Tugas 1: Praktikum 2 - Tugas Praktimum 2 Machine Learning

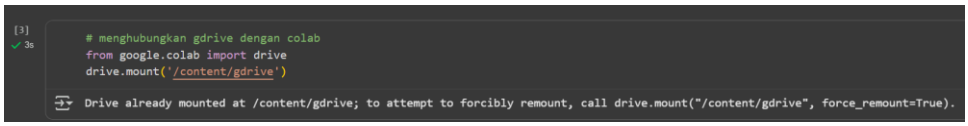
Jamilatun Khoerunnisa - 010222254

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: Jami22254ti@student.nurulfikri.ac.id

Praktikum

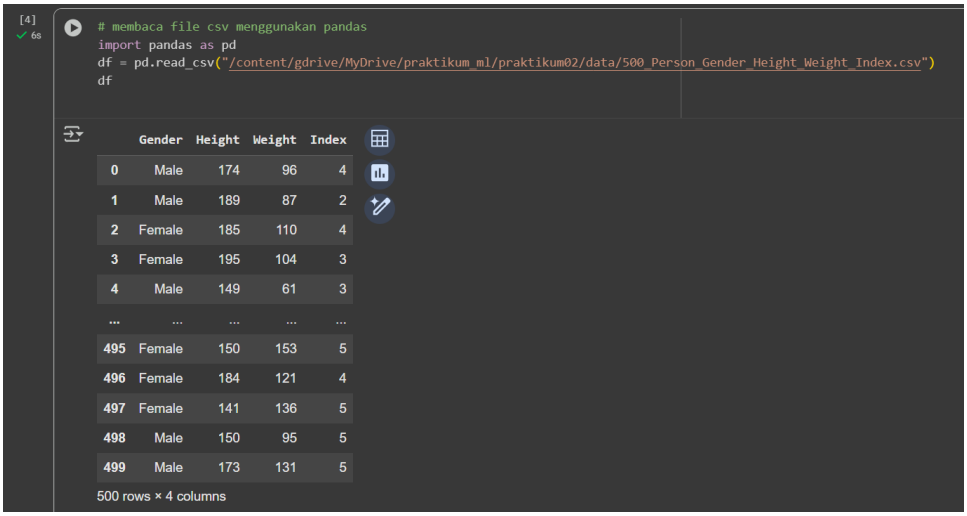
1. Langkah 1



```
[3]
✓ 3s
    # menghubungkan gdrive dengan colab
    from google.colab import drive
    drive.mount('/content/gdrive')

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

Pada Langkah pertama pengerjaan adalah menghubungkan gdrive dan gcolab. **from google.colab import drive drive.mount('/content/gdrive')** adalah perintah untuk menghubungkan.

2. Langkah 2



```
[4]
✓ 6s
# membaca file csv menggunakan pandas
import pandas as pd
df = pd.read_csv("/content/gdrive/MyDrive/praktikum_ml/praktikum02/data/500_Person_Gender_Height_Weight_Index.csv")
df
```

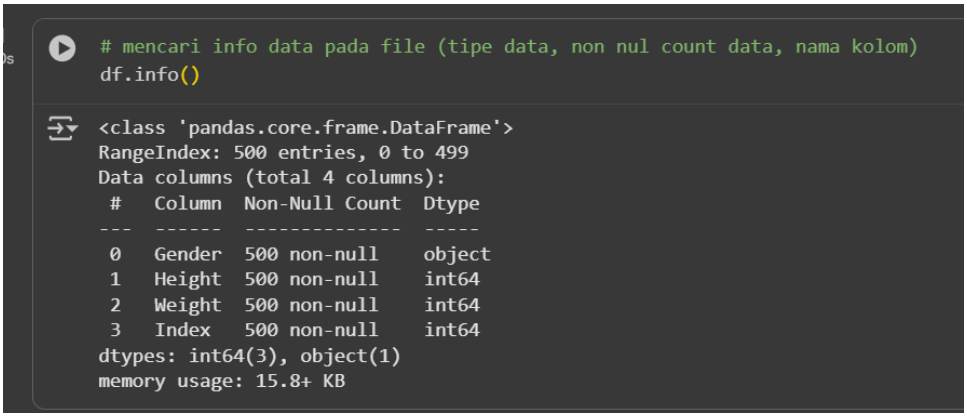
	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3
...	...	...	...	...
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

500 rows × 4 columns

Langkah kedua yaitu membaca data (file csv) menggunakan pandas, **df = pd.read_csv("/content/gdrive/MyDrive/praktikum_ml/praktikum02/data/500_Person_**

Gender_Height_Weight_Index.csv") untuk membaca file csv yang diberikan dan menyimpan file pada dataframe (df). **Df** untuk menampilkan isi dari dataframe.

3. **Langkah 3**



```
# mencari info data pada file (tipe data, non nul count data, nama kolom)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64
 2   Weight  500 non-null    int64
 3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

Pada Langkah ketiga **df.info()** untuk menampilkan ringkasan informasi tentang dataframe.

