

## BAB 3

### METODE PENELITIAN

#### 3.1 ALAT YANG DIGUNAKAN

Alat yang digunakan dalam penelitian ini terdiri dari dataset, perangkat keras (*hardware*), dan perangkat lunak (*software*). Berikut adalah rincian dari alat yang digunakan.

##### 3.1.1 Dataset

Dataset yang digunakan berasal dari penelitian [14] dengan nama *Tea Sickness Dataset*. Penggunaan dataset dari situs Mendelay Data dikarenakan dataset tersebut digunakan untuk penelitian tentang klasifikasi penyakit daun teh. Dataset berisi citra penyakit daun teh yang digunakan sebagai proses klasifikasi. Dari dataset tersebut terdapat delapan jumlah kelas jenis penyakit daun teh. Dataset ini berjumlah 800 dengan masing-masing kelas memiliki 100 citra. Citra penyakit daun teh tersebut terbagi menjadi delapan kelas yaitu *algal leaf spot*, *anthracnose*, *bird eye spot*, *brown blight*, *gray blight*, *red leaf spot*, *white leaf spot*, dan *healthy*.

##### 3.1.2 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini yaitu sebuah laptop. Adapun spesifikasi laptop yang digunakan yaitu pada Tabel 3.1 sebagai berikut:

**Tabel 3. 1 Spesifikasi laptop.**

No	Detail	Spesifikasi
1	Merek ( <i>tipe</i> )	HP 15S-DU3577TU
2	<i>Prosesor</i>	11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz 3.00 GHz
3	<i>Installed memory / RAM</i>	8 GB

No	Detail	Spesifikasi
4	<i>Tipe system</i>	64-bit
5	<i>Operating System/OS</i>	Windows 11 <i>Home</i>
6	VGA	Intel UHD Graphics

Pada penelitian ini dibutuhkan bantuan laptop untuk menjalankan *google colab* dengan spesifikasi seperti Tabel 3.1. Spesifikasi dari laptop juga berpengaruh terhadap waktu pemrosesan atau running program dalam *google colab*. Sehingga dengan menggunakan spesifikasi laptop yang cukup baik akan mempercepat pemrosesan datanya. Namun jika menggunakan spesifikasi laptop yang minimal dengan data yang banyak, maka proses *training* data akan berjalan lebih lama. Untuk *operating system* yang digunakan pada penelitian ini adalah *Windows 11*.

### 3.1.3 Perangkat Lunak (*Software*)

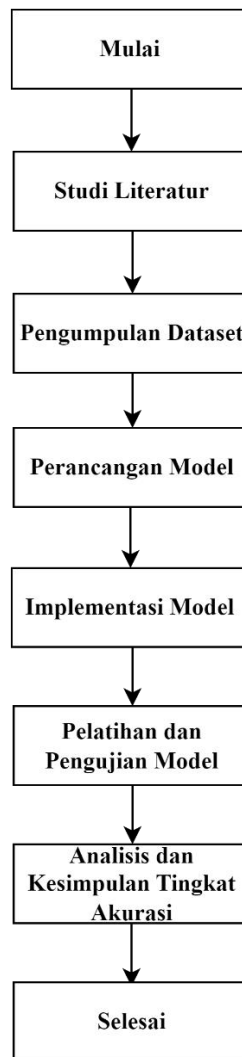
Perangkat lunak yang digunakan pada penelitian ini yaitu sebagai berikut :

**Tabel 3. 2 Perangkat lunak yang digunakan.**

No	Nama <i>Software</i>
1	<i>Google Colaboratory</i>
2	Bahasa Pemrograman Python 3.11
3	<i>Library : numpy, os, tensorflow, keras, matplotlib, seaborn, glob, sklearn.</i>

## 3.2 ALUR PENELITIAN

Alur penelitian klasifikasi penyakit daun teh pada Gambar 3.1 dimulai dari studi literatur, pengumpulan dataset, perancangan model, implementasi model, pengujian model, dan analisis serta kesimpulan. Adapun diagram alir dari penelitian ini disajikan pada gambar sebagai berikut.



**Gambar 3. 1 Diagram alur penelitian.**

### **3.2.1 Studi Literatur**

Alur penelitian yang pertama dilakukan dalam penelitian ini adalah melakukan studi literatur. Studi literatur dilakukan dengan mengacu pada jurnal, buku, dan sumber online terkait *deep learning*, klasifikasi penyakit tanaman, dan CNN. Studi literatur ini berfungsi sebagai referensi untuk mendapatkan pemahaman dan informasi yang terkait dengan penelitian sebelumnya, serta sebagai sumber referensi untuk penelitian yang sedang dilakukan.

