

Tugas 1: Judul tugas **Support Vector Machines (SVM)**

Siti aisah -0110222129 ^{1*}

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

² Sistem Informasi, STT Terpadu Nurul Fikri, Depok

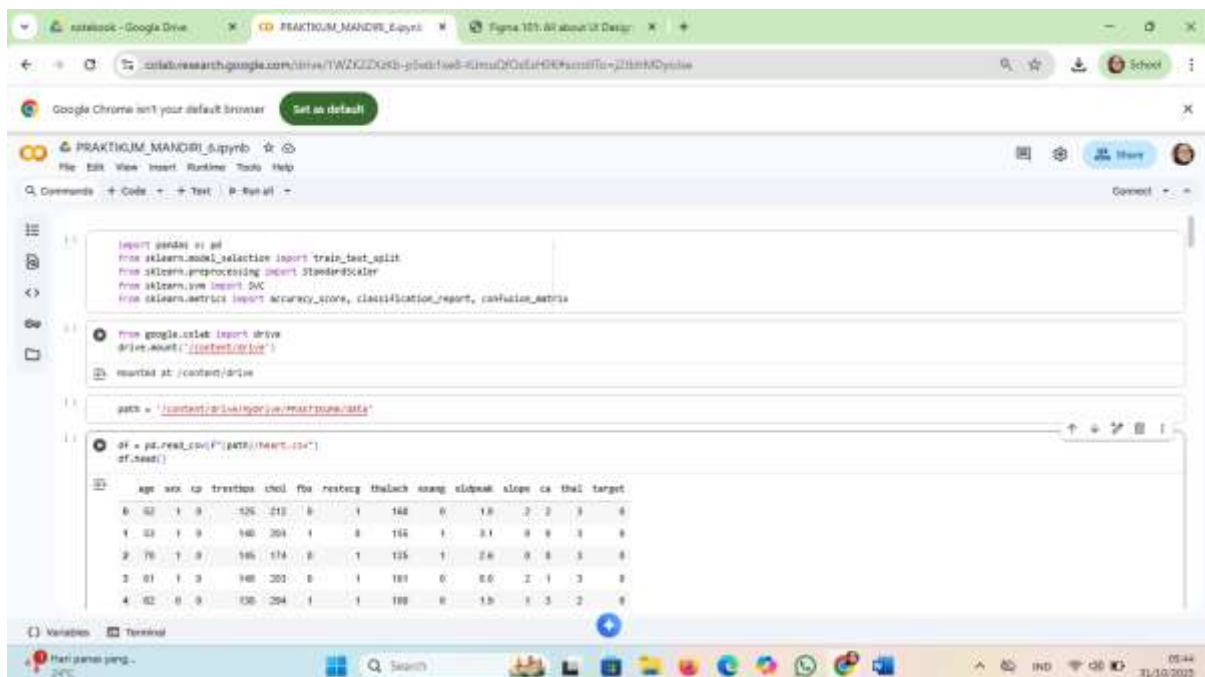
*E-mail: siti22129ti@student.nurulfikri.ac.id

Abstract.

Machine Learning (Pembelajaran Mesin) adalah cabang dari Kecerdasan Buatan (AI) yang memungkinkan sistem untuk belajar secara mandiri dari data, mengenali pola, dan membuat keputusan atau prediksi tanpa diprogram secara **eksplisit** untuk setiap

Tugas mandiri 1

1. Menyiapkan semua *tools* (library) yang dibutuhkan dan Mengakses data dari Google Drive.



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from google.colab import drive
drive.mount('/content/drive')

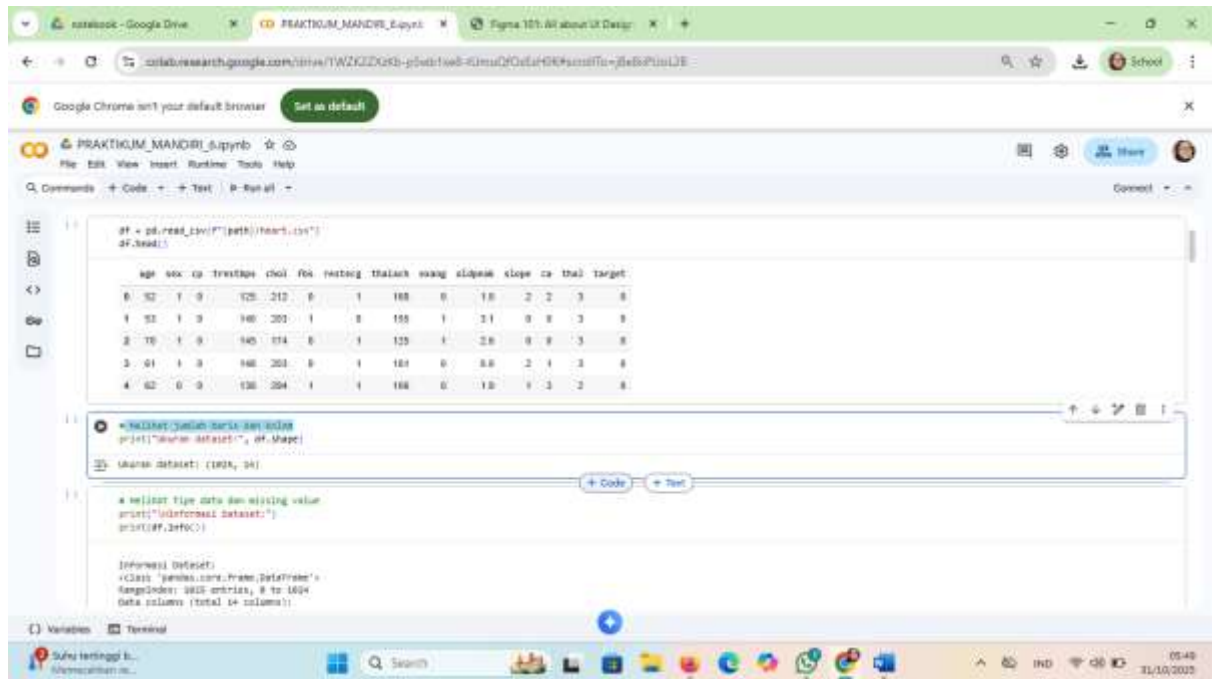
path = '/content/drive/MyDrive/PRAKTIKUM/SAHA'

