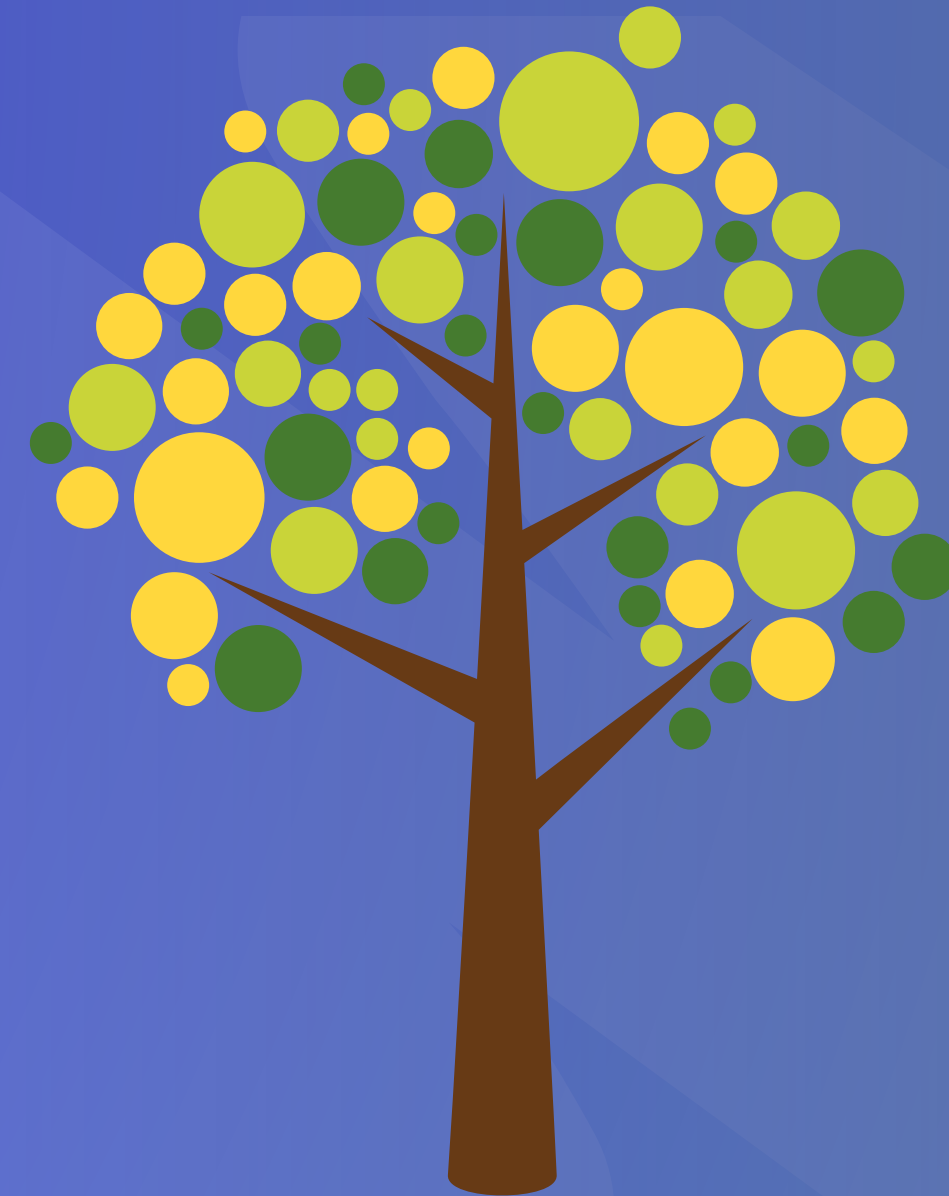


TI 5B

Decision Tree (Studi Kasus)

▶ ▶ KELOMPOK 4

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



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.

