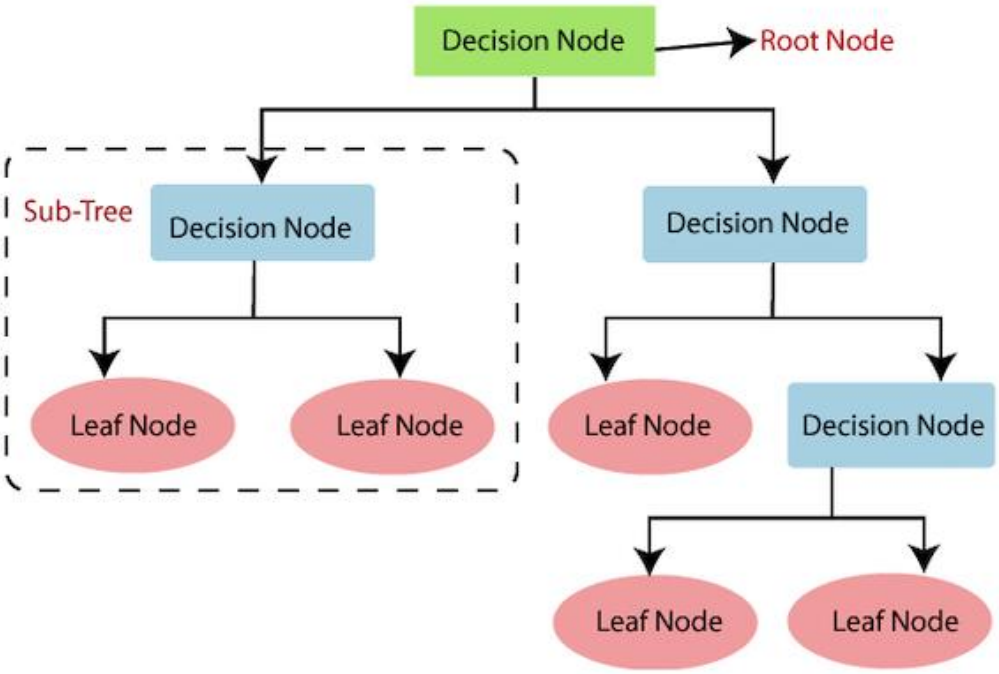
NIM	: 2311523007	Tanggal	: Senin, 03 Maret 2025
Nama	: Vania Zhafira Zahra	Asisten	: Annisa Nurul Hakim Dhiya Gustita Aqila
Mata Kuliah	: Praktikum Data Mining		
Modul	: 03		
Kelas	: A		

Intruksi “Decision Tree”

Decision Tree adalah sebuah cara/pemikiran/pembuatan keputusan yang berbentuk sekumpulan simpul seperti pohon yang dapat memberikan suatu jawaban dari beberapa pilihan Tindakan. Biasanya decision tree dimulai dari satu node atau satu simpul. Kemudian node tersebut bercabang untuk memberikan pilihan-pilihan Tindakan yang lain. Selanjutnya node tersebut akan memiliki cabang-cabang baru. Dalam pembuatan node atau cabang baru akan terus di ulang sampai kriteria berhenti dipenuhi. Decision tree biasanya dapat memperoses dataset yang berisi atribut nominal atau numerik. Label attribute harus berbentuk nominal untuk proses klasifikasi dan berbentuk numerik untuk regresi.



Struktur Decision Trees terdiri dari beberapa elemen utama yang bekerja bersama untuk menghasilkan keputusan atau prediksi. Berikut adalah elemen-elemen tersebut

- Elemen
- Deskripsi
- Root node
- Titik awal pohon yang mewakili keseluruhan dataset
- Internal nodes
- Titik percabangan yang mewakili pengujian atribut
- Branches
- Hasil dari pengujian atribut yang menghubungkan antar node
- Leaf nodes
- Titik akhir pohon yang mewakili keputusan atau hasil klasifikasi

Node Keputusan (Decision Nodes)

Node keputusan adalah simpul yang digunakan untuk menentukan atribut mana yang akan diuji pada setiap langkah. Anda dapat menganggapnya sebagai titik di mana pohon mulai bercabang.

Cabang (Branches)

Cabang adalah jalur yang menghubungkan node satu dengan node lainnya. Cabang ini menunjukkan hasil dari pengujian atribut pada node sebelumnya.

