Tugas 2: Machine Learning – Statistika Deskriptif dan Probabilitas

Muhammad Zaidan Ramdhan - 0110222040

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: muha22040ti@student.nurulfikri.ac.id

Laporan ini bertujuan untuk melakukan analisis eksplorasi data dari sebuah *dataset* menggunakan *library* Pandas dan Matplotlib di Python, dengan fokus pada pemahaman distribusi statistik dan hubungan antar variabel. Analisis dimulai dengan perhitungan Matriks Korelasi, yang secara spesifik menyertakan hanya kolom numerik (numeric_only=True), menunjukkan adanya korelasi positif yang kuat antara *Weight* dan *Index* (≈0.80), dan korelasi negatif yang moderat antara *Height* dan *Index* (≈−0.42). Selanjutnya, distribusi data divisualisasikan melalui Box Plot untuk *Height* dan *Weight*, menyoroti bahwa *Height* memiliki sebaran data yang lebih homogen (variansi kecil) dibandingkan dengan *Weight*. Analisis distribusi *Height* diperdalam dengan Histogram yang menunjukkan pola sebaran menyerupai lonceng, dengan frekuensi tertinggi berada pada rentang 175-187. Terakhir, hubungan linier antara variabel divisualisasikan menggunakan Scatter Plot, di mana satu plot secara eksplisit mendemonstrasikan korelasi positif sempurna dan plot lainnya menunjukkan korelasi negatif sempurna. Secara keseluruhan, analisis ini berhasil mengidentifikasi variabilitas data, mengkonfirmasi distribusi nilai *Height*, dan memvisualisasikan sifat serta kekuatan hubungan korelasi antar variabel.

1. Praktikum Mandiri

```
#membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + '/praktikum_02/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

Gambar 1. 1

Pada *Gambar 1.1* di atas, merupakan sebuah code untuk menampilkan dataset pada sebuah tabel menggunakan dataset 500_Person_Gender_Height_Weight_Index.csv.

- **df = pd.read_csv('../praktikum_02/data/500_Person_Gender_Height_Weight_Index.csv)** merupakan sebuah code untuk membaca file csv yang terdapat pada folder data dengan diikuti file 500_Person_Gender_Height_Weight_Index.csv.

```
#mencari info data pada file
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
```

Gambar 1. 2

Pada *Gambar 1.2* di atas, merupakan sebuah code untuk menampilkan tipe data pada dataset dengan output seperti gambar di atas.

```
#menghitung mean pada nilai numerik
df['Height'].mean()
```

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

Gambar 1. 3

Pada *Gambar 1.3* di atas, merupakan sebuah code untuk menampilkan mean pada kolom Height dengan output seperti gambar di atas.

- df[‘height’] = mengambil kolom height pada dataFrame.
- mean() = merupakan sebuah function untuk menentukan niai mean pada numerik

```
#menghitung median pada nilai numerik
df['Height'].median()
```

```
170.5
```

Gambar 1. 4

Pada *Gambar 1.4* di atas, merupakan sebuah code untuk menampilkan median pada kolom Height dengan output seperti gambar di atas.

- df[‘height’] = mengambil kolom height pada dataFrame.
- median() = merupakan sebuah function untuk menentukan niai median pada numerik

```
#menghitung variansi & standard devisiasi
df.var(numeric_only=True)
```

```
                0
Height  268.149162
Weight 1048.633267
Index     1.836168
dtype: float64
```

Gambar 1. 5

Pada *Gambar 1.5* di atas, merupakan sebuah code untuk menghitung variansi dengan output seperti gambar di atas.

- var() = method var berguna untuk menghitung variansi.
- (numeric_only = True) = parameter memastikan perhitungan hanya dilakukan pada kolom yang berisi angka (numerik).

```
#menghitung standar deviasi
df.std(numeric_only=True)
```

	0
Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

Gambar 1. 6

Pada *Gambar 1.6* di atas, merupakan sebuah code untuk menghitung standard devisiasi dengan output seperti gambar di atas.

- std() = method std berguna untuk menghitung standar deviasi.
- (numeric_only = True) = parameter memastikan perhitungan hanya dilakukan pada kolom yang berisi angka (numerik).

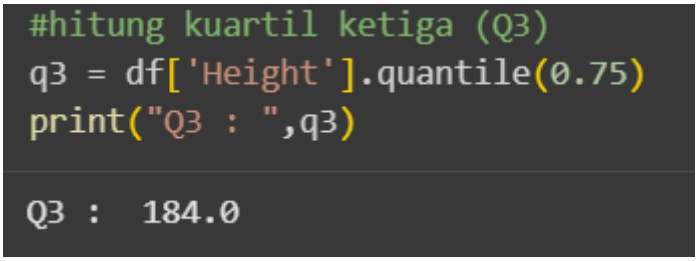
```
#hitung kuartil pertama (Q1)
q1 = df['Height'].quantile(0.25)
print("Q1 : ", q1)
```

