

Tugas 1: Judul tugas – Decision Tree

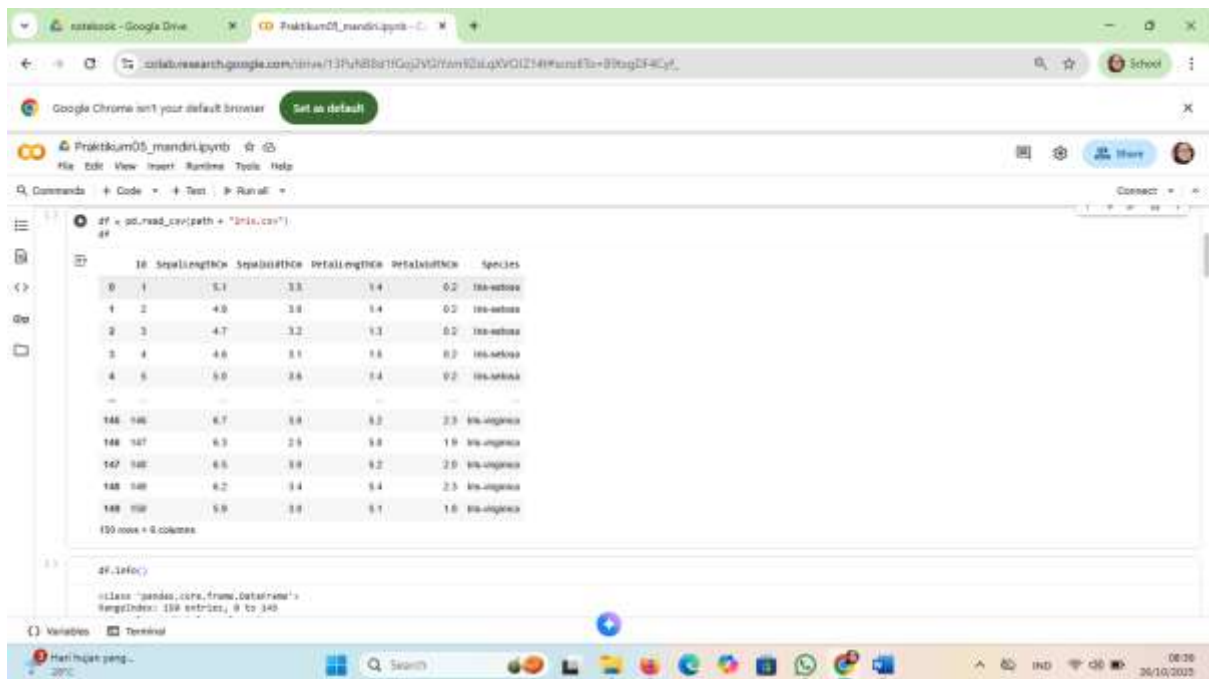
Siti aisah - 0110222129¹

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

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

Abstract. Pembelajaran *Machine Learning* merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan algoritma dan model statistik untuk memungkinkan sistem komputer belajar dari data dan membuat prediksi atau keputusan secara otomatis tanpa pemrograman eksplisit..