Node Daun (Leaf Nodes)

Node daun adalah simpul terakhir dalam pohon. Node ini mewakili hasil akhir, seperti kategori dalam klasifikasi atau nilai numerik dalam regresi.

Root Node (Node Akar)

Root Node adalah node pertama atau node utama dalam Decision Tree. Node ini mewakili fitur atau atribut yang pertama kali digunakan untuk membagi data.

Parent Node (Node Induk) & Child Node (Node Anak)

Parent Node adalah node yang memiliki cabang (anak/child node). Node ini merupakan hasil dari pembagian (split) root node atau node lainnya dalam pohon. Child Node adalah node yang dihasilkan dari pembagian parent node. Setiap parent node dapat memiliki dua atau lebih child node, tergantung pada jumlah kategori atau kondisi dalam dataset.

Proses Kerja Decision Tree

1. Pengumpulan dan Persiapan Data
2. Pembentukan Decision Tree
3. Pruning (Pemangkasan)
4. Evaluasi Model
5. Implementasi Model

Keunggulan Model Decision Tree

1. Mudah Dipahami dan Diinterpretasikan
2. Cocok untuk Data Non-linier
3. Tidak Memerlukan Normalisasi Data
4. Mampu Menangani Variabel Kategorikal & Data Numerik
5. Dapat Digunakan untuk Klasifikasi atau Regresi
6. Mengidentifikasi Variabel Paling Informatif

Kelemahan Model Desicion Tree

1. Rentan terhadap Overfitting
2. Sensitif terhadap Perubahan Data
3. Kurang Efektif untuk Data Kontinu
4. Sensitif terhadap Noise
5. Memerlukan Tuning Parameter

Entropy

Entropy adalah ukuran ketidakpastian atau keacakan dalam suatu dataset. Dalam Decision Tree, entropy digunakan untuk menentukan seberapa murni (homogen) suatu kelompok data, yang membantu dalam proses pemisahan (splitting) node.

Information Gain

Information Gain (IG) adalah ukuran yang digunakan untuk menentukan fitur terbaik dalam membagi data pada Decision Tree. IG mengukur seberapa besar penurunan entropy setelah suatu atribut digunakan untuk membagi data.

Penggunaan Entropy dan Information Gain dalam Decision Tree

a. Hitung entropy dataset saat ini
b. Untuk setiap atribut, hitung information gain yang diperoleh jika kita memisahkan data berdasarkan atribut tersebut
c. Pilih atribut dengan information gain tertinggi sebagai node keputusan
d. Ulangi proses secara rekursif untuk setiap subset data hasil pemisahan

Nomor 1

Dataset "ikan.csv" terdiri dari 4 atribut yaitu panjang tubuh, lebar tubuh, panjang sirip, dan lebar sirip. Terdapat beberapa spesies ikan sebagai kelas target pada dataset ini. Data ini dipakai untuk masalah klasifikasi, di mana kita bisa memprediksi spesies dari sebuah ikan berdasarkan atribut-atribut yang diberikan.

Tahapan yang ada pada tugas ini antara lain:

1. Load dataset 'ikan.csv' dan ubah ke dalam dataframe.

input

```
#import library
import pandas as pd
ikan = pd.read_csv('ikan.csv')
ikan
```

Output

	panjang_tubuh	lebar_tubuh	panjang_sirip	lebar_sirip	spesies
0	5.1	3.5	1.4	0.2	Ikan Nila
1	4.9	3.0	1.4	0.2	Ikan Nila
2	4.7	3.2	1.3	0.2	Ikan Nila
3	4.6	3.1	1.5	0.2	Ikan Nila
4	5.0	3.6	1.4	0.2	Ikan Nila
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Ikan Lele
146	6.3	2.5	5.0	1.9	Ikan Lele
147	6.5	3.0	5.2	2.0	Ikan Lele
148	6.2	3.4	5.4	2.3	Ikan Lele
149	5.9	3.0	5.1	1.8	Ikan Lele

150 rows × 5 columns

2. Pisahkan antara atribut (fitur) dan label (spesies ikan).

```
# Pisahkan antara atribut (fitur) dan label (spesies ikan)
x = ikan [['panjang_tubuh','lebar_tubuh','panjang_sirip','lebar_sirip']]
y = ikan ['spesies']

# Split data untuk training dan testing
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
```