4. **Langkah 4**

```
[6]
0s
# menghitung mean semua kolom numerik
df['Height'].mean()
```

```
np.float64(169.944)
```

```
[7]
0s
# menghitung median semua kolom numerik
df['Height'].median()
```

```
170.5
```

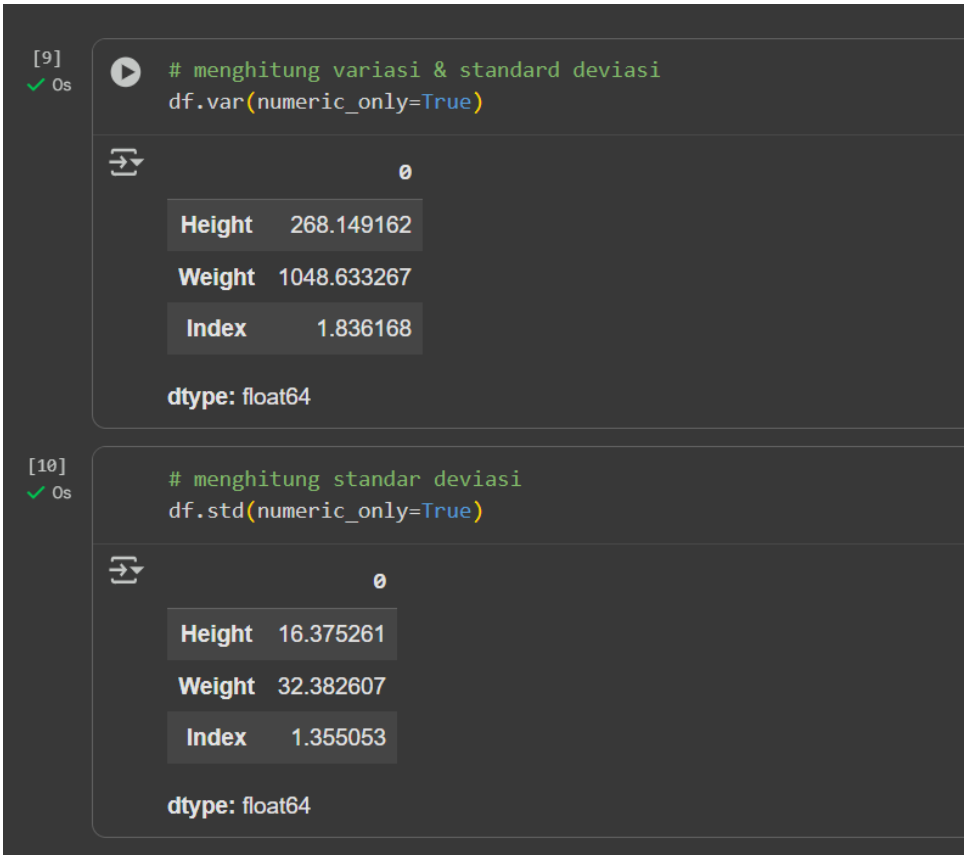
```
[8]
0s
# mencari modus
df['Height'].mode()
```

