

# Praktikum 2: Machine Learning – **Statistika Deskriptif dan Probabilitas**

**Muhammad Shiddiq 1 - 0110222199 <sup>1</sup>**

<sup>1</sup> Teknik Informatika, STT Terpadu Nurul Fikri, Depok

E-mail: muhammadshiddiq785@gmail.com

**Abstract.** Statistika deskriptif dan probabilitas merupakan dua konsep dasar dalam analisis data yang saling melengkapi. Statistika deskriptif berfungsi untuk menyajikan dan meringkas data melalui ukuran pemusatan, ukuran penyebaran, serta visualisasi seperti tabel dan grafik, sehingga pola dan karakteristik data dapat dipahami secara lebih sederhana. Sementara itu, probabilitas digunakan untuk mengukur peluang atau kemungkinan terjadinya suatu peristiwa, yang menjadi dasar bagi pengambilan keputusan dalam kondisi ketidakpastian. Kombinasi keduanya memungkinkan peneliti tidak hanya memahami data yang ada, tetapi juga melakukan prediksi terhadap peristiwa di masa mendatang.

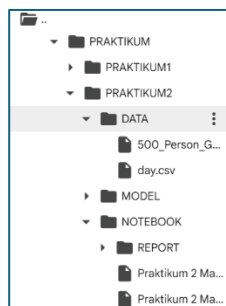
## **1. Praktikum 2 – Analisis Statistika Deskriptif dan Visualisasi Data**

Analisis statistika deskriptif dan visualisasi data merupakan tahap awal dalam memahami suatu kumpulan data. Statistika deskriptif berfungsi untuk menggambarkan atau merangkum karakteristik utama data tanpa melakukan penarikan kesimpulan yang bersifat umum. Analisis ini mencakup ukuran pemusatan seperti mean, median, dan modus, serta ukuran penyebaran seperti range, varians, dan standar deviasi. Selain itu, juga dapat digunakan ukuran bentuk distribusi seperti skewness dan kurtosis untuk mengetahui pola distribusi data.

Sementara itu, visualisasi data digunakan untuk menyajikan hasil analisis dalam bentuk grafik atau diagram, seperti histogram, diagram batang, pie chart, dan scatter plot. Melalui visualisasi, pola, tren, dan hubungan antarvariabel dapat terlihat dengan lebih jelas. Kombinasi antara analisis statistika deskriptif dan visualisasi data membantu peneliti memahami data secara mendalam serta menyampaikan informasi dengan cara yang lebih informatif dan menarik.

### **1.1 Membuat Folder**

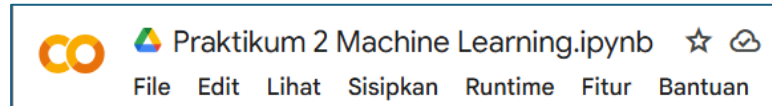
Langkah pertama kita harus membuat folder yang terstruktur dan juga rapih di google drive.



**Gambar 1.** Membuat folder di google drive, agar mudah untuk diakses

## 1.2 Membuat file notebook google colab

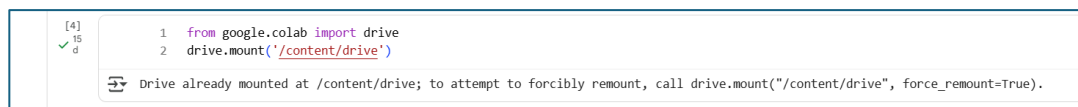
Selanjutnya membuat file notebook di google colab untuk praktikum.



Gambar 2. Membuat file google colab

## 1.3 Menghubungkan google colab dengan google drive

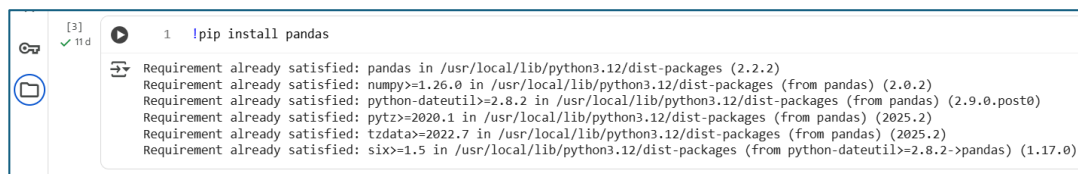
Selanjutnya menghubungkan google colab dengan google drive menggunakan perintah "From google.colab import drive  
Drive.mount('/content/drive')".



