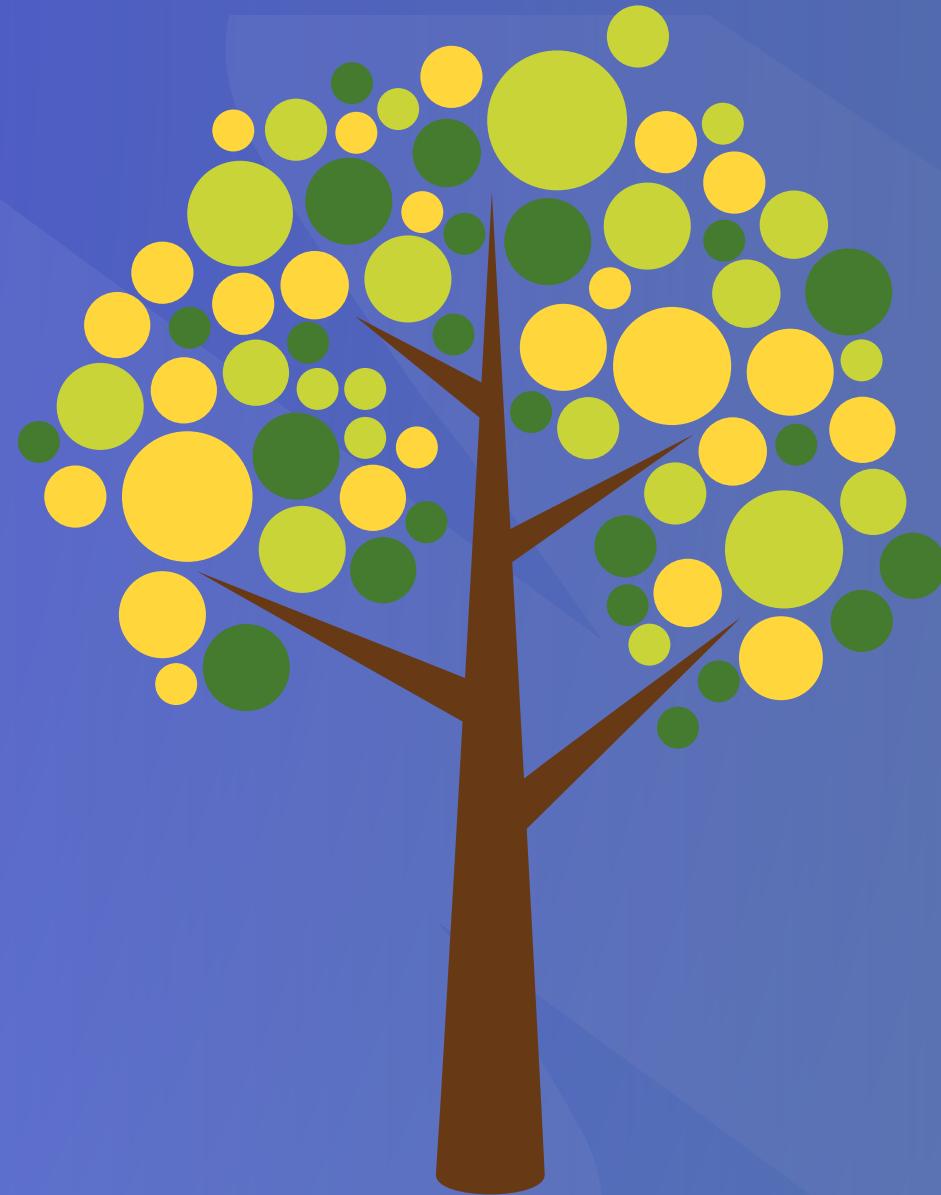


TI 5B

Decision Tree

▶ ▶ KELOMPOK 4

- 1. Amanda Rizky Yorina Prayoga
- 2. Dayinta Ayu Faj'rin
- 3. Diaz Azkha Varissa
- 4. Fadillah Dwi Anggraini