	Height
0	188

dtype: int64

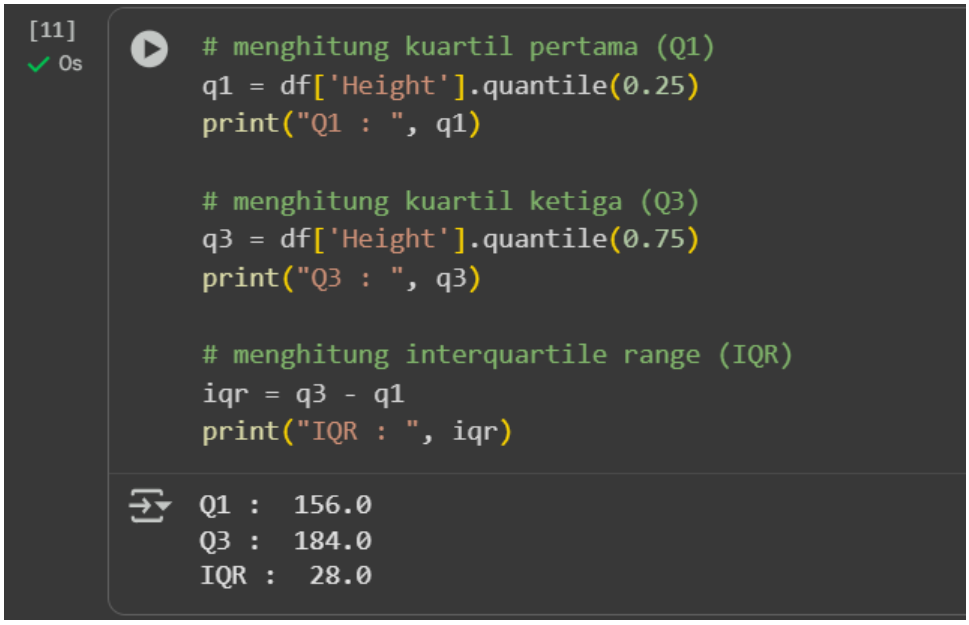
Pada Langkah 4 menghitung nilai-nilai sentral yaitu **Mean, Median, Modus.** Hasil yang didapat adalah Mean (169.944), Median (170.5), Modus (188).

5. Langkah 5



Langkah ini untuk menghitung metrik dan meilihat persebaran data. **.var** untuk mengukur titik data dan rata-rata. **.std** (akar kuadrat dari variasi), ukuran persebaran sama dengan data asli

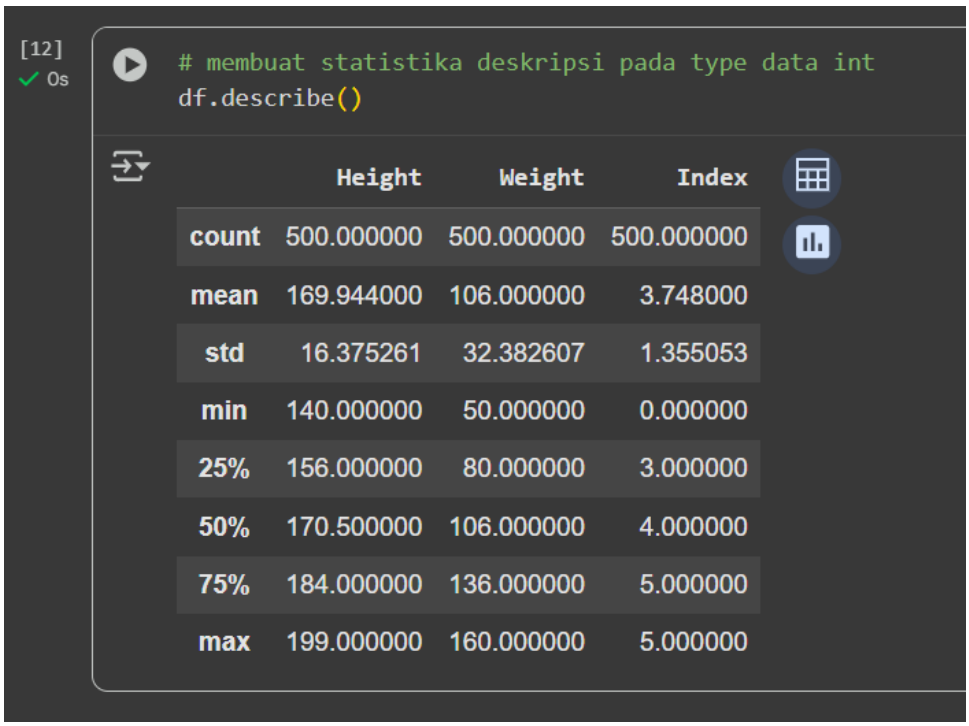
6. **Langkah 6**



```
[11]
✓ 0s

# menghitung kuartil pertama (Q1)
q1 = df['Height'].quantile(0.25)
print("Q1 : ", q1)

# menghitung kuartil ketiga (Q3)
q3 = df['Height'].quantile(0.75)
print("Q3 : ", q3)

# menghitung interquartile range (IQR)
iqr = q3 - q1
print("IQR : ", iqr)

Q1 :  156.0
Q3 :  184.0
IQR :  28.0
```

Langkah ini untuk menghitung kuartil 1 (Q1), dan kuartil 3 (Q3), hasil yang didapat adalah 156.0 untul Q1, dan 184.0 untuk Q2. Hasil IQR dihitung dari Q3 dikurang Q1.

