

BAB 3

METODE PENELITIAN

3.1 PEMODELAN SISTEM

Penelitian ini menggunakan suatu metode dalam mengklasifikasi penyakit daun kapas menggunakan *Convolutional Neural Network* dengan arsitektur *Alexnet*. Klasifikasi penyakit daun kapas dibagi menjadi empat kelas, yaitu *Bacterial Blight*, *Curl Virus*, *Fusarium Wilt*, dan sehat.

3.1.1 Dataset

Penelitian ini menggunakan dataset yang diperoleh dari situs *Kaggle* pada tahun 2021 [4]. Penggunaan dataset dari *Kaggle* dikarenakan dataset tersebut digunakan dalam penelitian tentang klasifikasi penyakit daun kapas. Dataset berisi total 1.710 citra penyakit pada daun kapas. Citra penyakit daun kapas tersebut terbagi menjadi empat kelas yaitu *bacterial blight*, *curl virus*, *fusarium wilt*, dan *healthy*.

3.1.2 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini yaitu sebuah *laptop* dengan spesifikasi sebagai berikut:

1. Intel® Core™ i5 5600U dual-core 2.3 GHz up to 2.8 GHz
2. Windows 10 (64-Bit)
3. Ram 8 GB
4. SSD 512 GB
5. GPU NVIDIA GeForce GTX 930MX

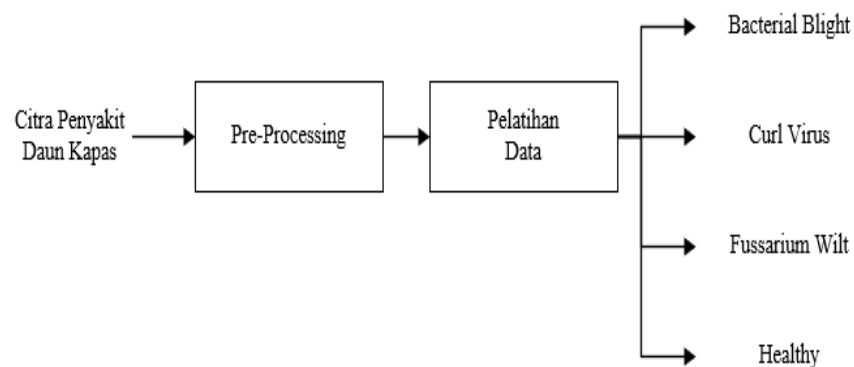
3.1.3 Perangkat Lunak (*Software*)

Software yang digunakan dalam simulasi dan analisis citra penyakit pada daun kapas adalah *Google Colab* dengan menggunakan bahasa pemrograman *Python*. *Software* ini banyak digunakan dalam penelitian dibidang pengembangan sistem, desain sistem, dan lain-lain. *Google Colab* merupakan *software* yang digunakan membuat dan menampilkan hasil

kinerja sistem klasifikasi penyakit pada daun kapas menggunakan *Convolutional Neural Network* dengan arsitektur *Alexnet*.

3.1.4 Perancangan Sistem

Perancangan sistem melalui beberapa tahapan yaitu data masukan yang berupa citra penyakit daun kapas. Citra penyakit daun kapas tersebut kemudian dilakukan tahap *preprocessing*, pada tahap ini dilakukan *proses resize* yang digunakan untuk menyamakan ukuran citra penyakit pada daun kapas. Data citra penyakit pada daun kapas dilatih agar dapat menghasilkan klasifikasi dengan tepat. Sistem diharuskan dapat mengklasifikasikan data citra penyakit daun kapas sesuai dengan kelas yang ditentukan, yaitu *bacterial blight*, *curl virus*, *fusarium wilt*, dan *healthy*.