Pengertian Decission Tree

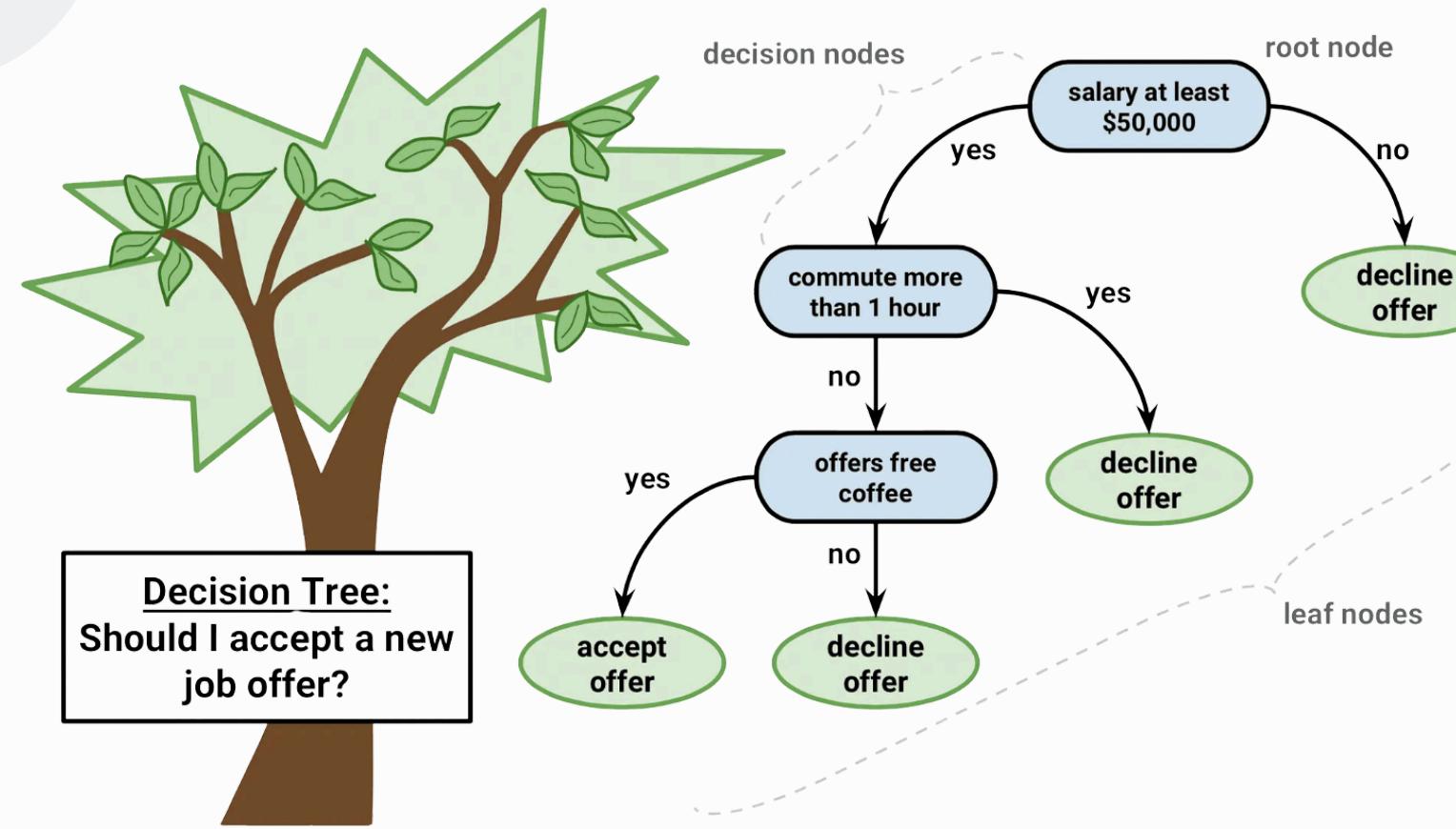
Decision tree adalah salah satu teknik klasifikasi yang populer dalam data mining. Metode decision tree ini menggunakan representasi pohon dimana setiap simpul atau node menggambarkan atribut, setiap cabang mewakili nilai analisa.

