

## LAPORAN PERTEMUAN 4

### (Data Preparation)

NAMA : YULIANA KRISTINA LEPAN OLEONA

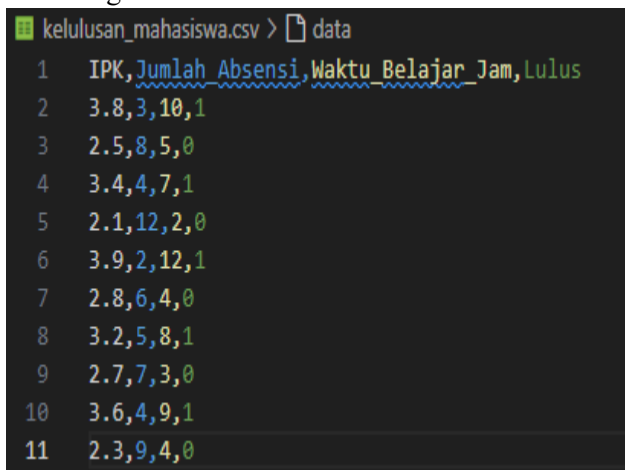
NIM : 231011402400

MATKUL : MACHINE LEARNING

#### A. LANGKAH – LANGKAH PROGRAM

##### 1. Membuat Dataset CSV

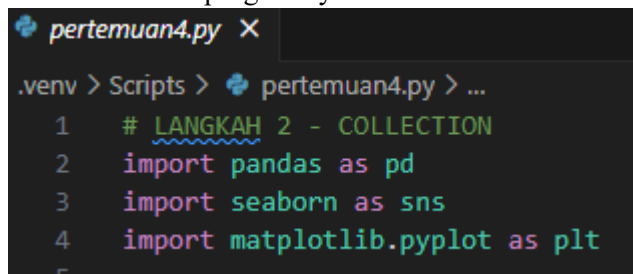
Dataset yang kita dapat, dibuat ke dalam notepad kemudian disimpan dengan nama kelulusan-mahasiswa.csv. Setelah itu disimpan ke folder D machine learning. Datasetnya sebagai berikut :



```
kelulusan_mahasiswa.csv > data
1  IPK,Jumlah Absensi,Waktu Belajar Jam,Lulus
2  3.8,3,10,1
3  2.5,8,5,0
4  3.4,4,7,1
5  2.1,12,2,0
6  3.9,2,12,1
7  2.8,6,4,0
8  3.2,5,8,1
9  2.7,7,3,0
10 3.6,4,9,1
11 2.3,9,4,0
```

##### 2. Collection

Pengumpulan data yang mungkin dibutuhkan dengan beberapa library yang ada di Python. Berikut ini code programnya :



```
pertemuan4.py X
.venv > Scripts > pertemuan4.py > ...
1  # LANGKAH 2 - COLLECTION
2  import pandas as pd
3  import seaborn as sns
4  import matplotlib.pyplot as plt
5
```

- Import pandas as pd ini digunakan untuk memanipulasi dan analisis data terutama data dalam format tabel (Dataframe).
- Import seaborn as sns dan matplotlib digunakan dalam menampilkan hasil analisis data dalam bentuk data statistik dan visual data yang lebih fleksibel.

```
# Baca file csv
df = pd.read_csv("kelulusan_mahasiswa.csv")

# Cek info dan tampilkan 5 baris pertama
print(df.info())
print(df.head())
```

Code diatas untuk membaca file cvs dan menampilkan informasi dataframe dan menampilkan 5 baris dari dataframe.

