

ASSIGNMENT 02
PYTHON PROGRAMMING



NAVIRA NAWA KUNNIASIH
03411940000002
GEOPHYSICAL ENGINEERING DEPARTMENT

INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

Python Programming

Untuk memudahkan pengerjaan data table maka diperlukan script import, iris, dan matplotlib. Fungsi import digunakan sebagai multi file yaitu dapat memanggil file lain di dalam satu modul yang berbeda, jadi fungsi import digunakan untuk memunculkan file lain di dalam program yang sudah ada. Fungsi iris dalam python digunakan sebagai dataset. Sedangkan, matplotlib adalah library plotting 2D Python yang menghasilkan gambar publikasi, dapat juga membuat plot, histogram, power spectra, grafik batang, grafik error, dll.

Dalam mengerjakan data table yaitu Hooke Law's dan memunculkan grafiknya maka scriptnya yang pertama kali ditulis adalah

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

Setelah dirun, maka akan muncul baris baru yang akan kita isi dengan tulisan fungsi “iris = pd.read_csv('halaman web tabular data yang akan kita masukkan’). Dan hasilnya akan seperti ini.

```
In [2]: iris = pd.read_csv('https://people.sc.fsu.edu/~jburkardt/data/csv/hooke.csv')
```

Setelah dirun, akan muncul baris baru lagi yang kita isi dengan “iris.head()”, fungsi ini nantinya akan memunculkan data tabular yang akan kita tampilkan grafiknya. Setelah dirun maka hasilnya akan seperti ini.

Out[3]:

	Index	"Mass (kg)"	"Spring 1 (m)"	"Spring 2 (m)"
0	1	0.00	0.050	0.050
1	2	0.49	0.066	0.066
2	3	0.98	0.087	0.080
3	4	1.47	0.116	0.108
4	5	1.96	0.142	0.138

Setelah itu menuliskan fungsi “iris.tail()” untuk memunculkan data yang belum bisa ditampilkan pada fungsi diatas. Dan hasilnya setelah dirun adalah.

Out[4]:

	Index	"Mass (kg)"	"Spring 1 (m)"	"Spring 2 (m)"
5	6	2.45	0.166	0.158
6	7	2.94	0.193	0.174
7	8	3.43	0.204	0.192
8	9	3.92	0.226	0.205
9	10	4.41	0.238	0.232

Setelah semua data telah ditampilkan, lalu menuliskan fungsi “iris.info()”. Dan setelah dirun tampilannya akan seperti ini.

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
Index                10 non-null int64
"Mass (kg)"          10 non-null float64
"Spring 1 (m)"       10 non-null float64
"Spring 2 (m)"       10 non-null float64
dtypes: float64(3), int64(1)
memory usage: 448.0 bytes
```

Kita butuh mendeskripsikan semua data yang ingin kita tampilkan pada grafik, maka harus menuliskan fungsi “iris.describe()”, dan ketika dirun akan menghasilkan output-an sebagai berikut.

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

Out[6]:

	Index	"Mass (kg)"	"Spring 1 (m)"	"Spring 2 (m)"
count	10.00000	10.000000	10.000000	10.000000
mean	5.50000	2.205000	0.148800	0.140300
std	3.02765	1.483549	0.067307	0.062429
min	1.00000	0.000000	0.050000	0.050000
25%	3.25000	1.102500	0.094250	0.087000
50%	5.50000	2.205000	0.154000	0.148000
75%	7.75000	3.307500	0.201250	0.187500
max	10.00000	4.410000	0.238000	0.232000

Di gambar di atas terdapat semua informasi yang kita butuhkan. Namun kita belum tahu semua tabular data yang akan kita buat grafiknya, maka untuk mengetahui semua datanya harus menuliskan fungsi. Dalam script ini fungsi yang kita tuliskan adalah

“print(iris.groupby('Index').mean()), dan setelah dirun akan muncul sebagai berikut.

```
In [7]: print(iris.groupby('Index').mean())
```

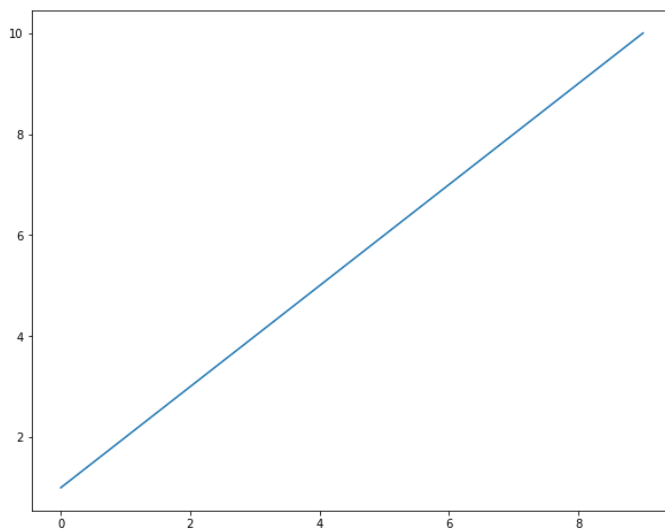
	"Mass (kg)"	"Spring 1 (m)"	"Spring 2 (m)"
Index			
1	0.00	0.050	0.050
2	0.49	0.066	0.066
3	0.98	0.087	0.080
4	1.47	0.116	0.108
5	1.96	0.142	0.138
6	2.45	0.166	0.158
7	2.94	0.193	0.174
8	3.43	0.204	0.192
9	3.92	0.226	0.205
10	4.41	0.238	0.232

Setelah semua data sudah didapatkan dan berhasil ditampilkan, maka harus memplot, caranya dengan menuliskan fungsi “import matplotlib.pyplot as plt” pada baris baru, dan setelah dirun kita harus menuliskan fungsi kembali yang ditampilkan pada gambar di bawah ini.

```
In [9]: import matplotlib.pyplot as plt
```

```
In [10]: plt.figure(figsize=(10,8))
plt.plot(iris['Index'], label='Index')
plt.plot(iris['Mass'], label='Mass')
plt.plot(iris['Spring 1'], label='Spring 1')
plt.plot(iris['Spring 2'], label='Spring 2')
# Keep in mind we dont plot renewable energy since it is an object
plt.xlabel('Data Count') #giving name to x axis
plt.ylabel('Value') #giving name to y axis
plt.title('Plot of each parameter') #title
plt.legend() # taggle on the legend of each line
plt.show() # to show the plot
```

Setelah dirun maka akan muncul grafik.



Grafik tersebut merupakan grafik hubungan antara massa dan panjang pegas setelah diberi massa pada percobaan Hukum Hooke.