7. **Langkah 7**

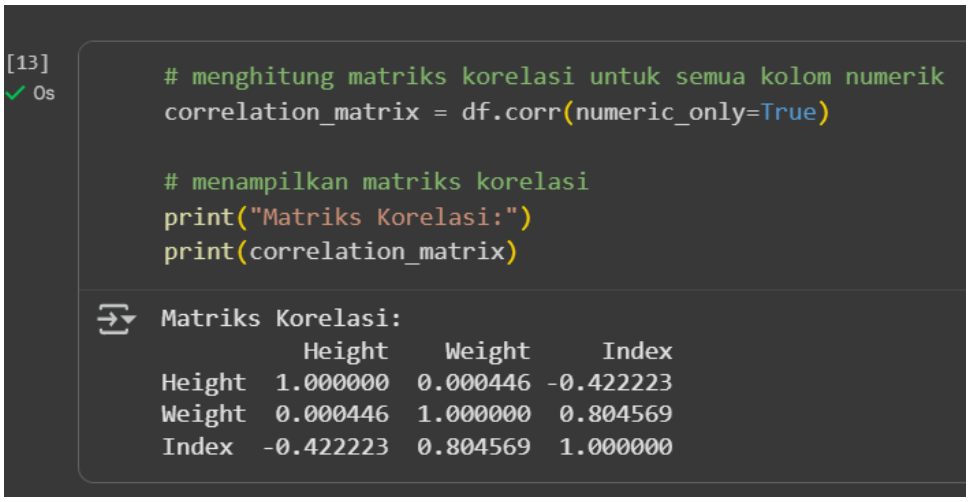


```
[12]
✓ 0s

# membuat statistika deskripsi pada type data int
df.describe()
```

	Height	Weight	Index
count	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000
std	16.375261	32.382607	1.355053
min	140.000000	50.000000	0.000000
25%	156.000000	80.000000	3.000000
50%	170.500000	106.000000	4.000000
75%	184.000000	136.000000	5.000000
max	199.000000	160.000000	5.000000

Df.describe() perintah ini digunakan untuk menampilkan hasil dari statistic deskriptif dari kolom numerik pada dataframe. Hasil yang didapat ada count, mean, std, min, 25%, 50%, 75%, dan max.

8. Langkah 8

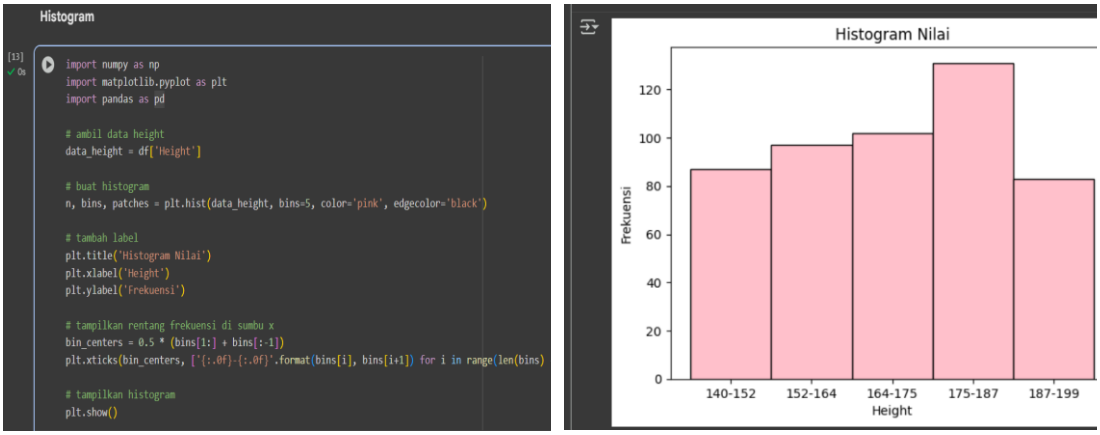


Pada perintah ini menunjukan variable height, weight, dan index saling berhubungan secara statistik dengan menggunakan korelasi.

9. Langkah 9

Pada Langkah ini membuat boxplot (diagram kotak) dari kolom height dan weight untuk melihat distribusi datanya. **Impor pandas as pd** untuk memanggil library pandas dan mengelola serta menganalisis dataframe, **import numpy as np** memanggil library numpy untuk operasi numerik.