1. Membaca data set



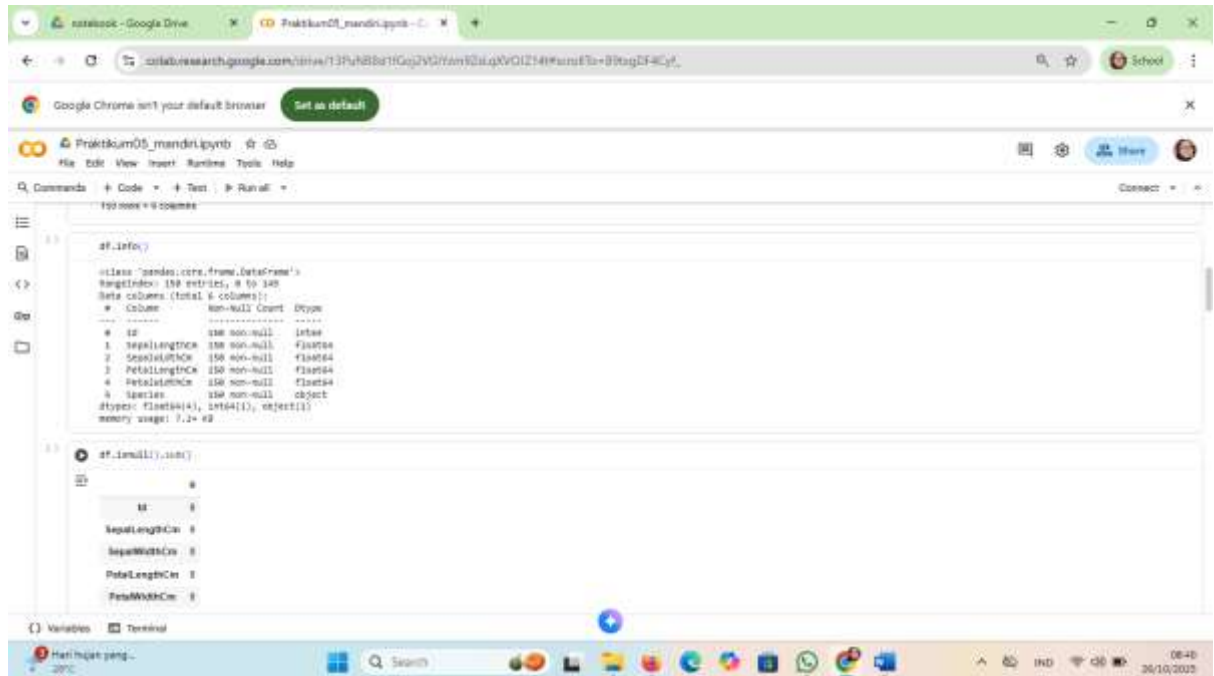
```
df = pd.read_csv(path + "iris.csv")
df
```

ID	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	iris-setosa
1	4.9	3.0	1.4	0.2	iris-setosa
2	4.7	3.2	1.3	0.2	iris-setosa
3	4.6	3.1	1.6	0.2	iris-setosa
4	5.0	3.6	1.4	0.2	iris-setosa
...
146	6.7	3.0	5.2	2.3	iris-virginica
147	6.3	2.9	5.0	1.9	iris-virginica
148	6.5	3.0	5.2	2.0	iris-virginica
149	6.2	3.4	5.4	2.3	iris-virginica
150	5.9	3.0	5.1	1.8	iris-virginica

```
df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150 entries, 0 to 149
```