Dalam penerapannya, algoritma decision tree memerlukan atribut kelas karena masuk ke dalam kategori supervised learning. Oleh karena itu, data latih harus cukup besar dan beragam untuk memperoleh hasil yang optimal.

Decision Tree



Komponen Utama Decision Tree



Root Node

Titik awal dari pohon keputusan, yang memecah dataset berdasarkan fitur pertama

Decision Node

Titik percabangan dalam pohon keputusan di mana dataset dibagi lebih lanjut berdasarkan fitur tertentu.

Edges

Garis yang menghubungkan simpul-simpul dalam pohon, yang menunjukkan keputusan yang diambil.

Leaf Node

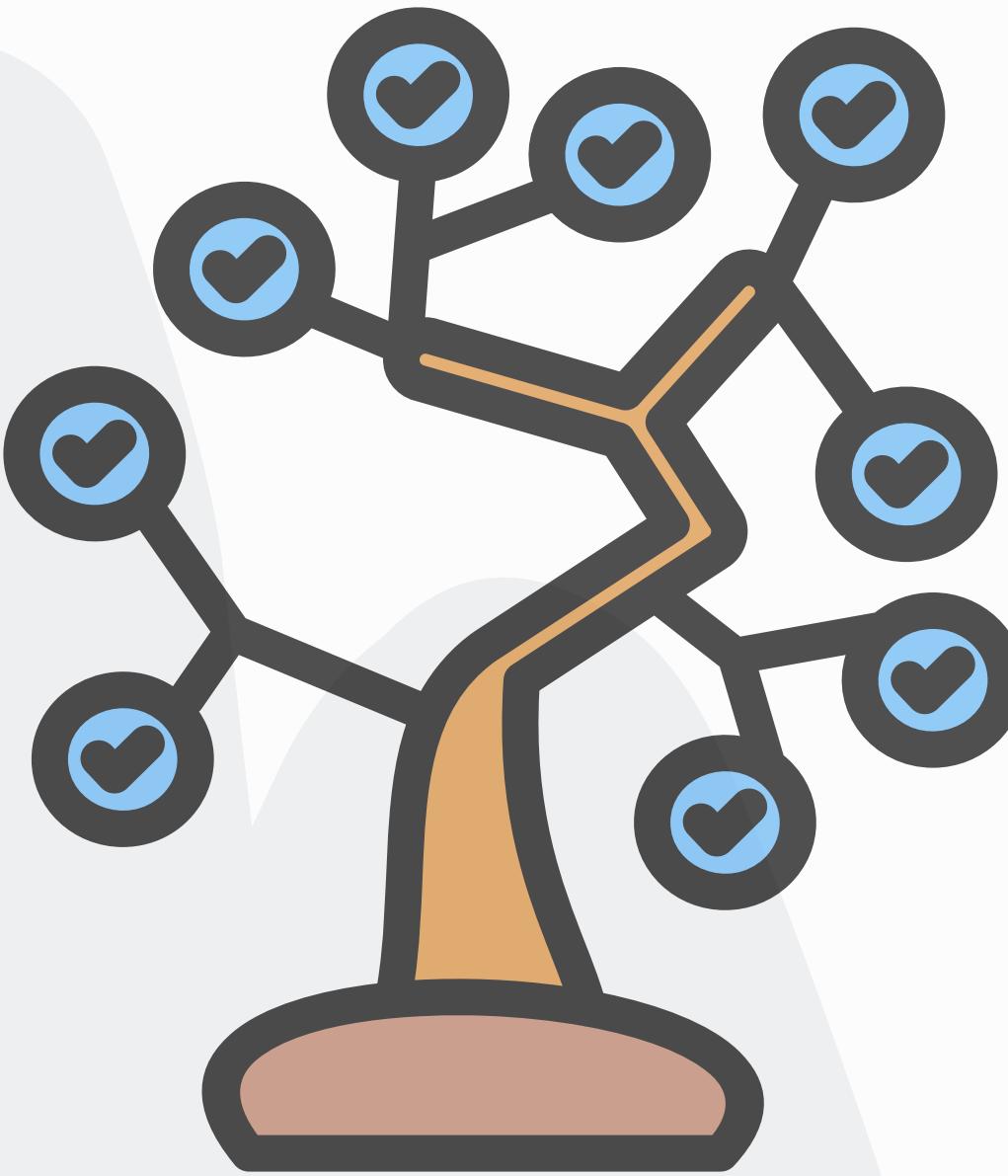
Simpul akhir yang menunjukkan label kelas atau nilai yang diprediksi.

