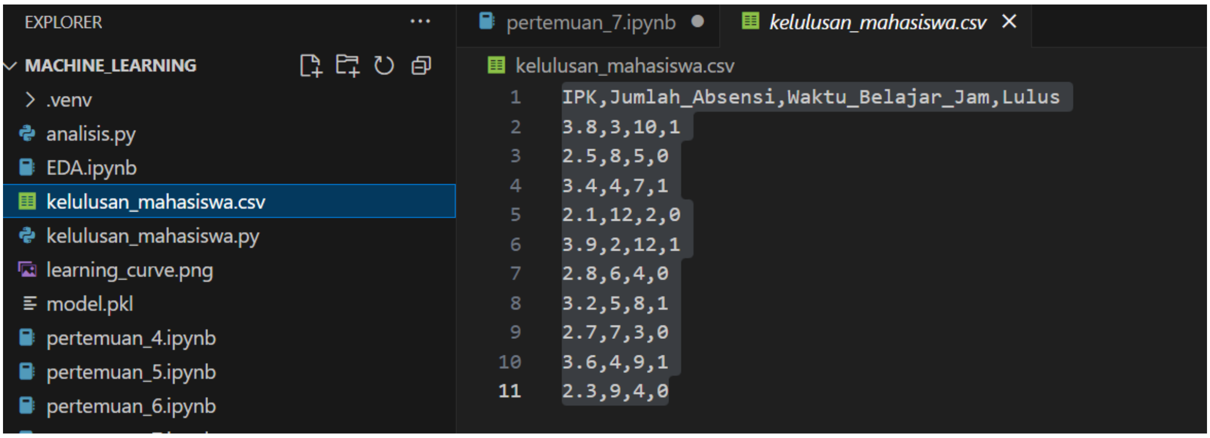
Laporan pengerjaan Pertemuan 4

Langkah 1 — Buat Dataset CSV

1. Ketikkan dataset berikut di file teks baru, lalu simpan dengan nama kelulusan_mahasiswa.csv:

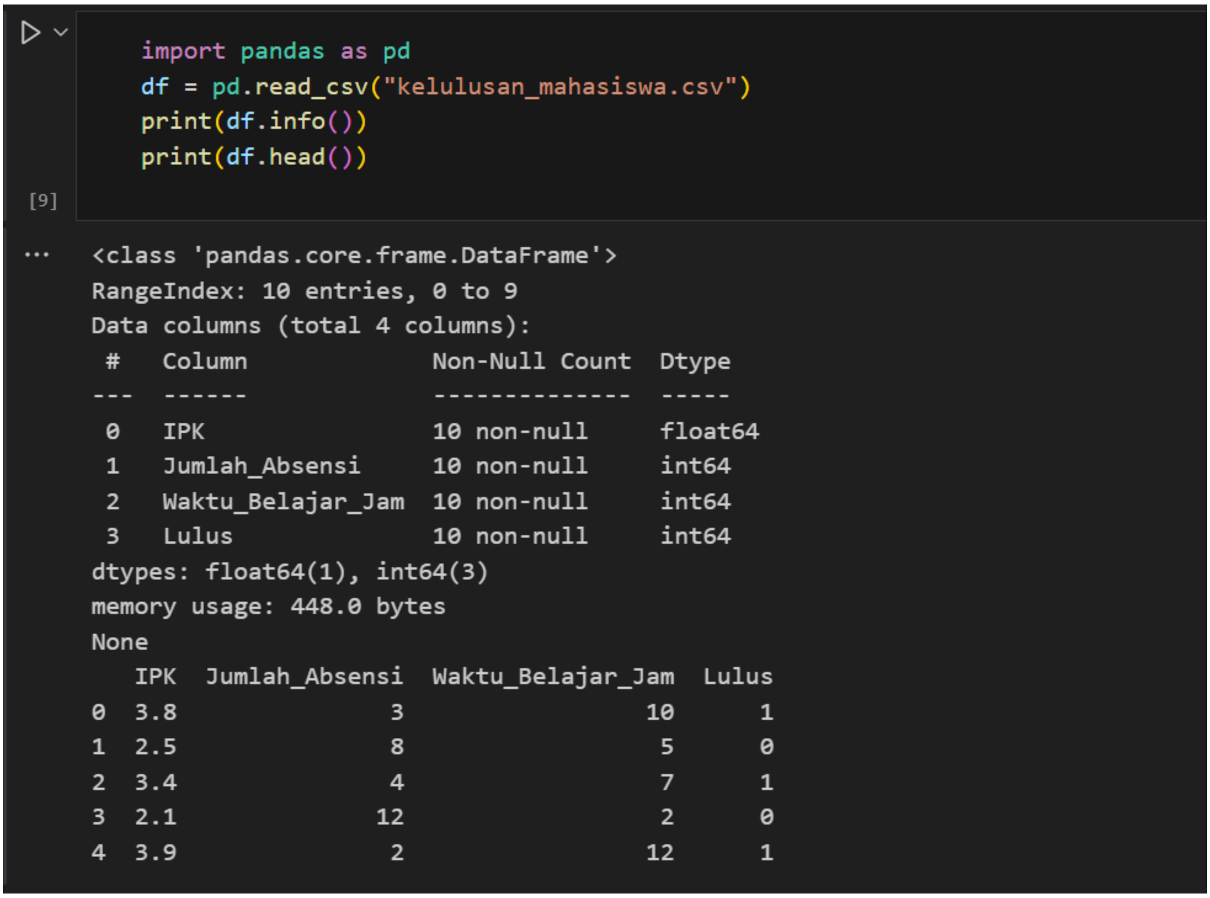


```
IPK,Jumlah_Absensi,Waktu_Belajar_Jam,Lulus
3.8,3,10,1
2.5,8,5,0
3.4,4,7,1
2.1,12,2,0
3.9,2,12,1
2.8,6,4,0
3.2,5,8,1
2.7,7,3,0
3.6,4,9,1
2.3,9,4,0
```