df = pd.read_csv(path + 'heart.csv')
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	0	125	212	0	1	168	0	1.6	2	2	3	0
1	63	1	0	140	201	1	0	155	1	3.1	0	0	3	0
2	70	1	0	180	174	0	1	135	1	2.6	0	0	3	0
3	61	1	0	140	201	0	1	180	0	0.0	2	1	3	0
4	62	0	0	130	204	1	1	180	0	1.5	1	3	2	0

1.1 hasil dari menyiapkan semua tools dan menghubungkan ke drive

2. melihat data ,Melihat jumlah baris dan kolom



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
# df = pd.read_csv("../path/heart.csv")
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restingt	thalach	exang	slope	ca	thal	target
0	52	1	0	125	212	0	1	165	0	1.0	2	2	0
1	53	1	0	140	201	1	0	155	1	2.1	0	0	0
2	70	1	0	145	174	0	1	125	1	2.0	0	0	0
3	61	1	0	140	201	0	1	161	0	0.0	2	1	0
4	62	0	0	130	204	1	1	168	0	1.0	1	3	0

```
# melihat jumlah baris dan kolom
print("ukuran dataset", df.shape)
```

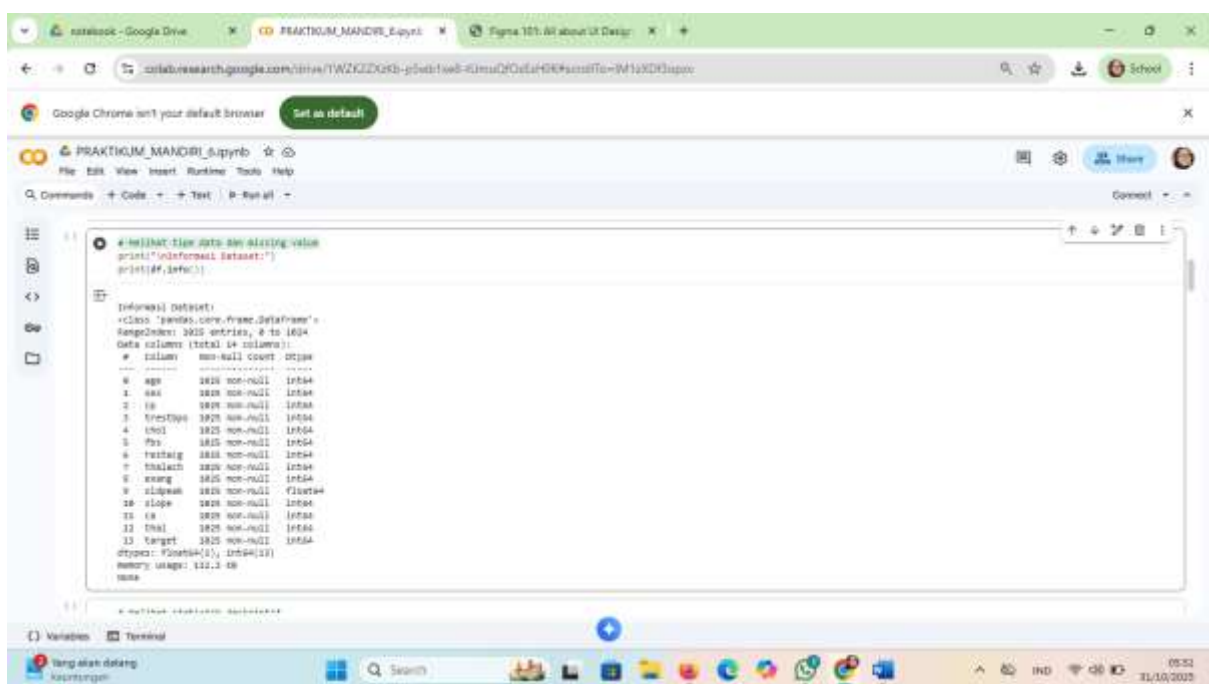
ukuran dataset: (1025, 14)

```
# melihat tipe data dan missing value
print("informasi dataset:")
print(df.info())
```

Informasi dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
column non-null count dtype
--- ---
0 age 1025 non-null int64
1 sex 1025 non-null int64
2 cp 1025 non-null int64
3 trestbps 1025 non-null int64
4 chol 1025 non-null float64
5 fbs 1025 non-null int64
6 restingt 1025 non-null int64
7 thalach 1025 non-null int64
8 exang 1025 non-null int64
9 slope 1025 non-null float64
10 ca 1025 non-null int64
11 thal 1025 non-null int64
12 target 1025 non-null int64
dtypes: float64(1), int64(13)
memory usage: 112.3 KB
None