10. Langkah 10

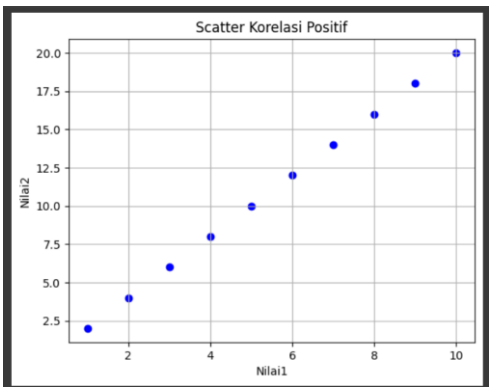


Gambar diatas hasil dari height dan frekuensinya, height dibagi kedalam interval 140-152 dsb. Sedangkan frekuensi adalah jumlah data yang masuk kedalam interval

11. Langkah 11

Scatter Plot

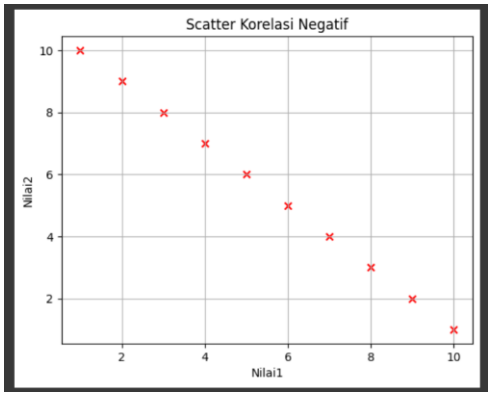
```python
import pandas as pd
import matplotlib.pyplot as plt

# buat dataframe
data = {
    'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,],
    'Nilai2': [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
}

df2 = pd.DataFrame(data)

# buat scatter plot
plt.scatter(df2['Nilai1'], df2['Nilai2'], color='blue', marker='o')

# tambah label
plt.title('Scatter Korelasi Positif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# tambah grid
plt.grid(True)

# tampilkan scatter plot
plt.show()
```



Gambar diatas menunjukan scatter plot pada nilai1(sumbu x) dan nilai2(sumbu y), pada gambar terdapat titik biru ke kanan atas artinya semakin besar nilai1 maka semakin besar juga nilai2 dan bisa dibilang korelasi sempurna. **pandas (pd)** untuk mengelola data dalam bentuk DataFrame, **matplotlib.pyplot (plt)** untuk membuat grafik/visualisasi.

12. Langkah 12

```python
import pandas as pd
import matplotlib.pyplot as plt

# buat dataframe
data = {
    'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Nilai2': [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
}
df3 = pd.DataFrame(data)

# buat scatter plot
plt.scatter(df3['Nilai1'], df3['Nilai2'], color='red', marker='x')

# tambah label
plt.title('Scatter Korelasi Negatif')
plt.xlabel('Nilai1')
plt.ylabel('Nilai2')

# tambah grid
plt.grid(True)

# tampilkan scatter plot
plt.show()
```



Gambar diatas adalah scatter plot yang menunjukan korelasi negatif antara nilai1(sumbu x) dan nilai2(sumbu y). Pada gambar terdapat titik merah yang menurun ke kanan bawah, artinya semakin besar nilai1 maka semakin kecil nilai2 ini bisa dibilang hubungan yang tidak sempurna.

Tugas Praktimum 2

1. **Langkah 1**

```
# menghubungkan gdrive dengan colab
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

Pada Langkah pertama pengerjaan adalah menghubungkan gdrive dan gcolab**. from google.colab import drive drive.mount('/content/gdrive')** adalah perintah untuk menghubungkan.

2. **Langkah 2**

```
# memanggil dataset lewat gdrive
path = "/content/gdrive/MyDrive/praktikum_ml/praktikum02"
```

Pada Langkah ini digunakan untuk menentukan Lokasi dataset pada gdrive agar lebih mudah untuk dipanggil.

3. **Langkah 3**

```
# membaca file csv menggunakan pandas
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.read_csv("/content/gdrive/MyDrive/praktikum_ml/praktikum02/data/day.csv")
df
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

731 rows × 16 columns

Langkah ketiga yaitu membaca data (file csv) menggunakan pandas, **df = pd.read_csv("/content/gdrive/MyDrive/praktikum_ml/praktikum02/data/day.csv")** untuk membaca file csv yang diberikan dan menyimpan file pada dataframe (df). **Df** untuk menampilkan isi dari dataframe.