```
Q1 :  156.0
```

Gambar 1. 7

Pada *Gambar 1.7* di atas, merupakan sebuah code untuk menghitung kuartil pertama dengan output seperti gambar di atas.

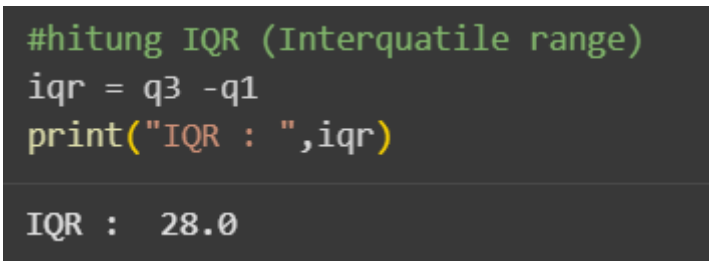
- df[‘height’] = mengambil kolom height pada dataFrame.
- Quantile(0.25) = sebuah function untuk menghitung kuartil pertama.



Gambar 1. 8

Pada *Gambar 1.8* di atas, merupakan sebuah code untuk menghitung kuartil ketiga dengan output seperti gambar di atas.

- df[‘height’] = mengambil kolom height pada dataFrame.
- Quantile(0.75) = sebuah function untuk menghitung quartil ketiga.



Gambar 1. 9

Pada *Gambar 1.9* di atas, merupakan sebuah code untuk menghitung interquatile range dengan output seperti gambar di atas.

- iqr = q3 – q1 =sebuah pengurangan untuk menentukan interquatile range.

```
#menghitung korelasi untuk semua kolom numerik
correlation_matrix = df.corr(numeric_only=True)

print("Matriks Korelasi:")
print(correlation_matrix)
```

```
Matriks Korelasi:
          Height    Weight     Index
Height  1.000000  0.000446 -0.422223
Weight  0.000446  1.000000  0.804569
Index  -0.422223  0.804569  1.000000
```

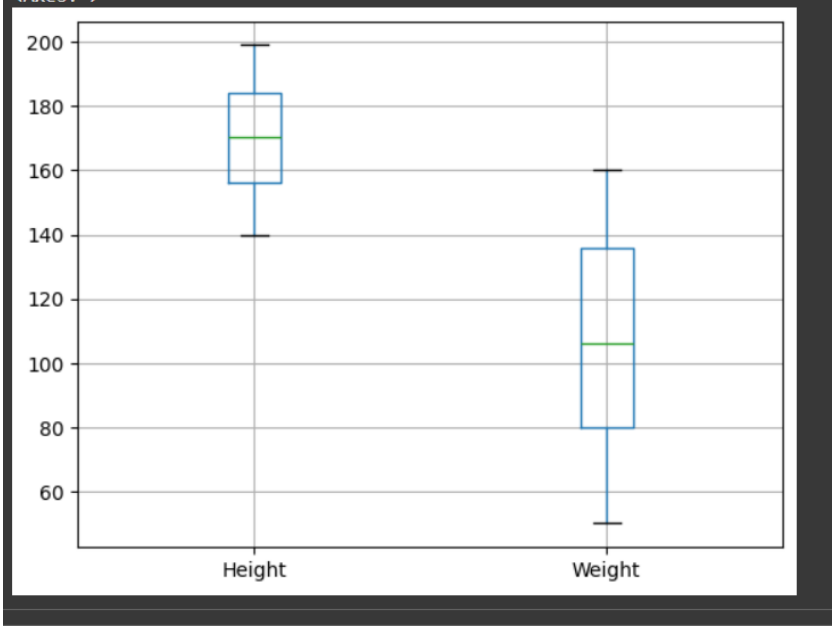
Gambar 1. 10

Pada *Gambar 1.10* di atas, merupakan sebuah code untuk menghitung korelasi semua kolom numerik dengan output seperti gambar di atas.

- df.corr() = Fungsi utama yang melakukan perhitungan korelasi (menggunakan metode Pearson secara default).
- (numeric_only=True) = parameter memastikan perhitungan hanya dilakukan pada kolom yang berisi angka (numerik).

