BAB IV
HASIL DAN PEMBAHASAN

4.1 Dataset

Pada penelitian ini peneliti menggunakan data dari pengepul kelapa sawit yang ada di desa Raman fajar. Pengumpulan data dilakukan melalui wawancara langsung ke pengepul yang ada di desa Raman fajar dan juga melalui observasi.

4.1.1 Labeling

Labeling merupakan proses penentuan label pada data. Pada studi kasus ini "KESIMPULAN" akan menjadi labelnya dan "CUACA, HARI, BERATAWAL dan BERATAKHIR" akan menjadi atribut. Kemudian diberikan kelas pada label tersebut dengan melakukan convert data yang dimana variabel yang awalnya berbentuk kategori menjadi numerik. Berikut ini adalah hasil dari pemberian label pada dataset :

Tabel 4. 1 Tabel labeling cuaca.

Cuaca	Cuaca cerah	Cuaca hujan
	1	0

Tabel 4. 2 Tabel labeling Hari.

Hari	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	minggu
	1	2	3	4	5	6	7

Tabel 4. 3 Tabel labeling Kesimpulan.

Kesimpulan	Pengepulan kelapa sawit diangkut tepat waktu	Pengepulan kelapa sawit ditunda
	1	0

4.2 Hasil Penelitian

4.2.1 Analisis Algoritma

Pada tahapan ini akan dianalisa algoritma machine learning yang digunakan yakni support vector machine yang memanfaatkan bahasa pemrograman python dengan jupyter notebook sebagai text editor. Adapun hal yang pertama dilakukan yakni

penggunaan library python pada jupyter notebook untuk dapat menerapkan algoritma SVM(Suport Vector Mechine).

SVM ditemukan oleh Vladimir N. Vapnik dan Alexey Ya. Chervonenkis pada tahun 1963. Sejak itu, SVM telah digunakan dalam klasifikasi teks, hiperteks dan gambar. SVM dapat bekerja dengan karakter tulisan tangan dan algoritma ini telah digunakan di laboratorium biologi untuk melakukan tugas seperti menyortir protein. SVM bekerja untuk mencari hyperplane atau fungsi pemisah (decision boundary) terbaik untuk memisahkan dua buah kelas atau lebih pada ruang input.

4.2.2 Implementasi

Seluruh pengkodean atau proses dalam pembuat model prediksi ini dapat dilihat dan di download pada platform GitHub dengan URL berikut : https://github.com/anggayogap/model-prediksi

Berikut ini adalah langkah – langkah dalam pembuatan model prediksi waktu panen tanaman kelapa sawit dengan menggunakan metode machine learning :

1. Input Library

Pada tahapan ini akan dilakukan proses input library yang dibutuhkan dalam pembuatan model prediksi waktu panen kelapa sawit untuk mempermudah pengepulan. Berikut adalah library yang dibutuhkan :

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

2. Membaca/Input Dataset

Input dataset adalah memasukkan atau membaca dataset yang telah disiapkan dan telah melalui proses cleaning dan labeling. Berikut adalah proses input data :

```
kelapasawit = pd.read_excel('datasetsawit.xlsx')
```

Setelah melalui proses input dataset telah dilakukan dan sukses yang selanjutnya adalah menampilkan data tersebut. Berikut adalah proses menampilkan dataset yang telah diinput :

```
kelapasawit
```

Output yang dihasilkan :

Cuaca	Hari	Berat awal	Berat akhir	Kesimpulan
1	1	3902	3898	1
1	6	3984	3950	0
0	2	4010	4998	1
0	3	4761	4751	1
1	4	5985	5970	1
1	4	3605	3600	1
1	6	4815	4800	1
1	6	4767	4750	0
0	3	5980	5968	0
1	3	5950	5945	1

Langkah selanjutnya adalah untuk menampilkan jumlah label yaitu "KESIMPULAN" :

```
kelapasawit['KESIMPULAN'].value_counts()
```

Output yang dihasilkan :

Label	Jumlah
1	126
0	66
Name : KESIMPULAN, dtype : int64	

3. Memisahkan Data

Pada tahapan yang akan dilakukan adalah memisahkan untuk data sebagai atribut dan data sebagai label agar dapat melakukan prediksi. Berikut adalah code untuk memisahkan nilai (x = data atribut, dan y = data label) :

```
x = kelapasawit.drop (columns='KESIMPULAN', axis=1)
y = kelapasawit['KESIMPULAN']
```

Setelah proses memisahkan data selesai, selanjutnya adalah menampilkan data x dan y guna untuk mengetahui bahwa data telah berhasil dipisahkan.

