

Digital Fair Skill 35.0 - Data Science



# Basic Machine Learning Using Iris Datasets From Scikit-learn Library

BY FIKRI NENDRA  
FADHLURRAHMAN

# About me

Halo !!

Saya Fikri Nendra Fadhlurrahman. Saya adalah seorang mahasiswa Teknik Informatika Di Universitas Negeri Surabaya. Saat ini saya sedang menjalani Perkuliahan di semester 4



# Iris Datasets

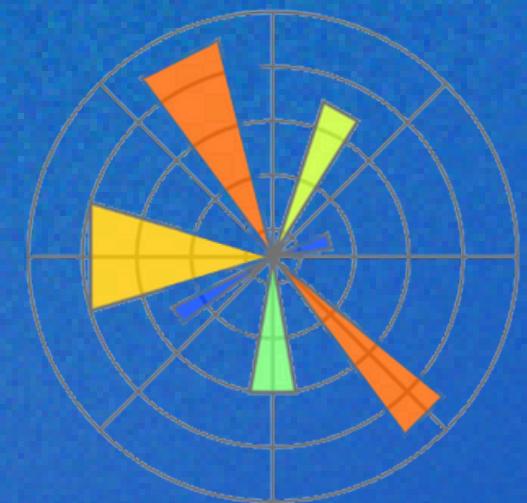
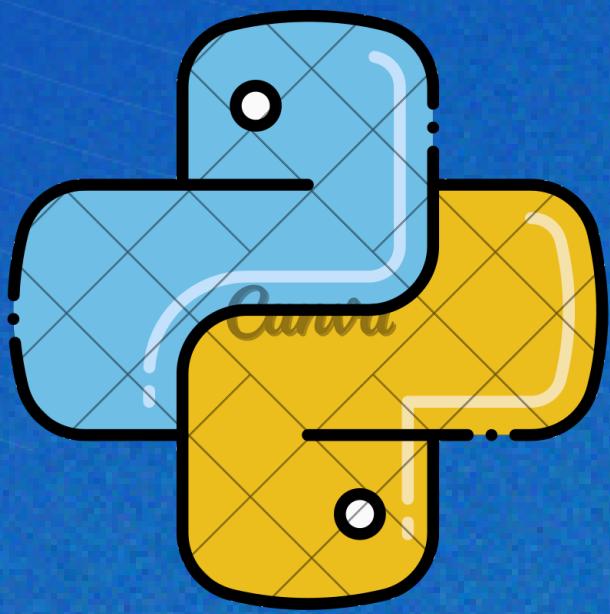
IRIS DATASET DI SCIKIT-LEARN ADALAH KUMPULAN DATA KLASIK DALAM PEMBELAJARAN MESIN YANG SERING DIGUNAKAN UNTUK KLASIFIKASI. DATASET INI BERISI INFORMASI TENTANG 3 SPESIES BUNGA IRIS (IRIS SETOSA, IRIS VERSICOLOR, DAN IRIS VIRGINICA), DENGAN 150 SAMPEL DAN 4 FITUR UNTUK SETIAP SAMPEL, YAITU:

1. SEPAL LENGTH (PANJANG KELOPAK BUNGA)
2. SEPAL WIDTH (LEBAR KELOPAK BUNGA)
3. PETAL LENGTH (PANJANG MAHKOTA BUNGA)
4. PETAL WIDTH (LEBAR MAHKOTA BUNGA)

SETIAP SAMPEL MEMILIKI LABEL KELAS YANG MENUNJUKKAN SPESIESNYA. DATASET INI SERING DIGUNAKAN UNTUK MENGUJI ALGORITMA KLASIFIKASI SEPERTI K-NEAREST NEIGHBORS (KNN), DECISION TREE, SVM, DAN NAIVE BAYES.



