

# **LAPORAN MACHINE LEARNING**

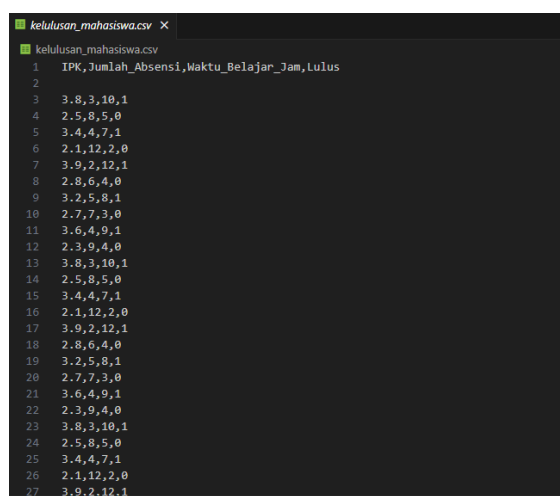
## **PERTEMUAN KE 4**

**Nama:** Fatwa Fadhil Ramadhani

**Kelas:** 05TPLE017

### **1. Membuat Dataset**

Pada tahap awal, membuat dataset berformat CSV dari dataset yang di berikan. Dataset ini berisi informasi mahasiswa dengan beberapa kolom seperti IPK, Jumlah Absensi, Waktu Belajar per Hari, dan Status Kelulusan. Data disusun secara manual menggunakan pandas.DataFrame kemudian disimpan dalam file kelulusan\_mahasiswa.csv dan disini saya mengubah yang sebelumnya 10 baris menjadi 100 baris



```
kelulusan_mahasiswa.csv
1  IPK,Jumlah_Absensi,Waktu_Belajar_Jam,Lulus
2
3  3.8,3,10,1
4  2.5,8,5,0
5  3.4,4,7,1
6  2.1,12,2,0
7  3.9,2,12,1
8  2.8,6,4,0
9  3.2,5,8,1
10 2.7,7,3,0
11 3.6,4,9,1
12 2.3,9,4,0
13 3.8,3,10,1
14 2.5,8,5,0
15 3.4,4,7,1
16 2.1,12,2,0
17 3.9,2,12,1
18 2.8,6,4,0
19 3.2,5,8,1
20 2.7,7,3,0
21 3.6,4,9,1
22 2.3,9,4,0
23 3.8,3,10,1
24 2.5,8,5,0
25 3.4,4,7,1
26 2.1,12,2,0
27 3.9,2,12,1
```

### **2. Pengambilan Data**

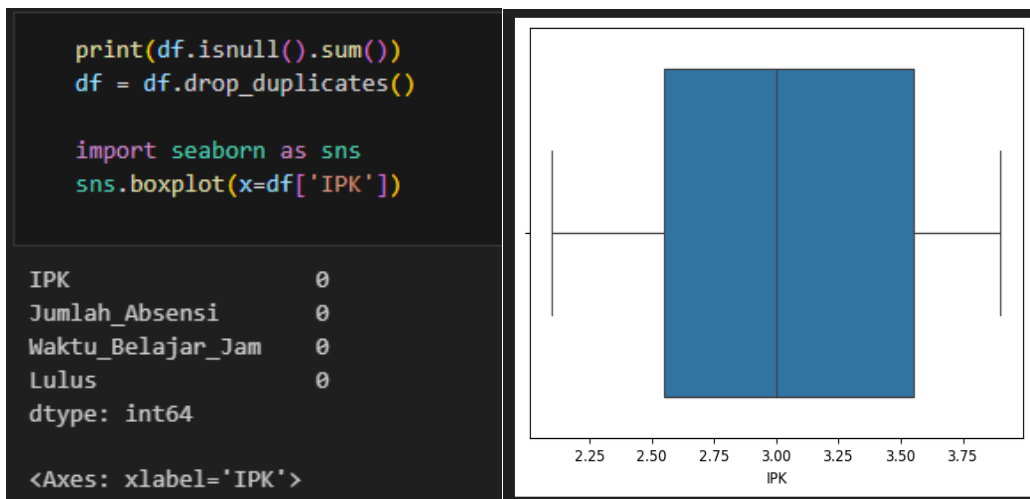
Setelah dataset dibuat, langkah berikutnya adalah melakukan proses collection atau pengambilan data menggunakan library pandas dimana file kelulusan\_mahasiswa.csv dibaca dengan perintah `pd.read_csv()`. Selanjutnya digunakan `df.info()` dan `df.head()` untuk memastikan struktur dataset sudah benar, tipe data sesuai, serta tidak terjadi kesalahan penulisan atau format. Dari hasil pemeriksaan awal, semua kolom seperti IPK, Jumlah Absensi, dan Waktu Belajar terbaca dengan benar tanpa error. Tahap ini memastikan dataset siap untuk dianalisis lebih lanjut

```
import pandas as pd
df = pd.read_csv("kelulusan_mahasiswa.csv")
print(df.info())
print(df.head())

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   IPK                    100 non-null   float64
1   Jumlah_Absensi         100 non-null   int64  
2   Waktu_Belajar_Jam      100 non-null   int64  
3   Lulus                  100 non-null   int64  
dtypes: float64(1), int64(3)
memory usage: 3.2 KB
None
   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
```