1. Menampilkan nilai x

```
print(x)
```

Output yang dihasilkan :

Cuaca	Hari	Berat awal	Berat akhir
1	1	3902	3898
1	6	3984	3950
0	2	4010	4998
0	3	4761	4751
1	4	5985	5970
1	4	3605	3600
1	6	4815	4800
1	6	4767	4750
0	3	5980	5968

[192 rows × 4 columns]

2. Menampilkan nilai y

```
print(y)
```

Output yang dihasilkan :

Kesimpulan
1
0
1
1
1
1
1
0
0

Name : KESIMPULAN, length 192, dtype : int64

4. Standarisasi Data

`Tahapan selanjutnya adalah standarisasi data, langkah pertama yang dilakukan proses standarscaler. StandarScaler` adalah bahwa ia akan

mengubah data sedemikian rupa sehingga distribusinya akan memiliki nilai rata-rata 0 dan standar deviasi 1 . Mengingat distribusi data, setiap nilai dalam dataset akan memiliki nilai rata-rata sampel dikurangi, dan kemudian dibagi dengan standar deviasi dari seluruh dataset. Berikut adalah proses melakukan StandarScaler :

```
scaler = StandardScaler()
scaler.fit(x)
```

Output yang dihasilkan :

StandarScaler
StandarScler()

Langkah selanjutnya adalah proses standarisasi data untuk menyamakan range data yang berbeda-beda. Berikut adalah proses code standarisasi data :

```
standarized_data = scaler.transform(x)
print(standarized_data)
```

Output yang dihasilkan :

Cuaca	Hari	Berat awal	Berat akhir
[8.81917104e-01	-1.62814793e+00	-1.47154384e+00	-1.47652578e+00]
[8.81917104e-01	9.94372224e-01	1.38167822e+00	-1.41879989e+00]
[-1.13389342e+00	-1.10364390e+00	-1.35318424e+00	-2.55401210e-01]
[-1.13389342e+00	-5.79139867e-01	-5.30146661e-01	-5.29599181e-01]
[8.81917104e-01	-5.46358365e-02	8.11262124e-01	8.23628862e-01]

Setelah proses standarisasi data selesai tahapan selanjutnya adalah menampilkan data yang telah distandarisasi untuk membuktikan bahwa proses standarisasi data telah berhasil. Berikut adalah proses code dan hasil menampilkan data yang telah distandarisasi :

```
x = standarized_data
y = kelapasawit['KESIMPULAN']
print(x)
print(y)
```

Output yang dihasilkan :

Cuaca	Hari	Berat awal	Berat akhir
[8.81917104e-01	-1.62814793e+00	-1.47154384e+00	-1.47652578e+00]
[8.81917104e-01	9.94372224e-01	1.38167822e+00	-1.41879989e+00]

[-1.13389342e+00	-1.10364390e+00	-1.35318424e+00	-2.55401210e-01]
[-1.13389342e+00	-5.79139867e-01	-5.30146661e-01	-5.29599181e-01]
[8.81917104e-01	-5.46358365e-02	8.11262124e-01	8.23628862e-01]

5. Spliting Data

Pada tahapan ini akan dilakukan proses spliting data (memcah data) menjadi data training dan data testing. Data training akan dimasukkan sebesar 80% dan sedangkan untuk data testing sebesar 20%. Berikut adalah proses code dan menampilkan hasil spliting data dapat dilihat pada gambar 4.8 :

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.2, stratify=y, random_state=2)
```

Output yang dihasilkan :

Jenis data	Rows	Columns
Jumlah data	192	4
Data training	153	4
Data testing	39	4

6. Input Algoritma (SVM)

Pada tahapan ini akan dilakukan proses input algoritma yang akan digunakan yaitu SVM (support vector mechine). Jenis kernel pada algoritma yang digunakan adalah linear. Kernel linear adalah untuk data yang dapat dipisahkan secara linear, dan dataset dapat diklasifikasi menjadi dua kelas dengan menggunakan sebuah garis lurus tunggal. Berikut adalah proses code dan menampilkan hasil input algoritma SVM :