### 3. CLEANING

Proses menghapus data yang duplikat dan melihat nilai yang hilang.

```
# LANGKAH 3 - CLEANING
# Cek missing values
print("Missing values:\n", df.isnull().sum())

# Hapus duplikat jika ada
df = df.drop_duplicates()

# Visualisasi boxplot untuk deteksi outlier
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=df['IPK'])
plt.show()
```

### 4. Exploratory Data Analysis (EDA)

EDA (Exploratory Data Analysis) adalah proses mengeksplorasi dan memahami data sebelum dilakukan modeling. Tujuannya untuk:

- Mengetahui distribusi data
- Mendeteksi outlier
- Melihat hubungan antar variabel
- Memahami pola-pola penting dalam data

Berikut code programnya :

```
# LANGKAH 4 - EXPLORATORY DATA ANALYSIS ( EDA )
# Statistik deskriptif
print(df.describe())

# Histogram distribusi IPK
sns.histplot(df['IPK'], bins=10, kde=True)
plt.show()

# Scatterplot hubungan IPK vs Waktu Belajar, dengan hue=Lulus
sns.scatterplot(x='IPK', y='Waktu_Belajar_Jam', data=df, hue='Lulus')
plt.show()

# Heatmap korelasi
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.show()
```

**Penjelasannya :**

```
*print(df.describe())
```

**Fungsi:**

Menampilkan **statistik deskriptif** dari kolom-kolom numerik dalam DataFrame df.

```
*sns.histplot(df['IPK'], bins=10, kde=True)
```

**Fungsi:**

Membuat **histogram** dari kolom IPK, dibagi dalam 10 interval (bins), dan menampilkan **kurva KDE (Kernel Density Estimate)** untuk menunjukkan distribusi IPK secara halus.

```
*sns.scatterplot(x='IPK', y='Waktu_Belajar_Jam', data=df, hue='Lulus')
```

**Fungsi:**

Membuat **scatter plot** (diagram sebar)

```
*sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
```

**Fungsi:**

Membuat **heatmap (peta panas)** yang menunjukkan **korelasi antar variabel numerik** dalam DataFrame.

## 5. Feature Engineering

Feature engineering adalah proses membuat fitur baru dari data yang sudah ada untuk:

- Memberikan informasi tambahan kepada model
- Meningkatkan kinerja prediksi atau analisis

Berikut code programnya :

```
# LANGKAH 5 - FEATURE ENGINEERING
# Buat fitur turunan
df['Rasio_Absensi'] = df['Jumlah_Absensi'] / 14
df['IPK_x_Study'] = df['IPK'] * df['Waktu_Belajar_Jam']

# Simpan dataset baru
df.to_csv("processed_kelulusan.csv", index=False)
```

**Penjelasannya :**

\*df['Rasio\_Absensi'] = df['Jumlah\_Absensi'] / 14 : Mengubah nilai absensi dari **jumlah mentah** menjadi **rasio (proporsi kehadiran)** yang lebih representatif, dengan rentang antara 0 dan 1.

\*df['IPK\_x\_Study'] = df['IPK'] \* df['Waktu\_Belajar\_Jam'] : Untuk melihat apakah kombinasi antara **kemampuan dan usaha** memiliki hubungan yang kuat dengan kelulusan.

\*df.to\_csv("processed\_kelulusan.csv", index=False) : Menyimpan **dataset yang sudah diproses**, termasuk fitur baru.

## 6. Splitting Dataset

Memisahkan dataset menjadi tiga bagian:

- Train set → Melatih model
- Validation set → Menguji performa model saat tuning
- Test set → Mengevaluasi akhir performa model

Berikut codenya :

```
# LANGKAH 6 - SPLITTING DATASET
from sklearn.model_selection import train_test_split

# Pisahkan fitur (X) dan label (y)
X = df.drop('Lulus', axis=1)
y = df['Lulus']

# Train (70%) dan sisanya 30% untuk val+test
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)

# Dari sisa 30%, bagi 50:50 -> masing-masing 15%
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42
)

print("Train:", X_train.shape, "Validation:", X_val.shape, "Test:", X_test.shape)
```

### Penjelasanya :

\*from sklearn.model\_selection import train\_test\_split

- Mengimpor fungsi train\_test\_split dari library scikit-learn.
- Digunakan untuk membagi dataset menjadi beberapa subset (train, validation, test).