Gambar 3. Menghubungkan google colab dengan google drive

## 1.4 Meng install pandas

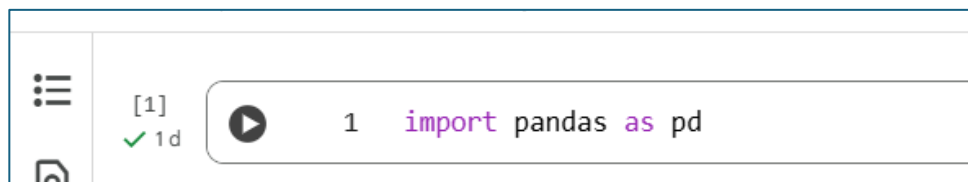
Selanjutnya meng install library pandas dengan perintah "!pip install pandas".



Gambar 4. Meng install pandas.

## 1.5 Meng import library pandas

Selanjutnya meng import library pandas dengan perintah "import pandas as pd". Pandas adalah perpustakaan Python sumber terbuka yang banyak digunakan untuk analisis dan manipulasi data. Perpustakaan ini menyediakan struktur data yang kuat dan fleksibel, terutama Series dan DataFrame, yang dirancang untuk menangani data terstruktur dengan efisien.



Gambar 5. Mengimport library pandas

## 1.6 Membaca dataset

Selanjutnya membaca dataset day.csv yang ada di google drive menggunakan perintah "df =

```
pd.read_csv('/content/drive/MyDrive/MACHINELEARNING/PRAKTIKUM/PRAKTIKUM
2/DATA/500_Person_Gender_Height_Weight_Index.csv')
df"
```

```
1 df = pd.read_csv(path + "500_Person_Gender_Height_Weight_Index.csv")
2 df
```

**Gambar 6.** Membaca dataset 500\_Person\_Gender\_Height\_Weight\_Index.csv.

**Tabel 1.** Berikut adalah hasil dataset yang telah dibaca.

	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	81	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 x 4 columns

### 1.7 Mengecek informasi dataset

Selanjutnya mengecek informasi dataset yang dibaca, dari total, jumlah kolom, missing value, dan type data menggunakan perintah “df.info()”

```
1 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 7.** Mengecek informasi dataset.

### 1.8 Mencari tau mean, median, dan mode

Selanjutnya menjadi nilai mean, median, dan modus dari kolom height menggunakan perintah df['Height'].mean(), untuk perintah selanjutnya hanya mengubah bagian mean menjadi median dan mode.

```
1 df['Height'].mean()
np.float64(169.944)

1 df['Height'].median()
170.5

1 df['Height'].mode()
Height
0      188
dtype: int64
```

**Gambar 8.** Mencari tau nilai mean, median dan modus.

### 1.9 Mencari tahu besaran nilai variasi dan standar deviasi

Selanjutnya mencari besaran nilai variasi dan standar deviasi menggunakan perintah `df.var(numeric_only=True)`, untuk menjadi nilai standar deviasi kita tinggal mengubah `var` menjadi `std`.

```
1 df.var(numeric_only=True)
```

Height	268.149162
Weight	1048.633267
Index	1.836168

dtype: float64

```
1 df.std(numeric_only=True)
```

Height	16.375261
Weight	32.382607
Index	1.355053

dtype: float64

Gambar 9. Mencari nilai variasi dan standar deviasi.

### 1.10 Mencari nilai kuartil 1 dan 3

Selanjutnya mencari nilai kuartil 1 dan 3.

```
1 q1 = df['Height'].quantile(0.25)
2 print("Q1 : ", q1)
3
4 q3 = df['Height'].quantile(0.75)
5 print("Q3 : ", q3)
6
7 iqr = q3 - q1
8 print("IQR : ", iqr)
```

Q1 : 156.0  
Q3 : 184.0  
IQR : 28.0

Gambar 10. Mencari nilai kuartil.

### 1.11 Mencari nilai statistik deskriptif secara cepat

Selanjutnya mencari nilai statistik dari dataset dengan cepat menggunakan perintah, `df.describe()`.

```
1 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

Gambar 11. Mencari nilai statistik deskriptif.