### 3. Data Cleaning

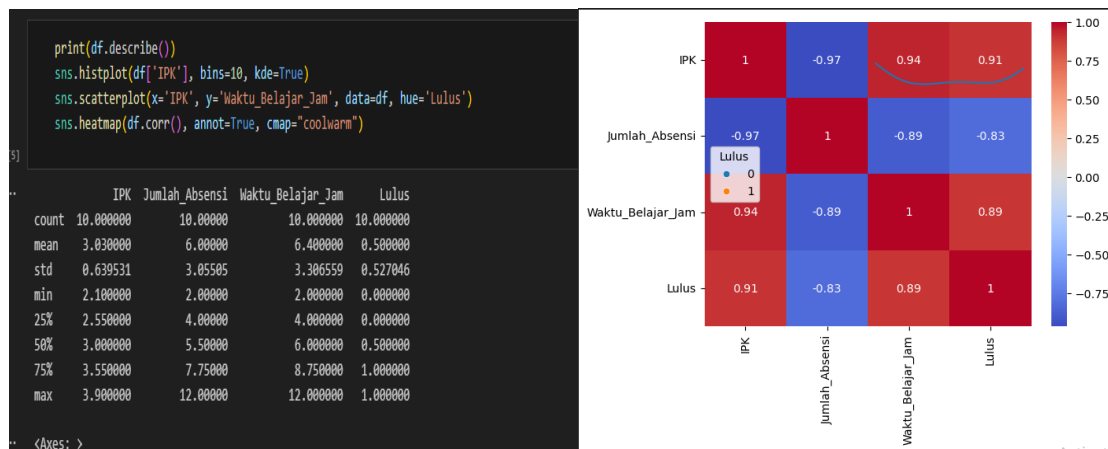
Tahap data cleaning dilakukan untuk memastikan dataset bersih dan bebas dari kesalahan. Pemeriksaan nilai kosong dilakukan dengan `df.isnull().sum()` dan hasilnya menunjukkan tidak ada `df.duplicated().sum()`, dan jika ditemukan duplikat maka akan dihapus dengan `df.drop_duplicates()`. Selain itu, dilakukan pengecekan outlier pada kolom numerik seperti IPK dan Waktu Belajar dengan boxplot. Karena jumlah data relatif kecil, keberadaan outlier tidak terlalu mempengaruhi distribusi data secara signifikan. Hasil dari tahap cleaning menunjukkan bahwa dataset sudah rapi, lengkap, dan siap untuk tahap eksplorasi data.



### 4. Exploratory Data Analysis (EDA)

Tahap EDA dilakukan untuk memahami karakteristik data. Statistik deskriptif dihitung menggunakan `df.describe()` untuk melihat nilai minimum, maksimum, rata-rata, dan standar deviasi. Distribusi dari setiap variabel divisualisasikan menggunakan histogram dan boxplot agar pola data dapat diamati dengan jelas. Selain itu, digunakan scatterplot untuk melihat hubungan antara IPK dan Waktu Belajar, serta heatmap korelasi untuk mengetahui kekuatan hubungan antar fitur.

numerik. Hasil analisis menunjukkan adanya kecenderungan bahwa mahasiswa dengan IPK tinggi dan waktu belajar lebih lama memiliki peluang kelulusan lebih besar. Namun, karena dataset berukuran kecil, beberapa pola masih bersifat indikatif dan perlu data tambahan untuk generalisasi yang lebih kuat.