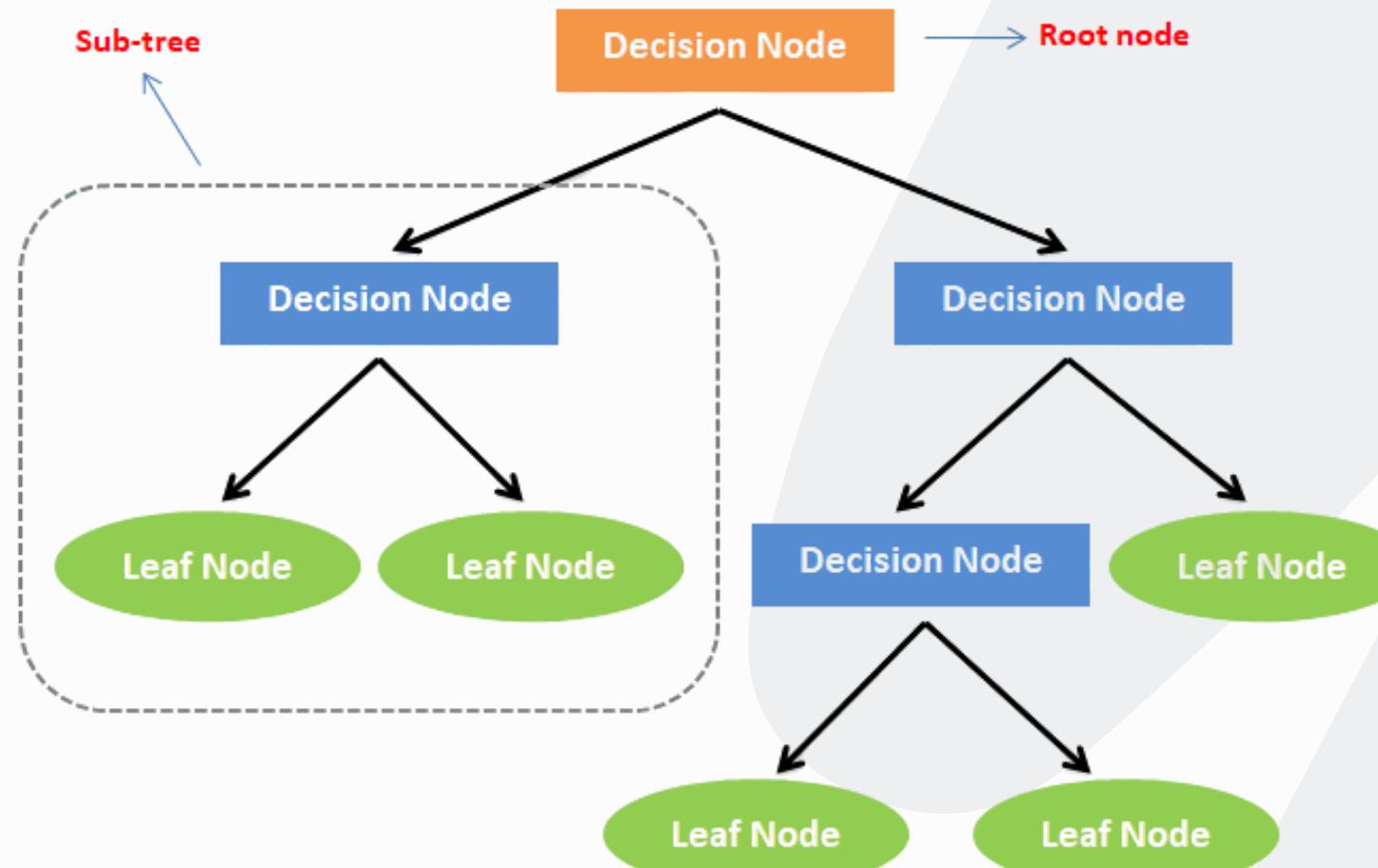


HIPOTESA FUNCTION

Hipotesa function pada Decision Tree adalah fungsi atau aturan yang digunakan model untuk memprediksi output (kelas atau nilai) berdasarkan input / pembelahan yang dilakukan. Hypothesis function mewakili “hipotesa” model mengenai bagaimana fitur-fitur dalam data menentukan hasil akhir.

Pada Decision Tree, hypothesis function bukan berupa persamaan matematis, melainkan serangkaian aturan (if-else) yang terbentuk sepanjang jalur dari root node ke leaf node.





Konsep hypothesis function adalah memetakan data input ke output dengan mengikuti struktur pohon keputusan. Setiap node di pohon memeriksa atribut tertentu, dan berdasarkan hasilnya, memilih cabang yang sesuai. Proses ini dilakukan hingga model mencapai leaf node, yang berisi hasil prediksi.

Secara umum, konsepnya yaitu:

1. Mulai dari root node.
2. Evaluasi kondisi pada node tersebut.
3. Ikuti cabang sesuai hasil kondisi.
4. Lanjutkan hingga mencapai leaf node.
5. Leaf node memberikan hasil prediksi → inilah output hypothesis function.

Dengan konsep ini, setiap prediksi pada Decision Tree dihasilkan dengan mengikuti urutan aturan yang terstruktur.