1.1 . Membaca file csv

2. melihat informasi data

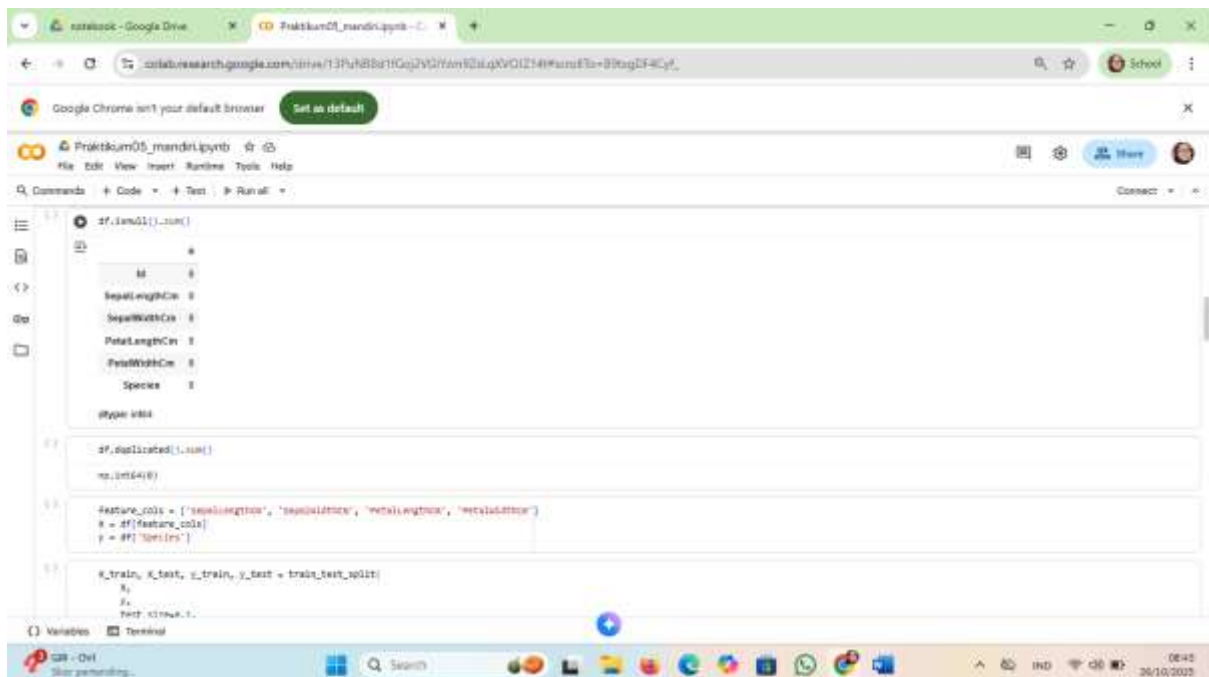


```
df.info()
Out[1]:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 148 entries, 0 to 147
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  --
 0   id            148 non-null    int64   
 1   sepalLengthCm 148 non-null    float64  
 2   sepalWidthCm  148 non-null    float64  
 3   petalLengthCm 148 non-null    float64  
 4   petalWidthCm  148 non-null    float64  
 5   species       148 non-null    object   
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ MB

df.isnull().sum()
Out[1]:
id                0
sepalLengthCm    0
sepalWidthCm     0
petalLengthCm    0
petalWidthCm     0
species          0
dtype: object
```

1.2 Hasil melihat informasi data

3. Cek Missing Value



```
df.isnull().sum()
Out[1]:
id                0
sepalLengthCm    0
sepalWidthCm     0
petalLengthCm    0
petalWidthCm     0
species          0
dtype: object

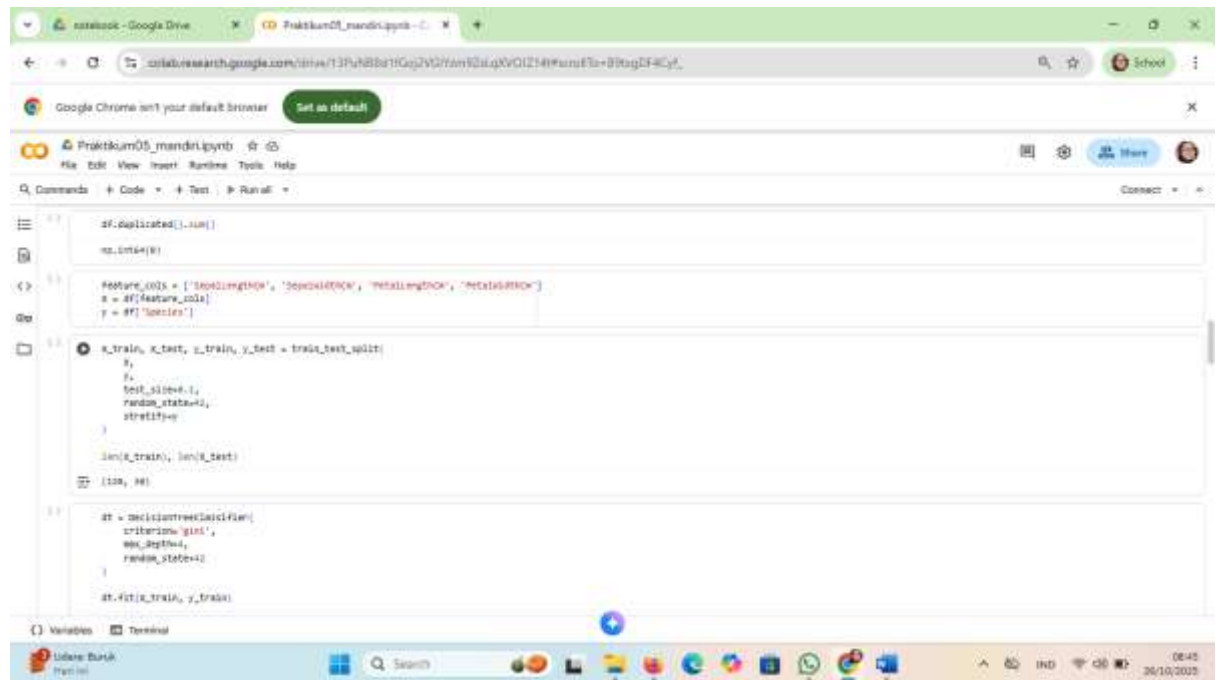
df.dropna(inplace=True)
Out[1]:
0

feature_cols = ['sepalLengthCm', 'sepalWidthCm', 'petalLengthCm', 'petalWidthCm']
x = df[feature_cols]
y = df['species']

x_train, x_test, y_train, y_test = train_test_split(
    x,
    y,
    test_size=0.2,
```