### **3.2.2 Pengumpulan Dataset**

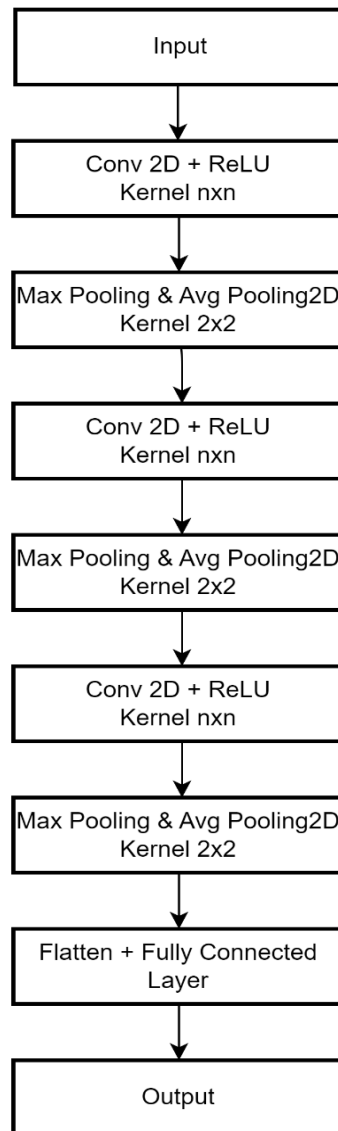
Pengumpulan dataset dilakukan dengan mengamati dataset yang telah tersedia di internet, yang sebelumnya telah diuji dengan metode berbeda tetapi

pada objek yang sama. Pengumpulan dataset berupa penyakit pada daun teh yang nantinya digunakan sebagai ekstraksi fitur citra digital sebagai data untuk klasifikasi. Pengumpulan dataset dilakukan melalui situs Mendeley Data yang dapat diakses melalui penelitian [14]. Citra yang digunakan adalah citra tunggal posisi diam. Dataset terdiri dari 800 citra penyakit daun teh dalam format JPG. Contoh citra dari jenis penyakit daun teh ditunjukkan dari Gambar 2.7 sampai Gambar 2.13. Berikut merupakan kode program pada proses *import dataset*.

```
from google.colab import drive #untuk menghubungkan ke drive
drive.mount('/content/gdrive')
algaleafspot = []
anthracnose = []
birdeyespot = []
brownblight = []
grayblight = []
healthy = []
redleafspot = []
whiteleafspot = []
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/algalleafspot/*'):
    algaleafspot.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/anthracnose/*'):
    anthracnose.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/birdeyespot/*'):
    birdeyespot.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/brownblight/*'):
    brownblight.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/grayblight/*'):
    grayblight.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/healthy/*'):
    healthy.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/redleafspot/*'):
    redleafspot.append(img)
for img in glob.glob('/content/gdrive/MyDrive/Tea Sickness
Dataset/whiteleafspot/*'):
    whiteleafspot.append(img)
```

### 3.2.3 Perancangan Model

Pada penelitian ini akan membahas terkait klasifikasi penyakit daun teh menggunakan metode *convolutional neural network* (CNN) yang mampu mendeteksi fitur secara otomatis dari data yang diberikan. Metode ini memiliki dua tahapan yang akan dilakukan. Tahap pertama adalah *feature learning*, yang terdiri dari dua lapisan utama yaitu lapisan konvolusi (*convolution layer*) dan *pooling layer*. Setelah melewati lapisan utama tersebut, digunakan fungsi aktivasi ReLU sebagai fungsi aktivasi dalam penelitian ini. Tahapan kedua yaitu *classification*, pada tahapan ini terdapat *layer fully connected* dengan fungsi aktivasi yang digunakan yaitu *softmax*.



**Gambar 3. 2 Rancangan model CNN.**

Pada Gambar 3.2 rancangan model CNN penelitian kali ini yaitu dimulai dengan input dataset selanjutnya menggunakan 3 *convolutional layer*, 3 *layer ReLU* dan *pooling layer* yang digunakan yaitu sebanyak tiga layer. Selanjutnya, proses klasifikasi dilakukan pada lapisan *fully connected*, dan hasil *output* dari metode CNN adalah tingkat akurasi klasifikasi.