Gambar 3.1 Diagram Blok Perancangan Sistem

1. *Input* Citra Penyakit Daun Kapas

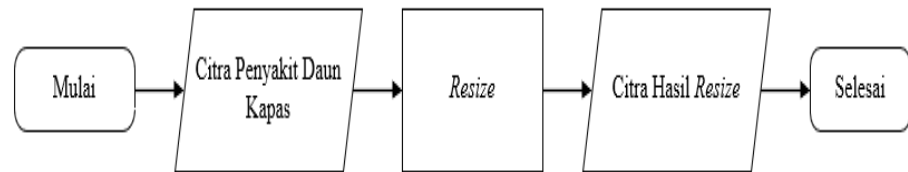


Gambar 3.2 *Flowchart* Input Citra Penyakit Daun Kapas

Proses input citra penyakit daun kapas dimulai dari mengunduh dari situs *Kaggle.com* seperti pada Gambar 3.2. Citra penyakit daun kapas yang digunakan adalah *bacterial blight*, *curl virus*, *fusarium wilt*, dan

healthy yang diperoleh dari *Kaggle*. Banyak citra yang digunakan sebanyak 1.710 data dengan format JPG.

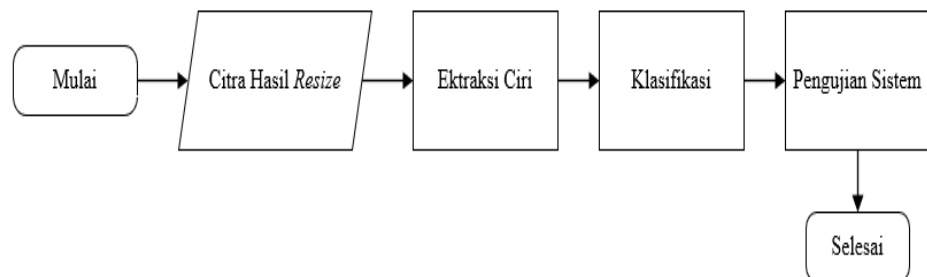
2. Preprocessing Data



Gambar 3.3 *Flowchart Preprocessing Data*

Preprocessing terdiri dari proses *resize*, seperti yang ditunjukkan pada Gambar 3.3. Dataset yang memiliki ukuran berbeda disamakan agar memiliki ukuran citra yang sama, yaitu menjadi 32x32, 64x64, 128x128, 256x256 piksel. Ukuran citra disamakan agar memudahkan sistem pada tahap selanjutnya karena memiliki ukuran citra yang sama.

3. Pelatihan Data

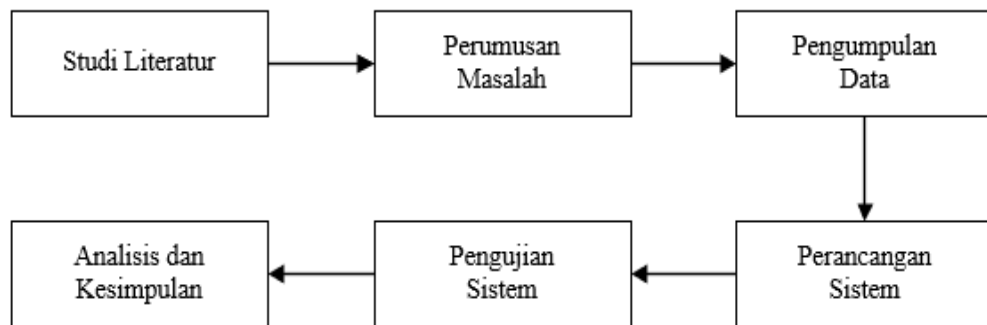


Gambar 3.4 *Flowchart Pelatihan Data*

Pada proses pelatihan data yang ditunjukkan pada Gambar 3.4, masukan terdiri dari 4 kelas citra penyakit daun kapas, yaitu *bacterial blight*, *curl virus*, *fusarium wilt*, dan *healthy*. Data citra masukan melewati proses *resizing* agar memiliki ukuran citra yang sama. Pada tahap klasifikasi, terdapat *fully connected layer* dan aktivasi *softmax* yang akan membangun jumlah keluaran yang diinginkan. Aktivasi *softmax* digunakan untuk klasifikasi *multiclass*.