Pastikan format CSV menggunakan koma (,) sebagai pemisah, baris pertama adalah header.

Langkah 2 — Collection

1. Buka file CSV dengan Pandas dan tampilkan info dataset:



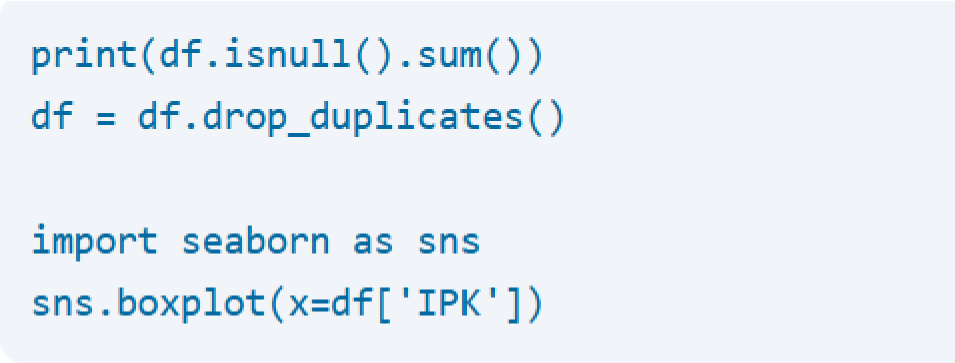
```
import pandas as pd
df = pd.read_csv("kelulusan_mahasiswa.csv")
print(df.info())
print(df.head())
```

```
<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: 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 3 — Cleaning