Pada proses *convolutional layer* ini menggunakan *kernel size* yang memiliki ukuran lebar  $n$  tinggi  $n$  dengan  $n$  merupakan nilai pada kernel yang akan diskenariokan. *Kernel* yang akan digunakan adalah memvariasikan *kernel size* 1x1, 2x2, 3x3, 4x4, dan 5x5 yang mana akan bergeser ke semua bagian piksel citra sesuai yang di *kernel size* inputkan, pergeseran yang terjadi tersebut yang disebut sebagai *stride*. Pada metode ini, lapisan konvolusi yang digunakan adalah Conv2D yang digunakan untuk menjalankan operasi konvolusional pada gambar latih. Setelah proses operasi konvolusi selesai diperoleh *output* yang disebut sebagai *activation maps* atau *features maps*. Untuk membantu pengolahan selanjutnya, hasil ini akan melewati fungsi aktivasi ReLU. Fungsi ini akan mengubah bilangan atau *output* yang bernilai negatif menjadi 0, sementara *output* yang bernilai positif akan mempertahankan nilai aktivasi yang diperoleh.

Pada proses *pooling layer* dalam penelitian ini menggunakan variasi *pooling* yaitu dengan membandingkan 2 variasi *pooling* yang ada yaitu *maximum pooling* dan *average pooling*. Dengan menggunakan *maximum pooling* dan *average pooling* ini maka setiap *output* dari lapisan konvolusi akan dibagi menjadi beberapa grid kecil dan kemudian mengambil nilai maximum untuk *maximum pooling* dan mengambil nilai rata rata untuk *average pooling*. Dari setiap grid tersebut kemudian disusun menjadi matriks. *Pooling* ini menggunakan *kernel* berukuran 2x2 yang artinya langkah akan bergeser sebanyak 2 dan bersifat konstanta.

Selanjutnya pada proses klasifikasi terdapat *flatten* yang digunakan sebagai operasi yang mengubah *array* multidimensi menjadi *array* satu dimensi. Selanjutnya terdapat *fully connected layer* dengan menerapkan metode *dense* untuk menghubungkan setiap neuronnya dengan neuron di layer sebelumnya dan layer setelahnya. Kemudian dilanjut menuju proses klasifikasi dengan menggunakan bantuan aktivasi *softmax* karena jumlah kelas sebanyak 8 kelas.

### 3.2.4 Implementasi Model

Pada model yang sudah dirancang akan diimplementasikan ke dalam *google colab* dengan menggunakan bahasa pemrograman *python*. Penelitian ini menggunakan metode *Conv2D*. Selanjutnya akan dilakukan beberapa tahapan dalam metode ini supaya mendapatkan hasil dari inputan yang diberikan. Penelitian ini melakukan skenario percobaan dengan memvariasikan *kernel size* pada lapisan konvolusi dan membandingkan 2 jenis *pooling layer*.

**Tabel 3. 3 Parameter model CNN.**

Parameter yang Digunakan	Konfigurasi <i>Max Pooling</i>	Konfigurasi <i>Avg Pooling</i>
Jumlah Filter	16,32,64	16,32,64
<i>Kernel Size</i>	1x1 - 5x5	1x1 - 5x5
<i>Pooling</i>	<i>Maximum Pooling</i>	<i>Average pooling</i>
Dense	128	128
Fungsi Aktivasi	<i>ReLu + Softmax</i>	<i>ReLu + Softmax</i>
<i>Neuron Output Layer</i>	8	8
Optimasi MPL	Adam	Adam

Pada Tabel 3.3 dapat diketahui perbedaan parameter dengan membandingkan *maximum pooling* dan *average pooling*. Dimana masing-masing jenis pooling akan melakukan skenario pada parameter *kernel size* yaitu 1x1, 2x2, 3x3, 4x4, dan 5x5. Selanjutnya untuk parameter lainnya yang digunakan sebagai berikut. Jumlah *filter* merupakan nilai *filter* yang digunakan pada inputan sebesar 16, 32, 64, sehingga untuk hasil operasi konvolusi yang dilakukan sebanyak jumlah *filter* yang di inputkan. *Kernel size* merupakan dimensi *kernel* yang digunakan untuk proses konvolusi sebesar yang diinputkan. *Pooling layer* berfungsi untuk mengurangi ukuran spasial dengan cara memilih *pixel* yang lebih penting, *maximum pooling* memilih *pixel* dengan cara menggunakan nilai *pixel* terbesar pada *pixel* sebelumnya sedangkan *average pooling* mengambil nilai rata-rata pada *pixel*. *Dense* berfungsi untuk menambahkan layer pada *fully connected layer* yang berupa neuron sebesar 128. *ReLu* merupakan fungsi aktivasi yang