# Tools Used



*matplotlib*



## # 1 Input Library dan Membaca Dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

target_names = iris.target_names
df['target_name'] = df['target'].apply(lambda x: target_names[x])

print("5 Data Teratas:")
print(df.head())
```

# INPUT LIBRARY & READ DATA

5 Data Teratas:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm) \
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

	target	target_name
0	0	setosa
1	0	setosa
2	0	setosa
3	0	setosa
4	0	setosa

# EKSPLORASI & ANALISIS

```
df.info()  
[8] ✓ 0.0s  
... <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 5 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   sepal length (cm)  150 non-null   float64  
 1   sepal width (cm)   150 non-null   float64  
 2   petal length (cm)  150 non-null   float64  
 3   petal width (cm)   150 non-null   float64  
 4   target            150 non-null   int64  
dtypes: float64(4), int64(1)  
memory usage: 6.0 KB
```

```
df['target'].unique()  
[9] ✓ 0.0s  
... array([0, 1, 2])
```

# DATA

```
df.describe()  
[10] ✓ 0.0s  
...  


|       | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target     |
|-------|-------------------|------------------|-------------------|------------------|------------|
| count | 150.000000        | 150.000000       | 150.000000        | 150.000000       | 150.000000 |
| mean  | 5.843333          | 3.057333         | 3.758000          | 1.199333         | 1.000000   |
| std   | 0.828066          | 0.435866         | 1.765298          | 0.762238         | 0.819232   |
| min   | 4.300000          | 2.000000         | 1.000000          | 0.100000         | 0.000000   |
| 25%   | 5.100000          | 2.800000         | 1.600000          | 0.300000         | 0.000000   |
| 50%   | 5.800000          | 3.000000         | 4.350000          | 1.300000         | 1.000000   |
| 75%   | 6.400000          | 3.300000         | 5.100000          | 1.800000         | 2.000000   |
| max   | 7.900000          | 4.400000         | 6.900000          | 2.500000         | 2.000000   |


```

+ Code

+ Markdown

# SPLIT DATA

```
# 2 Split Data (80% Training - 20% Testing)
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=100)
```

- Fungsi ini berasal dari Scikit-Learn (`sklearn.model_selection.train_test_split`) dan digunakan untuk membagi dataset menjadi data latih dan data uji.

## Parameter yang Digunakan

- `X = iris.data`  
→ Variabel independen (fitur: panjang & lebar sepal/petal).
- `y = iris.target`  
→ Variabel dependen (label kelas: Setosa, Versicolor, Virginica).
- `test_size=0.2`  
→ 20% data digunakan sebagai data uji, sedangkan 80% sebagai data latih.
- `random_state=100`  
→ Menjaga konsistensi pembagian data. Jika kode dijalankan ulang, hasil pembagian tetap sama.

### 3. Training Model

```
# 3 Training Model & Prediksi  
model=DecisionTreeClassifier(random_state=100)  
model.fit(X_train, y_train)
```

▼ DecisionTreeClassifier i ?

```
DecisionTreeClassifier(random_state=100)
```

Decision Tree Classifier adalah algoritma machine learning berbasis pohon keputusan yang digunakan untuk klasifikasi.

- ◆ Ide dasar: Membagi data menjadi subgrup berdasarkan fitur tertentu sampai didapatkan keputusan akhir.

- ◆ Output: Model berbentuk struktur pohon dengan node dan cabang.

💡 Contoh Sederhana:

Bagaimana cara menentukan spesies bunga berdasarkan panjang dan lebar petalnya?

1. Jika petal panjang  $\leq 2.45$  cm  $\rightarrow$  Iris Setosa

2. Jika tidak, periksa petal width untuk menentukan antara Versicolor atau Virginica

# TRAINING MODEL

```
y_pred = model.predict(X_test)
print("nilai prediksi:", y_pred)
print('nilai sebenarnya:', y_test)
print('\n')
# Evaluasi Model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred, target_names=iris.target_names)

print("Akurasi Model:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

# PREDIKSI DAN EVALUSI MODEL

```
nilai prediksi: [2 0 2 0 2 2 0 0 2 0 0 2 0 0 2 1 1 2 2 2 2 0 2 0 1 2 1 0 1 2]
nilai sebenarnya: [2 0 2 0 2 2 0 0 2 0 0 2 0 0 2 1 1 1 2 2 2 0 2 0 1 2 1 0 1 2]
```

- Model memprediksi hampir semua dengan benar.
- Ada satu kesalahan pada indeks ke-17, di mana model memprediksi kelas 2, tetapi sebenarnya kelas 1.