3.2 ALUR PENELITIAN

Penelitian dilakukan dalam beberapa tahap seperti pada Gambar 3.5, yaitu tahap perancangan sistem, tahap pengambilan dataset, tahap pengujian, dan yang terakhir adalah tahap analisis dari hasil pengujian simulasi.



Gambar 3.5 Diagram Blok Alur Penelitian

3.2.4 Studi Literatur

Penelitian ini dimulai dengan studi literasi yang berkaitan dengan topik yang akan diteliti. Pada tahapan ini dilakukan pengkajian informasi mengenai penyakit pada daun kapas, metode *Convolutional Neural Network*, arsitektur *Alexnet* berdasarkan penelitian yang pernah dilakukan sebelumnya. Tahap ini dilakukan evaluasi, identifikasi, dan perumusan masalah yang telah dikaji oleh peneliti sebelumnya. Dari hasil tersebut kemudian dilakukan analisa yang kemudian dapat dijadikan sebagai sumber pedoman pada penelitian ini.

3.2.5 Perumusan Masalah

Perumusan masalah mencakup tujuan yang akan dicapai pada penelitian ini, objek dan metode yang akan digunakan. Perumusan masalah pada penelitian ini membahas tentang cara klasifikasi penyakit dan mengetahui tingkat akurasi yang diperoleh pada citra daun kapas menggunakan *Convolutional Neural Network* dengan arsitektur *Alexnet*.

3.2.6 Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan observasi dataset yang terdapat pada internet yang pernah dilakukan uji dengan metode yang berbeda namun pada objek yang sama. Data pada penelitian ini berasal dari dataset yang diperoleh dari Kaggle [4]. Dataset terdiri dari 1.710 citra penyakit daun kapas dalam format JPG.

3.2.7 Perancangan Sistem

Perancangan sistem merupakan suatu gambaran sistem yang akan digunakan untuk melakukan pengolahan data. Rancangan sistem pada penelitian ini dibuat dalam diagram alur pengujian sistem dengan tahapan – tahapan seperti, *preprocessing* sinyal yaitu proses *resize* untuk menyamakan ukuran citra penyakit pada daun kapas. Tujuan dari tahap *pre-processing* adalah menyamakan ukuran dataset, yang sebelumnya memiliki ukuran berbeda agar memudahkan pada tahap selanjutnya, karena memiliki ukuran yang sama. Selanjutnya, data citra penyakit pada daun kapas dilatih agar dapat menghasilkan klasifikasi dengan tepat. Data citra kemudian digunakan sebagai masukan pada proses klasifikasi menggunakan *Convolutional Neural Network*, dengan menggunakan bantuan *software Google Colab* dengan bahasa *Python*.

1. Pengumpulan Data

Dataset yang digunakan adalah citra penyakit daun kapas yang diperoleh dari Kaggle pada tahun 2021 [4]. Dataset berisi total 1.710 citra penyakit pada daun kapas. Citra penyakit daun kapas tersebut terbagi menjadi empat kelas yaitu *bacterial blight*, *curl virus*, *fusarium wilt*, dan *healthy*.

```
from google.colab import drive
drive.mount('/content/drive')
path_ds = '/content/drive/MyDrive/Dataset'
imagePaths = (paths.list_images(path_ds))
data = []
labels = []
```

2. Perancangan Sistem

Pada perancangan sistem, terdapat *Pre-processing* terdiri dari proses *resize*. Dataset yang memiliki ukuran berbeda disamakan agar memiliki ukuran citra yang sama, agar memudahkan sistem pada tahap selanjutnya karena memiliki ukuran citra yang sama. Pada sintaks program, ukuran citra dilakukan proses *resize* menjadi ukuran 256x256 piksel.

