandas* artinya bersifat *open source* di bahasa pemrograman untuk pemrosesan data serta analisis data.

3.2.3.1 INPUT DATASET

Penelitian ini menggunakan data gambar hama daun *apoderus_javanicus*, *aulacaspis_tubercularis*, *dialeuropora_decempuncta* dan normal. Data yang digunakan diperoleh dari *Mendeley* Dataset tahun 2020. Proses penginputan dataset pada *google collab* dapat dilakukan sebagai berikut:

```
load data
```

3.2.3.2 PREPROCESSING

Proses *preprocessing* ini menggunakan normalisasi. Normalisasi memiliki 2 fungsi *def* dimana *def load_image* sendiri artinya mendefinisikan suatu jalur file yang akan membuka gambar. Setelah itu semua ukuran gambar diubah menjadi ukuran 224 ke 224 dengan sisi panjang lebar sama. Proses fungsi pertama pada normalisasi data dilakukan dengan *script* sebagai berikut:

```
def load_images(file_paths):  
    images = []  
    for path in file_paths:
```

```
img = Image.open(path)
img = img.resize((224, 224)) # Resize the image if necessary
img = np.array(img)
img = img / 255
images.append(img)
return np.array(images)
```

Proses normalisasi fungsi kedua ini yaitu *def flatten_images*. Fungsi kedua ini dikonversikan ke *array* multi dimensi dimana mengubah yang sebelum gambar menjadi angka. Oleh karena itu proses *preprocessing* tersebut dilakukan sebelum ke proses *modelling*. Adapun *script* proses fungsi kedua normalisasi data sebagai berikut:

```
def flatten_images(images):
    return images.reshape(images.shape[0], -1)
```

3.2.3.3 MODELLING

Proses *modelling* berada di *svm.fit* dimana merupakan tahapan dari *modelling*. Sebelum masuk tahapan ini, *svm.fit* tidak dapat langsung masuk melalui proses *preprocessing* yang prosesnya hanya mengubah gambar ke angka. Apabila proses normalisasi tersebut sudah maka muncul gambar/225 atau angka seperti 0 1 15 yang masuk ke pendekatan 1 ke 0 sama dengan *scalling*. Proses ini dengan sis panjang dan lebar 3x3 dibuat menjadi 1 dimensi sampai bawah dipindah pada fungsi 1. Adapun *script* proses *modelling* seperti berikut ini:

```
Svm.fit(flattened_train_images, train_labels)
```

3.2.3.4 AKURASI MODEL

Akurasi model dilakukan untuk visualisasi akurasi perfoma model. Selanjutnya proses ini dilakukan guna melihat tingkat akurasi yang dihasilkan. Akurasi yang dihasilkan pada skenario dari 1 sampai 7 berbeda. Skenario 1 menghasilkan akurasi sekitar 57% sedangkan skenario 2 sampai dengan 10

menghasilkan akurasi 60%. Apabila akurasi model tersebut lebih dari 50% maka otomatis langsung selesai pada programnya. Sedangkan akurasi yang dihasilkan kurang dari 50% maka akan kembali ke proses *modelling* agar nantinya akurasi yang dihasilkan baik. Adapun *script* menentukan akurasi model sebagai berikut:

```
print(classification_report(test_labels, svm_predictions))
```

3.2.4 PENGUJIAN SISTEM

A. *HOLDOUT VALIDATION*

Holdout validation yang digunakan dalam penelitian ini dengan total data gambar yaitu 2800. Setelah itu data dilakukan pembagian menjadi 70% data *training* dan 30% data *testing*. Maka dari itu dibutuhkan *holdout validation* karena memiliki jumlah data yang cukup besar sehingga data tersebut dapat dibagi. *Holdout validation* pada program ini terdapat diproses *training* menggunakan parameter yang sudah ditentukan dan melakukan prediksi terhadap model. Adapun *script holdout validation* sebagai berikut:

```
svm.fit(flattened_train_images, train_labels)
svm_predictions = svm1.predict(flattened_test_images)
```