3. Buat dan latih model Decision Tree.

```
# Buat dan latih model Decision Tree
from sklearn.tree import DecisionTreeClassifier

tree_model = DecisionTreeClassifier()
tree_model = tree_model.fit(x_train, y_train)
```

4. Lakukan prediksi dengan model yang telah dilatih.

input

```
#Prediksi model yang telah dilatih
y_pred = tree_model.predict(x_test)
print(y_pred)
```

Output

```
['Ikan Mas' 'Ikan Nila' 'Ikan Lele' 'Ikan Mas' 'Ikan Mas' 'Ikan Nila'
 'Ikan Mas' 'Ikan Lele' 'Ikan Mas' 'Ikan Mas' 'Ikan Lele' 'Ikan Nila'
 'Ikan Nila' 'Ikan Nila' 'Ikan Nila' 'Ikan Mas' 'Ikan Lele' 'Ikan Mas'
 'Ikan Mas' 'Ikan Lele' 'Ikan Nila' 'Ikan Lele' 'Ikan Nila' 'Ikan Lele'
 'Ikan Lele' 'Ikan Lele' 'Ikan Lele' 'Ikan Lele' 'Ikan Nila' 'Ikan Nila']
```

5. Hitung akurasi model.

Input

```
#hitung akurasi
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)
print ("akurasi:", accuracy)
```

Output

```
akurasi: 1.0
```

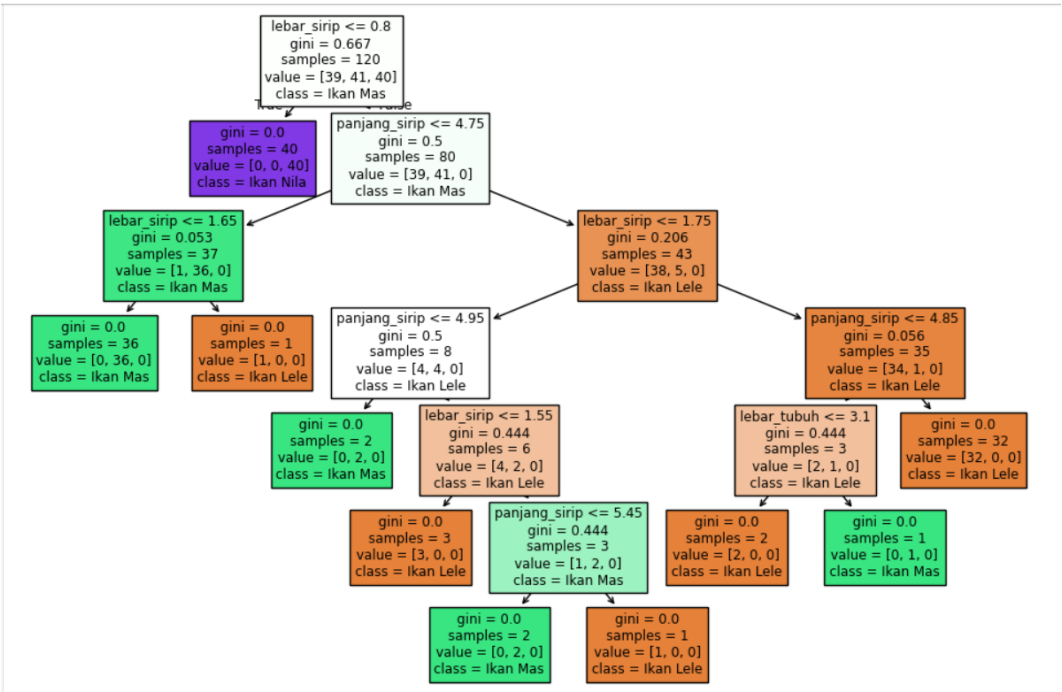
6. Visualisasi model Decision Tree yang telah dilatih.