### 1.12 Mencari nilai korelasi dari setiap kolom

Selanjutnya mencari nilai korelasi dari setiap kolom.

```
1 correlation_matrix = df.corr(numeric_only=True)
2
3 print("Matriks Korelasi:")
4 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 12. Mencari nilai korelasi.

### 1.13 Mengimport library numpy

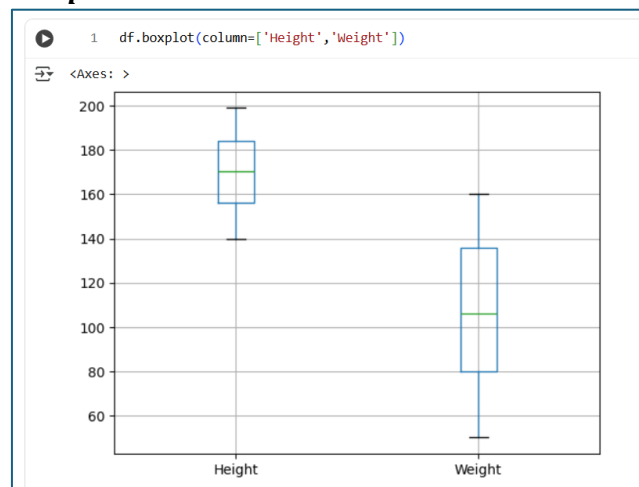
NumPy (Numerical Python) adalah salah satu library penting di Python yang digunakan untuk melakukan komputasi numerik dan pengolahan data berbasis array. Library ini menyediakan struktur data utama bernama ndarray, yaitu array multidimensi yang memungkinkan operasi matematika dilakukan dengan cepat dan efisien dibandingkan dengan list biasa di Python. NumPy juga memiliki banyak fungsi untuk melakukan operasi matematika, seperti aljabar linear, statistik, transformasi Fourier, dan manipulasi bentuk array. Karena kemampuannya dalam menangani data numerik besar dengan efisien, NumPy menjadi fondasi utama bagi banyak library lain di Python, seperti Pandas, SciPy, dan scikit-learn, yang banyak digunakan dalam analisis data dan machine learning.

```
1 import numpy as np
```

Gambar 13. Mengimport library numpy.

### 1.14 Visualisasi Data

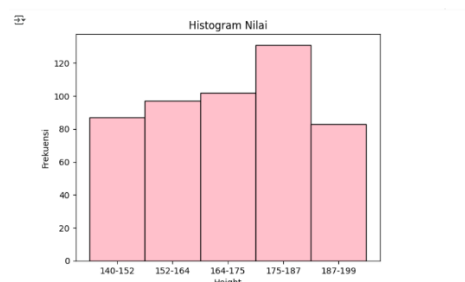
#### 1.14.1 Boxplot



Gambar 14. Visualiasasi boxplot.

#### 1.14.2 Histogram

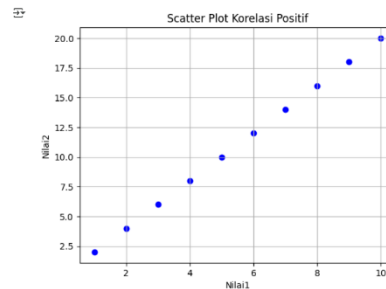
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Ambil data Height
6 data_height = df["Height"]
7
8 # Buat histogram
9 n, bins, patches = plt.hist(data_height, bins=5, color='pink', edgecolor='black')
10
11 # Tambahkan label
12 plt.title("Histogram Nilai")
13 plt.xlabel("Height")
14 plt.ylabel("Frekuensi")
15
16 # Tampilkan rentang frekuensi di sumbu x
17 bin_centers = 0.5 * (bins[1:] + bins[:-1])
18 plt.xticks(bin_centers, ['{:.0f}-{:.0f}'.format(bins[i], bins[i+1]) for i in range(4)])
19
20 # Tampilkan histogram
21 plt.show()
```



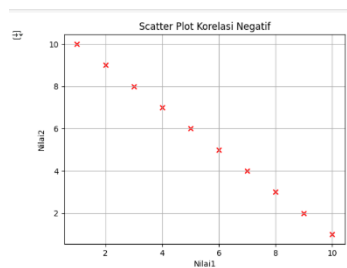
Gambar 15. Visualisasi histogram.