1.3 hasil dari Cek Missing Value

4. Untuk menghitung jumlah data duplikat



```
df.duplicated().sum()

no_duplicates()

feature_cols = ['SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth']
x = df[feature_cols]
y = df['Species']

x_train, x_test, y_train, y_test = train_test_split(
    x,
    y,
    test_size=0.1,
    random_state=42,
    stratify=y
)

len(x_train), len(x_test)

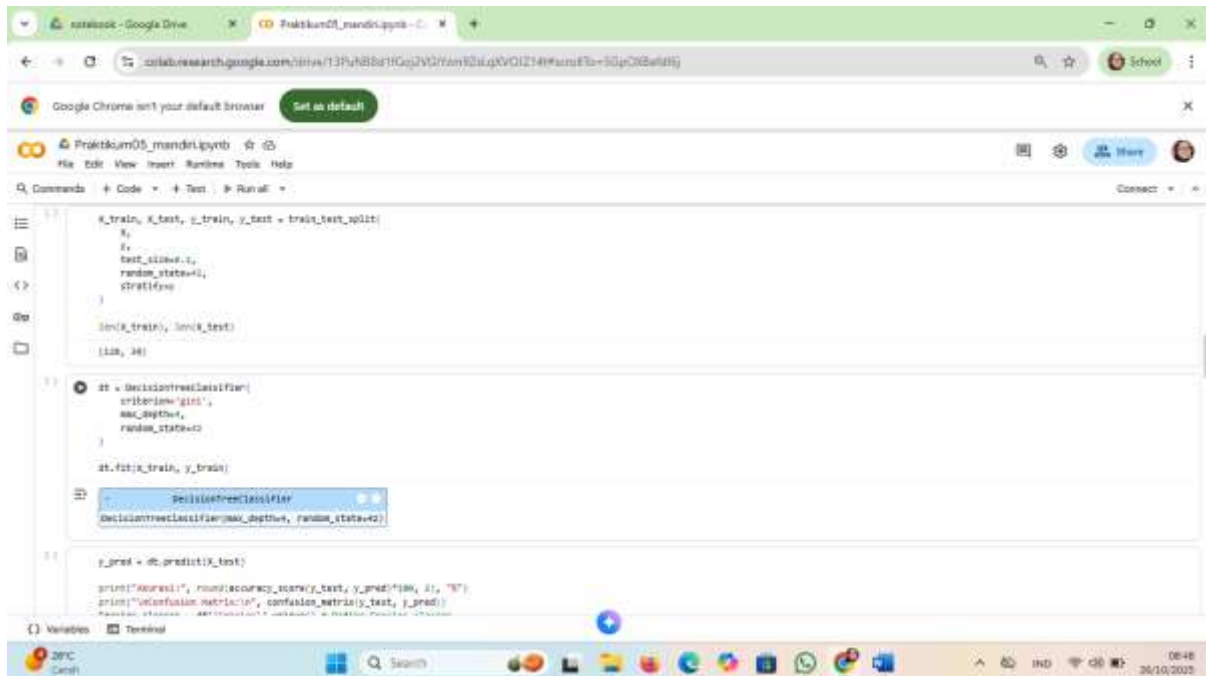
(120, 30)

dt = DecisionTreeClassifier(
    criterion='gini',
    max_depth=4,
    random_state=42
)

dt.fit(x_train, y_train)
```