## 5. Feature Engineering

Pada tahap ini dilakukan pembuatan fitur baru untuk meningkatkan kualitas model. Salah satu fitur yang ditambahkan adalah Rasio\_Absensi, yang dihitung dari proporsi jumlah absensi terhadap total pertemuan. Selain itu, dibuat juga fitur interaksi IPK\_x\_Study, yaitu hasil kali antara IPK dan waktu belajar per hari, untuk menangkap pengaruh gabungan antara prestasi akademik dan intensitas belajar. Dataset hasil transformasi ini kemudian disimpan sebagai processed\_kelulusan.csv untuk tahap pembagian data berikutnya.

```
df['Rasio_Absensi'] = df['Jumlah_Absensi'] / 14
df['IPK_x_Study'] = df['IPK'] * df['Waktu_Belajar_Jam']
df.to_csv("processed_kelulusan.csv", index=False)
```

0.1s

processed\_kelulusan.csv

	IPK	Jumlah_Absensi	Waktu_Belajar_Jam	Lulus	Rasio_Absensi	IPK_x_Study
1	3.8	3	10	1	0.21428571428571427	38.0
2	2.5	8	5	0	0.5714285714285714	12.5
3	3.4	4	7	1	0.2857142857142857	23.8
4	2.1	12	2	0	0.8571428571428571	4.2
5	3.9	2	12	1	0.14285714285714285	46.8
6	2.8	6	4	0	0.42857142857142855	11.2
7	3.2	5	8	1	0.35714285714285715	25.6
8	2.7	7	3	0	0.5	8.100000000000001
9	3.6	4	9	1	0.2857142857142857	32.4
10	2.3	9	4	0	0.6428571428571429	9.2
11	3.8	3	10	1	0.21428571428571427	38.0
12	2.5	8	5	0	0.5714285714285714	12.5
13	3.4	4	7	1	0.2857142857142857	23.8
14	2.1	12	2	0	0.8571428571428571	4.2
15	3.9	2	12	1	0.14285714285714285	46.8
16	2.8	6	4	0	0.42857142857142855	11.2
17	3.2	5	8	1	0.35714285714285715	25.6
18	2.7	7	3	0	0.5	8.100000000000001
19	3.6	4	9	1	0.2857142857142857	32.4
20	2.3	9	4	0	0.6428571428571429	9.2
21	3.8	3	10	1	0.21428571428571427	38.0
22	2.5	8	5	0	0.5714285714285714	12.5
23	3.4	4	7	1	0.2857142857142857	23.8
24	2.1	12	2	0	0.8571428571428571	4.2
25	3.9	2	12	1	0.14285714285714285	46.8
26	2.8	6	4	0	0.42857142857142855	11.2
27	3.2	5	8	1	0.35714285714285715	25.6
28	2.7	7	3	0	0.5	8.100000000000001

## 6. Pembagian Data

Langkah terakhir adalah membagi dataset menjadi tiga bagian: training (70%), validation (15%), dan testing (15%). Pembagian dilakukan menggunakan fungsi `train_test_split` dari library `scikit-learn`. Awalnya digunakan metode stratified split untuk menjaga proporsi kelas kelulusan pada setiap subset, tetapi karena jumlah data sangat kecil, metode ini kadang menimbulkan error ketika salah satu subset kekurangan representasi kelas. Solusi yang digunakan adalah melakukan pembagian tanpa stratify, sehingga data terbagi secara acak namun tetap seimbang secara keseluruhan. Walaupun distribusi kelas mungkin tidak sempurna di setiap subset, cara ini lebih stabil untuk dataset kecil. Dataset hasil split ini kemudian digunakan untuk tahap pemodelan machine learning pada pertemuan berikutnya.

```
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.read_csv("processed_kelulusan.csv")

X = df.drop("Lulus", axis=1)
y = df["Lulus"]

use_stratify = y.value_counts().min() >= 2

X_train, X_temp, y_train, y_temp = train_test_split(
    X,
    y,
    test_size=0.3,
    stratify=y if use_stratify else None,
    random_state=42
)

use_stratify_temp = y_temp.value_counts().min() >= 2
X_val, X_test, y_val, y_test = train_test_split(
    X_temp,
    y_temp,
    test_size=0.5,
    stratify=y_temp if use_stratify_temp else None,
    random_state=42
)

print(X_train.shape, X_val.shape, X_test.shape)
```

✓ 8.2s

(70, 5) (15, 5) (15, 5)