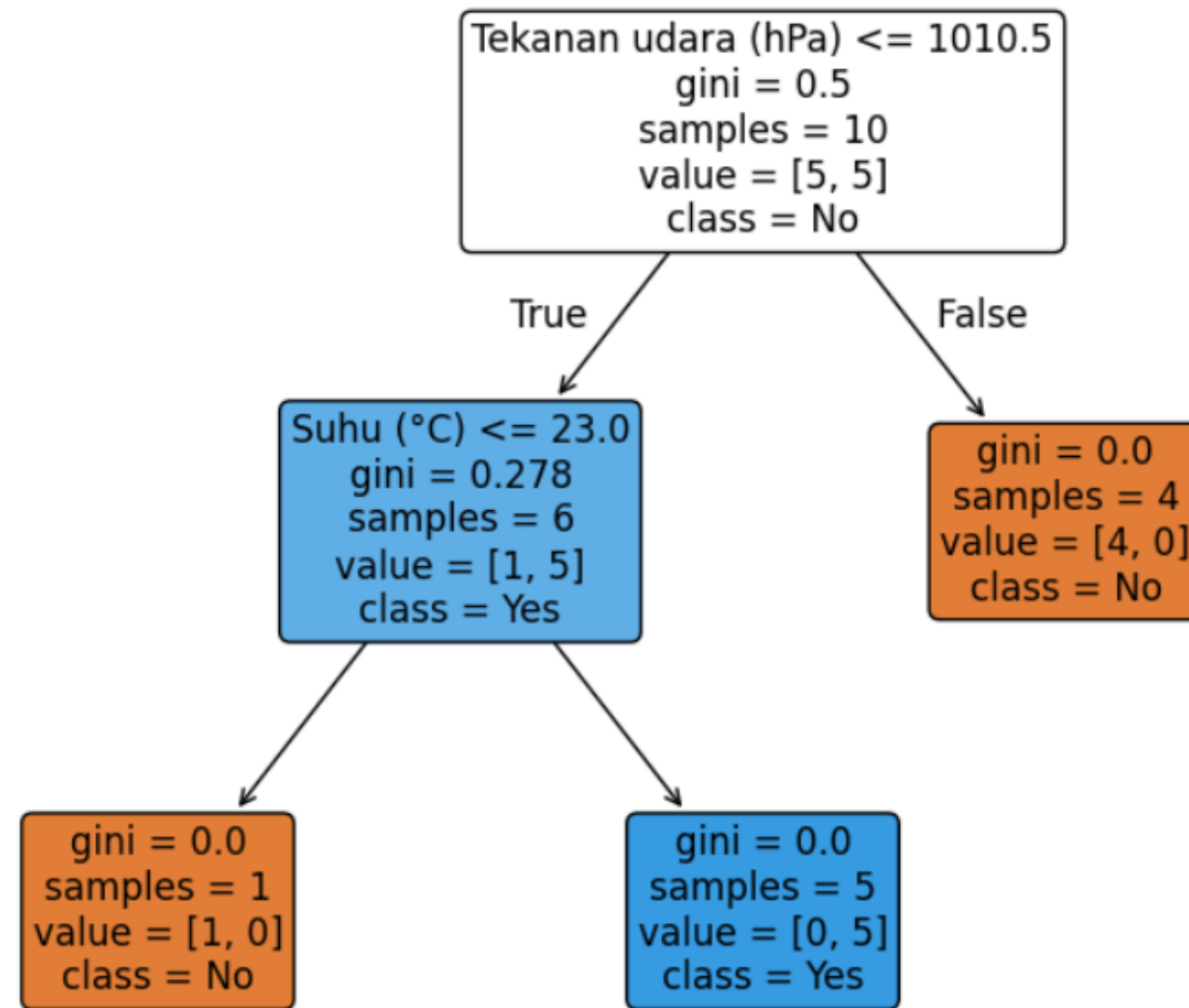
Pada kasus ini, model menghasilkan akurasi sekitar 0.6, yang berarti model hanya benar sekitar 60% dalam menebak apakah akan turun hujan atau tidak.

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.



- 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