

PEMBUATAN GRAFIK PHYTON  
KOMPUTASI GEOFISIKA



DAFINA AJENG KINANTI  
03411940000003

Pada penugasan kali ini telah dipelajari cara pembuatan grafik dan pengolahan data pada python melalui fungsi pandas. Salah satu contohnya, seperti yang tertera pada gambar dibawah ini:

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [2]: import pandas as pd
```

```
In [3]: curah_hujan=pd.read_csv('https://git.bandung.go.id/opendatabdg/databdg/raw/master/resources/48d/5c1/cuaca-dan-curah-hujan-di-kot
```

```
In [4]: curah_hujan.head()
```

```
Out[4]:
```

	Bulan	Temperatur_ratarata(oC)	Temperatur_max(oC)	Temperatur_min(oC)	Curah_Hujan(mm)	Hari_Hujan(hari)	LPM(%)
0	Januari	23.4	28.3	21.0	216.9	26	49
1	Februari	23.4	28.6	20.3	250.0	23	52
2	Maret	23.8	29.6	20.4	305.0	24	58
3	April	3.7	29.1	20.7	286.0	26	52
4	Mei	23.5	28.7	20.3	171.0	23	53

```
In [5]: curah_hujan.tail()
```

```
Out[5]:
```

	Bulan	Temperatur_ratarata(oC)	Temperatur_max(oC)	Temperatur_min(oC)	Curah_Hujan(mm)	Hari_Hujan(hari)	LPM(%)
7	Agustus	3.2	29.4	18.5	74.0	9	79
8	September	23.7	30.1	19.1	172.0	10	77
9	Oktober	23.8	30.0	19.6	234.0	21	69
10	Nopember	23.8	29.8	20.0	164.0	19	64

Gambar 1 Pengolahan data head dan tail

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [5]: curah_hujan.tail()
```

```
Out[5]:
```

	Bulan	Temperatur_ratarata(oC)	Temperatur_max(oC)	Temperatur_min(oC)	Curah_Hujan(mm)	Hari_Hujan(hari)	LPM(%)
7	Agustus	3.2	29.4	18.5	74.0	9	79
8	September	23.7	30.1	19.1	172.0	10	77
9	Oktober	23.8	30.0	19.6	234.0	21	69
10	Nopember	23.8	29.8	20.0	164.0	19	64
11	Desember	23.1	28.4	20.1	418.0	27	40

```
In [6]: curah_hujan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 7 columns):
Bulan                12 non-null object
Temperatur_ratarata(oC)  12 non-null float64
Temperatur_max(oC)     12 non-null float64
Temperatur_min(oC)     12 non-null float64
Curah_Hujan(mm)       12 non-null float64
Hari_Hujan(hari)       12 non-null int64
LPM(%)                12 non-null int64
dtypes: float64(4), int64(2), object(1)
memory usage: 800.0+ bytes
```

```
In [7]: curah_hujan.describe()
```

Gambar 2 pengolahan data curah hujan info

A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8890/notebooks/tugas%20ajeng%202.ipynb'. The notebook title is 'tugas ajeng 2' with a checkpoint from 2 hours ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for file operations, running, and code execution. The code cell contains 'curah\_hujan.describe()'. The output is a detailed statistical summary of the DataFrame, including counts for each month, mean and standard deviation for temperature and rainfall, and minimum/maximum values for temperature, rainfall, and days with rain.

```
In [7]: curah_hujan.describe
```

```
Out[7]: <bound method NDFrame.describe of      Bulan  Temperatur_ratarata(oC)  Temperatur_max(oC) \
0  Januari      23.4      28.3
1  Februari      23.4      28.6
2  Maret      23.8      29.6
3  April      3.7      29.1
4  Mei      23.5      28.7
5  Juni      23.6      28.5
6  Juli      22.5      28.0
7  Agustus      3.2      29.4
8  September      23.7      30.1
9  Oktober      23.8      30.0
10 Nopember      23.8      29.8
11 Desember      23.1      28.4

Temperatur_min(oC)  Curah_Hujan(mm)  Hari_Hujan(hari)  LPM(%)
0      21.0      216.9      26      49
1      20.3      250.0      23      52
2      20.4      305.0      24      58
3      20.7      286.0      26      52
4      20.3      171.0      23      53
5      20.3      231.5      16      61
6      19.0      159.0      16      58
7      18.5      74.0      9      79
8      19.1      172.0      10      77
9      19.6      234.0      21      69
10     20.0      164.0      19      64
11     20.1      418.0      27      40 >
```

