

LAPORAN HASIL PEMROGRAMAN PHYTON



Oleh :

Nama : Kharisma Amalia

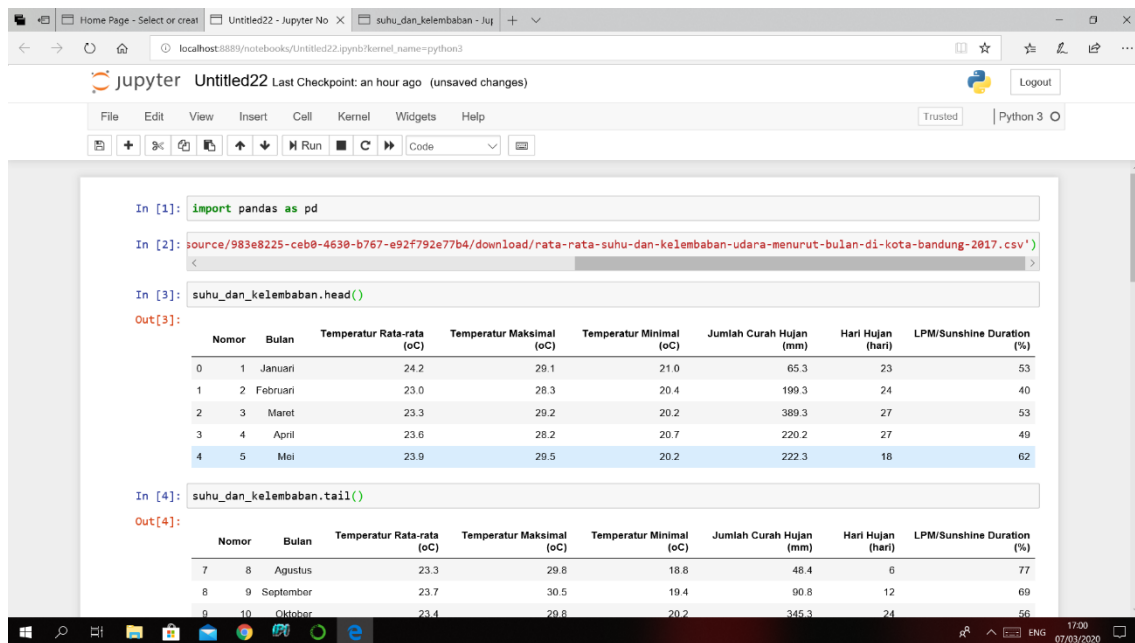
NRP : 03411940000013

DEPARTEMEN TEKNIK GEOFISIKA

FAKULTAS TEKNIK SIPIL PERENCANAAN DAN KEBUMIHAN

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

Curah hujan yang tak menentu membuat masyarakat harus mengetahui bagaimana intensitas dari factor-faktor di dalamnya, untuk mengetahui intensitas dari factor-faktor tersebut dapat dipermudah dengan pembuatan grafik. Pada kesempatan kali ini telah dipelajari cara membuat grafik pada python menggunakan fungsi library terutama fungsi pandas. Berikut merupakan bagaimana fungsi pandas digunakan



```
In [1]: import pandas as pd
In [2]: source/983e8225-ceb8-4638-b767-e92f792e77b4/download/rata-rata-suhu-dan-kelembaban-udara-menurut-bulan-di-kota-bandung-2017.csv')
In [3]: suhu_dan_kelembaban.head()
Out[3]:
```

