1. Membuat dataset dengan nama kelulusan_mahasiswa.csv’

	IPK	Jumlah_Absensi	Waktu_Belajar_Jam	Lulus
1	3.8	3	10	1
2	2.5	8	5	0
3	3.4	4	7	1
4	2.1	12	2	0
5	3.9	2	12	1
6	2.8	6	4	0
7	3.2	5	8	1
8	2.7	7	3	0
9	3.6	4	9	1
10	2.3	9	4	0

2. Collection (Membaca Data)
   Membuka dan memahami struktur dataset menggunakan Pandas. Data ini nantinya akan digunakan untuk proses analisis dan pemodelan Machine Learning pada pertemuan berikutnya.

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

```
<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: 452.0 bytes
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
```

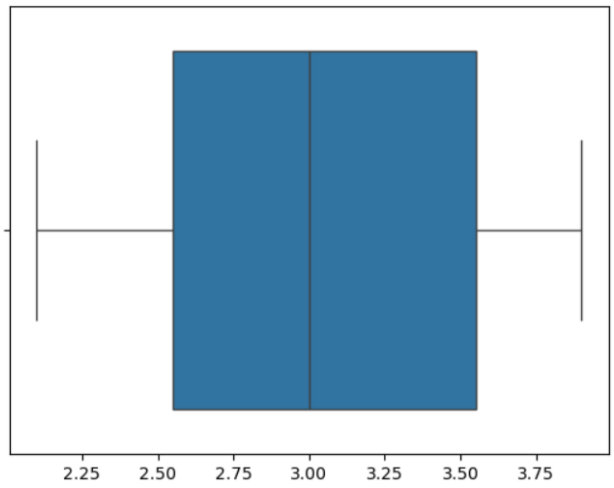
3. Cleaning (Pembersihan Data)
   Pemeriksaan data untuk memastikan tidak terdapat nilai yang kosong (null) maupun duplikat. Selanjutnya dilakukan visualisasi menggunakan boxplot untuk memeriksa distribusi nilai IPK dan memastikan tidak terdapat nilai yang menyimpan (outlier)

```
print(df.isnull().sum())
df = df.drop_duplicates()

import seaborn as sns
sns.boxplot(x=df['IPK'])
```

```
IPK                 0
Jumlah_Absensi      0
Waktu_Belajar_Jam   0
Lulus               0
dtype: int64
```

```
<Axes: xlabel='IPK'>
```

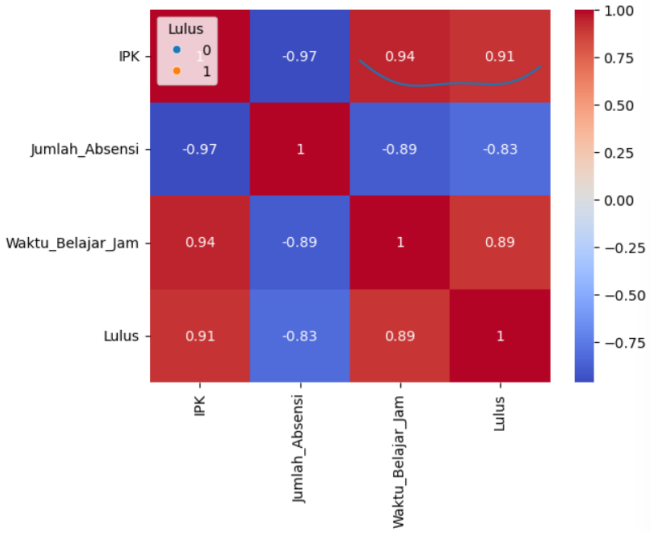


4. Explorotory Data Analysis (EDA)

   Analisis statistic deskriptif dilakukan dengan describe() untuk melihat ringkasan data, serta visualisasi berupa histogram, scatter plot, dan heatmap untuk memahami distribusi, hubungan dan korelasi antar variable

```
print(df.describe())
sns.histplot(df['IPK'], bins=10, kde=True)
sns.scatterplot(x='IPK', y='Waktu_Belajar_Jam', data=df, hue='Lulu
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
```

```
              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
<Axes: >
```



5. Feature Engineering
   - Rasio_Absensi, hasil pembagian jumlah absensi dengan total pertemuan (14) untuk melihat tingkat kehadiran mahasiswa
   - IPK_x_Study, hasil perkalian antara IPK dan waktu belajar untuk menggambarkan hubungan antara performa akademik dan durasi belajar

   Datset hasil transformasi kemudian disimpan ke file processed_kelulusan.csv

```
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)

print(df.head())
```

	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	0.4285714285714285	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

6. Splitting Dataset

   Error ini terjadi karena nilai pada kolom lulus (0 atau 1) terlalu sedikit saat dibagi, sedangkan dataset hanya memiliki 10 baris data. Akibatnya dilakukan pembagian data (70%-15%-15%) salah satu subset hanya memiliki 1data dari kelas tertentu, sehingga menyebabkan error ini

```
---> 13 X_val, X_test, y_val, y_test = train_test_split(
     14     X_temp, y_temp, test_size=0.5, stratify=y_temp,
     16 print(X_train.shape, X_val.shape, X_test.shape)

-> 2681     train, test = next(cv.split(X=arrays[0], y=stratify))

ValueError: The least populated class in y has only 1 member, which is too few. The minimum number of groups for any class cannot be less than 2.
```

   Akhirnya solusinya adalah dengan menghapus parameter stratify pada pembagian data kedua, proses pembagian dataset dapat berjalan karena tidak lagi memaksa keseimbangan kelas pada data yang jumlahnya sangat sedikit

```python
from sklearn.model_selection import train_test_split

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

# split pertama: train (70%) + temp (30%)  — pakai stratify supaya proporsi kelas terjaga
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42)

# split kedua: val & test — TIDAK pakai stratify (karena y_temp terlalu kecil)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=42)

print("Shapes:", X_train.shape, X_val.shape, X_test.shape)
print("y_train counts:\n", y_train.value_counts())
print("y_val counts:\n", y_val.value_counts())
print("y_test counts:\n", y_test.value_counts())
```

```
Shapes: (7, 5) (1, 5) (2, 5)
y_train counts:
 Lulus
1    4
0    3
Name: count, dtype: int64
y_val counts:
 Lulus
1    1
Name: count, dtype: int64
y_test counts:
 Lulus
0    2
Name: count, dtype: int64
```