Akurasi Model: 0.9666666666666667

- Interpretasi: Model berhasil mengklasifikasikan 96.67% data uji dengan benar.

## Confusion Matrix:

```
[[11  0  0]
 [ 0  5  1]
 [ 0  0 13]]
```

### Struktur Confusion Matrix:

	Prediksi Setosa (0)	Prediksi Versicolor (1)	Prediksi Virginica (2)
Setosa (0)	11 (✓)	0 (✗)	0 (✗)
Versicolor (1)	0 (✗)	5 (✓)	1 (✗)
Virginica (2)	0 (✗)	0 (✗)	13 (✓)

- 11 data Setosa diprediksi benar sebagai Setosa.
- 5 data Versicolor diprediksi benar sebagai Versicolor, tapi 1 salah sebagai Virginica.
- 13 data Virginica diprediksi benar sebagai Virginica.

Classification Report:				
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	11
versicolor	1.00	0.83	0.91	6
virginica	0.93	1.00	0.96	13
accuracy			0.97	30
macro avg	0.98	0.94	0.96	30
weighted avg	0.97	0.97	0.97	30

Recall → Seberapa banyak data kelas tertentu yang diklasifikasikan dengan benar?

- Setosa: 1.00 (Semua data Setosa diklasifikasikan dengan benar).
- Versicolor: 0.83 (1 data Versicolor diklasifikasikan sebagai Virginica).
- Virginica: 1.00 (Semua data Virginica diklasifikasikan dengan benar).

Precision → Seberapa banyak prediksi yang benar dibandingkan total prediksi kelas tersebut?

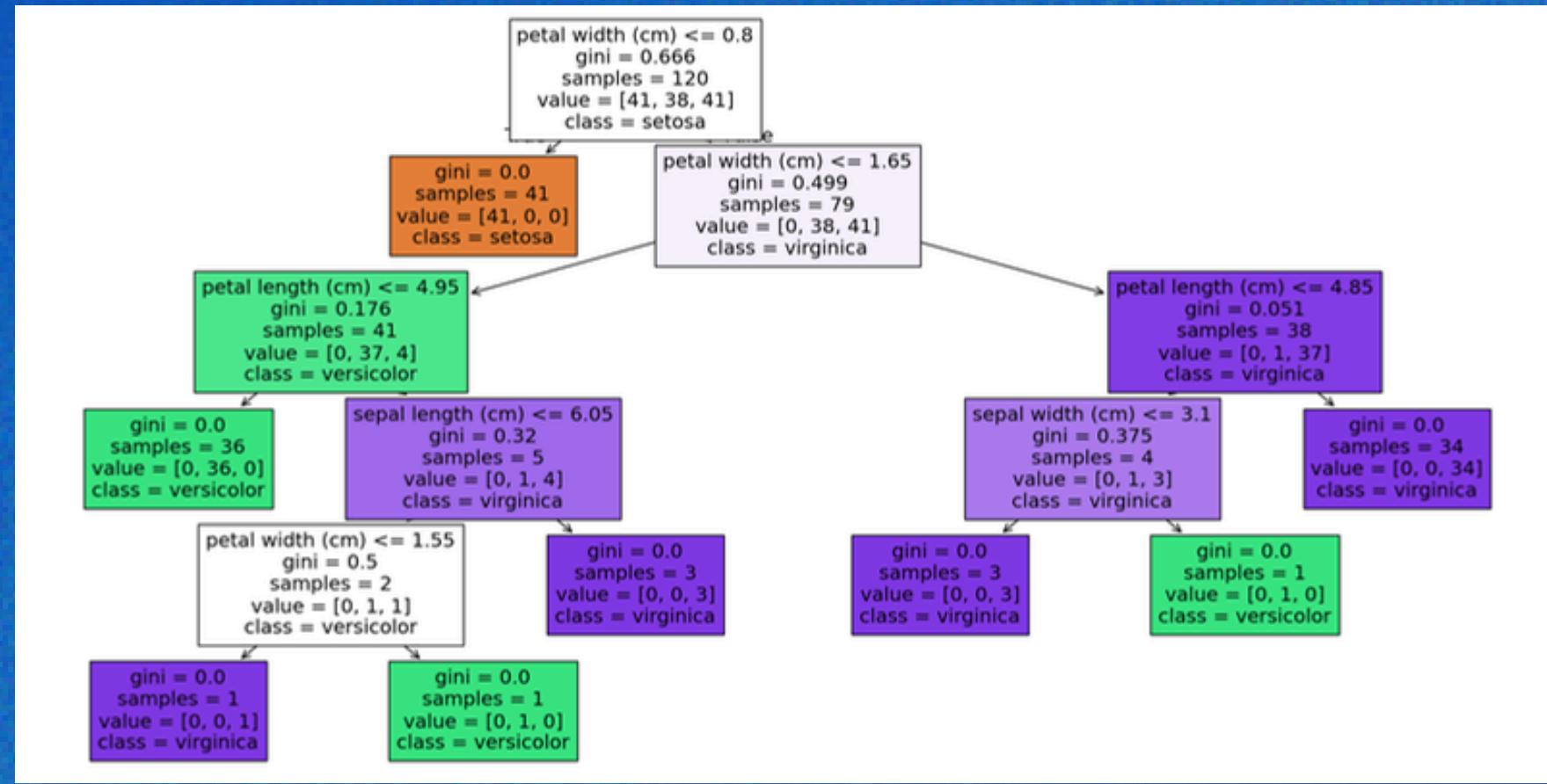
- Setosa: 1.00 (Semua yang diprediksi sebagai Setosa benar).
- Versicolor: 1.00 (Semua prediksi Versicolor benar).
- Virginica: 0.93 (Ada satu data yang seharusnya Versicolor, tetapi diprediksi sebagai Virginica).