Input

```
# Visualisasi Decision Tree
import matplotlib.pyplot as plt
from sklearn import tree

plt.figure(figsize = (12, 8))
tree.plot_tree(tree_model, feature_names = x.columns, class_names = tree_model.classes_, filled = True)
plt.show()
```

Output



7. Tarik kesimpulan yang diperoleh dari model.

Input

```python
# Kesimpulan Model
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, tree_model.predict(x_test))

print("=== Kesimpulan Decision Tree ===")
print(f"Jumlah Node: {tree_model.tree_.node_count}")
print(f"Kedalaman Maksimum: {tree_model.tree_.max_depth}")
print(f"Akurasi Model: {accuracy:.2f}")

# Menampilkan fitur yang paling berpengaruh
import numpy as np
importances = tree_model.feature_importances_
feature_importance_df = pd.DataFrame({
    'Fitur': x.columns,
    'Pentingnya': importances
}).sort_values(by="Pentingnya", ascending=False)

print("\nFitur yang paling berpengaruh dalam klasifikasi ikan:")
print(feature_importance_df.to_string(index=False))

# Evaluasi performa model berdasarkan akurasi
if accuracy > 0.8:
    print("\nModel memiliki performa yang baik dalam mengklasifikasikan spesies ikan.")
elif accuracy > 0.6:
    print("\nModel cukup baik, namun masih bisa ditingkatkan.")
else:
    print("\nModel kurang akurat, perlu optimasi lebih lanjut.")

# Analisis visualisasi decision tree
print("\nBerdasarkan visualisasi pohon keputusan, fitur utama dalam klasifikasi adalah fitur dengan nilai penting tertinggi.")
```

Output

```
=== Kesimpulan Decision Tree ===
Jumlah Node: 19
Kedalaman Maksimum: 6
Akurasi Model: 1.00

Fitur yang paling berpengaruh dalam klasifikasi ikan:
        Fitur  Pentingnya
  lebar_sirip    0.577395
panjang_sirip    0.405935
  lebar_tubuh    0.016670
panjang_tubuh    0.000000

Model memiliki performa yang baik dalam mengklasifikasikan spesies ikan.

Berdasarkan visualisasi pohon keputusan, fitur utama dalam klasifikasi adalah fitur dengan nilai penting tertinggi.
```

Dari analisis feature importance, fitur yang paling berpengaruh dalam menentukan spesies ikan dapat diidentifikasi. Fitur ini memainkan peran utama dalam pemisahan data pada setiap node dalam Decision Tree. Jika akurasi model cukup tinggi (≥80%), berarti model mampu mengenali pola dengan baik. Namun, jika akurasi masih rendah, beberapa langkah seperti pruning, penyesuaian parameter (max_depth, min_samples_split), atau penambahan data dapat dilakukan untuk meningkatkan performa model.

Nomor 2

Dataset “Iris.csv” terdiri dari 4 atribut yaitu panjang sepal, lebar sepal, panjang petal, dan lebar petal. Terdapat 3 kelas target pada dataset ini. Data ini dipakai untuk masalah klasifikasi, di mana kita bisa memprediksi spesies dari sebuah bunga berdasarkan atribut-atribut yang diberikan.

Tahapan yang ada pada tugas ini antara lain:

1. Ubah dataset ke dalam dataframe.

input

```
#import library
import pandas as pd
Iris = pd.read_csv('Iris.csv')
Iris
```

output

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

2. Hapus kolom 'Id' pada dataframe dan pisahkan antara atribut dan label.

Input

```
# Hapus kolom 'Id'
Iris = Iris.drop(columns=['Id'], errors='ignore')

# Pisahkan atribut (X) dan label (y)
X = Iris.drop(columns=['Species'])
y = Iris['Species']

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)

print (Iris.head())
print(y.head())
```

Output

```
   SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      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            1.5           0.2  Iris-setosa
4            5.0           3.6            1.4           0.2  Iris-setosa
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

3. Buat dan latih model Decision Tree.