berfungsi mengubah nilai negatif pada filter menjadi nol. *Neuron output layer* merupakan jumlah kelas yang digunakan pada klasifikasi yaitu sebanyak 8 kelas. Adam merupakan metode menghitung *learning rate* untuk parameter yang berbeda. Setiap *kernel size* pada proses konvolusi dan perbedaan penggunaan *pooling layer* dilakukan untuk mengetahui pengaruhnya terhadap nilai akurasi klasifikasi pada penyakit daun teh. Berikut kode program yang digunakan pada implementasi model.

```
#Mendefinisikan Model CNN Max Pooling
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras.models import Sequential

# membuat objek model
model = Sequential()

# layer konvolusi 1
model.add(Conv2D(16, (3, 3), activation='relu',
input_shape=(150, 150, 3)))
# layer pooling
model.add(MaxPooling2D(pool_size=(2, 2)))
# layer konvolusi 2
model.add(Conv2D(32, (3, 3), activation='relu'))
# layer pooling
model.add(MaxPooling2D(pool_size=(2, 2)))
# layer konvolusi 3
model.add(Conv2D(64, (3, 3), activation='relu'))
# layer pooling
model.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening
model.add(Flatten())
# Full connection
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dense(units = 8, activation = 'softmax'))

# Compiling the CNN
model.compile(loss = 'categorical_crossentropy', optimizer =
'Adam', metrics = ['accuracy'])

# summary dari model
model.summary()
```

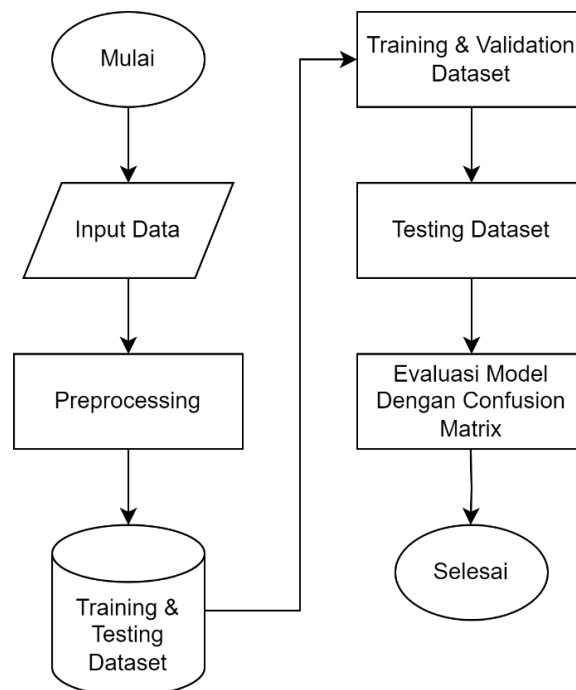


### 3.2.5 Pelatihan dan Pengujian Model

**Tabel 3. 4 Pembagian dataset.**

Data	Pembagian	Jumlah Dataset	Total Dataset
<i>Training</i>	70%	560	800
<i>Validation</i>	10%	80	
<i>Testing</i>	20%	160	

Pada tahapan pembagian data dalam model klasifikasi ini menggunakan *hold out cross validation*. Dalam proses ini, data akan dibagi menjadi tiga bagian yaitu *Training*, *Validation*, dan *Testing*. Data *training* digunakan untuk melatih model, dan selama proses pelatihan model mencoba mempelajari pola dalam data sehingga dapat membuat prediksi yang baik. Data *validation* digunakan untuk mengukur kinerja model selama pelatihan dan menentukan parameter terbaik untuk model. Data *testing* digunakan untuk menguji performa akhir model setelah model dilatih dan memberikan perkiraan akurasi yang lebih realistis pada data yang belum pernah dilihat sebelumnya. Pembagian data pada Tabel 3.4 dilakukan dengan persentase pembagian masing-masing sebesar 70% untuk data *Training*, 10% untuk data *Validation*, dan 20% untuk data *Testing*.