Cost Function

Terdapat beberapa cara untuk mengukur cost function, yaitu:

1. Gini Impurity Gini Gain

Mengukur ketidakmurnian atau keberagaman data dalam sebuah node. Semakin rendah nilai Gini, semakin baik pembelahan / pemisahan tersebut.

dengan keterangan rumus :

IG(t): Impurity Gini pada node t.

c: Jumlah kelas (misalnya, 2 kelas: "Ya" atau "Tidak").

p_i: Proporsi (probabilitas) pengamatan yang termasuk dalam kelas i di node t.

Node yang murni (semua observasi kelasnya sama) akan memiliki IG = 0.

2. Entropy Information Gain

Mengukur ketidakpastian dalam suatu data atau node. Entropy semakin rendah menunjukkan bahwa data dalam node tersebut lebih homogen.

$$H(t) = - \sum_{i=1}^c p_i \log_2(p_i)$$

- H(t): Entropi pada node t.
- c: Jumlah kelas.
- p_i: Proporsi pengamatan yang termasuk dalam kelas i di node t.

Node yang murni memiliki H = 0. Node yang paling tidak murni (kelasnya terdistribusi merata) memiliki H = 1 (untuk 2 kelas).



Pembelahan terbaik adalah yang meminimalkan nilai cost function (Gini atau Entropy), yang berarti data dalam masing-masing node akan lebih homogen (semua sample memiliki kelas yang sama).