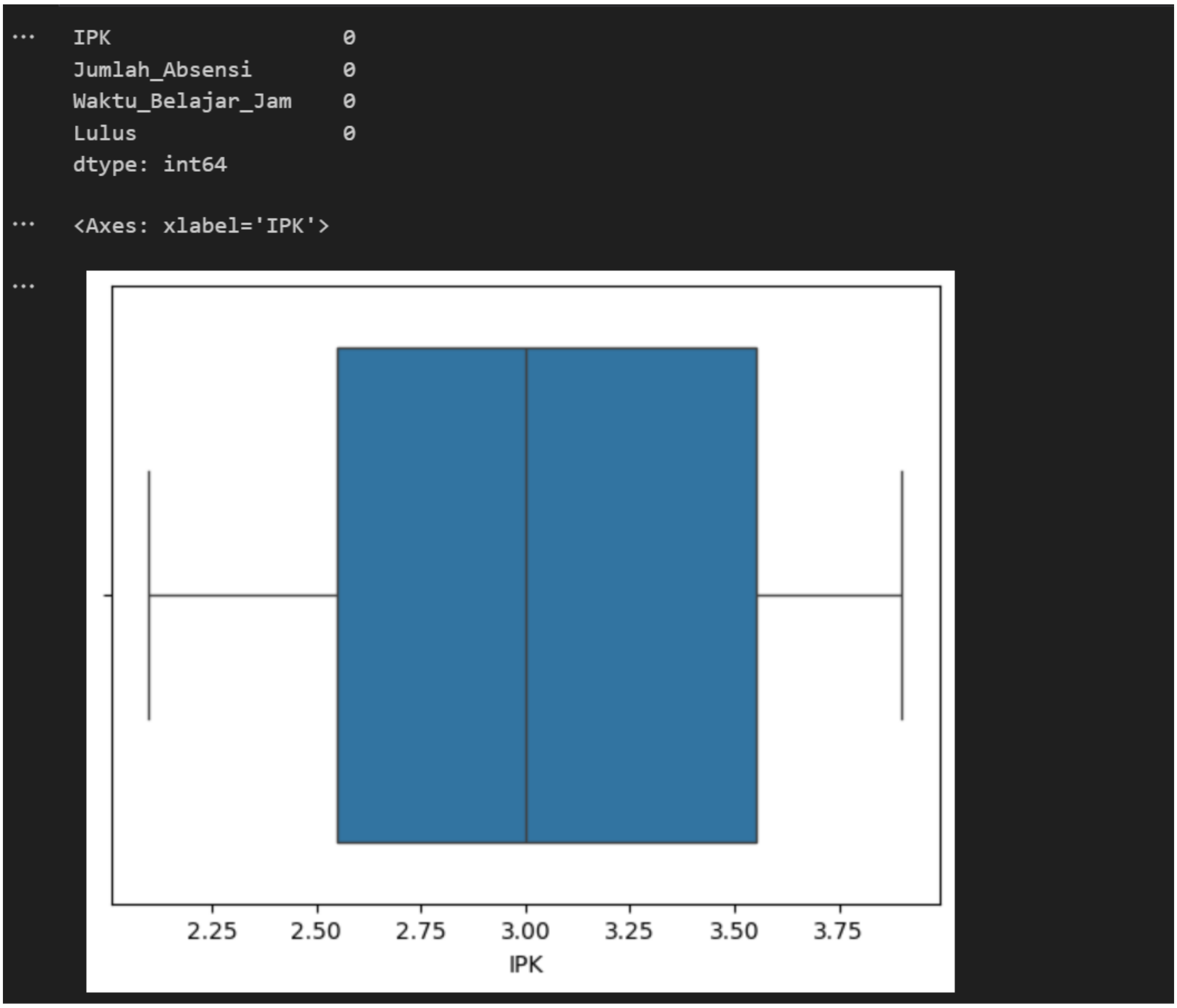
- Periksa *missing value* dan tangani (isi median/modus).
- Hapus data duplikat.
- Identifikasi outlier dengan boxplot.

1. copy paste kode dari lembar kerja ke vscode



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

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

2. Setelah itu dapat hasilnya :

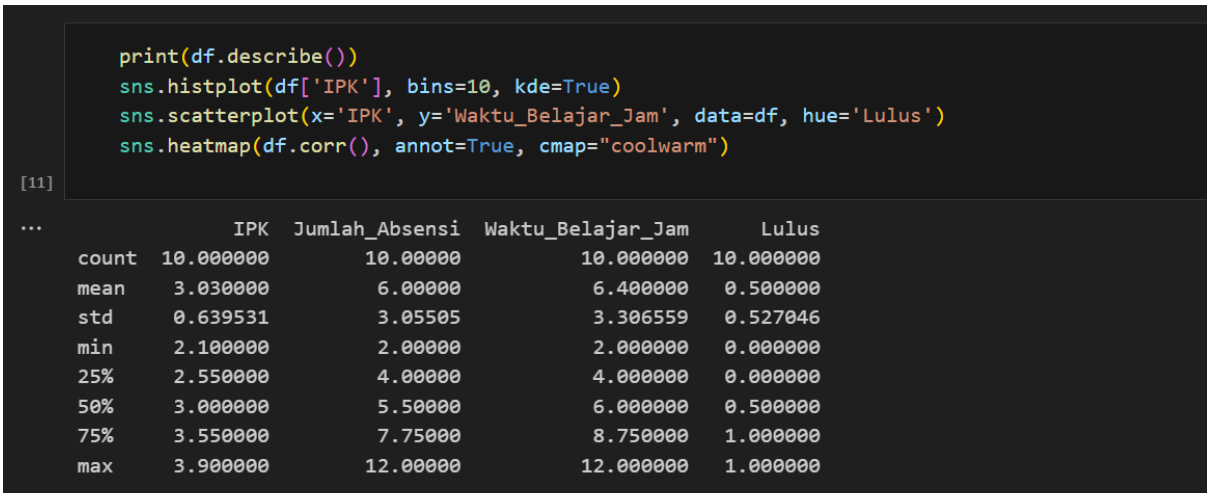