Pembagian rasio data, berguna untuk menentukan presentase data yang digunakan. Data dibagi menjadi tiga yaitu, data latih 80%, data validasi 10%, dan data tes 10%. Arsitektur Alexnet dengan 8 lapisan, 5 *layer* konvolusi dan 3 *fully connected layer*.

```
for imagePath in imagePaths:
    image = Image.open(imagePath).convert('L')
    image = np.array(image.resize((256, 256)))
    data.append(image)
    label = imagePath.split(os.path.sep)[-2]
    labels.append(label)
train_ratio = 0.80
validation_ratio = 0.10
test_ratio = 0.10

x_train, x_test, y_train, y_test = train_test_split(
    np.array(data), np.array(labels), test_size=1 -
    train_ratio)

x_val, x_test, y_val, y_test = train_test_split(x
_test, y_test, test_size=test_ratio/(test_ratio +
validation_ratio))

print(x_train.shape, x_val.shape, x_test.shape)
```

```

class AlexNet(Sequential):
    def __init__(self, input_shape, num_classes):
        super().__init__()
        self.add(Conv2D(96, kernel_size=(11,11), strides=
(4,4), padding= 'same', activation= 'relu', input_
shape= (input_shape), kernel_initializer= 'he_norm
al'))

        self.add(BatchNormalization())
        self.add(MaxPooling2D(pool_size=(3,3), strides=
(2,2), padding= 'same', data_format= None))

        self.add(Conv2D(256, kernel_size=(5,5), strides
= (1,1), padding= 'same', activation= 'relu', kern
el_initializer= 'he_normal'))
        self.add(BatchNormalization())

        self.add(MaxPooling2D(pool_size=(3,3), strides=
(2,2), padding= 'same', data_format= None))

        self.add(Conv2D(384, kernel_size=(3,3), strides
= (1,1), padding= 'same', activation= 'relu', kern
el_initializer= 'he_normal'))
        self.add(BatchNormalization())

        self.add(Conv2D(384, kernel_size=(3,3), strides
= (1,1), padding= 'same', activation= 'relu', kern
el_initializer= 'he_normal'))
        self.add(BatchNormalization())

        self.add(Conv2D(256, kernel_size=(3,3), strides
= (1,1), padding= 'same', activation= 'relu', kern
el_initializer= 'he_normal'))
        self.add(BatchNormalization())

        self.add(MaxPooling2D(pool_size=(3,3), strides=
(2,2), padding= 'same', data_format= None))

```

3. Pengujian sistem

Pada pengujian sistem, masukan terdiri dari 4 kelas citra penyakit daun kapas, yaitu *bacterial blight*, *curl virus*, *fusarium wilt*, dan *healthy*. Data citra masukan melewati proses *resizing* agar memiliki ukuran citra yang sama. Pada tahap klasifikasi, terdapat *fully connected layer* dan aktivasi *softmax* yang akan membangun

jumlah keluaran yang diinginkan. Aktivasi softmax digunakan untuk klasifikasi *multiclass*.

```
from tensorflow.keras.callbacks import EarlyStopping
keras_callbacks = [
    EarlyStopping(monitor='val_accuracy', patience=20, mode='max', min_delta=0.001),
]
H = model.fit(x_train, y_train, validation_data=(x_val, y_val), epochs=50, batch_size=8, shuffle = True, callbacks=[keras_callbacks])
import matplotlib.pyplot as plt
print(H.history.keys())
```

4. Analisis

Akurasi model dari proses pelatihan data yang berupa grafik akurasi. *Confusion matrix* digunakan untuk melakukan uji kinerja sistem yang telah dibuat, *confusion matrix* membantu dalam mendapatkan nilai akurasi.

```
# summarize history for accuracy
plt.plot(H.history['accuracy'])
plt.plot(H.history['val_accuracy'])
plt.title('MODEL ACCURACY')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training Data', 'Validation Data'], loc='lower right')
X = np.linspace(0, 100, 100)
plt.ylim(.1, 1)
plt.show()

# confusion matrix
print("[INFO] evaluating network...")
target_names=['bacterial_blight', 'healthy', 'fusarium_wilt', 'curl_virus']
```