### 1.14.3 Scatter Plot Positif dan Negatif

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Buat DataFrame contoh
5 data = {
6     'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
7     'Nilai2': [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
8 }
9
10 df2 = pd.DataFrame(data)
11
12 # Buat scatter plot
13 plt.scatter(df2['Nilai1'], df2['Nilai2'], color='blue', marker='o')
14
15 # Tambahkan label
16 plt.title('Scatter Plot Korelasi Positif')
17 plt.xlabel('Nilai1')
18 plt.ylabel('Nilai2')
19
20 # Tambahkan grid
21 plt.grid(True)
22
23 # Tampilkan plot
24 plt.show()
```



Gambar 16. Visualisasi scatter plot positif.



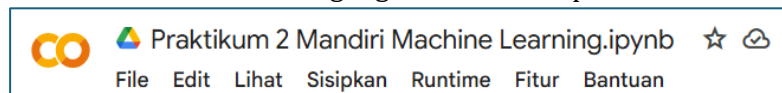
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Buat DataFrame contoh
5 data = {
6     'Nilai1': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
7     'Nilai2': [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
8 }
9
10 df3 = pd.DataFrame(data)
11
12 # Buat scatter plot
13 plt.scatter(df3['Nilai1'], df3['Nilai2'], color='red', marker='x')
14
15 # Tambahkan label
16 plt.title('Scatter Plot Korelasi Negatif')
17 plt.xlabel('Nilai1')
18 plt.ylabel('Nilai2')
19
20 # Tambahkan grid
21 plt.grid(True)
22
23 # Tampilkan plot
24 plt.show()
```

Gambar 17. Visualisasi scatter plot negatif.

## 2. Praktikum 2 Mandiri – Membagi data

### 2.1 Membuat file notebook google colab

Selanjutnya membuat file notebook di google colab untuk praktikum.



Gambar 18. Membuat file google colab

### 2.2 Menghubungkan google colab dengan google drive

Selanjutnya menghubungkan google colab dengan google drive menggunakan perintah "From google.colab import drive  
Drive,mount('/content/drive')".

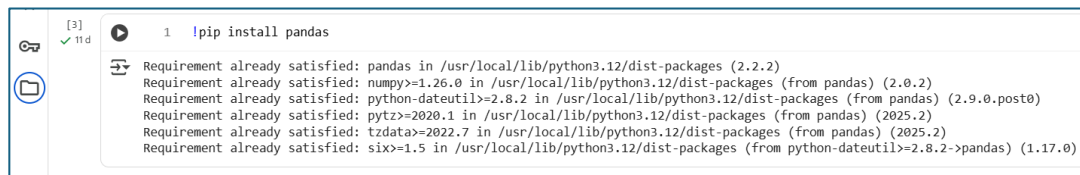
```
[4]
✓ 15
d 1 from google.colab import drive
  2 drive.mount('/content/drive')

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

Gambar 19. Menghubungkan google colab dengan google drive

### 2.3 Meng install pandas

Selanjutnya meng install library pandas dengan perintah "!pip install pandas".

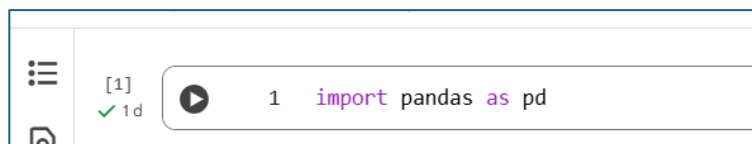


```
[3] 1 pip install pandas
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

Gambar 20. Meng install pandas.

## 2.4 Meng import library pandas

Selanjutnya meng import library pandas dengan perintah “import pandas as pd”. Pandas adalah perpustakaan Python sumber terbuka yang banyak digunakan untuk analisis dan manipulasi data. Perpustakaan ini menyediakan struktur data yang kuat dan fleksibel, terutama Series dan DataFrame, yang dirancang untuk menangani data terstruktur dengan efisien.

