Nama : Aerio David Tirta Atmdjo

NIM : 434231081

Kelas : C3

Laporan Pratikum ML Prak

1. Import Library

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

Panda = membaca & mengelola data excel

Streamlit = sebagai tampilan

Sckit-learn = untuk pemodelan Decision Tree, split data, evaluasi akurasi.

Matplotlib = untuk visualisasi pohon Keputusan

2. Load Dataset

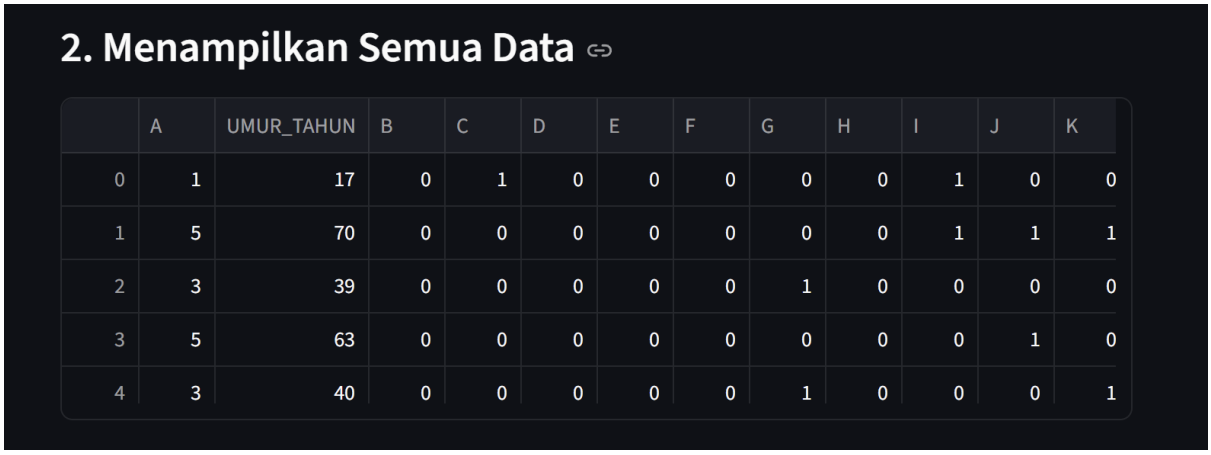
```
file_path = "BlaBla.xlsx"
df = pd.read_excel(file_path)
df = df.replace({"Y": 1, "N": 0})
```

Dataset dibaca dari file “BlaBla.xlsx”
Nilai kategori "Y" dan "N" dikonversi ke angka 1 dan 0 supaya bisa diproses oleh algoritma.

3. Menampilkan Data

```
st.subheader("2. Menampilkan Semua Data")
st.dataframe(df.head())
```

Menampilkan **5 baris pertama dataset** (karena memakai head) agar pengguna tahu bentuk datanya.



2. Menampilkan Semua Data

| | A | UMUR_TAHUN | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 17 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 5 | 70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 2 | 3 | 39 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 5 | 63 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 3 | 40 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

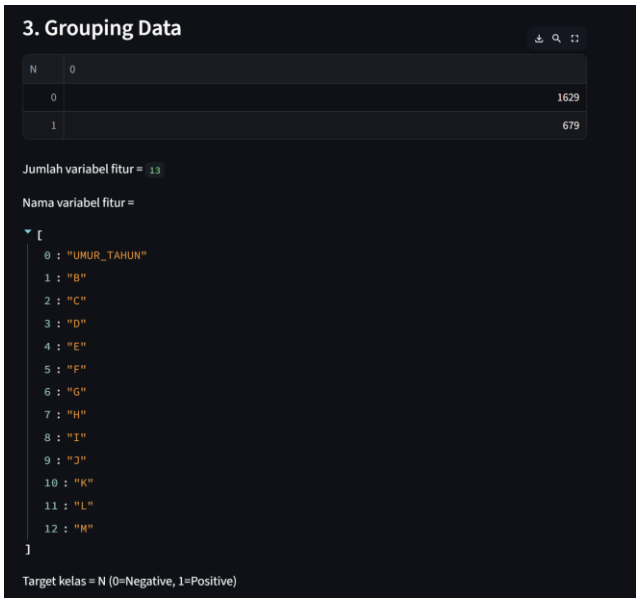
4. Grouping Data

```
st.subheader("3. Grouping Data")
st.write(df.groupby("N").size())
st.write("Jumlah variabel fitur =", len(df.columns) - 2)
st.write("Nama variabel fitur =", list(df.drop(columns=["N", "A"]).columns))
st.write("Target kelas = N (0=Negative, 1=Positive)")
```

Mengecek distribusi data target N (misalnya jumlah pasien **Negative** vs **Positive**).