Studi Kasus

Dataset ini berisi 15 catatan mengenai kondisi cuaca yang terdiri dari beberapa fitur yang relevan dengan prediksi apakah akan terjadi hujan atau tidak. Setiap baris dalam dataset mencakup data cuaca dengan beberapa fitur seperti suhu, kelembapan, tekanan udara, arah angin. juga label “Turun Hujan” berdasarkan fitur yang menunjukkan apakah hujan terjadi atau tidak.

Berdasarkan kombinasi keempat fitur tersebut, dapat menentukan apakah berpotensi turun hujan atau tidak. Dataset ini dapat digunakan untuk melatih model Decision Tree agar mampu memahami pola-pola cuaca dan memprediksi kemungkinan hujan pada hari-hari berikutnya.

	Suhu (°C)	Kelembapan (%)	Tekanan udara (hPa)	Arah angin (m/s)	Turun Hujan
0	30	65	1015	5.0	No
1	28	85	1010	2.5	Yes
2	25	90	1008	3.0	Yes
3	32	55	1020	4.0	No
4	26	80	1005	1.5	Yes
5	29	70	1012	6.0	No
6	27	75	1013	4.5	No
7	24	95	1007	3.5	Yes
8	31	60	1016	5.5	No
9	33	50	1021	4.5	No
10	27	72	1009	3.0	Yes
11	26	78	1014	5.0	Yes
12	22	88	1006	6.5	No
13	30	68	1017	4.0	Yes
14	29	77	1011	3.5	No

Penjelasan Kode

```
import pandas as pd  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import accuracy_score
```

Kode dimulai dengan mengimpor beberapa library penting yang dibutuhkan dalam proses pembuatan model Decision Tree.



- Library pandas digunakan untuk membaca, menyimpan, dan mengolah data dalam bentuk tabel sehingga memudahkan proses analisis.
- DecisionTreeClassifier dari scikit-learn diimpor sebagai algoritma utama yang digunakan untuk membangun model klasifikasi berbasis pohon keputusan.
- train_test_split digunakan untuk membagi dataset menjadi dua bagian, yaitu data pelatihan dan data pengujian, agar model dapat dilatih dan diuji secara adil.
- accuracy_score diimpor untuk menghitung tingkat akurasi dari hasil prediksi model sehingga dapat mengetahui seberapa baik model tersebut bekerja dalam memprediksi data yang belum pernah dilihat sebelumnya.



Penjelasan Kode

```
df = pd.read_excel('dataset_hujan_numerik.xlsx')  
df
```

Kode diatas digunakan untuk membaca file Excel bernama dataset_hujan_numerik.xlsx dan menyimpannya ke dalam sebuah tabel Python yang disebut DataFrame dengan nama df. Setelah itu, df ditampilkan supaya bisa melihat isi datanya langsung

Decision Tree



	Suhu (°C)	Kelembapan (%)	Tekanan udara (hPa)	Arah angin (m/s)	Turun Hujan
0	30	65	1015	5.0	No
1	28	85	1010	2.5	Yes
2	25	90	1008	3.0	Yes
3	32	55	1020	4.0	No
4	26	80	1005	1.5	Yes
5	29	70	1012	6.0	No
6	27	75	1013	4.5	No
7	24	95	1007	3.5	Yes
8	31	60	1016	5.5	No
9	33	50	1021	4.5	No
10	27	72	1009	3.0	Yes
11	26	78	1014	5.0	Yes
12	22	88	1006	6.5	No
13	30	68	1017	4.0	Yes
14	29	77	1011	3.5	No

Penjelasan Kode

```
# Mengubah target variabel ke dalam bentuk numerik
df['Turun Hujan'] = df['Turun Hujan'].map({'Yes': 1, 'No': 0})

# Membagi data menjadi fitur dan target
X = df.drop('Turun Hujan', axis=1)
y = df['Turun Hujan']

# Membagi data menjadi train dan test set
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)

# Membuat dan melatih model decision tree
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Melakukan prediksi dan mengevaluasi model
y_pred = model.predict(X_test)

# Menghitung akurasi
print("Akurasi:", accuracy_score(y_test, y_pred))

...
Akurasi: 0.6
```

Decision Tree



Kode di samping digunakan untuk menyiapkan data, melatihnya menggunakan model Decision Tree, lalu mengecek seberapa baik model tersebut bekerja dalam memprediksi hujan.

