

**FAKULTAS SAINS & TEKNOLOGI - UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**UJIAN AKHIR SEMESTER GANJIL TA. 2023/2024**

---

**PROGRAM STUDI : INFORMATIKA PROGRAM SARJANA**

<b>Mata Ujian</b>	<b>: Pengantar Analisis Data (I-XII)</b>
<b>Hari, Tanggal</b>	<b>: Selasa, 16 Januari 2024</b>
<b>Dosen</b>	<b>: Canggih Puspo Wibowo, S.T., M.Eng. Fadil Indra Sanjaya, S.Kom., M.Kom. Muhammad Fachrie, S.T., M.Cs. Muhammad Kopravi, S.Kom., M.Eng. Muhammad Zakariyah, S.Kom., M.Kom. Saucha Diwandari, S.Kom., M.Eng. Yoga Sahria, S.Kom., M.Kom.</b>
<b>Waktu</b>	<b>: 100 Menit</b>
<b>Sifat</b>	<b>: Take Home (Buka buku, internet)</b>

---

**PERATURAN**

1. Jawaban untuk setiap soal pada UAS ini dibuat menggunakan aplikasi pengolah kata (Ms. Word atau semacamnya) lalu hasilnya disimpan dalam format PDF. Cantumkan juga kode program yang Anda buat di dalam lembar jawaban tersebut pada setiap nomor soal.
2. Uraikanlah jawaban Anda dengan lengkap dan sistematis. Jawaban harus dibuat berurutan sesuai nomor urut soal.
3. Tidak diperkenankan melakukan *copy-paste* jawaban dari ChatGPT atau aplikasi semacamnya.
4. Tidak diperkenankan berdiskusi dengan siapapun, namun Anda diperbolehkan membuka buku atau sumber belajar dari internet.
5. Jawaban UTS dalam format PDF yang telah disertai dengan kode program dicetak pada kertas A4 dan dikumpulkan sesuai jadwal ujian, yakni Hari Selasa, 16 Januari 2024 pukul 13.20 WIB di ruangan yang telah ditentukan pada jadwal ujian.
6. Lembar jawaban yang tidak ditandatangani tidak akan diperiksa dan akan diberikan nilai 0.

**SOAL**

Unduhlah dataset **pilkada-1000-uas.csv** dan **pilkada-100-uas.csv** pada link berikut: [https://bit.ly/uas\\_analisis\\_data](https://bit.ly/uas_analisis_data). Pada ujian ini, Anda diminta mengekstrak pengetahuan berupa *decision tree* dari dataset **pilkada-1000-uas.csv**. Kemudian, *decision tree* yang sudah diperoleh akan Anda gunakan untuk memprediksi kemenangan paslon pada dataset **pilkada-100-uas.csv**.

1. Berdasarkan logika dan intuisi, atribut manakah yang harus dihapus dari dataset **pilkada-1000.csv**? Jelaskan alasan Anda!
2. Buatlah kode program dalam bahasa Python untuk mengekstrak pohon keputusan (*decision tree*) dari dataset **pilkada-1000-uas.csv** dan lakukan prediksi terhadap data **pilkada-100-uas.csv** menggunakan *decision tree* yang sudah dibuat. Aturilah nilai pada parameter *max\_depth* dan *criterion* sampai Anda mendapatkan akurasi yang paling tinggi pada prediksi dataset **pilkada-100.csv** (Anda bisa juga mengubah parameter lainnya).
3. Catatlah setiap hasil prediksi yang Anda dapatkan untuk setiap parameter dalam sebuah tabel lalu berikan analisis Anda terhadap hasil pengujian yang Anda peroleh!
4. Tampilkan visualisasi pohon keputusan yang telah berhasil dibangun!
5. Lampirkan semua kode program yang Anda buat pada ujian ini!

##### SELAMAT MENGERJAKAN #####

# Lembar Jawaban UAS Pengantar Analisis Data

Nama : Muhammad Zaki Wicaksono

NIM : 5220411248

Kelas : VII

Pernyataan :

Dengan mengerjakan ujian ini, maka saya menyatakan bahwa semua jawaban **SAYA KERJAKAN SENDIRI** tanpa bekerjasama maupun meminta bantuan siapapun. Saya bersedia diberi **nilai E** jika pernyataan ini terbukti salah.