```
classifier = svm.SVC(kernel='linear')
classifier.fit(x_train, y_train)
```

Output yang dihasilkan :

SVC
SVC (kernel = ’linear’)

7. Uji Akurasi Prediksi

Pada tahapan ini akan lakukan proses uji tingkat akurasi pada prediksi yang akan dilakukan. Akurasi yang akan dilakukan adalah pada akurasi pada data training dan

akurasi pada data testing. Berikut adalah proses code dan menampilkan hasil dari melakukan uji akurasi pada data :

a. Data training

```
x_train_prediction = classifier.predict(x_train)
training_data_accuracy = accuracy_score(x_train_prediction, y_train)
print('akurasi data training adalah = ', training_data_accuracy)
```

Output yang dihasilkan :

Akurasi data training
akurasi data training adalah = 0.6601307189542484

Dapat dilihat pada hasil diatas tingkat akurasi pada data training adalah sebesar 0,66 atau 66%.

b. Data testing

```
x_test_prediction = classifier.predict(x_test)
test_data_accuracy = accuracy_score(x_test_prediction, y_test)
print('akurasi data testing adalah = ', test_data_accuracy)
```

Output yang dihasilkan :

Akurasi data training
akurasi data testing adalah = 0.6666666666666666

Dapat dilihat pada hasil diatas nilai akurasi pada data testing adalah sebesar 0,66 atau sebesar 66%.

8. Form Input Data

Pada tahap ini akan dilakukan proses membuat form untuk inputan data yang akan di lakukan proses prediksi. Berikut adalah proses code membuat form input data dan menampilkan hasil dari form input data dapat dilihat pada gambar 4.12 :

```
CUACA = eval(input ('input nilai cuaca ='))
print(CUACA)
HARI = eval(input('input nilai hari ='))
print(HARI)
BERATAWAL = eval(input ('input nilai berat awal ='))
print(BERATAWAL)
BERATAKHIR = eval(input('input nilai berat akhir ='))
print(BERATAKHIR)
```

Output yang dihasilkan :

Input form data

```
Input nilai cuaca = 1
Input nilai hari = 1
Input nilai berat awal = 3898
Input nilai berat akhir = 3898
```

9. Prediksi

Pada tahapan ini akan dilakukan proses prediksi pada dataset yang telah di input, dan akurasi telah diukur melalui proses sebelumnya. Dan untuk input atribut data yang akan diprediksi telah dilakukan pada proses sebelumnya. Berikut adalah proses melakukan prediksi dapat dilihat pada gambar 4.13 :

```
input_data = (CUACA, HARI, BERATAWAL, BERATAKHIR)
input_data_as_numpy_array = np.array(input_data)
input_data_reshape = input_data_as_numpy_array.reshape(1,-1)
std_data = scaler.transform(input_data_reshape)
print(std_data)
prediction = classifier.predict(std_data)
print(prediction)
if (prediction[0] == 0 ):
  print('pengepulan kelapa sawit ditunda')
else :
  print('pengepulan kelapa sawit tepat waktu')
```

Output yang dihasilkan :

Hasil prediksi
[[0.8819171 -1.62814793 -1.47592753 -1.47652578]] [1] pengepulan kelapa sawit tepat waktu

10. Menyimpan File Code

Pada tahapan ini akan dilakukan proses menyimpan hasil code model prediksi waktu panen kelapa sawit untuk mempermudah pengepulan dalam bentuk “.sav” agar dapat di jalankan dalam bentuk web sederhana dengan menggunakan tools streamlit. Berikut adalah proses menyimpan file hasil code model yang telah dibuat :

```
import pickle
filename = 'kelapasawit_model.sav'
pickle.dump(classifier, open(filename,'wb'))
```

11. Tampilan Website Sederhana

Setelah proses srcript code untuk prediksi selesai, langkah selanjutnya adalah membuat tampilan interface website sederhana untuk agar supaya penggunaan dari

model lebih mudah dan interaktif. Berikut adalah proses code pembuatan website sederhana menggunakan tools streamlit :

a. Import library

Berikut adalah proses dari import library yang dibutuhkan :

```
import pickle
import streamlit as st
```

b. Input model

Berikut adalah proses memasukan (input) model yang telah dibuat dan disimpan dalam bentuk (.sav) :

```
kelapasawit_model = pickle.load(open('kelapasawit_model.sav', 'rb'))
```

c. Judul website dan form input data

Berikut adalah proses membuat judul website dan membuat form untuk input data yang akan diprediksi :