```
import pandas as pd
import numpy as np

df.boxplot(column=['Height','Weight'])
```

```
<Axes: >
```

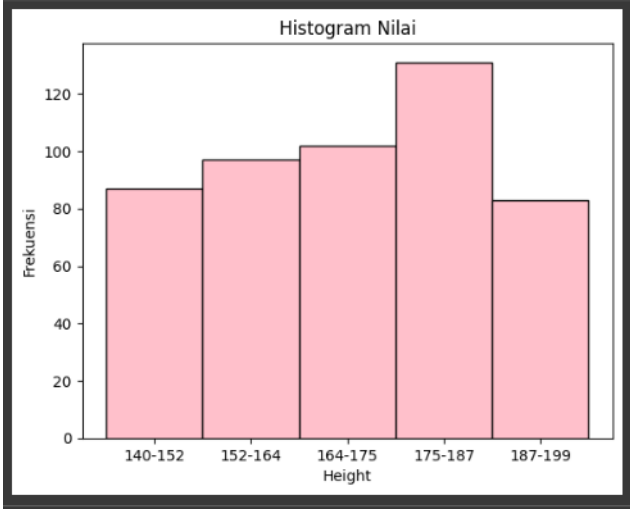


Gambar 1. 11

Pada *Gambar 1.11* di atas, merupakan sebuah code untuk membuat diagram kotak pada Height dan Weight dengan output seperti gambar di atas.

- boxplot()= fungsi yang membuat diagram kotak.
- column=['Height', 'Weight']= argumen ini menentukan bahwa diagram kotak hanya akan dibuat untuk kolom 'Height' dan 'Weight'.

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

# Ambil data Height
data_height = df["Height"]

# Buat histogram
n, bins, patches = plt.hist(data_height, bins=5, color='pink', edgecolor='black')

# Tambahkan label
plt.title('Histogram Nilai')
plt.xlabel('Height')
plt.ylabel('Frekuensi')

# Tampilkan rentang frekuensi di sumbu x
bin_centers = 0.5 * (bins[:-1] + bins[1:])
plt.xticks(bin_centers, ['{:.0f}-{:.0f}'.format(bins[i], bins[i+1]) for i in range(len(bins)-1)])

# Tampilkan histogram
plt.show()
```



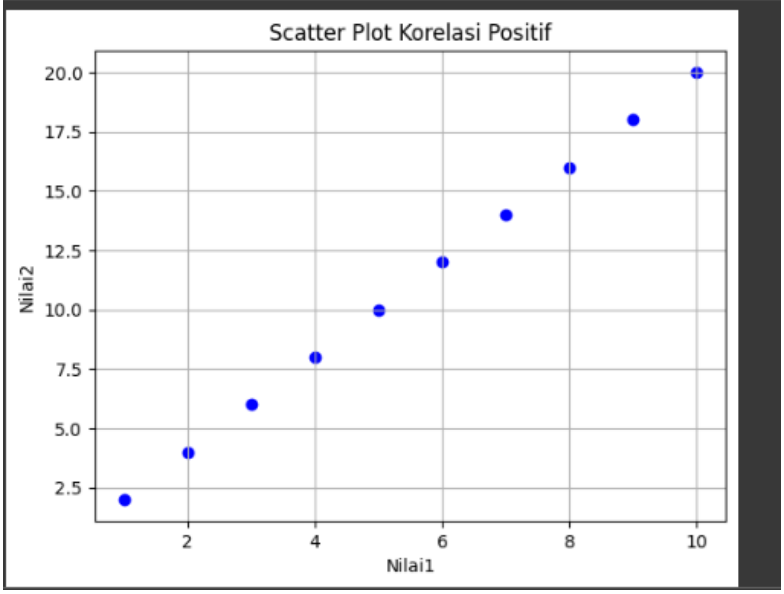
Gambar 1. 12

Pada *Gambar 1.12* di atas, merupakan sebuah code untuk menghitung nilai frekuensi dengan output seperti gambar di atas.

- numpy (as np): digunakan untuk operasi numerik.
- matplotlib.pyplot (as plt): ini adalah library visualisasi data yang digunakan untuk menggambar histogram.
- pandas (as pd): digunakan untuk mengelola data dalam format DataFrame (df).
- data_height = df["Height"] = mengambil data dari kolom 'Height' dalam DataFrame (df) dan menyimpannya dalam variabel baru bernama data_height.
- plt.hist()= fungsi untuk membuat histogram.
- data_height= data yang akan diplot.
- bins=5= menentukan bahwa data akan dibagi menjadi 5 interval (kotak/bar) yang sama lebarnya.
- color='pink', edgecolor='black'= mengatur warna isian bar dan garis tepi.