```
# Buat dan latih model Decision Tree
from sklearn.tree import DecisionTreeClassifier

tree_model = DecisionTreeClassifier()
tree_model = tree_model.fit(x_train, y_train)
```

4. Lakukan prediksi dengan model yang telah dilatih,

Input

```
#prediksi dengan model yang telah dilatih
y_pred = tree_model.predict(x_test)
print(y_pred)

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print ("akurasi:", accuracy)
```

Output

```
['Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa']
akurasi: 1.0
```

5. Visualisasi model Decision Tree yang telah dilatih.

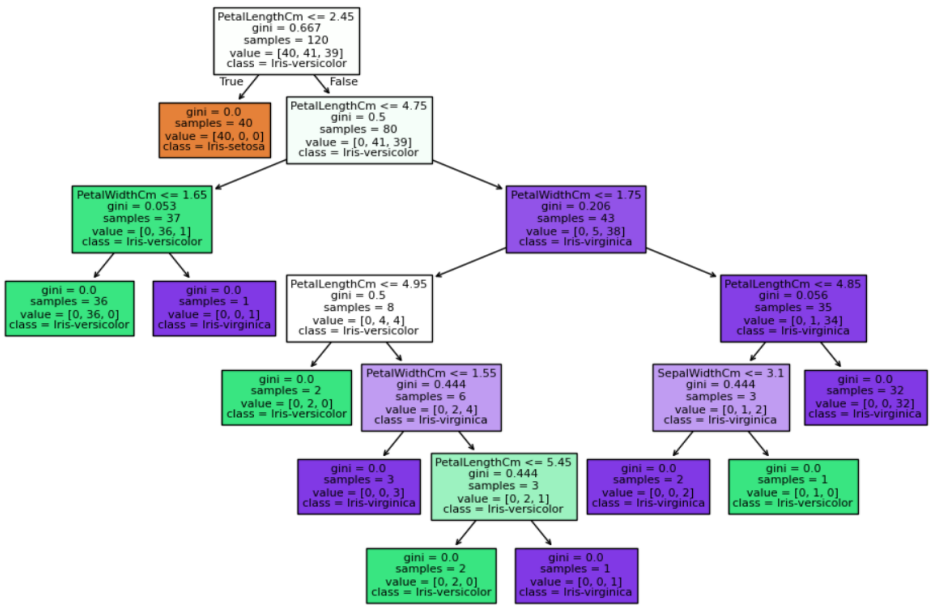
Input

```
# Visualisasi model Decision Tree yang telah dilatih.
import matplotlib.pyplot as plt
from sklearn import tree

plt.figure(figsize = (12, 8))
tree.plot_tree(tree_model, feature_names = x.columns, class_names = tree_model.classes_, filled = True)
plt.show()
```

6. Sehingga hasil output terakhir nya berupa decision tree dibawah ini.

Output



7. Tarik kesimpulan yang diperoleh

input

```python
# Kesimpulan Model
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, tree_model.predict(x_test))

print("=== Kesimpulan Decision Tree ===")
print(f"Jumlah Node: {tree_model.tree_.node_count}")
print(f"Kedalaman Maksimum: {tree_model.tree_.max_depth}")
print(f"Akurasi Model: {accuracy:.2f}")

# Menampilkan fitur yang paling berpengaruh
import numpy as np
importances = tree_model.feature_importances_
feature_importance_df = pd.DataFrame({
    'Fitur': x.columns,
    'Pentingnya': importances
}).sort_values(by="Pentingnya", ascending=False)

print("\nFitur yang paling berpengaruh dalam klasifikasi ikan:")
print(feature_importance_df.to_string(index=False))

# Evaluasi performa model berdasarkan akurasi
if accuracy > 0.8:
    print("\nModel memiliki performa yang baik dalam mengklasifikasikan spesies ikan.")
elif accuracy > 0.6:
    print("\nModel cukup baik, namun masih bisa ditingkatkan.")
else:
    print("\nModel kurang akurat, perlu optimasi lebih lanjut.")

# Analisis visualisasi decision tree
print("\nBerdasarkan visualisasi pohon keputusan, fitur utama dalam klasifikasi adalah fitur dengan nilai penting tertinggi.")
```

Output

```
=== Kesimpulan Decision Tree ===
Jumlah Node: 19
Kedalaman Maksimum: 6
Akurasi Model: 1.00

Fitur yang paling berpengaruh dalam klasifikasi ikan:
        Fitur  Pentingnya
PetalLengthCm    0.906143
 PetalWidthCm    0.077186
 SepalWidthCm    0.016670
SepalLengthCm    0.000000

Model memiliki performa yang baik dalam mengklasifikasikan spesies ikan.

Berdasarkan visualisasi pohon keputusan, fitur utama dalam klasifikasi adalah fitur dengan nilai penting tertinggi.
```