Menampilkan jumlah variabel fitur dan nama kolom fitur.

Menjelaskan target yang dipakai (N).



3. Grouping Data

| N | 0 |
|---|---|
| 0 | 1629 |
| 1 | 679 |

Jumlah variabel fitur = 13

Nama variabel fitur =

```
[
  0 : "UMUR_TAHUN"
  1 : "B"
  2 : "C"
  3 : "D"
  4 : "E"
  5 : "F"
  6 : "G"
  7 : "H"
  8 : "I"
  9 : "J"
  10 : "K"
  11 : "L"
  12 : "M"
]
```

Target kelas = N (0=Negative, 1=Positive)

5.Training & Testing

```
X = df.drop(columns=["N", "A"])
y = df["N"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

**X** = fitur (semua kolom kecuali N target & A ID).

**y** = target (N).

Dataset dibagi menjadi **80% training** dan **20% testing**.


6. Informasi Split Data

```
st.subheader("4. Split Data Training & Testing")
st.write("Jumlah Data Training =", len(X_train))
st.write("Jumlah Data Testing =", len(X_test))
```

Menampilkan split datanya

7. Membuat Model Decision Tree

```
clf = DecisionTreeClassifier(criterion="gini", max_depth=5, random_state=42)
clf.fit(X_train, y_train)
st.subheader("5. Decision Tree")
st.success("Model berhasil dilatih.")
```

Membuat model **Decision Tree** dengan kriteria gini dan kedalaman maksimum 5.

Model dilatih dengan data training.


8. Prediksi & Contoh output

```
y_pred = clf.predict(X_test)
st.subheader("6. Instance Prediksi Decision Tree")
st.write("Beberapa hasil prediksi vs data asli:")
for i in range(5):
    st.write("Prediksi =", y_pred[i], " | Asli =", list(y_test)[i])
```

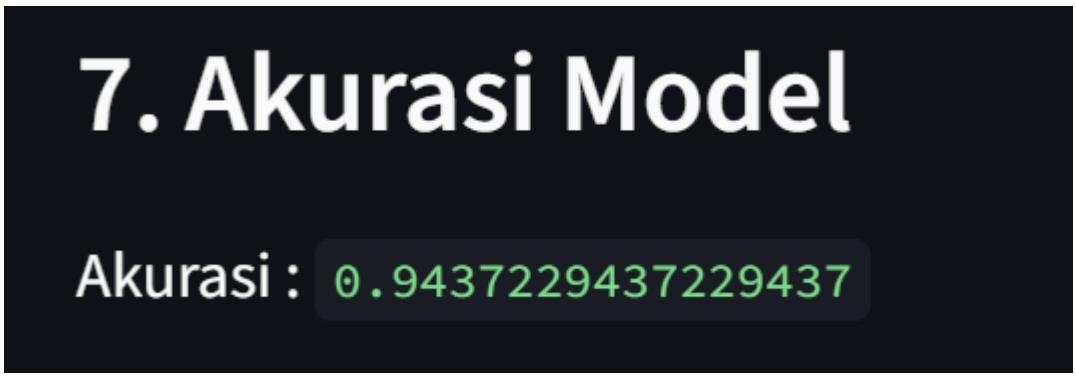
Menguji model dengan data testing.

Menampilkan 5 contoh perbandingan hasil prediksi vs label asli.

9. Akurasi Model

```
st.subheader("7. Akurasi Model")
st.write("Akurasi :", accuracy_score(y_test, y_pred))
```

Menghitung akurasi model (proporsi prediksi yang benar).




10. Classification Report