\*X = df.drop('Lulus', axis=1)

- Memisahkan fitur (variabel input) ke dalam X
- drop('Lulus', axis=1) artinya hapus kolom Lulus dari df, karena itu adalah label (target)

\*y = df['Lulus']

- Mengambil kolom Lulus sebagai target/output (y)

\*Membagi data menjadi train dan sisanya

- Membagi data menjadi 70% train dan 30% sisanya (temp).
- stratify=y: memastikan proporsi label (misalnya “Lulus” vs “Tidak Lulus”) tetap seimbang di setiap set.
- random\_state=42: untuk hasil pembagian yang reproducible (hasilnya akan sama setiap dijalankan).

\*Membagi sisa data (30%) menjadi validation dan test

- Sisa 30% dibagi dua menjadi:  
15% → validation  
15% → test
- stratify=y\_temp: menjaga proporsi label tetap seimbang di kedua set ini.

\*print("Train:", X\_train.shape, "Validation:", X\_val.shape, "Test:", X\_test.shape)

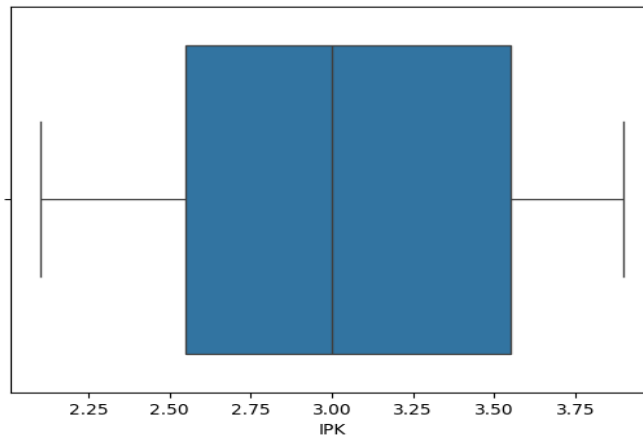
- Menampilkan jumlah baris dan kolom dari masing-masing subset data.
- Train set punya 70 data dan 8 fitur
- Validation dan Test set masing-masing punya 15 data dan 8 fitur

Hasil dari program yang dijalankan sebagai berikut :

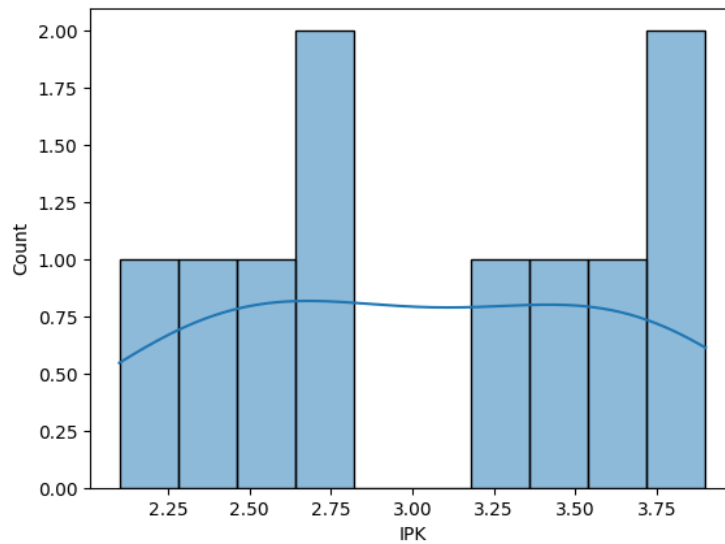
```
<class pandas.core.frame.DataFrame>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   IPK                    10 non-null    float64
1   Jumlah_Absensi        10 non-null    int64
2   Waktu_Belajar_Jam     10 non-null    int64
3   Lulus                  10 non-null    int64
dtypes: float64(1), int64(3)
memory usage: 448.0 bytes
```

