

# Tugas 2: Machine Learning – Decision Tree dan Random Forest

Muhammad Zaidan Ramdhan - 0110222040

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

\*E-mail: [muha22040ti@student.nurulfikri.ac.id](mailto:muha22040ti@student.nurulfikri.ac.id)

Laporan ini bertujuan melakukan penerapan algoritma *Decision Tree* untuk melakukan klasifikasi pada dataset Iris. Dataset ini terdiri dari empat fitur numerik yaitu *Sepal Length (Cm)*, *Sepal Width (Cm)*, *Petal Length (Cm)*, dan *Petal Width (Cm)*, serta satu kolom target berupa nama spesies bunga (*Species*). Karena kolom target masih berbentuk teks, maka dilakukan proses *label encoding* agar data target dapat diubah menjadi bentuk numerik sehingga dapat digunakan oleh model *machine learning*. Selanjutnya, dataset dibagi menjadi dua bagian yaitu 80% data training dan 20% data testing menggunakan fungsi `train_test_split` dari *scikit-learn*.

Model *Decision Tree Classifier* kemudian dilatih menggunakan data training untuk mempelajari pola hubungan antara fitur-fitur numerik dengan label spesies bunga. Setelah proses pelatihan selesai, model diuji menggunakan data testing untuk mengevaluasi performanya. Evaluasi dilakukan dengan menghitung nilai akurasi, menampilkan classification report, serta membuat confusion matrix untuk melihat hasil prediksi setiap kelas. Hasil evaluasi menunjukkan bahwa model mampu melakukan klasifikasi dengan baik terhadap data uji yang diberikan.

## 1. Tugas mandiri – membuat model Decision Tree

```
1. Melakukan mounting G-Drive

from google.colab import drive
drive.mount("/content/gdrive")

Mounted at /content/gdrive
```

Gambar 1. 1

Pada *Gambar 1.1* di atas, merupakan sebuah code untuk mounted atau menghubungkan google colab dengan google drive.

## 2. Membuat path untuk dataset

```
path = "/content/gdrive/MyDrive/machine_learning/pertemuan04"
```

*Gambar 1. 2*

Pada *Gambar 1.2* di atas, kita membuat sebuah path untuk mengatur folder yang akan digunakan, dalam case ini kita menggunakan folder pertemuan04.

## 3. Import seluruh library yang akan digunakan

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
```

*Gambar 1. 3*

Pada *Gambar 1.3* di atas, merupakan sebuah code untuk import beberapa library yang akan kita gunakan.

#### 4. Load dataset

```
df = pd.read_csv(path + "/praktikum_mandiri/data/Iris.csv")
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: [Generate code with df](#) [New interactive sheet](#)

Gambar 1. 4

Pada *Gambar 1.4* di atas, merupakan code untuk menampilkan dataset pada sebuah tabel menggunakan dataset calonpembelimobil.csv dan hanya menampilkan 5 data.

- **df = pd.read\_csv(path + '/praktikum\_mandiri/data/Iris.csv)** merupakan sebuah code untuk membaca file csv yang terdapat pada folder data dengan diikuti file Iris.csv.

#### 5. Membuat variabel X dan y

```
X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
y = df['Species']
```

Gambar 1. 5

Pada *Gambar 1.5* di atas, merupakan sebuah code untuk memisahkan variabel X dan Y.

## 6. Melakukan encode ke bentuk numerik pada variabel y

```
label_encoder = LabelEncoder()  
y_encoded = label_encoder.fit_transform(y)
```

Gambar 1. 6

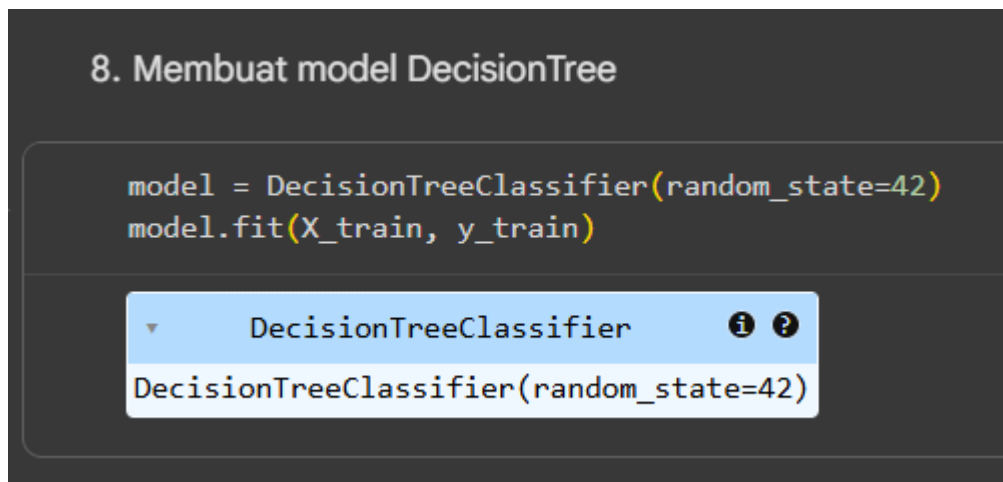
Pada *Gambar 1.6* di atas, merupakan sebuah code untuk melakukan encode column dataset menjadi numerik.