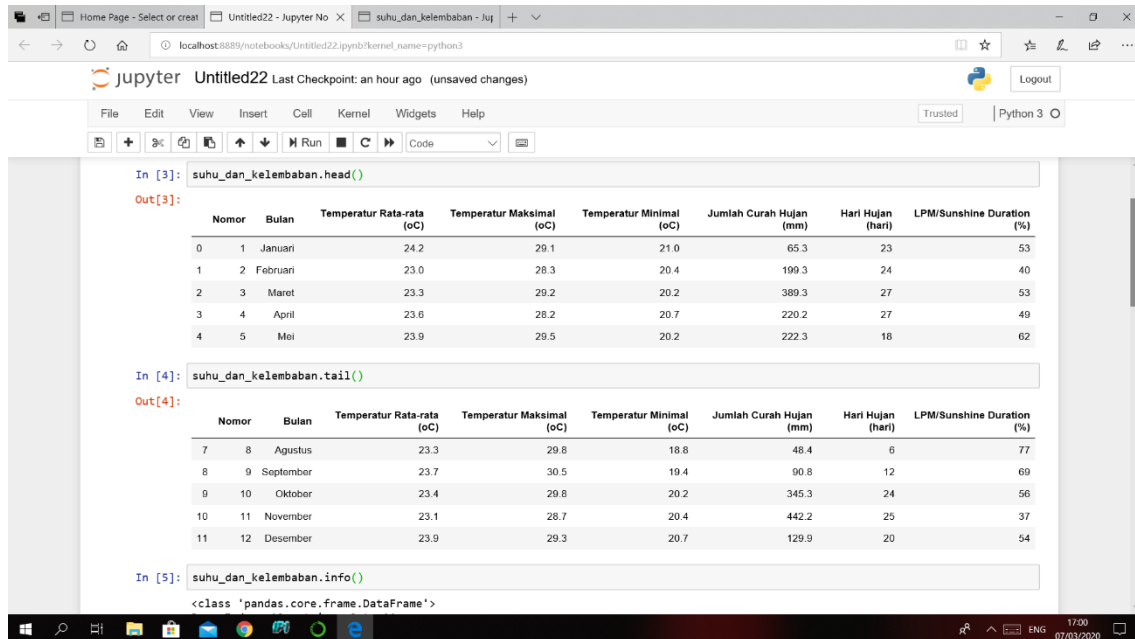
	Nomor	Bulan	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
0	1	Januari	24.2	29.1	21.0	65.3	23	53
1	2	Februari	23.0	28.3	20.4	199.3	24	40
2	3	Maret	23.3	29.2	20.2	389.3	27	53
3	4	April	23.6	28.2	20.7	220.2	27	49
4	5	Mai	23.9	29.5	20.2	222.3	18	62

```
In [4]: suhu_dan_kelembaban.tail()
Out[4]:
```

	Nomor	Bulan	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
7	8	Agustus	23.3	29.8	18.8	48.4	6	77
8	9	September	23.7	30.5	19.4	90.8	12	69
9	10	Oktober	23.4	29.8	20.2	345.3	24	56

Hal pertama yang harus dilakukan adalah mengimportkan fungsi pandas sebagai pd. Setelah itu kita menentukan nama untuk mewakili tentang apa grafik ini, selanjutnya mencantumkan sebuah link yang berisi data tabular untuk diolah sehingga nantinya akan keluar output berupa grafik. Perintah yang diinputkan adalah

suhu_dan_kelembaban=pd.csv_read(link yang bersangkutan).



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

```
In [3]: suhu_dan_kelembaban.head()
```

```
Out[3]:
```

	Nomor	Bulan	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
0	1	Januari	24.2	29.1	21.0	65.3	23	53
1	2	Februari	23.0	28.3	20.4	199.3	24	40
2	3	Maret	23.3	29.2	20.2	389.3	27	53
3	4	April	23.6	28.2	20.7	220.2	27	49
4	5	Mei	23.9	29.5	20.2	222.3	18	62

```
In [4]: suhu_dan_kelembaban.tail()
```