1.4 hasil dari Untuk menghitung jumlah data duplikat

5. Membuat model Decision Tree



```
x_train, x_test, y_train, y_test = train_test_split(
    x,
    y,
    test_size=0.1,
    random_state=42,
    stratify=y
)

len(x_train), len(x_test)

(120, 30)

dt = DecisionTreeClassifier(
    criterion='gini',
    max_depth=4,
    random_state=42
)

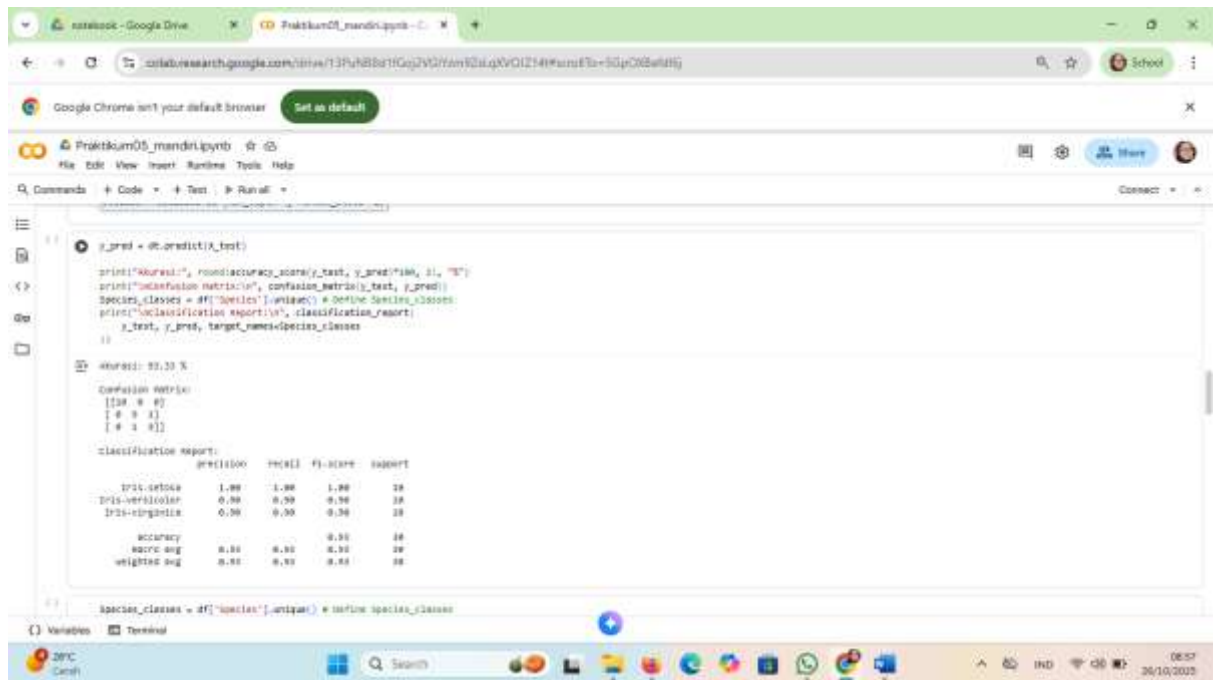
dt.fit(x_train, y_train)

dt.predict(x_test)

print("Accuracy: ", round(accuracy_score(y_test, y_pred), 2), "%")
print("Confusion Matrix: ", confusion_matrix(y_test, y_pred))
```

1.5 Membuat model Decision Tree

6. memprediksi kelas dari data uji berdasarkan model



```
y_pred = dt.predict(X_test)

print("Akurasi:", round(accuracy_score(y_test, y_pred)*100, 2))
print("Confusion matrix:", confusion_matrix(y_test, y_pred))
species_classes = df['species'].unique() # Define Species_classes
print("\nClassification report:\n", classification_report(y_test, y_pred, target_names=species_classes))
```