## 7. Membagi dataset menjadi data training dan testing

```
X_train, X_test, y_train, y_test = train_test_split(  
    X,  
    y_encoded,  
    test_size=0.2,  
    random_state=42  
)
```

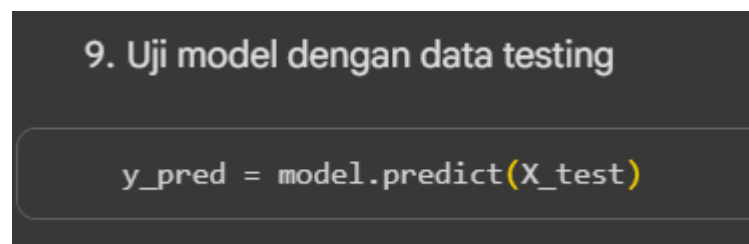
Gambar 1. 7

Pada *Gambar 1.7* di atas, merupakan sebuah code untuk melakukan pembagian data training dan data testing.



Gambar 1. 8

Pada *Gambar 1.8* di atas, membuat model DecisionTree.



Gambar 1. 9

Pada *Gambar 1.9* di atas, merupakan sebuah code untuk menguji model dengan data testing.

## 10. Evaluasi model

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

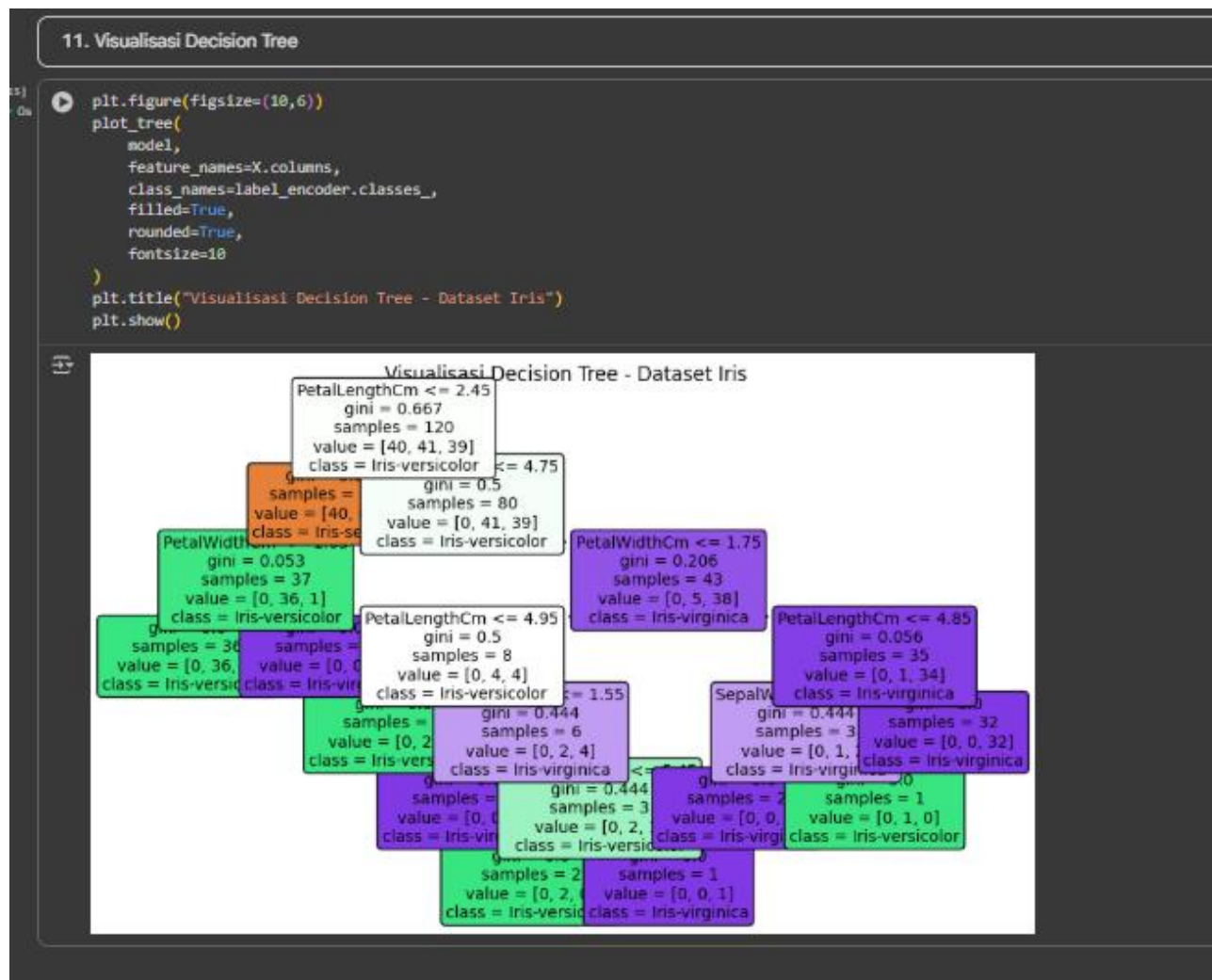
```
Classification Report:
              precision    recall  f1-score   support

     0       1.00      1.00      1.00        10
     1       1.00      1.00      1.00         9
     2       1.00      1.00      1.00        11

 accuracy          1.00          1.00      1.00        30
 macro avg          1.00          1.00      1.00        30
weighted avg          1.00          1.00      1.00        30
```

Gambar 1. 9

Pada *Gambar 1.9* di atas, merupakan sebuah code untuk evaluasi model dengan data uji atau testing.



Gambar 1. 10

Pada Gambar 1.10 di atas, merupakan sebuah kode membuat visualisasi dari decision tree.

Link github praktikum:

[https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan05/praktikum\\_05/notebooks/praktikum\\_05.ipynb](https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan05/praktikum_05/notebooks/praktikum_05.ipynb)

Link github tugas:

[https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan05/praktikum\\_mandiri/notebooks/latihan\\_05.ipynb](https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan05/praktikum_mandiri/notebooks/latihan_05.ipynb)