1.2 hasil dari melihat data dan melihat baris kolom

3.Melihat tipe data dan missing value



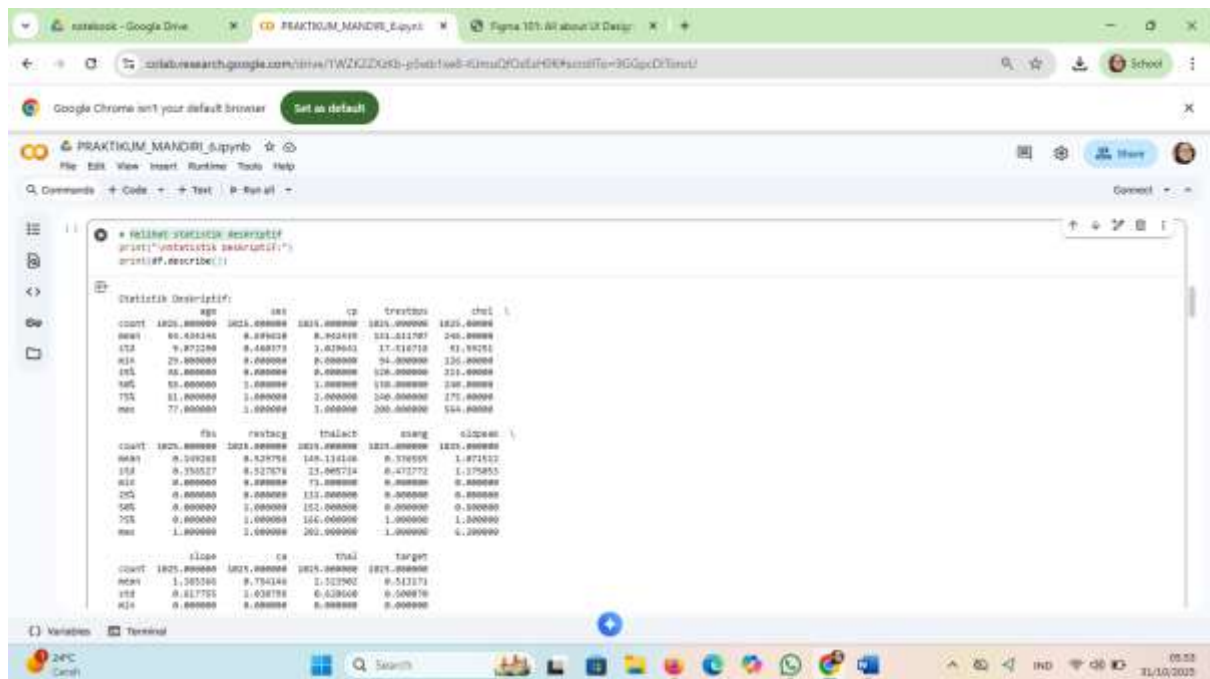
The screenshot shows a Jupyter Notebook interface with the following code and output:

```
# melihat tipe data dan missing value
print("informasi dataset:")
print(df.info())
```

Informasi dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
column non-null count dtype
--- ---
0 age 1025 non-null int64
1 sex 1025 non-null int64
2 cp 1025 non-null int64
3 trestbps 1025 non-null int64
4 chol 1025 non-null float64
5 fbs 1025 non-null int64
6 restingt 1025 non-null int64
7 thalach 1025 non-null int64
8 exang 1025 non-null int64
9 slope 1025 non-null float64
10 ca 1025 non-null int64
11 thal 1025 non-null int64
12 target 1025 non-null int64
dtypes: float64(1), int64(13)
memory usage: 112.3 KB
None

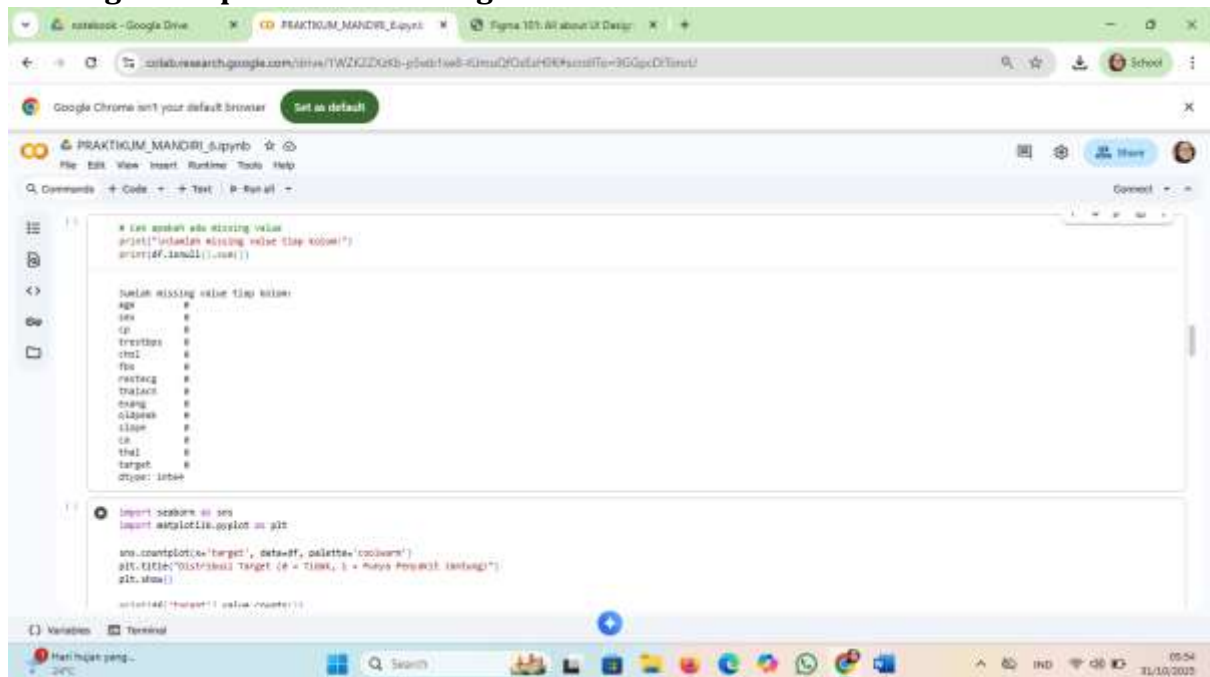
1.2 hasil dari melihat Melihat tipe data dan missing value