Gambar 3 pengolahan data curah hujan describe

A screenshot of a Jupyter Notebook interface showing two code cells and their outputs. The first code cell uses 'groupby' to calculate the mean for each day of the month. The second code cell uses 'groupby' to calculate the mean for each temperature range. The outputs are formatted as tables.

```
In [9]: curah_hujan.groupby('Hari_Hujan(hari)').mean()
```

```
Out[9]:
```

Hari_Hujan(hari)	Temperatur_ratarata(oC)	Temperatur_max(oC)	Temperatur_min(oC)	Curah_Hujan(mm)	LPM(%)
9	3.20	29.40	18.50	74.00	79.0
10	23.70	30.10	19.10	172.00	77.0
16	23.05	28.25	19.65	195.25	59.5
19	23.80	29.80	20.00	164.00	64.0
21	23.80	30.00	19.60	234.00	69.0
23	23.45	28.65	20.30	210.50	52.5
24	23.80	29.60	20.40	305.00	58.0
26	13.55	28.70	20.85	251.45	50.5
27	23.10	28.40	20.10	418.00	40.0

```
In [13]: curah_hujan.groupby('Temperatur_ratarata(oC)').mean()
```

```
Out[13]:
```

Temperatur_ratarata(oC)	Temperatur_max(oC)	Temperatur_min(oC)	Curah_Hujan(mm)	Hari_Hujan(hari)	LPM(%)
3.2	29.40	18.50	74.000000	9.000000	79.000000
3.7	29.10	20.70	286.000000	26.000000	52.000000
22.5	28.00	19.00	159.000000	16.000000	58.000000

Gambar 4 Pengolahan data curah hujan secara pengelompokan



Gambar 5 Pengolahan grafik

Berdasarkan gambar 1-5 dapat kita ketahui bahwa data tersebut adalah pengolahan data dari data curah hujan yang berada di daerah Bandung. Alasan saya mengambil data tersebut karena saya tertarik untuk menganalisis dan mempelajari tentang iklim di suatu daerah. Selain itu, ketika saya belajar tentang iklim dan intensitas hujan saya dapat mengetahui resiko bencana yang akan terjadi akibat iklim ekstrim serta tingginya curah hujan, seperti resiko tanah longsor dan banjir. Alasan lain saya memilih data ini juga karena data ini masih ada kaitannya dengan jurusan Teknik geofisika. Pada 5 gambar tersebut menggunakan beberapa jenis fungsi yang kegunaannya adalah sebagai berikut :

Pada gambar 1 terdapat fungsi head dan tail. Fungsi head diinputkan pada data tersebut untuk menunjukkan 5 data teratas. Sedangkan, fungsi tail diinputkan pada data tersebut untuk menunjukkan 5 data paling bawah.

Pada gambar 2 terdapat fungsi info. Fungsi ini digunakan pada data tersebut untuk menunjukkan nomer index beserta tipe datanya.

Pada gambar 3 terdapat fungsi describe. Fungsi ini diinputkan pada data tersebut untuk menunjukkan rangkuman statistik seperti rata-rata, median, dll pada kolom.

Pada gambar 4 terdapat fungsi groupby mean. Fungsi ini diinputkan untuk pengelompokan rata-rata pada data yang diinputkan.

Pada gambar kelima terdapat plotting untuk grafik dan juga terdapat interpretasi grafik dari data yang saya olah. Pada pembuatan grafik menggunakan fungsi berupa `matplotlib.pyplot` yang diimport sebagai `plt`. `Plt` disini berfungsi untuk memplot suatu grafiknya. Setelah selesai memplot didapatkan interpretasi grafik seperti gambar kelima. Grafik tersebut memuat 3 garis grafik yang merupakan grafik dari temperatur rata-rata, temperatur minimum, dan temperatur maximum. Dimana temperatur tersebut dimuat berdasarkan bulan. Alasan saya membuat grafik hanya berdasarkan temperatur tanpa menginput grafik curah hujan, hari hujan, dan LPM karena temperatur pada data `Cvs` nya merupakan data yang paling banyak dan paling beragam.

Pada grafik tersebut dapat dilihat bahwa temperatur rata-rata kota Bandung pada sepanjang tahun 2019 adalah sekitar 20-24°C. Sedangkan, untuk temperatur minimum kota Bandung pada sepanjang tahun 2019 berkisar 19-20°C. Serta, untuk temperatur maximumnya sekitar 27-30°C. Pada grafik tersebut juga bisa dilihat bahwa dalam 1 bulan temperatur di kota Bandung sangat tidak menentu terkadang bersuhu 19-20°C terkadang bisa melonjak ke angka 27-30°C. Serta pada grafik tersebut dapat dilihat bahwa pada tahun 2019 di kota Bandung temperaturnya tidak melebihi 30°C.