```
st.title('Prediksi Waktu Panen tanaman Kelapa Sawit untuk mempermudah Pengepulan')
col1, col2 = st.columns(2)
with col1 :
    CUACA = st.text_input ('Masukkan nilai cuaca')
with col2 :
   HARI = st.text_input ('Masukkan nilai Hari')
with col1 :
   BERATAWAL = st.text_input ('Masukkan nilai berat awal')
with col2 :
   BERATAKHIR = st.text_input ('Masukkan nilai berat akhir')
```

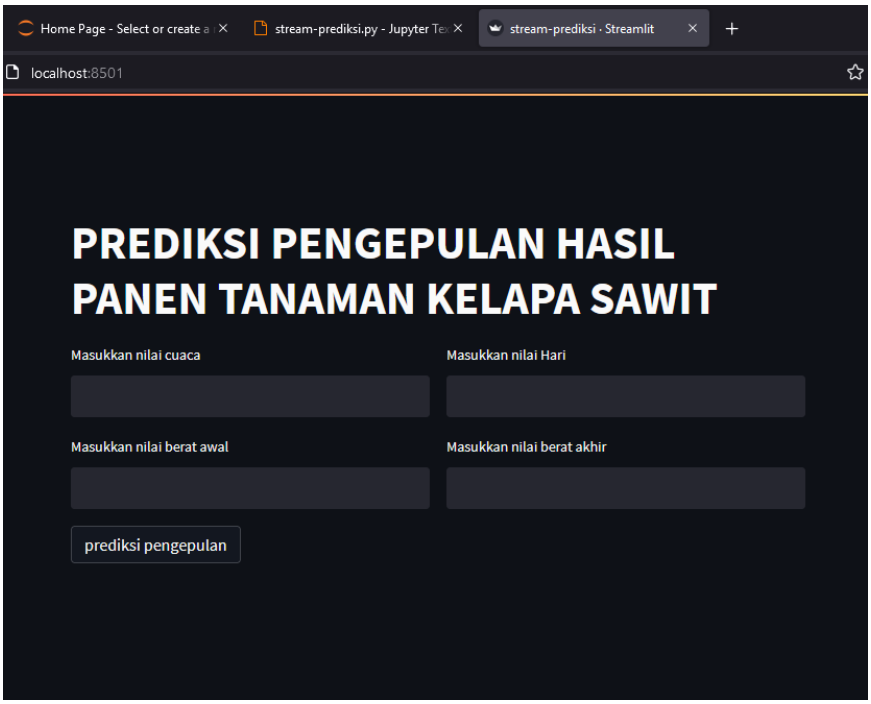
d. Hasil prediksi

Berikut adalah proses untuk membuat dan menampilkan hasil prediksi :