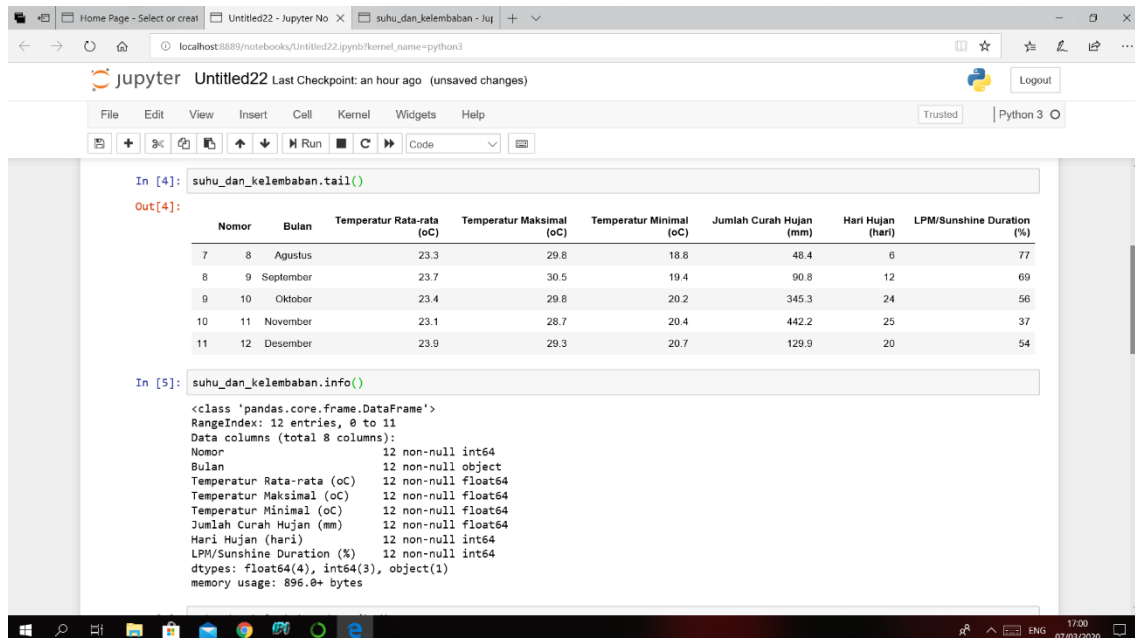
```
Out[4]:
```

	Nomor	Bulan	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
7	8	Agustus	23.3	29.8	18.8	48.4	6	77
8	9	September	23.7	30.5	19.4	90.8	12	69
9	10	Oktober	23.4	29.8	20.2	345.3	24	56
10	11	November	23.1	28.7	20.4	442.2	25	37
11	12	Desember	23.9	29.3	20.7	129.9	20	54

```
In [5]: suhu_dan_kelembaban.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Selanjutnya untuk mengetahui 5 data teratas dan 5 data terbawah dituliskan perintah `suhu_dan_kelembaban.head()` serta `suhu_kelembaban.tail()`. Setelah itu kita perlu unntuk mengetahui informasi mengenai data set yang akan kita gunakan. Informasi ini nantinya yang akan digunakan untuk membuat grafik. Perintah yang dituliskan serta `suhu_kelembaban.info()`.



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

```
In [4]: suhu_dan_kelembaban.tail()
```

```
Out[4]:
```

	Nomor	Bulan	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
7	8	Agustus	23.3	29.8	18.8	48.4	6	77
8	9	September	23.7	30.5	19.4	90.8	12	69
9	10	Oktober	23.4	29.8	20.2	345.3	24	56
10	11	November	23.1	28.7	20.4	442.2	25	37
11	12	Desember	23.9	29.3	20.7	129.9	20	54

```
In [5]: suhu_dan_kelembaban.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 8 columns):
Nomor                12 non-null int64
Bulan                12 non-null object
Temperatur Rata-rata (oC)  12 non-null float64
Temperatur Maksimal (oC)  12 non-null float64
Temperatur Minimal (oC)  12 non-null float64
Jumlah Curah Hujan (mm)  12 non-null float64
Hari Hujan (hari)       12 non-null int64
LPM/Sunshine Duration (%)  12 non-null int64
dtypes: float64(4), int64(3), object(1)
memory usage: 896.0+ bytes
```

Sedangkan untuk perintah serta `suhu_kelembaban.describe()` berfungsi untuk menampilkan table dari perintah `info` yang telah dideklarasikan.

```
In [5]: suhu_dan_kelembaban.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 8 columns):
Nomor          12 non-null int64
Bulan          12 non-null object
Temperatur Rata-rata (oC)  12 non-null float64
Temperatur Maksimal (oC)  12 non-null float64
Temperatur Minimal (oC)   12 non-null float64
Jumlah Curah Hujan (mm)   12 non-null float64
Hari Hujan (hari)        12 non-null int64
LPM/Sunshine Duration (%) 12 non-null int64
dtypes: float64(4), int64(3), object(1)
memory usage: 896.0+ bytes

In [6]: suhu_dan_kelembaban.describe()