**Gambar 3. 3 Flowchart pelatihan dan pengujian model.**

Pada Gambar 3.3 merupakan *flowchart* pelatihan dan pengujian model dimana terdapat dua bagian yaitu pelatihan dan pengujian. Proses pelatihan dan pengujian data dimulai dengan membagi antara data *training* dan *validation* melalui proses split dataset kemudian dilakukan pengujian. Data training akan melalui proses ekstrasi fitur kemudian dimasukkan kedalam model CNN yang sudah dibuat dan model dilatih agar dapat memperoleh hasilnya. Begitu juga untuk yang pengujian data berarti digunakan 20% dari total dataset sebagai data *testing*. Berikut kode program pelatihan dan pengujian pada model.

```
#train the model

epochs=75
history = model.fit(trainDataset, epochs=epochs,
validation_data=valDataset)

#visualize the model's training performance
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()

#visualize the model's training performance
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()

#evaluate the model on the test dataset
loss, acc = model.evaluate(testDataset)
print('Loss:', loss)
print('Accuracy:', acc)
```

### 3.2.6 Analisis dan Kesimpulan Tingkat Akurasi

Pada tahap ini, akan dilakukan proses *Confusion matrix* untuk membantu dalam mendapatkan nilai akurasi dari model yang telah dibuat sebelumnya. Nilai

akurasi yang didapatkan akan ditampilkan dalam sebuah tabel *confusion matrix* dan perhitungan *confusion matrix* pada klasifikasi *multiclass* untuk mendapatkan sebuah nilai akurasi. Berikut kode program untuk menampilkan hasil akurasi.

```

y_pred = model.predict(imagestotal)
y_pred = np.argmax(y_pred, axis=1)
y_pred

labelstotal
labelstotalint = [np.where(r==1)[0][0] for r in labelstotal]

cm = confusion_matrix(labelstotalint, y_pred )

sns.heatmap(cm, cmap="Reds",annot=True,
cbar_kws={"orientation":"vertical","label":"warna grafik"})
plt.xlabel('Predicted Label', color='black')
plt.ylabel('True Label', color='black')
plt.gcf().axes[0].tick_params(color='white')
plt.gcf().axes[1].tick_params(color='black')
plt.gcf().set_size_inches(10,6)
plt.show()

```

Selanjutnya tahap terakhir dalam penelitian ini yaitu akan didapatkan hasil persentase akurasi dari model CNN yang telah dibuat. Hasil tersebut akan digunakan sebagai dasar untuk menyimpulkan hasil dari penelitian yang telah dilakukan. Berikut tabel *confusion matrix* klasifikasi jenis penyakit pada daun teh.

**Tabel 3. 5 Confusion matrix pada klasifikasi penyakit daun teh.**

		Predicted Clasification							
		Class	Algal Leaf Spot	Anthra cnose	Bird Eye Spot	Brown Blight	Gray Blight	Healthy	Red Leaf Spot
Actual Clasification	Algal Leaf Spot	TALS_ALS	FAC_ALS	FBES_ALS	FBB_ALS	FGB_ALS	FHT_ALS	FRLS_ALS	FWLS_ALS
	Anthra Cnose	FALS_AC	TAC_AC	FBES_AC	FBB_AC	FGB_AC	FHT_AC	FRLS_AC	FWLS_AC
	Bird Eye Spot	FALS_BES	FAC_BES	TBES_BES	FBB_BES	FGB_BES	FHT_BES	FRLS_BES	FWLS_BES
	Brown Blight	FALS_BB	FAC_BB	FBES_BB	TBB_BB	FGB_BB	FHT_BB	FRLS_BB	FWLS_BB

<i>Gray Blight</i>	F <sub>ALS_GB</sub>	F <sub>AC_GB</sub>	F <sub>BES_GB</sub>	F <sub>BB_GB</sub>	T <sub>GB_GB</sub>	F <sub>HT_GB</sub>	F <sub>RLS_GB</sub>	F <sub>WLS_GB</sub>
<i>Healthy</i>	F <sub>ALS_HT</sub>	F <sub>AC_HT</sub>	F <sub>BES_HT</sub>	F <sub>BB_HT</sub>	F <sub>GB_HT</sub>	T <sub>HT_HT</sub>	F <sub>RLS_HT</sub>	F <sub>WLS_HT</sub>
<i>Red Leaf Spot</i>	F <sub>ALS_RLS</sub>	F <sub>AC_RLS</sub>	F <sub>BES_RLS</sub>	F <sub>BB_RLS</sub>	F <sub>GB_RLS</sub>	F <sub>RLS_RWS</sub>	T <sub>RLS_RLS</sub>	F <sub>WLS_RLS</sub>
<i>White Leaf Spot</i>	F <sub>ALS_WLS</sub>	F <sub>AC_WLS</sub>	F <sub>BES_WLS</sub>	F <sub>BB_WLS</sub>	F <sub>GB_WLS</sub>	F <sub>RLS_WLS</sub>	F <sub>WLS_WLS</sub>	T <sub>WLS_WLS</sub>