```
...   IPK                0
      Jumlah_Absensi     0
      Waktu_Belajar_Jam  0
      Lulus              0
      dtype: int64

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

...
```

Langkah 4 — Exploratory Data Analysis (EDA)

- Hitung statistik deskriptif.
- Buat histogram distribusi IPK.
- Visualisasi scatterplot (IPK vs Waktu Belajar).
- Tampilkan heatmap korelasi.

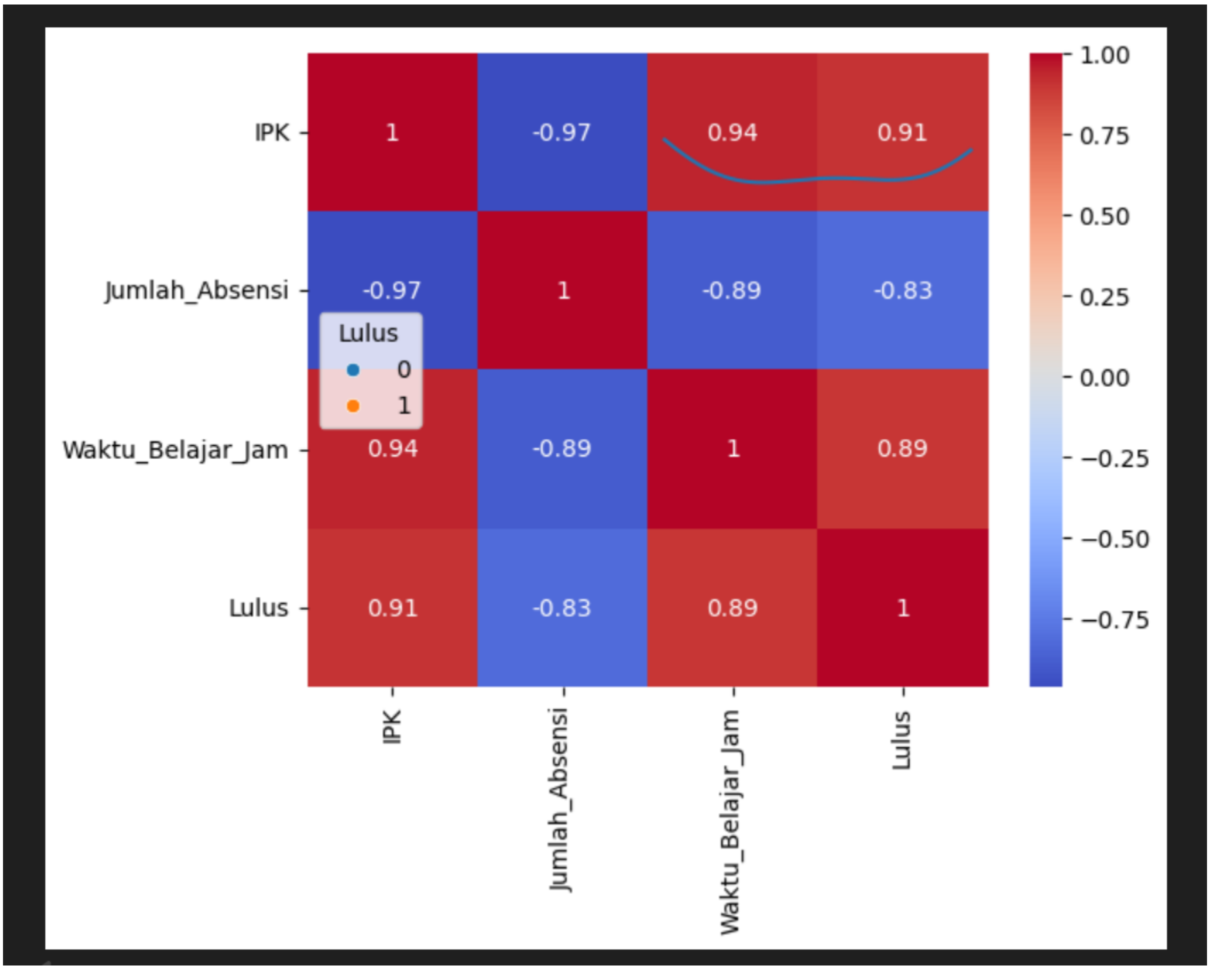
1. Copy paste dari lembar kerja ke vscode dan dapat hasilnya seperti ini