- bin_centers dan plt.xticks() = kode ini menghitung dan menampilkan rentang nilai untuk setiap bar agar pembacaan diagram lebih mudah.
- plt.show() = menampilkan hasil yang sudah selesai ke layar.

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

# Buat DataFrame contoh
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')

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

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()
```



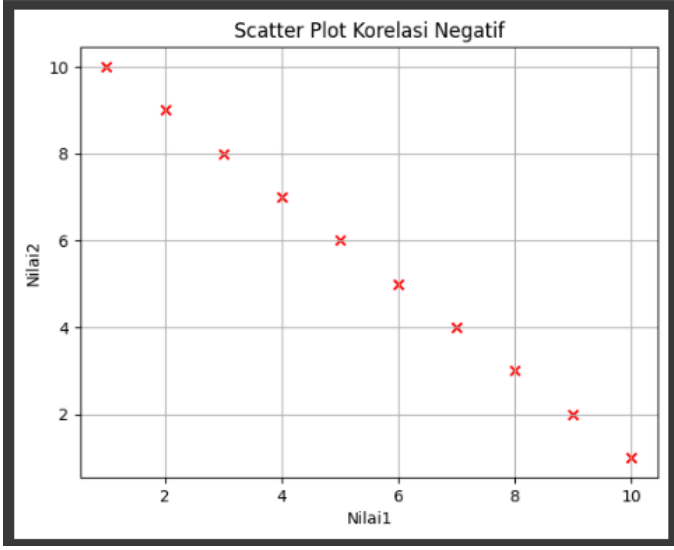
Gambar 1. 13

Pada *Gambar 1.13* di atas, merupakan sebuah code untuk menghitung korelasi secara positif dengan output seperti gambar di atas.

- numpy (as np): digunakan untuk operasi numerik.
- matplotlib.pyplot (sebagai plt): ini adalah library visualisasi data yang digunakan untuk menggambar histogram.
- pandas (as pd): digunakan untuk mengelola data dalam format DataFrame (df).

- matplotlib.pyplot (sebagai plt): library visualisasi yang digunakan untuk menggambar plot.
- variabel data = object yang berisi Nilai1 sebagai sumbu X dan Nilai 2 sebagai sumbu Y
- df2 = kedua daftar ini digabungkan menjadi sebuah tabel data (*DataFrame*) bernama df2.
- plt.scatter(): fungsi untuk membuat Scatter Plot (Diagram Tebar).
- df2['Nilai1']: data untuk sumbu horizontal (X).
- df2['Nilai2']: data untuk sumbu vertikal (Y).
- color='blue', marker='o': mengatur penampilan titik data (warna biru dan bentuk lingkaran).
- plt.grid(True) = Menambahkan garis kisi (grid) di latar belakang plot untuk membantu pembacaan nilai koordinat.
- plt.show() = menampilkan hasil yang sudah selesai ke layar.

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

# Buat DataFrame contoh
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')

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

# Tambahkan grid
plt.grid(True)

# Tampilkan plot
plt.show()
```



Gambar 1. 14

Pada *Gambar 1.14* di atas, merupakan sebuah code untuk menghitung korelasi secara negatif dengan output seperti gambar di atas.

- numpy (as np): digunakan untuk operasi numerik.
- matplotlib.pyplot (sebagai plt): ini adalah library visualisasi data yang digunakan untuk menggambar histogram.

2. Tugas mandiri – membagi data

```
from google.colab import drive
drive.mount("/content/gdrive")
```

```
Drive already mounted at /content/gdrive;
```

Gambar 2. 1

Pada *Gambar 2.1* di atas, merupakan sebuah code untuk mounted atau menghubungkan google colab dengan google drive.

```
#memanggil dataset lewat gdrive
path = "/content/gdrive/MyDrive/machine_learning/pertemuan02"
```

Gambar 2. 2

Pada *Gambar 2.2* di atas, kita membuat sebuah path untuk mengatur folder yang akan digunakan, dalam case ini ita menggunakan folder pertemuan02.

```
#membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + '/tugas_mandiri/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

Gambar 2. 3

Pada *Gambar 2.1* di atas, merupakan sebuah code untuk menampilkan dataset pada sebuah tabel menggunakan dataset day.csv.

- **df = pd.read_csv(path + '/tugas_mandiri/data/day.csv)** merupakan sebuah code untuk membaca file csv yang terdapat pada folder data dengan diikuti file day.csv.