3.2.5 ANALISIS SISTEM

B. *CONFUSION MATRIX*

Berdasarkan hasil dari *Holdout validation* akan diproses dengan *Confusion Matrix* yang terdapat dua kelas atau lebih. *Confusion Matrix* menggunakan metode sebuah informasi tentang klasifikasi yang digunakan untuk mengukur perhitungan akurasi di algoritma. Pemodelan yang telah dibuat nantinya dievaluasi dengan melihat beberapa kinerja dari algoritma, serta diukur pada tingkat akurasi yang sudah dihasilkan di data *testing*. Berikut merupakan perhitungan akurasi 4x4 :

$$\text{Akurasi} = \frac{T_{AA}+T_{BB}+T_{CC}+T_{NN}}{T_{AA}+T_{BA}+T_{CA}+T_{NA}+T_{AB}+T_{BB}+T_{CB}+T_{NB}+T_{AC}+T_{BC}+T_{CC}+T_{NC}+T_{AN}+T_{BN}+T_{CN}+T_{NN}} \times 100\% \quad (3.1)$$

Berikut ini adalah Tabel 3.1 *Multiple Confusion Matrix 4x4* :

Tabel 3.1 Confusion Matrix 4x4

		Hasil Prediksi			
		A (<i>Apoderus Javanicus</i>)	B (<i>Aulacaspis Tubercularis</i>)	C (<i>Dialeuropora Decempuncta</i>)	N (Normal)
Hasil Aktual	A (<i>Apoderus Javanicus</i>)	T_{AA}	T_{BA}	T_{CA}	T_{NA}
	B (<i>Aulacaspis Tubercularis</i>)	T_{AB}	T_{BB}	T_{CB}	T_{NB}
	C (<i>Dialeuropora Decempuncta</i>)	T_{AC}	T_{BC}	T_{CC}	T_{NC}
	N (Normal)	T_{AN}	T_{BN}	T_{CN}	T_{NN}

Tabel 3.1 diatas merupakan *confusion matrix 4x4*. Kemudian dijelaskan memiliki empat kelas yang mana sudah diketahui rumus-rumusny. Kelasnya sendiri terdiri dari kelas hama daun. Kategori kelas yaitu *apoderus javanicus*, *aulacaspis tubercularis*, *dialeuropora decempuncta* dan normal. Rumus tersebut diantaranya terdapat akurasi, presisi (*precision*), *recall* dan *f1-score*. Oleh karena itu tabel *confusion matrix 4x4* ini guna perfoma model.

- Presisi (*Precision*)

Presisi (*precision*) dimana menampilkan nilai akurasi dari data yang akan diminta dengan hasil prediksi. Berikut merupakan persamaan perhitungan *precision confusion matrix*:

$$\text{Precision} = \frac{(T_{AA}+T_{BB}+T_{CC}+T_{NN})}{(T_{AA}+T_{BB}+T_{CC}+T_{NN})+(T_{AB}+T_{AC}+T_{AN}+T_{BA}+T_{BC}+T_{BN}+T_{CA}+T_{CB}+T_{CN}+T_{NA}+T_{NB}+T_{NC})} \quad (3.2)$$

- *Recall*

Recall dalam *confusion matrix* ini diartikan ketika data tersebut tidak dapat diprediksi dengan benar. Berikut merupakan persamaan perhitungan *recall*:

$$Recall = \frac{(TAA+TBB+TCC+TNN)}{(TAA+TBB+TCC+TNN)+(TBA+TCA+TNA+TAB+TCB+TNB+TAC+TBC+TNC+TAN+TBN+TCN)} \quad (3.3)$$

- *F1-Score*

F1-score pada *confusion matrix* dikatakan sebagai perbandingan rata-rata dari *recall* dan *precision*. Berikut merupakan rumus perhitungan dari *f1-score*:

$$F1-Score = \frac{2x (Recall \times Precision)}{(Recall+Precision)} \quad (3.4)$$

Script ini dijelaskan untuk membuat visualisasi perfoma model dalam bentuk *confusion matrix*. Setelah itu variabel disini menggunakan variabel *cm*, dimana terdapat fungsi *confusion matrix*. Selanjutnya *confusion matrix* akan mengevaluasi akurasi yang telah dihasilkan pada klasifikasi. Adapun *script confusion matrix* sebagai berikut:

```
cm = confusion_matrix(test_labels, svm_predictions)
```