Out[6]:
```

	Nomor	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	6.500000	23.516667	29.183333	20.125000	191.541667	19.750000	58.250000
std	3.605551	0.353768	0.669916	0.623954	137.599481	6.552238	11.686239
min	1.000000	23.000000	28.200000	18.800000	38.100000	6.000000	37.000000
25%	3.750000	23.300000	28.675000	19.875000	84.425000	16.250000	52.000000
50%	6.500000	23.400000	29.200000	20.200000	164.600000	21.500000	55.000000

```
In [6]: suhu_dan_kelembaban.describe()

Out[6]:
```

	Nomor	Temperatur Rata-rata (oC)	Temperatur Maksimal (oC)	Temperatur Minimal (oC)	Jumlah Curah Hujan (mm)	Hari Hujan (hari)	LPM/Sunshine Duration (%)
count	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
mean	6.500000	23.516667	29.183333	20.125000	191.541667	19.750000	58.250000
std	3.605551	0.353768	0.669916	0.623954	137.599481	6.552238	11.686239
min	1.000000	23.000000	28.200000	18.800000	38.100000	6.000000	37.000000
25%	3.750000	23.300000	28.675000	19.875000	84.425000	16.250000	52.000000
50%	6.500000	23.400000	29.200000	20.200000	164.600000	21.500000	55.000000
75%	9.250000	23.750000	29.575000	20.475000	253.050000	24.250000	63.750000
max	12.000000	24.200000	30.500000	21.000000	442.200000	27.000000	77.000000