Tanda tangan:

1. Menurut logika dan intuisi saya, atribut yang sebaiknya dihapus dari dataset pilkada-1000.csv adalah kolom "Nama Kandidat (1)" dan "Nama Kandidat (2)". Alasan penghapusan kedua kolom ini adalah karena analisis yang kita lakukan bersifat global dan tidak memfokuskan pada data perorangan. Dengan demikian, kolom-kolom tersebut lebih relevan jika kita ingin memprediksi data per-objek.

tetapi perlu diperhatikan bahwa penjelasan di atas sesuai dengan instruksi pada soal saja. Dalam prakteknya saya menghapus banyak kolom, proses penghapusan atribut yang saya lakukan melibatkan evaluasi lebih lanjut menggunakan algoritma training untuk mengidentifikasi korelasi yang kurang baik setelah dilakukan treatment data.

2. Membuat pohon keputusan

```
#training
X_train_1000, X_test_1000, y_train_1000, y_test_1000 = train_test_split(df.drop("Hasil
Pilkada", axis=1), df["Hasil Pilkada"], test_size=0.2)

model = DecisionTreeClassifier(max_depth=3, criterion="entropy")
model.fit(X_train_1000, y_train_1000)

#cek akurasi training
prediksi = model.predict(X_test_1000)
akurasi = np.mean(prediksi == y_test_1000)
print("peforma training 1000 data:", akurasi)

plt.figure(figsize=(20, 10))
plot_tree(model, feature_names=df.drop("Hasil Pilkada", axis=1).columns, filled=True)
plt.show()

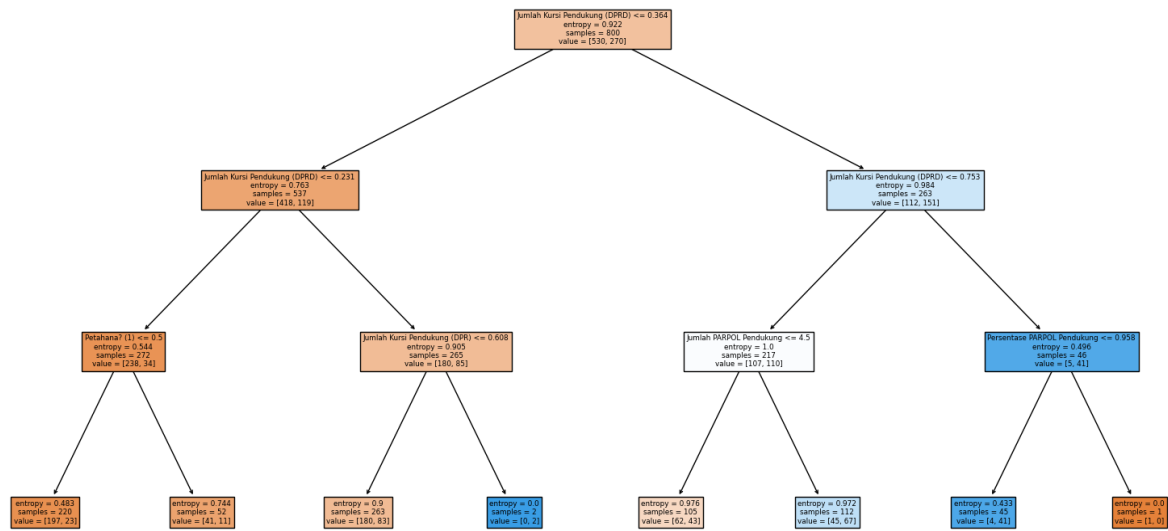
X_train_100, X_test_100, y_train_100, y_test_100 = train_test_split(dy.drop("Hasil Pilkada",
axis=1), dy["Hasil Pilkada"], test_size=0.2)
y_pred_100 = model.predict(X_test_100)

#hitung akurasi analisi
akurasiimplementasi = np.mean(y_pred_100 == y_test_100)
print("Akurasi hasil:", akurasiimplementasi)
```

3. Hasil analisis akhirnya

Running program ke-	Hasil
1	0,7
2	0,7
3	0,8
4	0,75
5	0,9
6	0,75
7	0,7
8	0,75
9	0,9
10	0,7