```
st.subheader("8. Classification Report")
st.text(classification_report(y_test, y_pred))
```

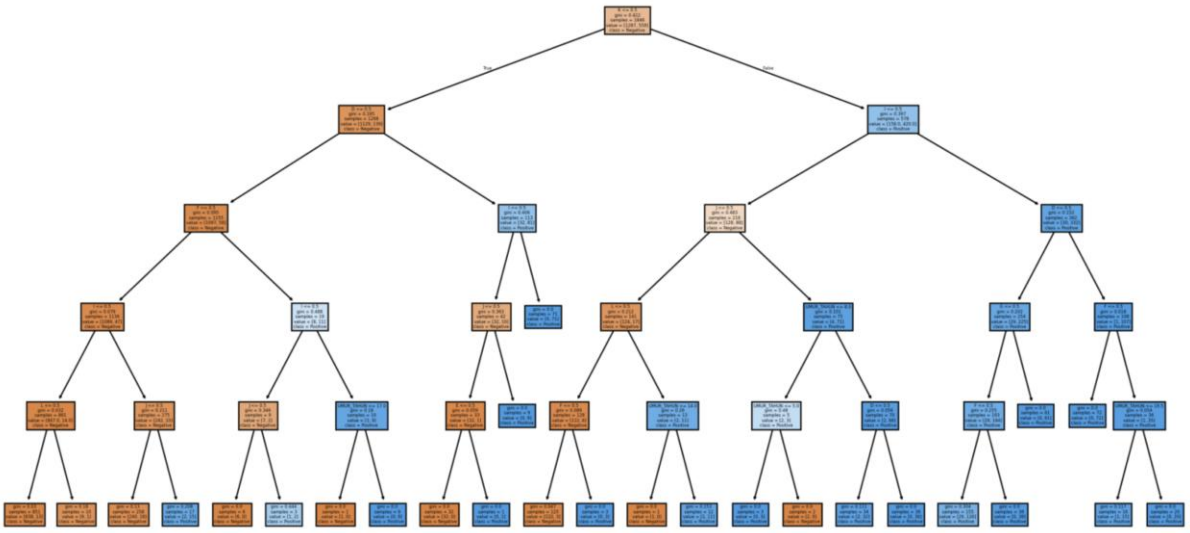
Menampilkan laporan metrik evaluasi: **precision, recall, f1-score, support**.

11. Visualisasi Pohon Keputusan

```
st.subheader("9. Visualisasi Pohon Keputusan")
fig, ax = plt.subplots(figsize=(16, 8))
plot_tree(clf, filled=True, feature_names=X.columns, class_names=["Negative", "Positive"])
st.pyplot(fig)
```

Membuat visualisasi grafis dari pohon keputusan yang sudah dilatih.

Node berwarna memperlihatkan aturan klasifikasi.



12. Form Input Pasien

```python
st.subheader("10. Prediksi Pasien Baru")

umur = st.number_input("Umur Pasien", min_value=1, max_value=100, step=1)
gender = st.radio("Jenis Kelamin", ["Perempuan", "Laki-laki"])
gender = 0 if gender == "Perempuan" else 1

gejala = []
for kolom in X.columns[2:]:  # ambil kolom C–M
    val = st.radio(f"Apakah pasien mengalami {kolom}?", ["Tidak", "Ya"], index=0)
    gejala.append(1 if val == "Ya" else 0)

# Kategori umur (A_k)
if umur < 21:
    A_k = 1
elif umur <= 30:
    A_k = 2
elif umur <= 40:
    A_k = 3
elif umur <= 50:
    A_k = 4
else:
    A_k = 5

if st.button("Prediksi Pasien"):
    data_input = [A_k, gender] + gejala
    hasil = clf.predict([data_input])[0]
    st.success("Hasil Prediksi: **Positive**" if hasil == 1 else "Hasil Prediksi: **Negative**")
    st.write("Kode umur =", A_k)
    st.write("Data input =", data_input)
```

- Membuat form interaktif: input **umur, gender, gejala** (kolom C–M).
- Umur diubah jadi **kategori umur** (A_k).
- Data input diproses ke model untuk diprediksi.
- Hasil prediksi ditampilkan ke user (Positive / Negative).