```
In [7]: import matplotlib.pyplot as plt

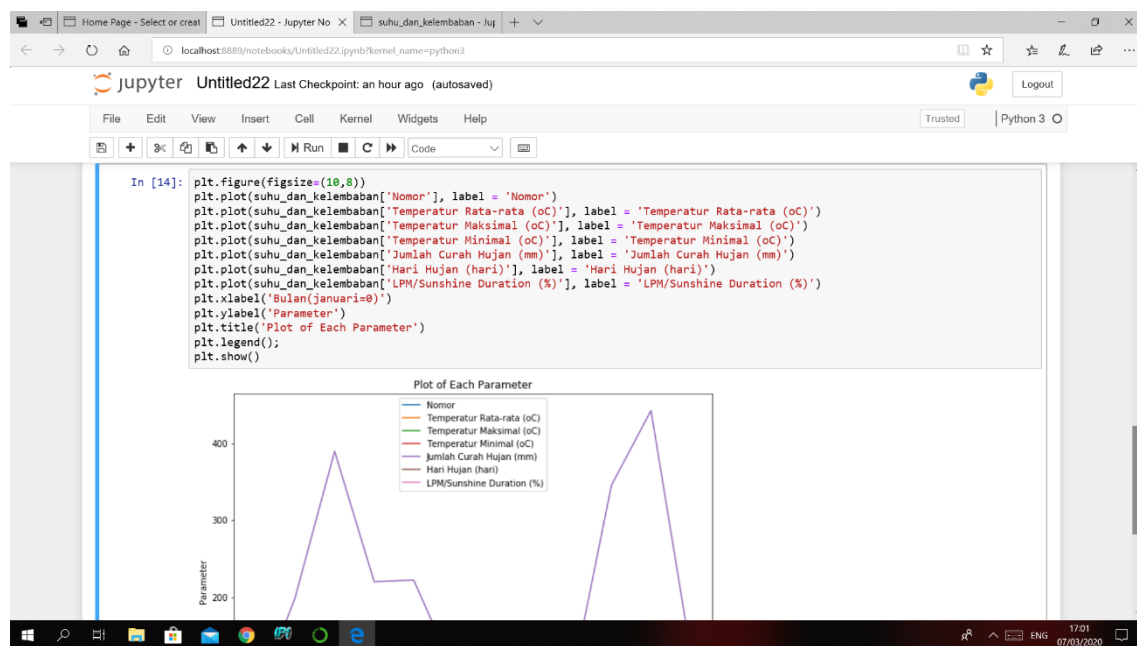
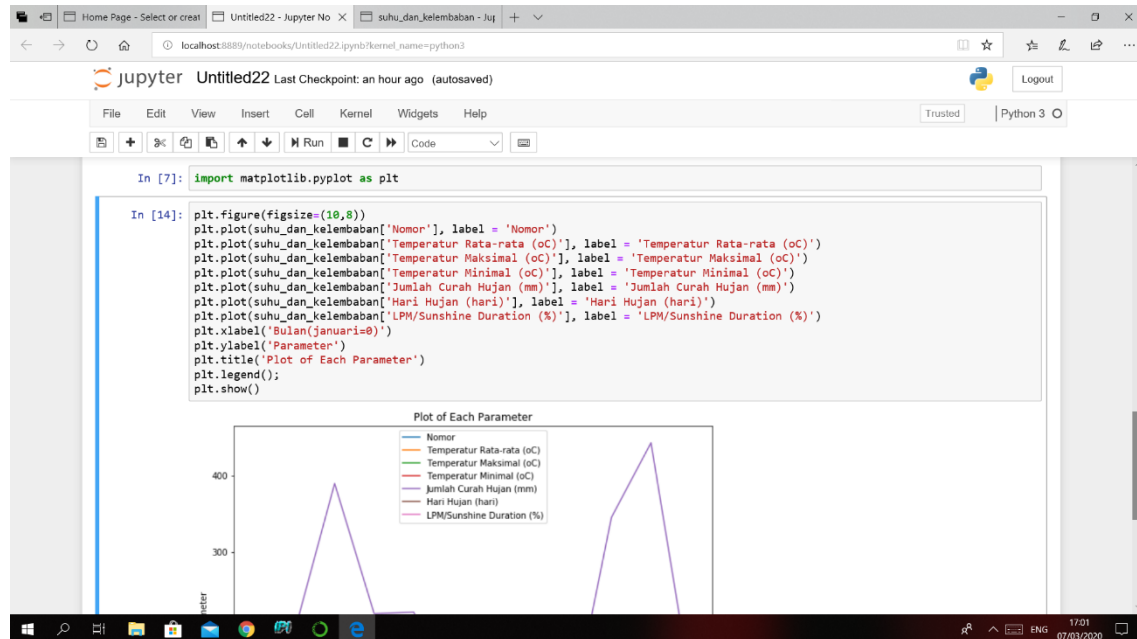
In [14]: plt.figure(figsize=(10,8))
plt.plot(suhu_dan_kelembaban['Nomor'], label = 'Nomor')
plt.plot(suhu_dan_kelembaban['Temperatur Rata-rata (oC)'], label = 'Temperatur Rata-rata (oC)')
plt.plot(suhu_dan_kelembaban['Temperatur Maksimal (oC)'], label = 'Temperatur Maksimal (oC)')
plt.plot(suhu_dan_kelembaban['Temperatur Minimal (oC)'], label = 'Temperatur Minimal (oC)')
plt.plot(suhu_dan_kelembaban['Jumlah Curah Hujan (mm)'], label = 'Jumlah Curah Hujan (mm)')
plt.plot(suhu_dan_kelembaban['Hari Hujan (hari)'], label = 'Hari Hujan (hari)')
plt.plot(suhu_dan_kelembaban['LPM/Sunshine Duration (%)'], label = 'LPM/Sunshine Duration (%)')
plt.xlabel('Bulan(januari=0)')
```

Setelah mengimportkan pandas dan melakukan serangkaian perintahnya, maka hal yang dilakukan setelahnya adalah mengimportkan matplotlib sebagai plt untuk melakukan sebuah plot untuk grafik. Perintah yang dituliskan adalah import

matplotlib.pyplot

as

plt.



Setelah itu, hal yang harus dilakukan adalah plotting objek yang telah tertuliskan pada info. Yaitu nomor, temperature rata-rata, temperature minimal, temperature maksimal, jumlah curah hujan, hari hujan, dan sunshine duration. Selain plotting juga terdapat perintah `plt.figure(figsize=(10,8))` perintah ini berarti adalah ukuran dari figure grafik yang diinginkan `plt.xlabel` untuk memberi label dari sumbu x, `plt.ylabel` untuk memberi label sumbu y, `plt.legend` untuk menampilkan legenda dari warna=warna garis grafik, dan

yang terakhir adalah `plt.show` untuk menampilkan grafik. Sehingga grafiknya kan jadi seperti pada gambar di bawah ini