F1-score → Rata-rata Precision & Recall (semakin tinggi, semakin baik).

- Setosa: 1.00
- Versicolor: 0.91
- Virginica: 0.96

Support → Jumlah data sebenarnya dalam setiap kelas.

# VISUALISASI DATA DECISION TREE



Decision tree ini menunjukkan bahwa:

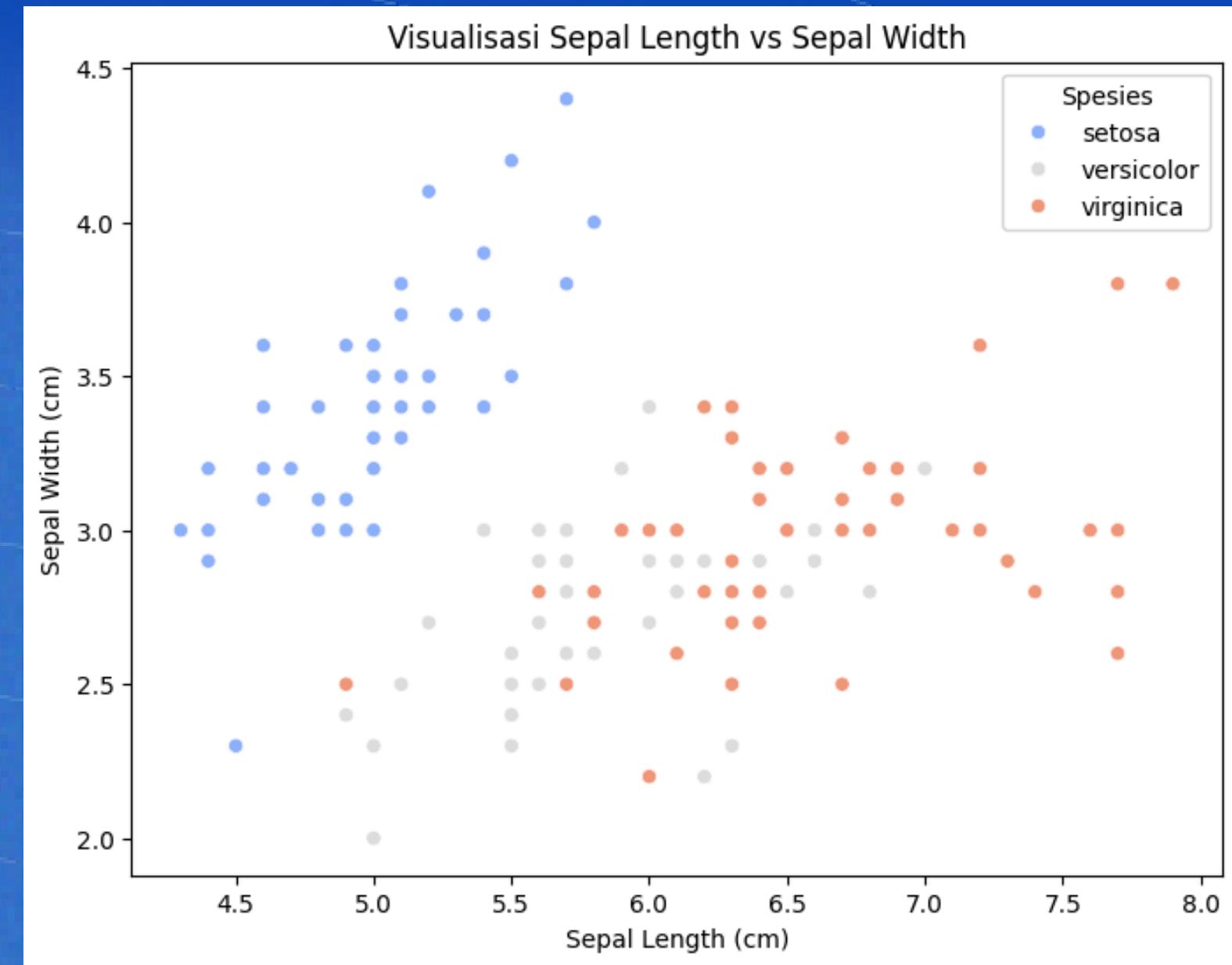
- Iris setosa dapat dipisahkan dengan sempurna menggunakan petal width  $\leq 0.8$  cm
- Pemisahan versicolor dan virginica lebih kompleks dan membutuhkan beberapa level keputusan
- Semakin ke bawah, node memiliki nilai Gini yang semakin kecil, menunjukkan pemisahan yang semakin baik

Pohon keputusan ini merupakan model yang cukup efektif untuk klasifikasi Iris karena mampu mencapai pemisahan yang baik dengan aturan-aturan yang relatif sederhana dan mudah diinterpretasi.