Adapun persamaan menghitung akurasi dari klasifikasi penyakit daun teh dengan menggunakan *confusion matrix* adalah sebagai berikut :

$$\text{Akurasi} = \frac{T_{ALS\_ALS} + T_{AC\_AC} + T_{BES\_BES} + T_{BB\_BB} + T_{GB\_GB} + T_{HT\_HT} + T_{RLS\_RLS} + T_{WLS\_WLS}}{\text{Jumlah seluruh data}} \times 100 \% \quad (3.1)$$

Keterangan :

T<sub>ALS\_ALS</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Algal Leaf Spot*

F<sub>AC\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthracnose* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>BES\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>BB\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>GB\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>HT\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>RLS\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>WLS\_ALS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Algal Leaf Spot*.

F<sub>ALS\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Anthracnose*.

- T<sub>AC\_AC</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Anthraco*se dan kelas sebenarnya *Anthraco*se.
- F<sub>BES\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Anthraco*se.
- F<sub>BB\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Anthraco*se.
- F<sub>GB\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Anthraco*se.
- F<sub>HT\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Anthraco*se.
- F<sub>RLS\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Anthraco*se.
- F<sub>WLS\_AC</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Anthraco*se.
- F<sub>ALS\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>AC\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthraco*se dan kelas sebenarnya *Bird Eye Spot*.
- T<sub>BES\_BES</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>BB\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>GB\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>HT\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>RLS\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>WLS\_BES</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Bird Eye Spot*.
- F<sub>ALS\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Brown Blight*.

- F<sub>AC\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthracnose* dan kelas sebenarnya *Brown Blight*.
- F<sub>BES\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Brown Blight*.
- T<sub>BB\_BB</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Brown Blight*.
- F<sub>GB\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Brown Blight*.
- F<sub>HT\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Brown Blight*.
- F<sub>RLS\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Brown Blight*.
- F<sub>WLS\_BB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Brown Blight*.
- F<sub>ALS\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Gray Blight*.
- F<sub>AC\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthracnose* dan kelas sebenarnya *Gray Blight*.
- F<sub>BES\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Gray Blight*.
- F<sub>BB\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Gray Blight*.
- T<sub>GB\_GB</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Gray Blight*.
- F<sub>HT\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Gray Blight*.
- F<sub>RLS\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Gray Blight*.
- F<sub>WLS\_GB</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Gray Blight*.
- F<sub>ALS\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Healthy*.

- F<sub>AC\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthracnose* dan kelas sebenarnya *Healthy*.
- F<sub>BES\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Healthy*.
- F<sub>BB\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Healthy*.
- F<sub>GB\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Healthy*.
- T<sub>HT\_HT</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Healthy*.
- F<sub>RLS\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Healthy*.
- F<sub>WLS\_HT</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Healthy*.
- F<sub>ALS\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>AC\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthracnose* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>BES\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>BB\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>GB\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>HT\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *Red Leaf Spot*.
- T<sub>RLS\_RLS</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>WLS\_RLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *Red Leaf Spot*.
- F<sub>ALS\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Algal Leaf Spot* dan kelas sebenarnya *White Leaf Spot*.

- F<sub>AC\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Anthraco*se dan kelas sebenarnya *White Leaf Spot*.
- F<sub>BES\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Bird Eye Spot* dan kelas sebenarnya *White Leaf Spot*.
- F<sub>BB\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Brown Blight* dan kelas sebenarnya *White Leaf Spot*.
- F<sub>GB\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Gray Blight* dan kelas sebenarnya *White Leaf Spot*.
- F<sub>HHT\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Healthy* dan kelas sebenarnya *White Leaf Spot*.
- F<sub>RSL\_WLS</sub> : Kondisi dimana hasil prediksi salah. Hasil prediksi adalah *Red Leaf Spot* dan kelas sebenarnya *White Leaf Spot*.
- T<sub>WLS\_WLS</sub> : Kondisi dimana hasil prediksi benar. Hasil prediksi adalah *White Leaf Spot* dan kelas sebenarnya *White Leaf Spot*.