4. **Langkah 4**

```
# Bagi data menjadi Training (80%) dan Testing (20%)
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42)

# Dari data Training, ambil 10% untuk Validation
train_data_final, val_data = train_test_split(train_data, test_size=0.1, random_state=42)
```

Langkah ini digunakan untuk membagi dataset day.csv menjadi 3(tiga) bagian. Data training 80%, data testing 20%, dan data validation 10%. Training digunakan untuk melatih model, validation untuk menguji model selama proses training, dan testing untuk mengevaluasi performa akhir model dengan data yang baru.

5. **Langkah 5**

```
# Tampilkan jumlah data dan 5 baris pertama untuk setiap set
# tampilan jumlah data Training
print("Jumlah Data Training:", len(train_data_final))
print(train_data_final.head(), "\n")
```

```
Jumlah Data Training: 525
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
657      658  2012-10-19       4   1    10        0        5           1
163      164  2011-06-13       2   0     6        0        1           1
305      306  2011-11-02       4   0    11        0        3           1
111      112  2011-04-22       2   0     4        0        5           1
538      539  2012-06-22       3   1     6        0        5           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
657           2  0.563333  0.537896  0.815000   0.134954     753        4671
163           1  0.635000  0.601654  0.494583   0.305350     863        4157
305           1  0.377500  0.390133  0.718750   0.082092     370        3816
111           2  0.336667  0.321954  0.729583   0.219521     177        1506
538           1  0.777500  0.724121  0.573750   0.182842     964        4859

      cnt
657  5424
163  5020
305  4186
111  1683
538  5823
```

Pada Langkah ini menampilkan hasil jumlah dataset training setelah dibagikan. **print(train_data_final.head()** untuk menampilkan 5(lima) baris pertama dari dataset training yang baru, **"\n"** untuk menambahkan baris yang kosong agar hasil lebih rapih.

6. **Langkah 6**

```
# tampilan jumlah data Validation
print("Jumlah Data Validation:", len(val_data))
print(val_data.head(), "\n")
```

```
Jumlah Data Validation: 59
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
325      326  2011-11-22       4   0    11        0        2           1
410      411  2012-02-15       1   1     2        0        3           1
92        93  2011-04-03       2   0     4        0        0           0
47        48  2011-02-17       1   0     2        0        4           1
508      509  2012-05-23       2   1     5        0        3           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
325           3  0.416667  0.421696  0.962500   0.118792      69        1538
410           1  0.348333  0.351629  0.531250   0.181600     141        4028
92            1  0.378333  0.378767  0.480000   0.182213    1651        1598
47            1  0.435833  0.428658  0.505000   0.230104     259        2216
508           2  0.621667  0.584612  0.774583   0.102000     766        4494

      cnt
325  1607
410  4169
92   3249
47   2475
508  5260
```

Pada Langkah ini menampilkan hasil jumlah dataset validation setelah dibagikan. **print(train_data_final.head()** untuk menampilkan 5(lima) baris pertama dari dataset training yang baru, **"\n"** untuk menambahkan baris yang kosong agar hasil lebih rapih.

7. **Langkah 7**

```
# tampilan jumlah data Testing
print("Jumlah Data Testing:", len(test_data))
print(test_data.head(),"\n")
```

```
Jumlah Data Testing: 147
     instant      dteday  season  yr  mnth  holiday  weekday  workingday  \
703      704  2012-12-04       4   1    12        0        2           1
33        34  2011-02-03       1   0     2        0        4           1
300      301  2011-10-28       4   0    10        0        5           1
456      457  2012-04-01       2   1     4        0        0           0
633      634  2012-09-25       4   1     9        0        2           1

     weathersit      temp     atemp       hum  windspeed  casual  registered  \
703           1  0.475833  0.469054  0.733750   0.174129     551        6055
33            1  0.186957  0.177878  0.437826   0.277752      61        1489
300           2  0.330833  0.318812  0.585833   0.229479     456        3291
456           2  0.425833  0.417287  0.676250   0.172267    2347        3694
633           1  0.550000  0.544179  0.570000   0.236321     845        6693

      cnt
703  6606
33   1550
300  3747
456  6041
633  7538
```

Langkah ini digunakan untuk meliat jumlah data testing dan melihat Sebagian isi dataset nya. **print("Jumlah Data Testing:", len(test_data))** untuk menghitung dan menampilkan jumlah baris data testing, **print(test_data.head(), "\n")** untuk mrnampilkan 5(lima)baris pertama dari data testing

Link Github:

https://github.com/Jamilatun/ti03_Mila_01101222254/tree/main/praktikum02