```
kelapasawit_prediksi = ''
if st.button('prediksi pengepulan'):
kelapasawit_prediction=kelapasawit_model.predict([[CUACA, HARI, BERATAWAL, BERATAKHIR]])
if(kelapasawit_prediction[0] == 0):
  kelapasawit_prediksi = 'pengepulan kelapa sawit tepat waktu'
else :
 kelapasawit_prediksi = 'pengepulan kelapa sawit ditunda'
 st.success(kelapasawit_prediksi)
```

e. Hasil website

Dapat dilihat pada gambar 4.1 adalah hasil dari pembuatan website sederhana menggunakan streamlit :



Gambar 4.1 Hasil pembuatan website sederhana.

f. Uji coba prediksi

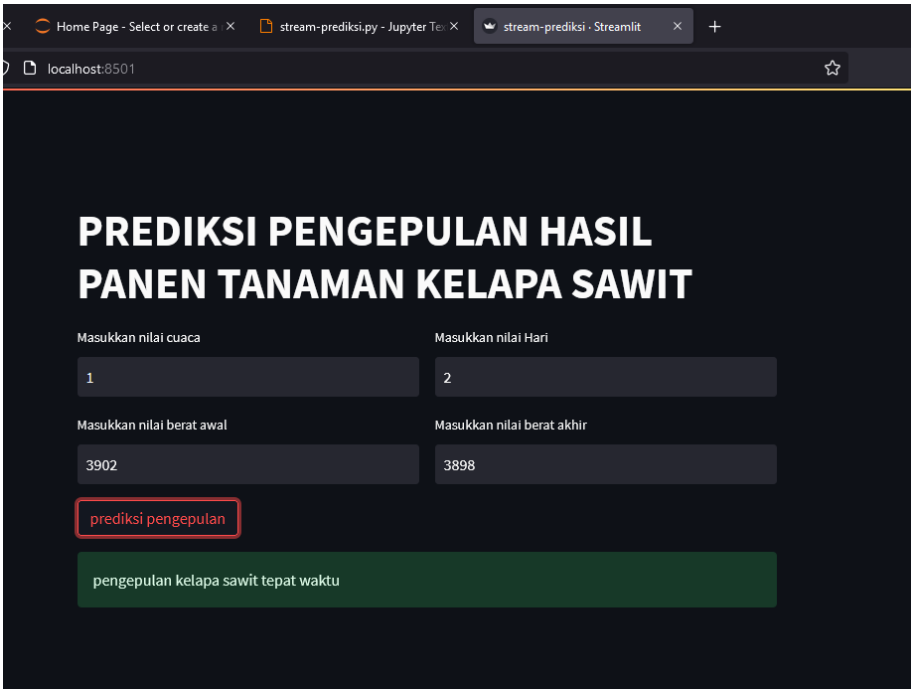
1. Website

Pengujian model prediksi pada website sederhana dengan contoh data yang digunakan adalah sebagai berikut :

Tabel 4.4 Tabel contoh data untuk prediksi

Variabel	Nilai
Cuaca	1
hari	1
Berat awal	3902
Berat akhir	3898
Hasil prediksi	Pengepulan tepat waktu

Dapat dilihat pada gambar 4.2 telah dilakukan uji coba prediksi dan uji coba fungsi pada website telah berjalan dengan sebagaimana mestinya.



Gambar 4.2 Hasil uji coba prediksi menggunakan website.

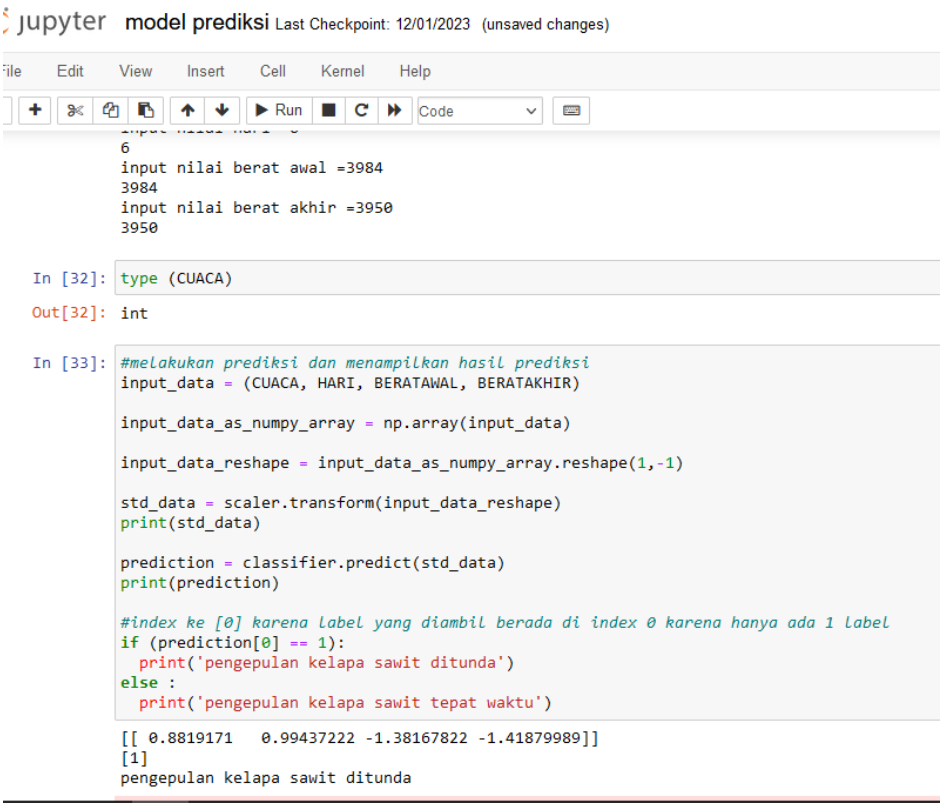
2. Jupyter notebook

Pengujian model prediksi pada jupyter notebook dengan contoh data yang digunakan adalah sebagai berikut :

Variabel	Nilai
Cuaca	1

hari	6
Berat awal	3984
Berat akhir	3950
Hasil prediksi	Pengepulan ditunda

Dapat dilihat pada gambar 4.3 telah dilakukan uji coba prediksi dan uji coba fungsi pada jupyter notebook telah berjalan dengan sebagaimana mestinya.



Gambar 4.3 Hasil uji coba prediksi menggunakan Jupyter notebook.