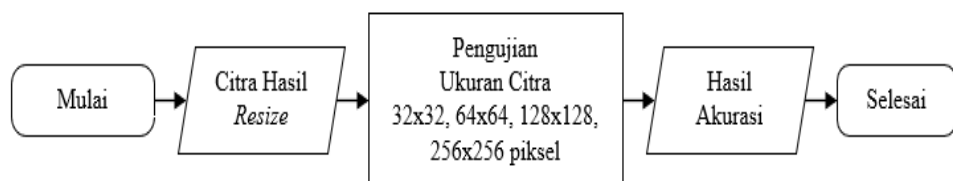
```

predictions = model.predict(x_test, batch_size=8)
print(predictions)
print(y_test)
print(classification_report(y_test.argmax(axis=1)
,
predictions.argmax(axis=1), target_names=target_
names))
import sklearn.metrics as metrics
import seaborn as sn
import matplotlib.pyplot as plt
confusion_matrix = metrics.confusion_matrix(y_tes
t.argmax(axis=1), predictions.argmax(axis=1))
x_axis_labels = ['bacterial_blight', 'healthy', 'fu
ssarium_wilt', 'curl_virus']
y_axis_labels = ['bacterial_blight', 'healthy', 'fu
ssarium_wilt', 'curl_virus']
sn.heatmap(confusion_matrix, xticklabels=x_axis_l
abels, yticklabels=y_axis_labels, annot=True, fmt
='d')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.tight_layout()
plt.show()

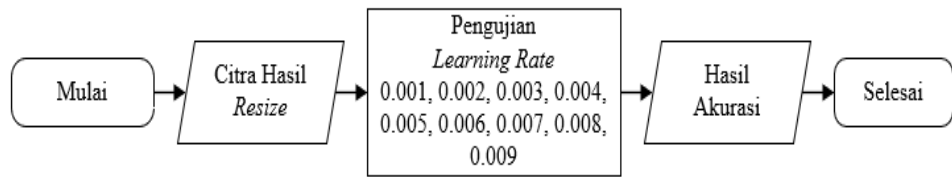
```

3.2.5 Pengujian Sistem

Tahap ini akan menguji kinerja dari sistem yang telah dibuat, tujuannya adalah agar dapat mengetahui seberapa efektif sistem dalam mengklasifikasi citra yang digunakan. Salah satu cara yang dapat digunakan untuk melakukan uji kinerja sistem adalah menggunakan *confusion matrix* yang membantu dalam mendapatkan nilai akurasi. Melalui dua skema pengujian yaitu pengujian ukuran citra dan *learning rate*, proses kedua skema pengujian dapat dilihat pada Gambar 3.6 dan 3.7.



Gambar 3.6 Flowchart Pengujian Ukuran Citra



Gambar 3.7 Flowchart Pengujian Learning Rate

Perhitungan confusion matrix pada klasifikasi *multiclass*, untuk mendapatkan nilai akurasi dapat dilakukan dengan menggunakan rumus berikut [28]:

$$\text{Akurasi} = \frac{\text{Total True Positive}}{\text{Total Dataset}} \quad (3.1)$$

Tabel 3.1 Confusion Matrix

	Prediksi				Akurasi	
	Kelas	<i>Bacterial Blight</i>	<i>Healthy</i>	<i>Fusarium Wilt</i>		<i>Curl Virus</i>
Aktual	<i>Bacterial Blight</i>	40	2	1	1	0.9715
	<i>Healthy</i>	0	33	2	1	0.9661
	<i>Fusarium Wilt</i>	1	1	42	0	0.9715
	<i>Curl Virus</i>	0	0	0	47	0.9884
Rata-rata					0.947	

TP merupakan data yang diklasifikasi sistem sesuai dengan kategori sebenarnya. Contoh dari TP adalah pada kelas klasifikasi *Bacterial Blight* pada Tabel 3.1 *confusion matrix*, nilai TP nya adalah 40. Total dataset merupakan jumlah seluruh nilai yang terdapat pada tabel *confusion matrix*, pada Tabel 3.1 total datasetnya sebanyak 171.

3.2.6 Analisis dan Kesimpulan

Tahap ini merupakan analisis terhadap data yang telah diolah oleh sistem yang telah dibuat dan memberikan kesimpulan dengan tujuan untuk menjawab perumusan masalah yang menjadi dasar pelaksanaan penelitian ini.