Data Preparation

1. **Collection**

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

[3] Python

```
<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
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
```

Langkah pertama yaitu membaca file **kelulusan_mahasiswa.csv** menggunakan library pandas.
Data ini berisi 10 baris dan 4 kolom, yaitu **IPK**, **Jumlah Absensi**, **Waktu Belajar Jam**, dan **Lulus**.
Dari hasil pengecekan awal, semua data sudah lengkap tanpa ada nilai kosong.
Dengan melihat lima data pertama, kita bisa mengetahui bentuk dan isi datanya secara umum sebelum diproses lebih lanjut.

2. **Cleaning**

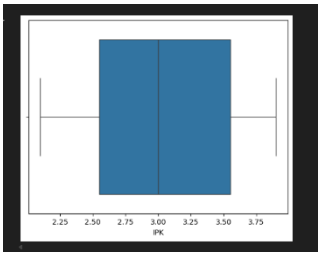
```python
print(df.isnull().sum())
df = df.drop_duplicates()

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

[5] Python

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

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



Setelah data dibaca, langkah berikutnya adalah memastikan bahwa data tidak memiliki nilai kosong ataupun data yang sama (duplikat).
Hasil pengecekan menunjukkan bahwa semua data lengkap dan tidak ada baris yang berulang.

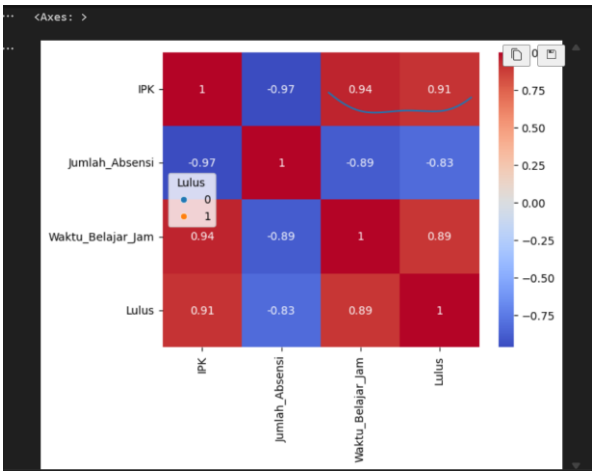
Selanjutnya, dibuat **boxplot** untuk kolom **IPK**.
Grafik ini membantu melihat apakah ada nilai yang terlalu ekstrem (*outlier*).
Hasilnya menunjukkan bahwa nilai IPK mahasiswa berkisar antara **2.1 hingga 3.9**, dan tidak ditemukan data yang menyimpang jauh.
Artinya, data IPK cukup bersih dan bisa langsung digunakan untuk analisis.

3. **Exploratory Data Analysis (EDA)**

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

[6] Python

	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



Pada tahap ini, dilakukan analisis deskriptif untuk melihat gambaran umum dari data, seperti nilai rata-rata, minimum, maksimum, dan sebaran datanya.
Dari hasilnya:

- Rata-rata IPK mahasiswa sekitar **3.03**
- Rata-rata absensi sekitar **6 kali**
- Rata-rata waktu belajar sekitar **6,4 jam**

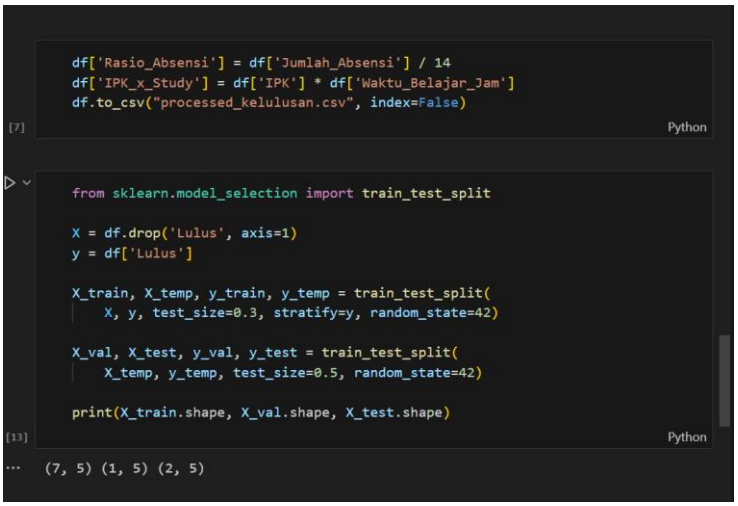
Selain itu, dibuat grafik seperti **histogram**, **scatterplot**, dan **heatmap** untuk melihat hubungan antar variabel.

Dari hasil korelasi, diperoleh:

- **IPK dan Lulus** memiliki korelasi positif kuat (**0.91**) → semakin tinggi IPK, semakin besar peluang lulus.
- **Jumlah Absensi dan IPK** memiliki korelasi negatif kuat (**-0.97**) → semakin banyak absen, IPK cenderung menurun.
- **Waktu Belajar dan IPK** juga punya korelasi positif (**0.94**) → makin sering belajar, makin tinggi IPK.

Kesimpulannya, variabel-variabel ini memiliki hubungan yang masuk akal dan saling berpengaruh terhadap kelulusan mahasiswa.

4. **Feature Engineering & Splitting Dataset**



```
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)
```

[7] Python

```
from sklearn.model_selection import train_test_split

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

X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42)

X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=42)

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

[13] Python

```
... (7, 5) (1, 5) (2, 5)
```

Untuk memperkaya informasi, dibuat dua fitur baru:

- **Rasio_Absensi**, yaitu jumlah absensi dibagi total pertemuan (14 kali).
- **IPK_x_Study**, yaitu hasil perkalian antara IPK dan waktu belajar.

Fitur ini membantu memberikan sudut pandang baru bagi model nantinya, misalnya melihat efek gabungan antara IPK dan intensitas belajar.

Setelah itu, data dibagi menjadi tiga bagian:

- **70% untuk data latih** (train)
- **15% untuk data validasi** (validation)
- **15% untuk data uji** (test)

Hasil pembagian data menunjukkan bentuk (7, 5), (1, 5), dan (2, 5), artinya pembagian berhasil dan siap digunakan untuk tahap pemodelan.

Kesimpulan :

Dari seluruh proses ini, dapat disimpulkan bahwa:

- Data sudah bersih, lengkap, dan tidak ada duplikasi.
- Nilai-nilai antar variabel menunjukkan hubungan yang logis, terutama antara IPK, waktu belajar, dan kelulusan.
- Penambahan fitur baru membantu memperkaya informasi dalam data.
- Dataset sudah dibagi dengan baik sehingga siap digunakan untuk proses **pembuatan model prediksi kelulusan mahasiswa** di tahap selanjutnya.