```
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")
```

[11]

```
             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
```

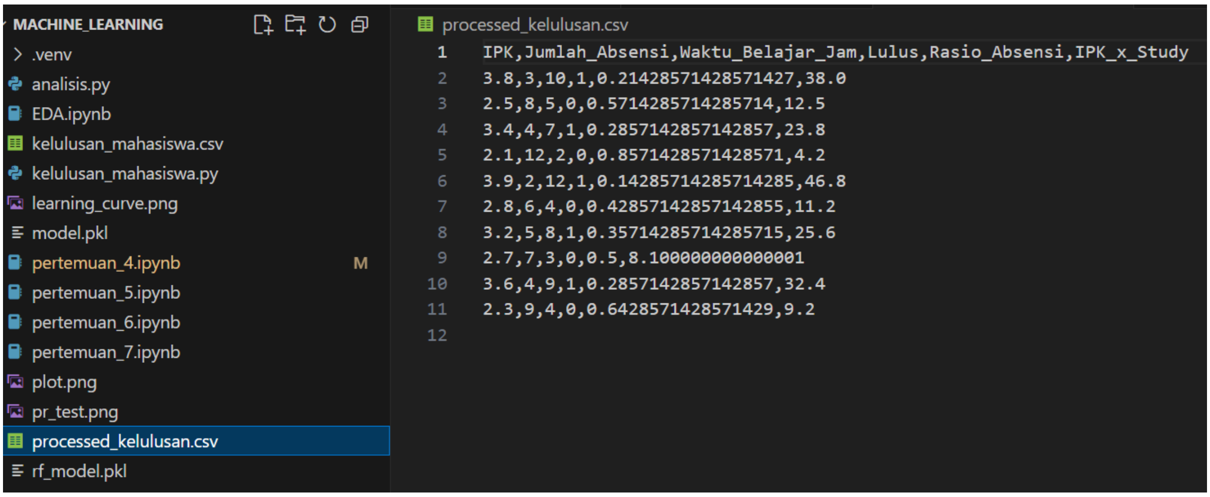


Langkah 5 — Feature Engineering

Buat fitur turunan baru:

```
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)
```

Masukkan kodenya dari lembar kerja ke vscode, hasilnya kita dapat file baru seperti ini