Akurasi: 93.33 %

Confusion matrix:

```
[[ 3  0  0]
 [ 0  3  0]
 [ 0  0  3]]
```

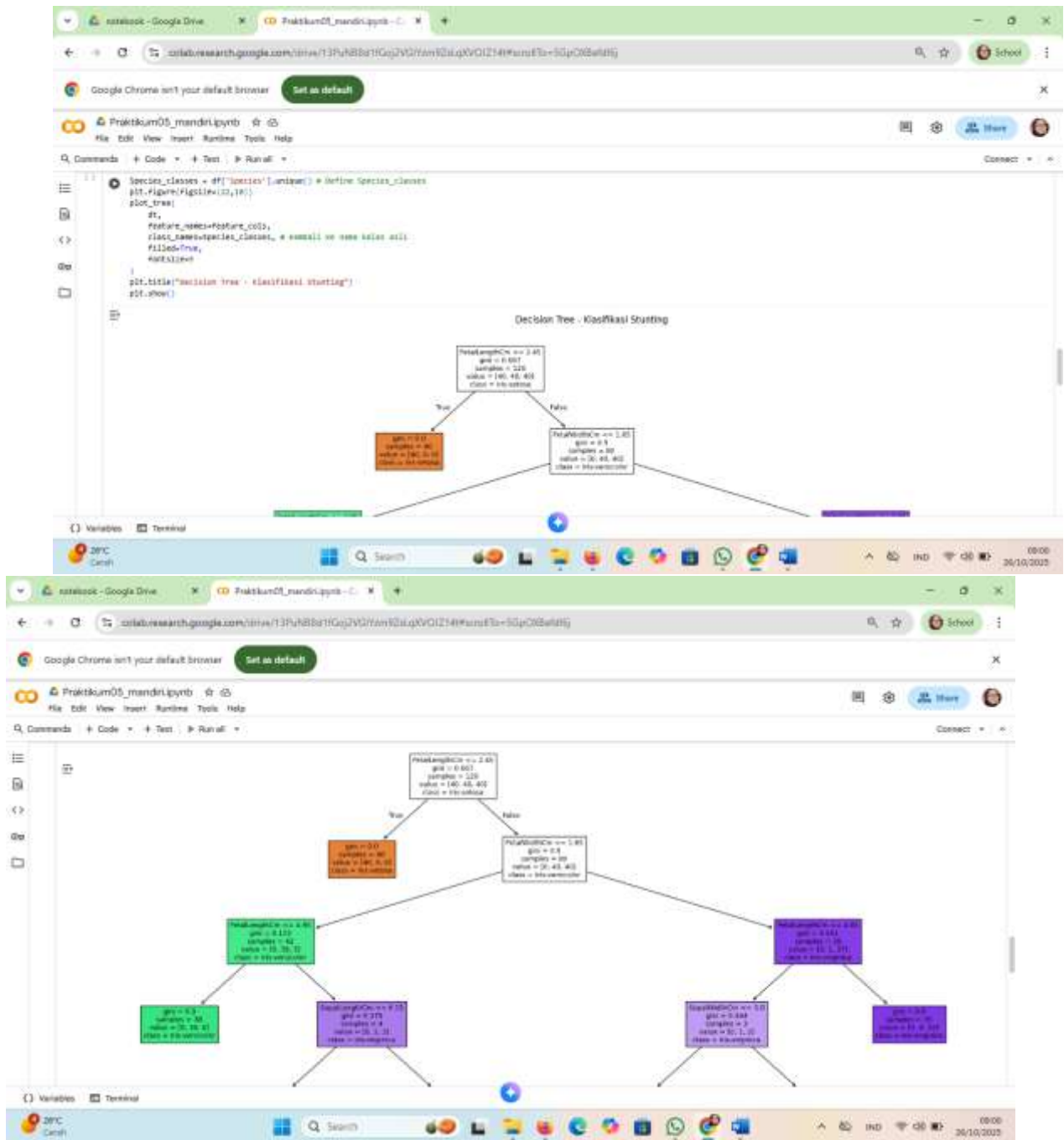
Classification report:

	precision	recall	f1-score	support
iris-setosa	1.00	1.00	1.00	10
iris-versicolour	0.90	0.90	0.90	10
iris-virginica	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.93	0.93	0.93	30
weighted avg	0.93	0.93	0.93	30

```
species_classes = df['species'].unique() # Define Species_classes
```