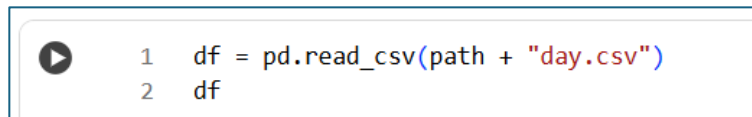


```
[1] 1 import pandas as pd
```

Gambar 21. Mengimport library pandas

## 2.5 Membaca dataset

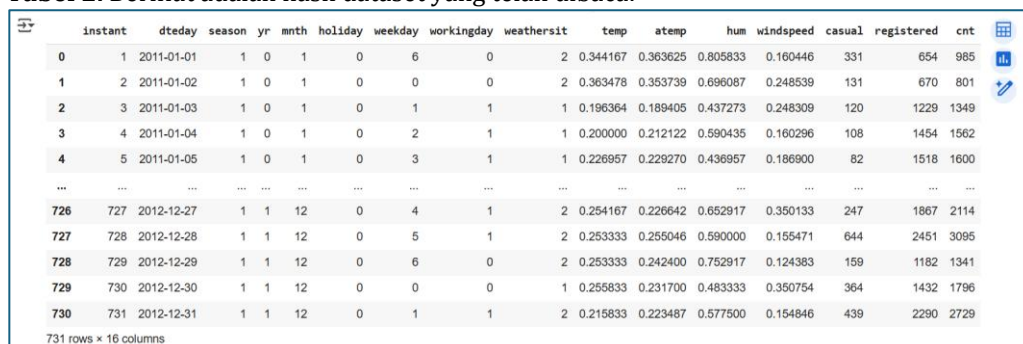
Selanjutnya membaca dataset day.csv yang ada di google drive menggunakan perintah “df = pd.read\_csv('/content/drive/MyDrive/MACHINELEARNING/PRAKTIKUM/PRAKTIKUM 2/DATA/day.csv') df”



```
1 df = pd.read_csv(path + "day.csv")
2 df
```

Gambar 22. Membaca dataset day.csv

Tabel 2. Berikut adalah hasil dataset yang telah dibaca.



	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

## 2.6 Mengecek informasi dataset

Selanjutnya mengecek informasi dataset yang dibaca, dari total, jumlah kolom, missing value, dan type data menggunakan perintah “df.info()”

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  -
0   instant     731 non-null    int64
1   dteday      731 non-null    object
2   season      731 non-null    int64
3   yr          731 non-null    int64
4   mnth        731 non-null    int64
5   holiday     731 non-null    int64
6   weekday     731 non-null    int64
7   workingday  731 non-null    int64
8   weathersit   731 non-null    int64
9   temp        731 non-null    float64
10  atemp       731 non-null    float64
11  hum         731 non-null    float64
12  windspeed   731 non-null    float64
13  casual      731 non-null    int64
14  registered  731 non-null    int64
15  cnt         731 non-null    int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB

```

**Gambar 23.** Mengecek informasi dataset.

## 2.7 Membagi Data menjadi training, validation, dan testing

```

1  from sklearn.model_selection import train_test_split
2
3  train_data, test_data = train_test_split(df, test_size=0.2, random_state=42)
4  train_data, val_data = train_test_split(train_data, test_size=0.1, random_state=42)
5
6  print("Jumlah total data:", len(df))
7  print("Jumlah data training:", len(train_data))
8  print("Jumlah data validation:", len(val_data))
9  print("Jumlah data testing:", len(test_data))

```

Jumlah total data: 731  
 Jumlah data training: 525  
 Jumlah data validation: 59  
 Jumlah data testing: 147

**Gambar 23.** Membagi dataset.

## 2.8 Mencari tahu hasil dari pembagian data

```

1  print("\n===== Data Training =====")
2  print(train_data.head())
3
4  print("\n===== Data Validation =====")
5  print(val_data.head())
6
7  print("\n===== Data Testing =====")
8  print(test_data.head())

```

**Gambar 24.** Mencari tahu hasil pembagian data.

```

===== Data Training =====
   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 25.** Hasil bagi data training.



===== Data Validation =====									
	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 26.** Hasil bagi data validation.

===== Data Testing =====									
	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 27.** Hasil bagi data testing.

**Link Github :** <https://github.com/Shid2iq/Machine-Learning>

## Referensi:

- Munir, S., Seminar, K. B., Sudradjat, Sukoco, H., & Buono, A. (2022). The Use of Random Forest Regression for Estimating Leaf Nitrogen Content of Oil Palm Based on Sentinel 1-A Imagery. *Information, 14*(1), 10. <https://doi.org/10.3390/info14010010>
- Seminar, K. B., Imantho, H., Sudradjat, Yahya, S., Munir, S., Kaliana, I., Mei Haryadi, F., Noor Baroroh, A., Supriyanto, Handoyo, G. C., Kurnia Wijayanto, A., Ijang Wahyudin, C., Liyantono, Budiman, R., Bakir Pasaman, A., Rusiawan, D., & Sulastri. (2024). PreciPalm: An Intelligent System for Calculating Macronutrient Status and Fertilizer Recommendations for Oil Palm on Mineral Soils Based on a Precision Agriculture Approach. *Scientific World Journal, 2024*(1). <https://doi.org/10.1155/2024/1788726>