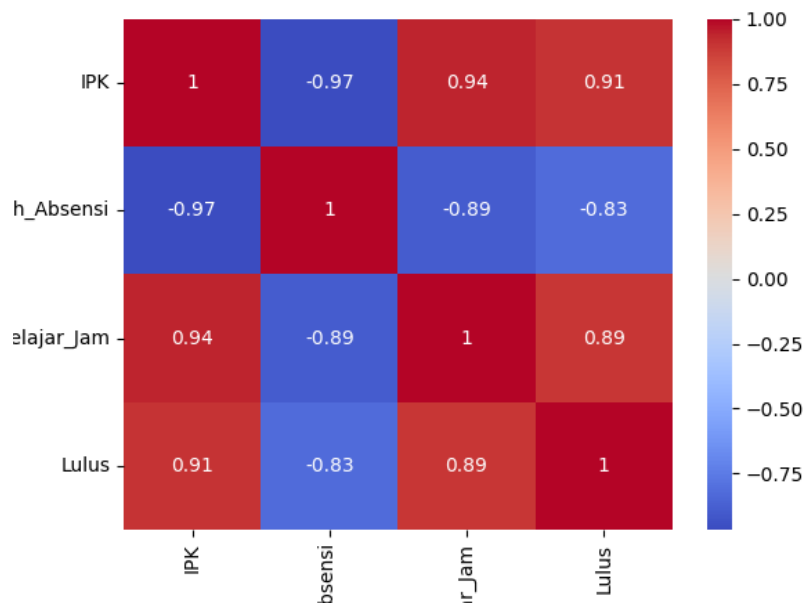
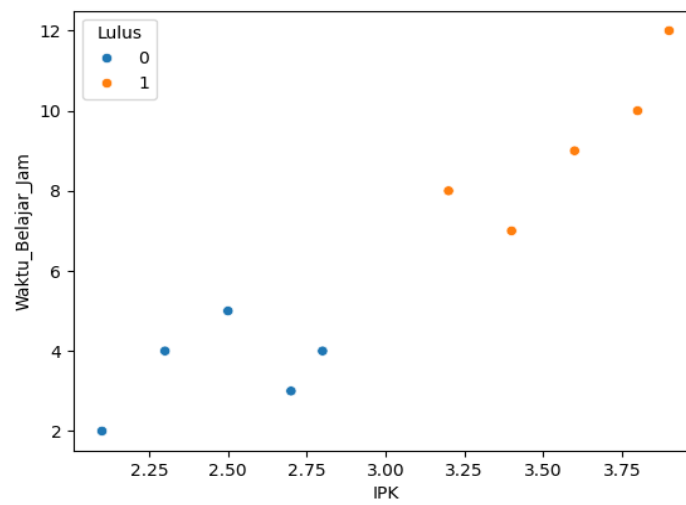
```
IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
0    3.8            3           10         1
1    2.5            8           5          0
2    3.4            4           7          1
3    2.1           12           2          0
4    3.9            2          12          1
```

```
IPK  Jumlah_Absensi  Waktu_Belajar_Jam  Lulus
0    3.8            3           10         1
1    2.5            8           5          0
2    3.4            4           7          1
3    2.1           12           2          0
4    3.9            2          12          1
Missing values:
IPK            0
Jumlah_Absensi 0
Waktu_Belajar_Jam 0
Lulus          0
dtype: int64
```



	IPK	Jumlah_Absensi	Waktu_Belajar_Jam	Lulus
count	10.000000	10.00000	10.000000	10.000000
mean	3.030000	6.00000	6.400000	0.500000
std	0.639531	3.05505	3.306559	0.527046
min	2.100000	2.00000	2.000000	0.000000
25%	2.550000	4.00000	4.000000	0.000000
50%	3.000000	5.50000	6.000000	0.500000
75%	3.550000	7.75000	8.750000	1.000000
max	3.900000	12.00000	12.000000	1.000000





```

Shapes:
X_train: (7, 5)
X_val: (1, 5)
X_test: (2, 5)
PS D:\machine_learning>

```