4. Melihat statistik deskriptif



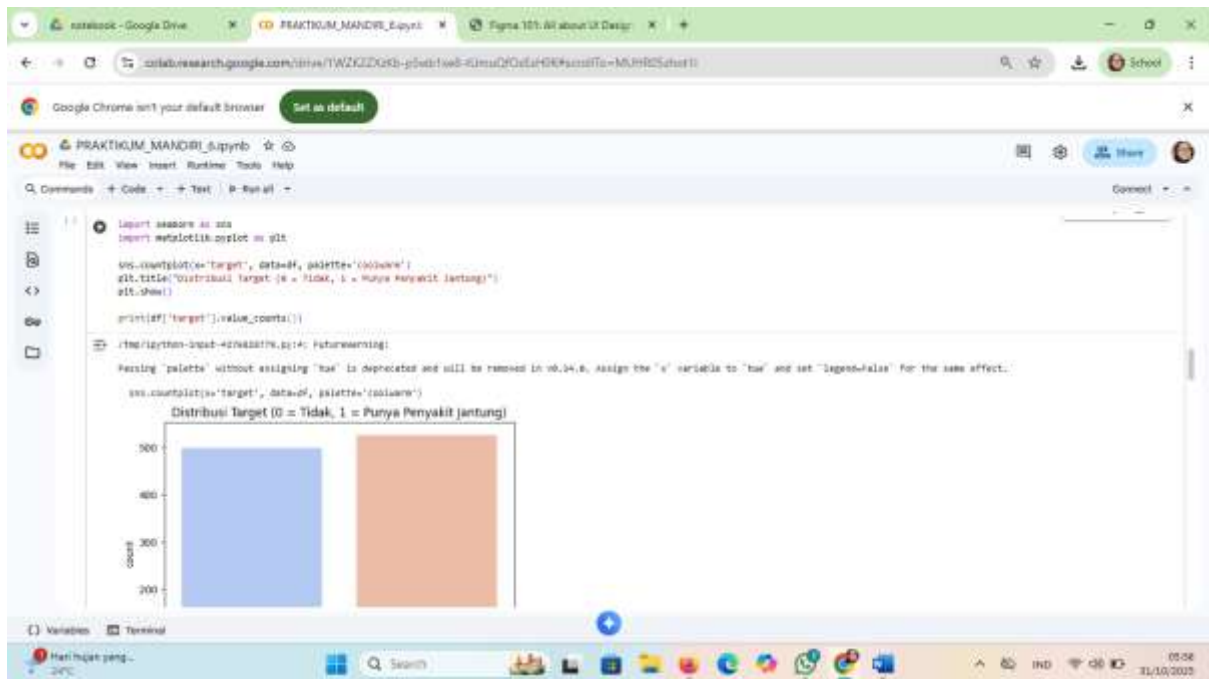
1.3 hasil dari Melihat statistik deskriptif