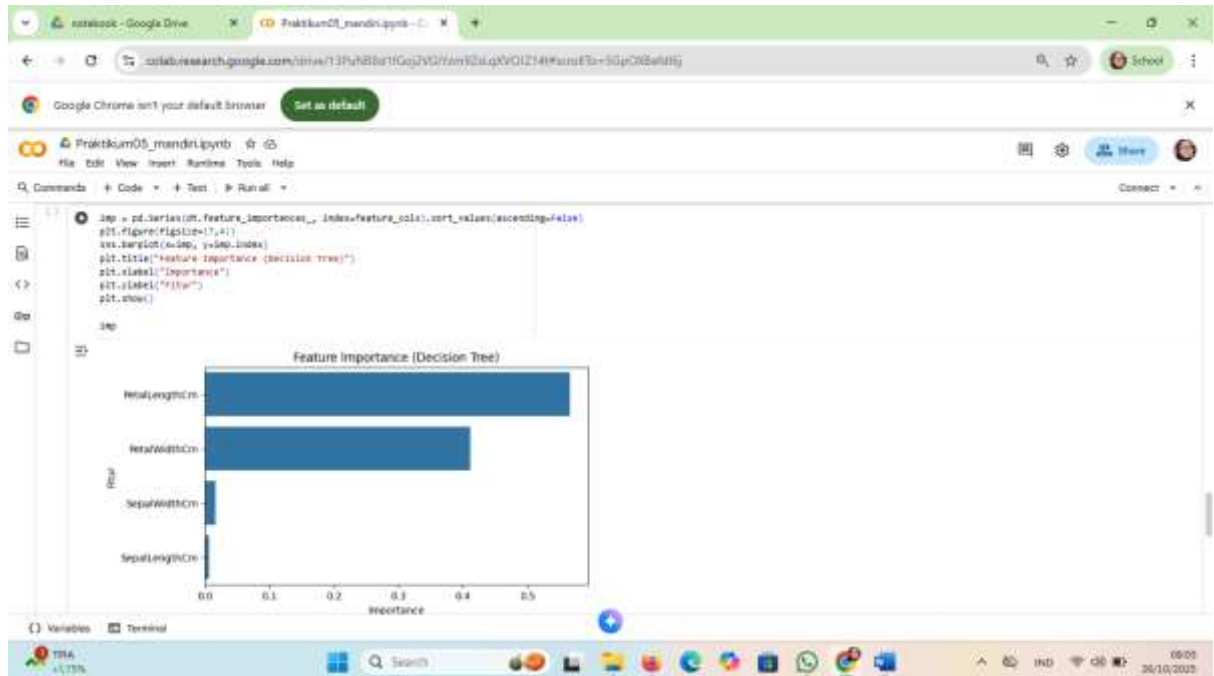
1.6 memprediksi kelas dari data uji berdasarkan model

7. klasifikasi stunting



1.7 hasil klasifikasi stunting

8. untuk menampilkan tingkat kepentingan (importance) setiap fitur dalam model Decision Tree.



1.8 hasil menampilkan tingkat kepentingan (importance) setiap fitur dalam model Decision Tree.

9. mencari nilai terbaik dari parameter

```
1 scores = []
2 for i in range(2, 10):
3     m = DecisionTreeClassifier(max_depth=i, random_state=0)
4     m.fit(x_train, y_train)
5     scores[i] = accuracy_score(y_test, m.predict(x_test))
6
7 scores
8 best_i = max(scores, key=scores.get)
9 print("Best max_depth: (best_i) | acc: (round(scores[best_i]*100, 2))%")
10
11 Best max_depth: 2 | acc: 86.67%
```

1.9 hasil mencari nilai terbaik dari parameter

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., Kaliana, 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/Sitiiaisah1604/machine-learning>