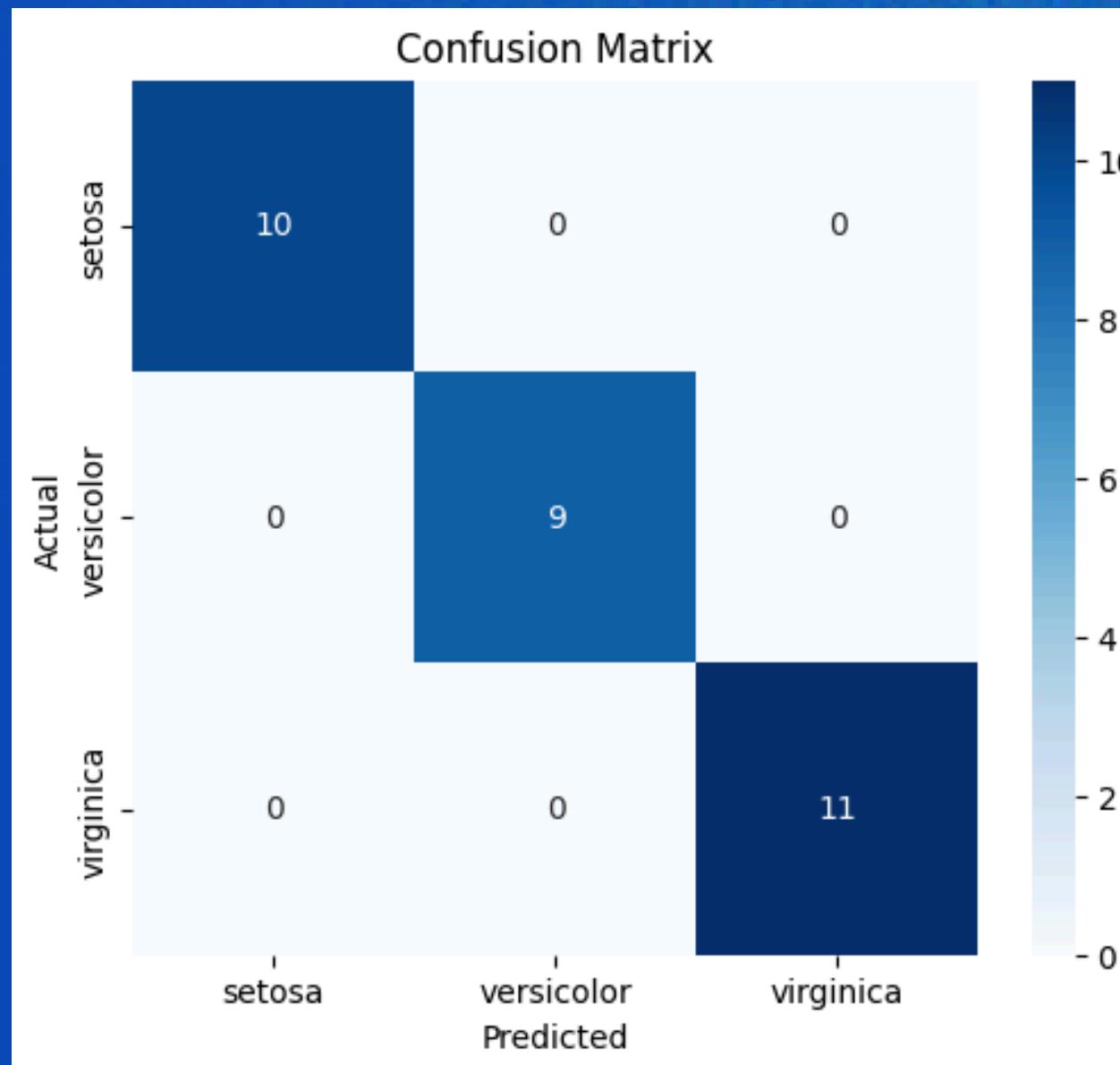
# VISUALISASI DATA SCATTER PLOT

- Terdapat pemisahan yang jelas untuk Iris setosa
- Versicolor dan virginica memiliki overlap yang signifikan
- Ada korelasi positif lemah antara sepal length dan width untuk setosa
- Setiap spesies memiliki karakteristik morfologi yang berbeda
- Plot ini menunjukkan bahwa sepal length dan width saja tidak cukup untuk memisahkan semua spesies dengan sempurna

Plot ini sangat berguna untuk memahami karakteristik morfologi dari ketiga spesies Iris dan bagaimana mereka dapat dibedakan berdasarkan ukuran sepal mereka.



# VISUALISASI DATA CONFUSION MATRIX



## Performa Model:

- Model memiliki akurasi sempurna (100%)
- Tidak ada false positives atau false negatives
- Semua kelas dapat diklasifikasikan dengan benar
- Total sampel yang diuji: 30 ( $10 + 9 + 11$ )

Confusion matrix ini menunjukkan bahwa model klasifikasi bekerja sangat baik dalam membedakan ketiga spesies Iris, tanpa ada kesalahan klasifikasi sama sekali. Ini mengindikasikan bahwa fitur-fitur yang digunakan (seperti panjang dan lebar sepal/petal) sangat efektif dalam membedakan ketiga spesies Iris.



# TERIMA KASIH

Digital Fair Skill 35.0 - Data Science