5. Mengecek apakah ada missing value



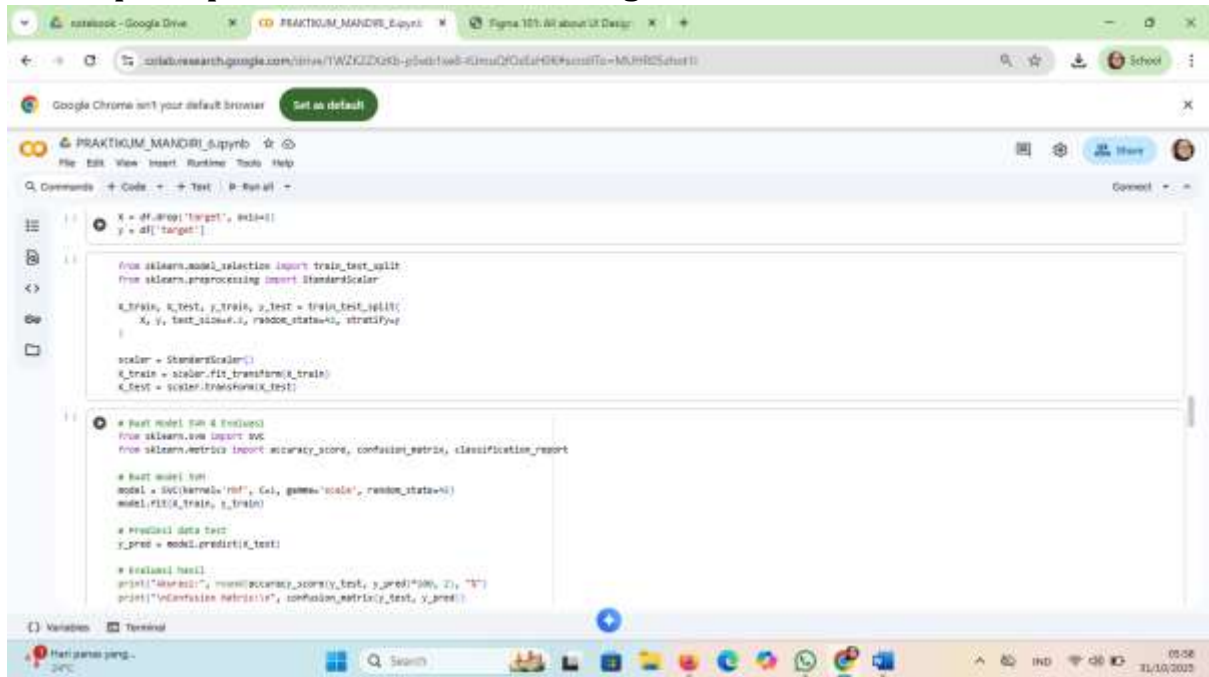
1.5 hasil dari mengecek apakah ada missing value

6. Analisis Data Eksploratif (EDA) dan memvisualisasikan distribusi variabel target



1.6 hasil dari Analisis Data Eksploratif (EDA) dan memvisualisasikan distribusi variabel target

7. mempersiapkan data, melatih, dan mengevaluasi



The screenshot shows a Jupyter Notebook interface with the following code:

```
x = df.drop('target', axis=1)
y = df['target']

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.3, random_state=42, stratify=y)

scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

# Part Model dan Evaluasi
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

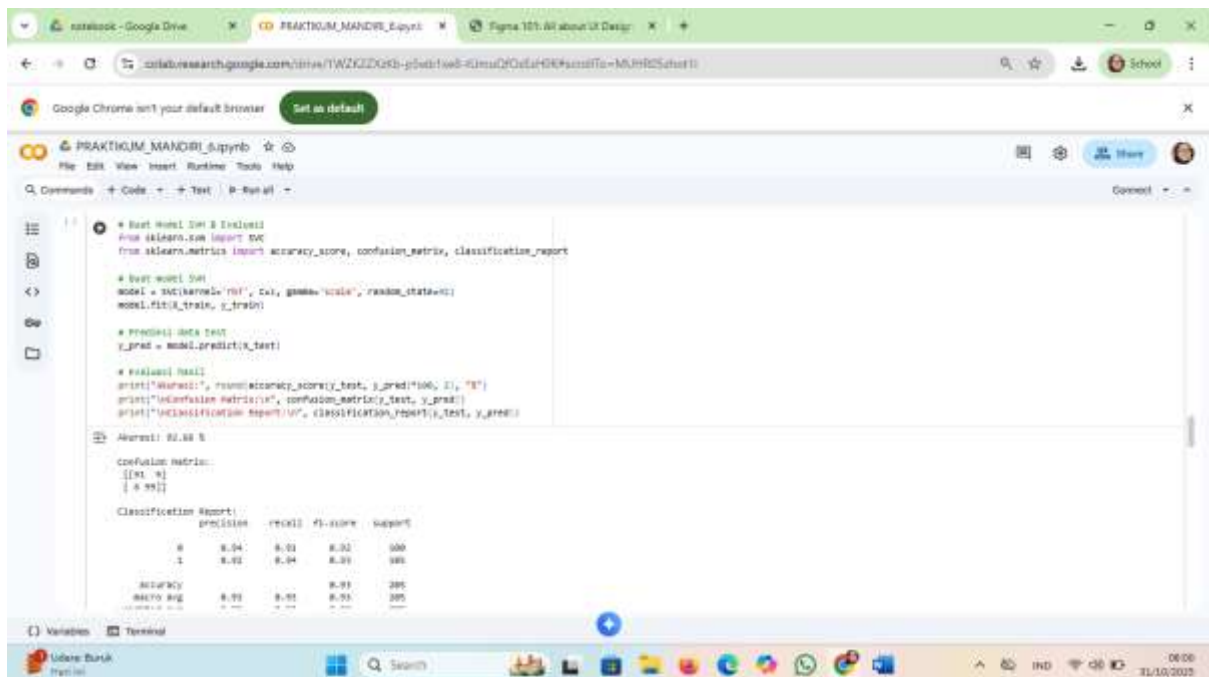
# Part model svm
model = SVC(kernel='rbf', C=1, gamma='scale', random_state=42)
model.fit(x_train, y_train)

# Prediksi data test
y_pred = model.predict(x_test)

# Evaluasi hasil
print("Akurasi:", round(accuracy_score(y_test, y_pred)*100, 2), "%")
print("Confusion Matrix:", confusion_matrix(y_test, y_pred))
```