- Pertama, kolom "Turun Hujan" diubah dari teks Yes dan No menjadi angka 1 dan 0, agar bisa dibaca oleh model.
- Setelah itu, data dipisahkan menjadi dua bagian: X sebagai fitur yang berisi variabel input (Suhu, Kelembapan, Tekanan Udara, dan Arah Angin), dan y sebagai target yang ingin ditebak, yaitu apakah hujan turun atau tidak.
- Data kemudian dibagi lagi menjadi data latih dan data uji, di mana sebagian besar data digunakan untuk mengajari model dan sisanya dipakai untuk mengetesnya.
- Setelah model Decision Tree selesai dilatih menggunakan data latih, model digunakan untuk memprediksi hasil pada data uji.
- Hasil prediksi tersebut dibandingkan dengan nilai sebenarnya menggunakan fungsi accuracy_score, sehingga bisa mengetahui tingkat ketepatan model.

Pada kasus ini, model menghasilkan akurasi sekitar 0.6, yang berarti model hanya benar sekitar 60% dalam menebak apakah akan turun hujan atau tidak.

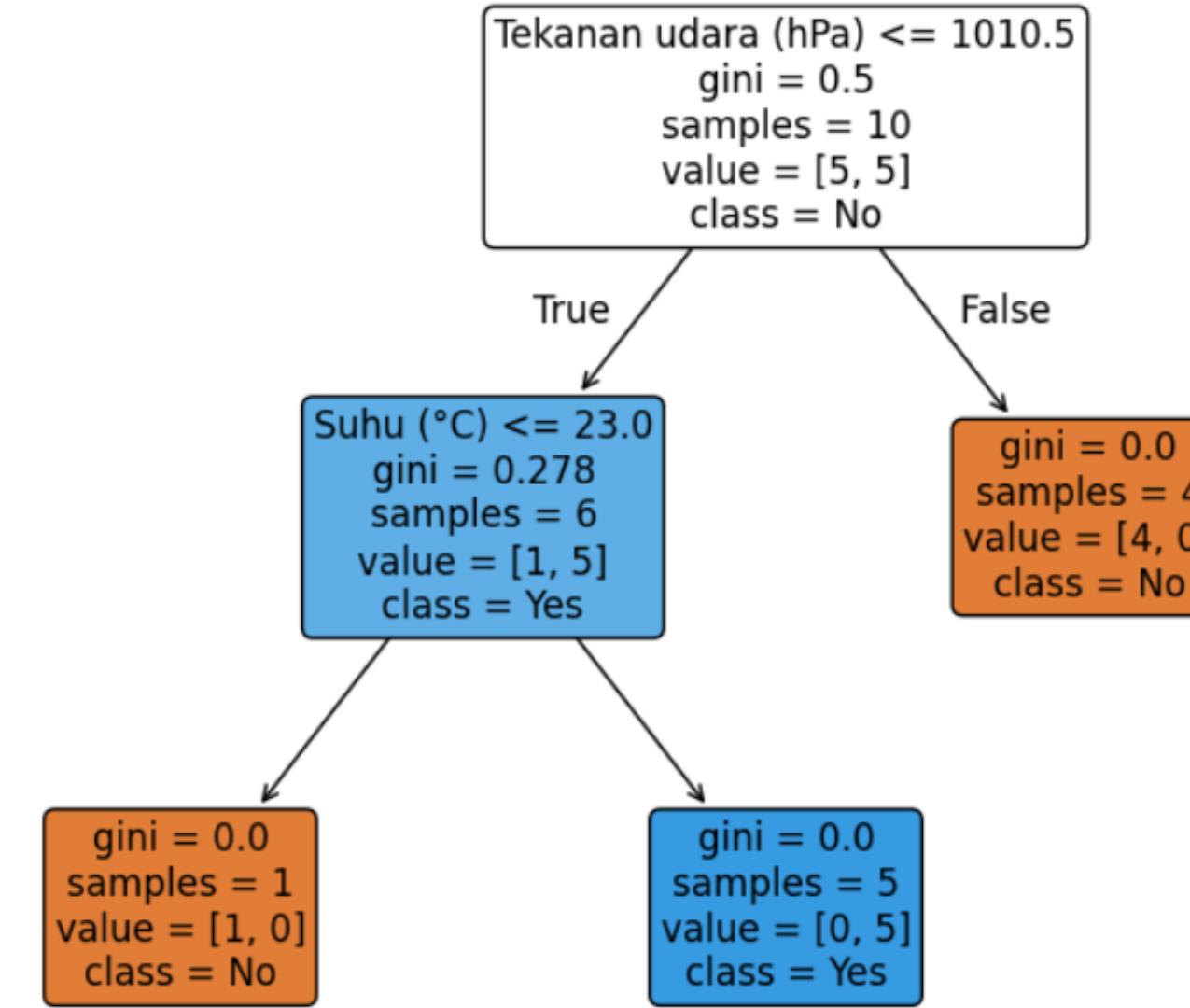
Penjelasan Kode

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

# Visualisasi pohon keputusan
plt.figure(figsize=(9,7))
plot_tree(
    model,
    filled=True,
    feature_names=X.columns,
    class_names=['No', 'Yes'],
    rounded=True
)
plt.show()
```

Selanjutnya, pada kode tersebut, untuk memvisualisasikan pohon keputusan yang telah dibangun menggunakan model yang telah dilatih.

= Decision Tree



- Node pertama menggunakan Tekanan udara (hPa) dengan threshold ≤ 1010.5 untuk membagi data.