```
IPK,Jumlah_Absensi,Waktu_Belajar_Jam,Lulus,Rasio_Absensi,IPK_x_Study
3.8,3,10,1,0.21428571428571427,38.0
2.5,8,5,0,0.5714285714285714,12.5
3.4,4,7,1,0.2857142857142857,23.8
2.1,12,2,0,0.8571428571428571,4.2
3.9,2,12,1,0.14285714285714285,46.8
2.8,6,4,0,0.42857142857142855,11.2
3.2,5,8,1,0.35714285714285715,25.6
2.7,7,3,0,0.5,8.100000000000001
3.6,4,9,1,0.2857142857142857,32.4
2.3,9,4,0,0.6428571428571429,9.2
```

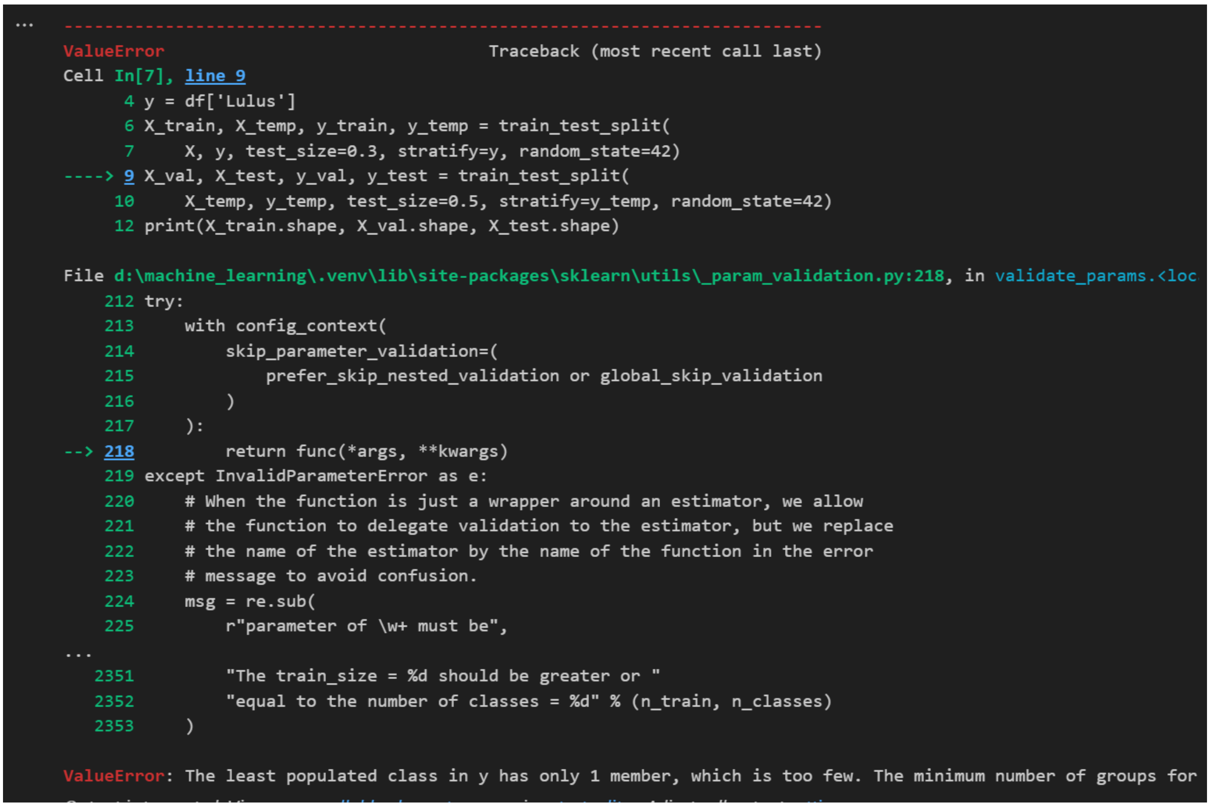
Langkah 6 — Splitting Dataset

Bagi dataset menjadi Train (70%), Validation (15%), Test (15%) menggunakan stratified split:

1. Copy paste dari lembar kerja ke vscode

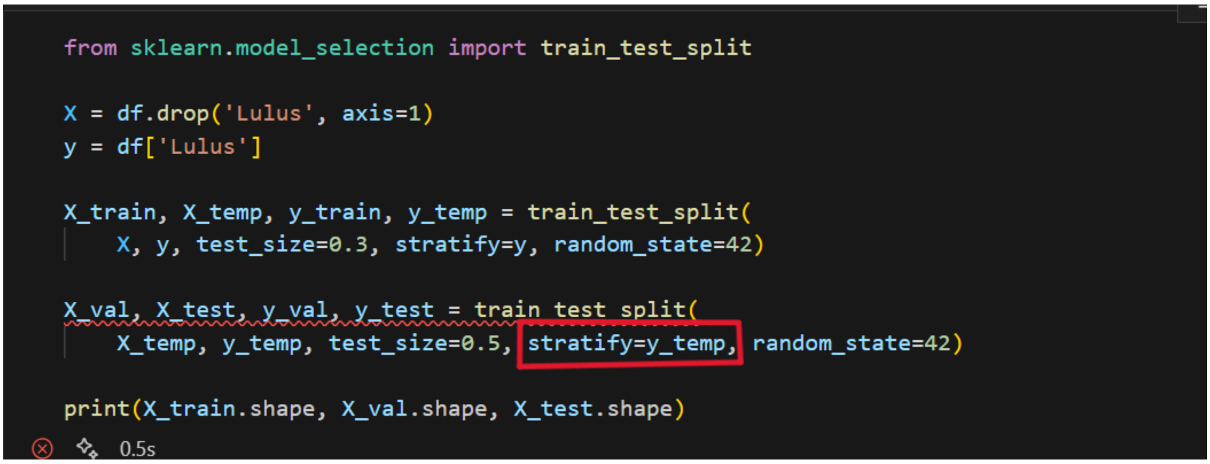
```
from sklearn.model_selection import train_test_split