1.7 hasil dari mempersiapkan data, melatih, dan mengevaluasi

8. Buat Model SVM & Evaluasi, Buat model SVM, Prediksi data test, Evaluasi hasil



The screenshot shows the same Jupyter Notebook with the following code and output:

```
# Part Model dan Evaluasi
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Part model svm
model = SVC(kernel='rbf', C=1, gamma='scale', random_state=42)
model.fit(x_train, y_train)

# Prediksi data test
y_pred = model.predict(x_test)

# Evaluasi hasil
print("Akurasi:", round(accuracy_score(y_test, y_pred)*100, 2), "%")
print("Confusion Matrix:", confusion_matrix(y_test, y_pred))
print("Classification Report:", classification_report(y_test, y_pred))
```

The output of the code is as follows:

```
Akurasi: 92.88 %
Confusion matrix:
[[ 91  10]
 [  0  99]]

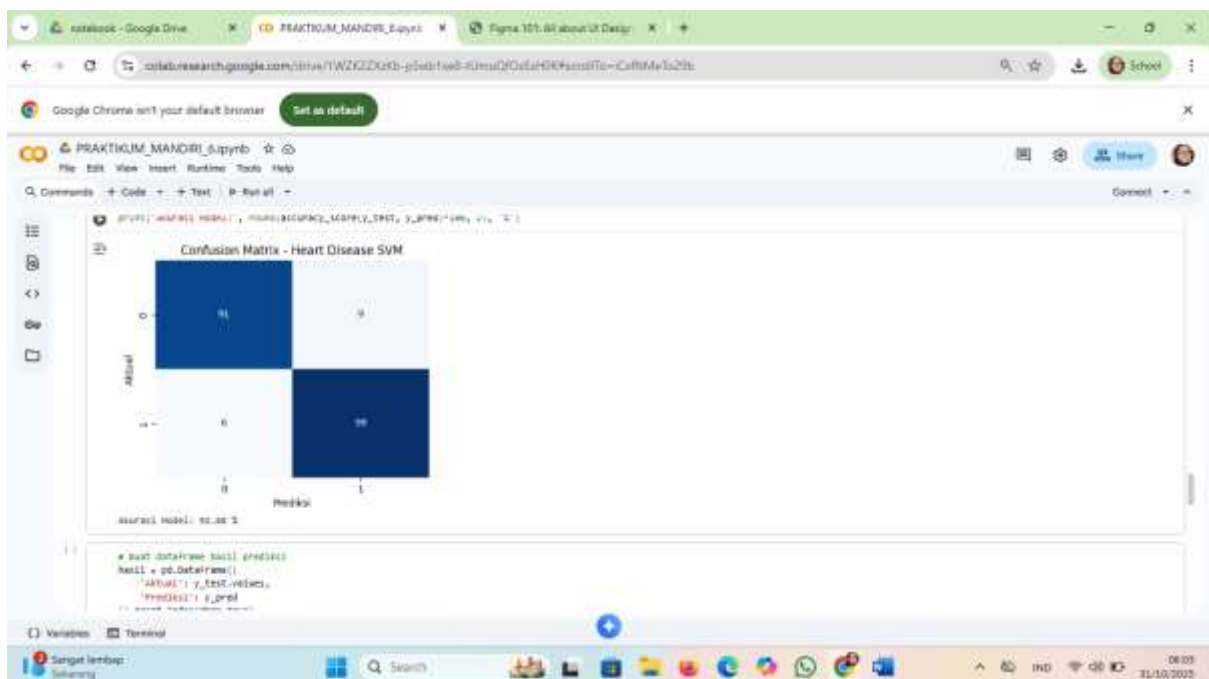
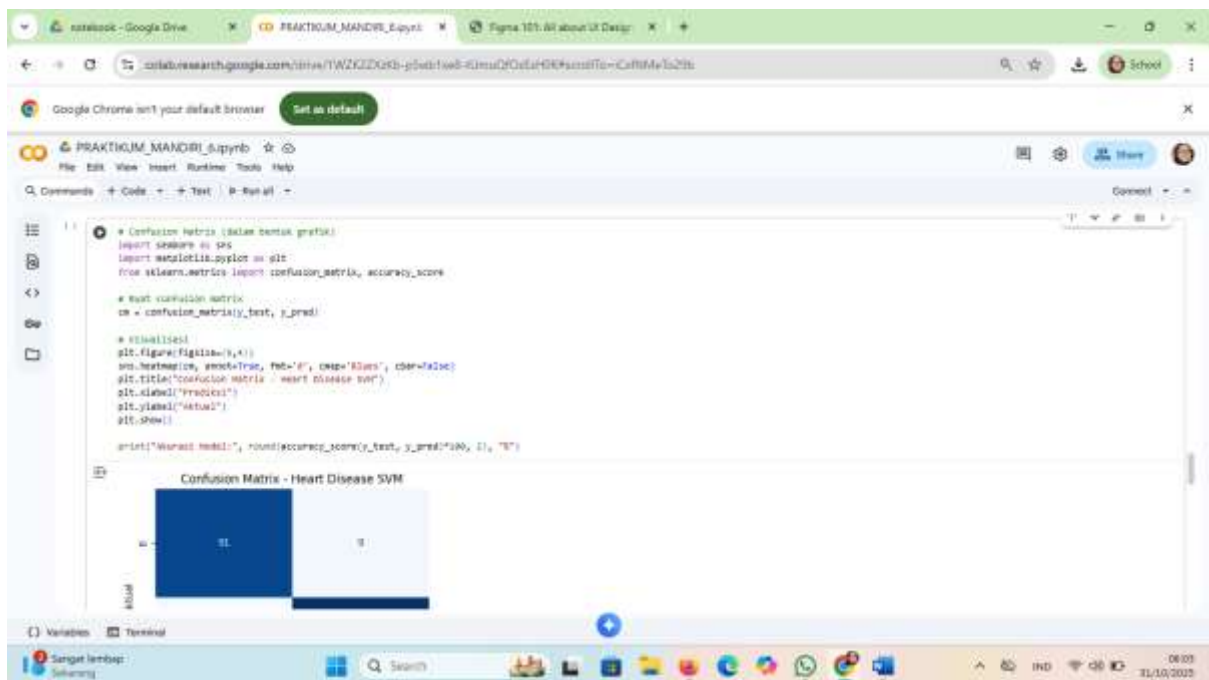
Classification Report:
      precision    recall  f1-score   support

     0       0.94      0.93      0.93       100
     1       0.92      0.94      0.93        99

 accuracy: 0.93
 avg  precision: 0.93
 avg  recall: 0.93
 avg  f1-score: 0.93
 avg  support: 199
```