- Cabang kiri (True) menggunakan Suhu ($^{\circ}\text{C}$) dengan threshold ≤ 23.0 dan Gini = 0.278, menunjukkan pembagian data yang lebih murni.
- Sub-cabang kiri (True) menggunakan Suhu ($^{\circ}\text{C}$) dengan threshold ≤ 23.0 dan Gini = 0, menunjukkan pembagian data murni.
- Sub-cabang kanan (False) menggunakan Suhu ($^{\circ}\text{C}$) dengan threshold > 23.0 dan Gini = 0, menunjukkan pembagian data murni.
- Cabang kanan (False) menggunakan Tekanan udara (hPa) dengan threshold > 1010.5 dan Gini = 0 juga menunjukkan pembagian data murni.

21112015

Friday

Nov 2025

Thank you



Penjelasan Kode

`df.describe()`

Perintah `df.describe()` digunakan untuk melihat ringkasan statistik dasar dan menampilkan informasi seperti jumlah data, nilai rata-rata, nilai minimum, nilai maksimum, dan seberapa besar variasi datanya untuk setiap kolom angka.

	Suhu (°C)	Kelembapan (%)	Tekanan udara (hPa)	Arah angin (m/s)	Turun Hujan
<code>count</code>	15.000000	15.000000	15.000000	15.000000	15.000000
<code>mean</code>	27.933333	73.866667	1012.266667	4.133333	0.466667
<code>std</code>	3.058166	12.944203	4.920317	1.355764	0.516398
<code>min</code>	22.000000	50.000000	1005.000000	1.500000	0.000000
<code>25%</code>	26.000000	66.500000	1008.500000	3.250000	0.000000
<code>50%</code>	28.000000	75.000000	1012.000000	4.000000	0.000000
<code>75%</code>	30.000000	82.500000	1015.500000	5.000000	1.000000
<code>max</code>	33.000000	95.000000	1021.000000	6.500000	1.000000