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, stratify=y_temp, random_state=42)

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

2. setelah di paste ternyata eror



```
...  ---------------------------------------------------------------------------
     ValueError                                Traceback (most recent call last)
     Cell In[7], line 9
           4 y = df['Lulus']
           6 X_train, X_temp, y_train, y_temp = train_test_split(
           7     X, y, test_size=0.3, stratify=y, random_state=42)
     ----> 9 X_val, X_test, y_val, y_test = train_test_split(
          10     X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42)
          12 print(X_train.shape, X_val.shape, X_test.shape)

     File d:\machine_learning\.venv\lib\site-packages\sklearn\utils\_param_validation.py:218, in validate_params.<loc
         212 try:
         213     with config_context(
         214         skip_parameter_validation=(
         215             prefer_skip_nested_validation or global_skip_validation
         216         )
         217     ):
     --> 218         return func(*args, **kwargs)
         219 except InvalidParameterError as e:
         220     # When the function is just a wrapper around an estimator, we allow
         221     # the function to delegate validation to the estimator, but we replace
         222     # the name of the estimator by the name of the function in the error
         223     # message to avoid confusion.
         224     msg = re.sub(
         225         r"parameter of \w+ must be",
     ...
        2351         "The train_size = %d should be greater or "
        2352         "equal to the number of classes = %d" % (n_train, n_classes)
        2353     )

     ValueError: The least populated class in y has only 1 member, which is too few. The minimum number of groups for
```

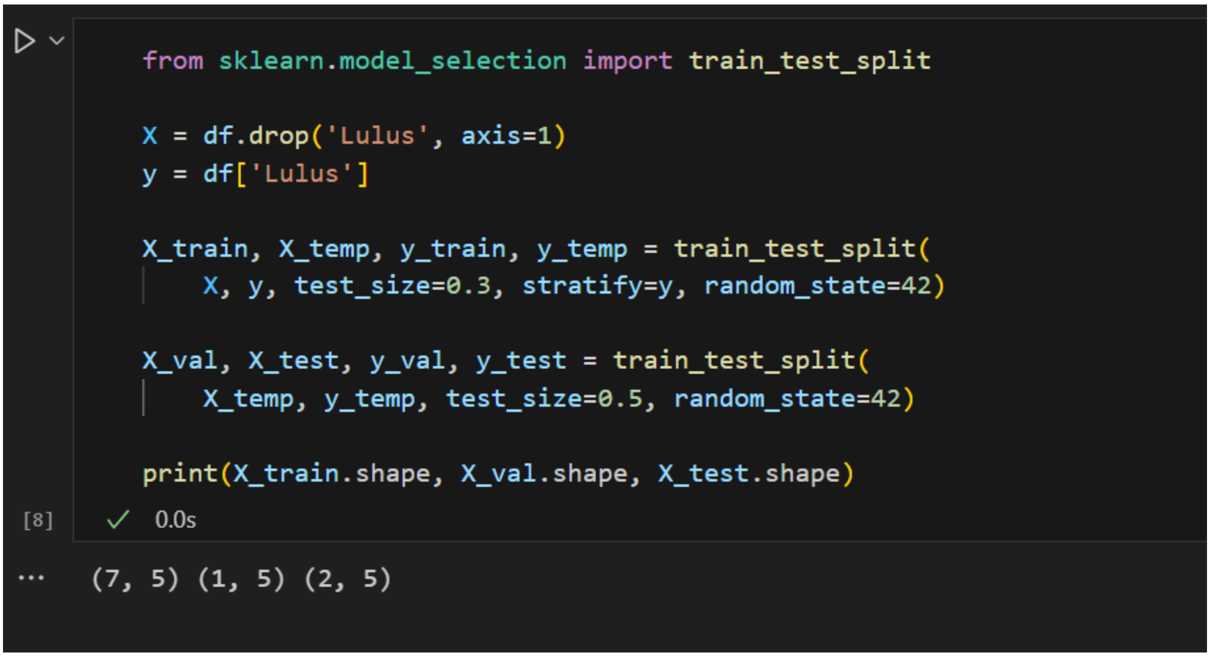
3. Setelah itu kita hapus yang kita tandai dibawah ini



```
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, stratify=y_temp, random_state=42)

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

0.5s

4. Akhirnya tidak eror dan ini hasilnya



```
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)
```

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