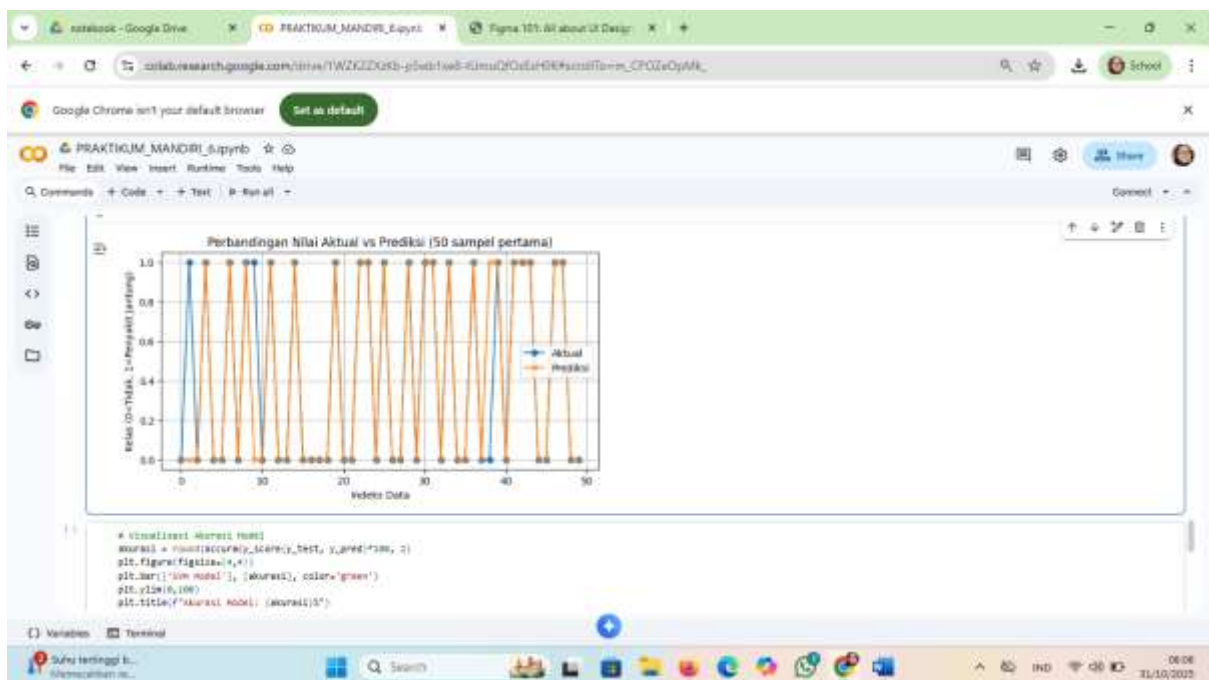
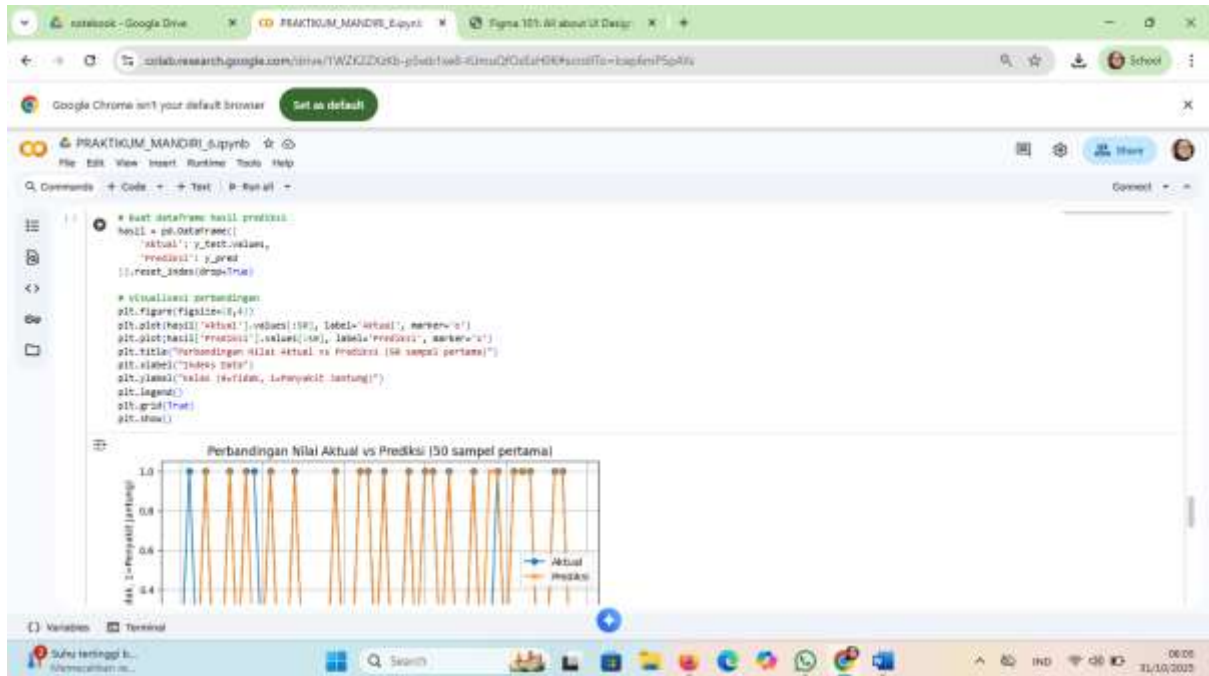
1.8 hasil dari Buat Model SVM & Evaluasi, Buat model SVM, Prediksi data test, Evaluasi hasil