4. visualisasi pohon keputusan yang telah berhasil dibangun



5. lampiran code

```
#import library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import plot_tree

#baca file
dy = pd.read_csv("pilkada-100-uas.csv")
df = pd.read_csv("pilkada-1000-uas.csv")

#melihat file
df

#ubah data non-numerik menjadi numerik
le = LabelEncoder()

df['Nama Daerah'] = le.fit_transform(df['Nama Daerah'])
df['Jalur Pencalonan'] = le.fit_transform(df['Jalur Pencalonan'])
df['Nama Kandidat (1)'] = le.fit_transform(df['Nama Kandidat (1)'])
df['Nama Kandidat (2)'] = le.fit_transform(df['Nama Kandidat (2)'])
df['Gender (1)'] = le.fit_transform(df['Gender (1)'])
df['Gender (2)'] = le.fit_transform(df['Gender (2)'])
df['Latar Belakang Profesi (1)'] = le.fit_transform(df['Latar Belakang Profesi (1)'])
```

```

df['Latar Belakang Profesi (2)'] = le.fit_transform(df['Latar Belakang Profesi (2)'])
df['Hasil Pilkada'] = le.fit_transform(df['Hasil Pilkada'])

dy['Nama Daerah'] = le.fit_transform(dy['Nama Daerah'])
dy['Jalur Pencalonan'] = le.fit_transform(dy['Jalur Pencalonan'])
dy['Nama Kandidat (1)'] = le.fit_transform(dy['Nama Kandidat (1)'])
dy['Nama Kandidat (2)'] = le.fit_transform(dy['Nama Kandidat (2)'])
dy['Gender (1)'] = le.fit_transform(dy['Gender (1)'])
dy['Gender (2)'] = le.fit_transform(dy['Gender (2)'])
dy['Latar Belakang Profesi (1)'] = le.fit_transform(dy['Latar Belakang Profesi (1)'])
dy['Latar Belakang Profesi (2)'] = le.fit_transform(dy['Latar Belakang Profesi (2)'])
dy['Hasil Pilkada'] = le.fit_transform(dy['Hasil Pilkada'])

#lihat data
df

#lihat korelasi untuk memutuskan data yang tidak berguna
for kolom in df.columns:
    if kolom != 'Hasil Pilkada':
        korelasi = df['Hasil Pilkada'].corr(df[kolom])
        print(f"korelasi Kolom 'Hasil Pilkada' dengan kolom '{kolom}': {korelasi}")

#lewati data yang korelasinya jelek berdasarkan penelusuran data pilkada 1000
df = df.drop(['Jumlah Paslon'], axis=1)
df = df.drop(['Jalur Pencalonan'], axis=1)
df = df.drop(['Nama Kandidat (1)', 'Nama Kandidat (2)'], axis=1)
df = df.drop(['Gender (1)', 'Gender (2)'], axis=1)
df = df.drop(['Usia (1)', 'Usia (2)'], axis=1)
df = df.drop(['Latar Belakang Profesi (1)', 'Latar Belakang Profesi (2)'], axis=1)

dy = dy.drop(['Jumlah Paslon'], axis=1)
dy = dy.drop(['Jalur Pencalonan'], axis=1)
dy = dy.drop(['Nama Kandidat (1)', 'Nama Kandidat (2)'], axis=1)
dy = dy.drop(['Gender (1)', 'Gender (2)'], axis=1)
dy = dy.drop(['Usia (1)', 'Usia (2)'], axis=1)
dy = dy.drop(['Latar Belakang Profesi (1)', 'Latar Belakang Profesi (2)'], axis=1)

#lihat data
df

#memastikan data sebelum diproses
#menangani str pada file 1000 data
for col in df.columns:
    if df[col].dtype == "object":
        df[col] = LabelEncoder().fit_transform(df[col])

#menangani str pada file 100 data
for col in dy.columns:
    if dy[col].dtype == "object":
        dy[col] = LabelEncoder().fit_transform(dy[col])

#melihat tipe data sebelum training (memastikan)
df.dtypes

#training
X_train_1000, X_test_1000, y_train_1000, y_test_1000 = train_test_split(df.drop("Hasil Pilkada", axis=1), df["Hasil Pilkada"], test_size=0.2)

```

```
model = DecisionTreeClassifier(max_depth=3, criterion="entropy")
model.fit(X_train_1000, y_train_1000)

#cek akurasi training
prediksi = model.predict(X_test_1000)
akurasi = np.mean(prediksi == y_test_1000)
print("peforma training 1000 data:", akurasi)

plt.figure(figsize=(20, 10))
plot_tree(model, feature_names=df.drop("Hasil Pilkada", axis=1).columns, filled=True)
plt.show()

X_train_100, X_test_100, y_train_100, y_test_100 = train_test_split(dy.drop("Hasil Pilkada",
axis=1), dy["Hasil Pilkada"], test_size=0.2)
y_pred_100 = model.predict(X_test_100)

#hitung akurasi analisi
akurasiimplementasi = np.mean(y_pred_100 == y_test_100)
print("Akurasi hasil:", akurasiimplementasi)
```

**CAPAIAN PEMBELAJARAN - MATAKULIAH :**

CP – MK			
CPMK1	Mahasiswa mampu memahami konsep dasar analisis data		
CPMK2	Mahasiswa mampu memahami missing value beserta cara penanganannya		
CPMK3	Mahasiswa mampu melakukan feature scalling pada data		
Kesesuaian Materi Dengan Silabus	Kelengkapan Informasi pada Soal	Catatan Perbaikan (jika ada)	Validasi Kaprodi Informatika