```python
# --- 1. Muat Dataset ---
path_file = path + '/tugas_mandiri/data/day.csv'
df = pd.read_csv(path_file)

# --- 2. Pembagian Dataset (Chained Split) ---

# Langkah 1: Pisahkan Data Testing (20%) dari Data Training Sisa (80%)
train_data, test_data = train_test_split(
    df,
    test_size=0.2,            # 20% untuk Testing
    random_state=42,          # Untuk hasil yang konsisten
    shuffle=True
)

# Langkah 2: Pisahkan Data Validation (10% dari train_data)
# train_data sisa 80% dari total, jadi 10% dari 80% = 8% dari total
train_data, val_data = train_test_split(
    train_data,
    test_size=0.1,            # 10% dari data yang tersisa (train_data)
    random_state=42,
    shuffle=True
)

# --- 3. Verifikasi dan Tampilkan Hasil ---
print("\n" + "="*50)
print("HASIL PEMBAGIAN DATASET (Skema Chained Split)")
print("="*50)
```

Gambar 2. 4

Pada *Gambar 2.4* di atas, merupakan sebuah code untuk membagi dataset menjadi data testing, validation dan training dataset day.csv.

```python
# Ambil total data
total_data = len(df)
len_train = len(train_data)
len_val = len(val_data)
len_test = len(test_data)
```

Gambar 2. 5

Pada *Gambar 2.5* di atas, merupakan sebuah code untuk mengambil total data.

```python
# a) Data Training
print(f"\n(a) Data Training (Rasio: {len_train / total_data * 100:.2f}%)")
print(f"Jumlah Data: {len_train} baris")
print("5 Baris Teratas:")
print(train_data.head())
print("-" * 50)
```

Gambar 2. 6

Pada *Gambar 2.5* di atas, merupakan sebuah code untuk mengambil data Training.

```
(a) Data Training (Rasio: 71.82%)
Jumlah Data: 525 baris
5 Baris Teratas:
     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
--------------------------------------------------
```

Gambar 2. 7

Pada *Gambar 2.7* di atas, merupakan hasil atau output dari code *gambar 2.6* yang menampilkan data Training.

```
# b) Data Validation
print(f"\n(b) Data Validation (Rasio: {len_val / total_data * 100:.2f}%)")
print(f"Jumlah Data: {len_val} baris")
print("5 Baris Teratas:")
print(val_data.head())
print("-" * 50)
```

Gambar 2. 8

Pada *Gambar 2.8* di atas, merupakan sebuah code untuk mengambil data Validation.

```
(b) Data Validation (Rasio: 8.07%)
Jumlah Data: 59 baris
5 Baris Teratas:
     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
--------------------------------------------------
```

Gambar 2. 9

Pada *Gambar 2.9* di atas, merupakan hasil atau output dari code *gambar 2.8* yang menampilkan data Validation.

```python
# c) Data Testing
print(f"\n(c) Data Testing (Rasio: {len_test / total_data * 100:.2f}%)")
print(f"Jumlah Data: {len_test} baris")
print("5 Baris Teratas:")
print(test_data.head())
print("=" * 50)
```

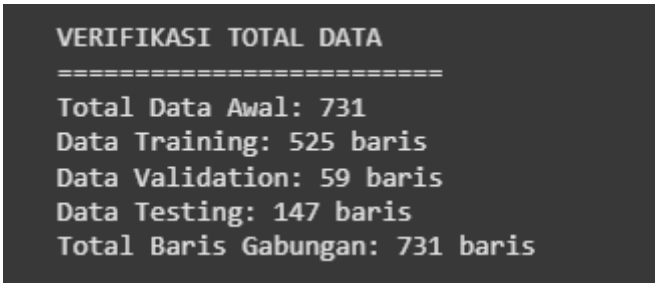
Gambar 2. 10

Pada *Gambar 2.10* di atas, merupakan sebuah code untuk mengambil data Testing.

```
(c) Data Testing (Rasio: 20.11%)
Jumlah Data: 147 baris
5 Baris Teratas:
     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
==================================================
```

Gambar 2. 11

Pada *Gambar 2.11* di atas, merupakan hasil atau output dari code *gambar 2.10* yang menampilkan data Testing.



Gambar 2. 12

Pada *Gambar 2.12* di atas, merupakan informasi tentang total data awal sampe hasil pembagian total data menjadi data (Training, Validation, Testing) dengan total baris gabungan 731 baris.