9. Confusion Matrix (dalam bentuk grafik)



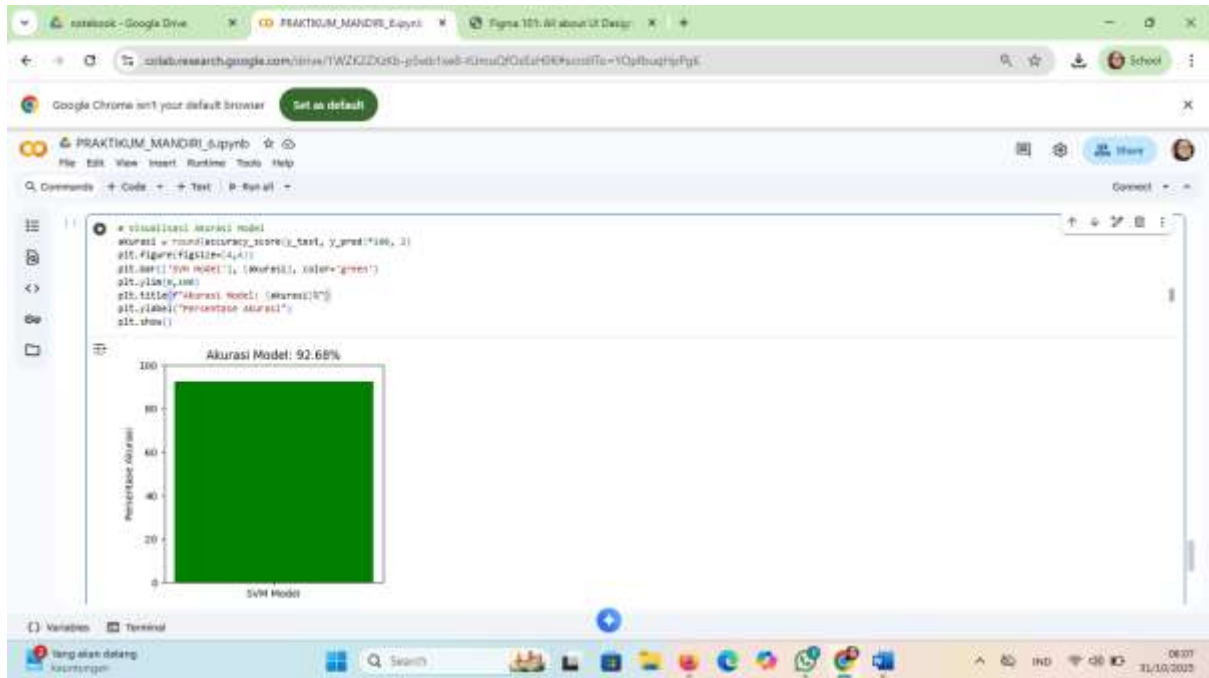
1.9 hasil dari Confusion Matrix (dalam bentuk grafik)

10. Membuat dataframe hasil prediksi



1.10 hasil dari Membuat dataframe hasil prediksi

11. Visualisasi Akurasi Model



1.11 hasil dari Visualisasi Akurasi Model

Referensi:

- Munir, S., Seminar, K. B., Sudradjat, Sukoco, H., & Buono, A. (2022). The Use of Random Forest Regression for Estimating Leaf Nitrogen Content of Oil Palm Based on Sentinel 1-A Imagery. *Information*, 14(1), 10. <https://doi.org/10.3390/info14010010>
- Seminar, K. B., Imantho, H., Sudradjat, Yahya, S., Munir, S., Kaliaana, I., Mei Haryadi, F., Noor Baroroh, A., Supriyanto, Handoyo, G. C., Kurnia Wijayanto, A., Ijang Wahyudin, C., Liyantono, Budiman, R., Bakir Pasaman, A., Rusiawan, D., & Sulastri. (2024). PreciPalm: An Intelligent System for Calculating Macronutrient Status and Fertilizer Recommendations for Oil Palm on Mineral Soils Based on a Precision Agriculture Approach. *Scientific World Journal*, 2024(1). <https://doi.org/10.1155/2024/1788726>

LINK GITHUB: <https://github.com/Sitiaisah1604/machine-learning>