

Pembuatan Plot Dasar dari Data Tabular Menggunakan Matplotlib dan Pandas

Pada materi kali ini, mahasiswa mencoba untuk membuat plot dasar beserta grafiknya dari data tabular menggunakan Matplotlib dan visualisasi Pandas. Kemampuan dalam memahami penulisan script ini sangat berguna untuk mengolah data tabular menjadi grafik yang dapat memvisualisasikan isi data serta lebih mudah dipahami oleh masyarakat. Untuk dapat menjalankan scriptnya, maka hal yang perlu dilakukan adalah mengimpor syntax Matplotlib dan Pandas.

```
import pandas as pd

iris = pd.read_csv('https://www.stats.govt.nz/assets/Uploads/Consumers-price-
index/Consumers-price-index-December-2019-quarter/Download-data/consumers-
price-index-December-2019-quarter-corrections.csv')

print(iris.head())
```

Hasil python:

	Series reference	Period	Initially published	Revised
0	CPIQ.SE907303	2017.03	1170.192308	1172
1	CPIQ.SE907303	2017.06	1000.000000	1000
2	CPIQ.SE907303	2017.09	1040.384615	1021
3	CPIQ.SE907303	2017.12	1105.000000	1111
4	CPIQ.SE907303	2018.03	1124.000000	1117

Pada baris pertama script tersebut terdapat perintah ***import pandas as pd*** merupakan syntax untuk mengimpor pandas supaya dapat digunakan pada script ini. Kemudian dibawahnya merupakan script untuk mengimpor data tabular dari internet dalam format csv. Pada tugas ini, saya menggunakan data tabular tentang Indeks Harga Konsumen yang diunduh dari website pemerintah Selandia Baru. Di baris ketiga terdapat ***print(iris.head())*** yang dimaksudkan untuk menampilkan data lima teratas dari data tabular tersebut. Dan hasilnya seperti yang telah tercantum diatas.

```
print(iris.tail())
```

Hasil python:

	Series reference	Period	Initially published	Revised
149	CPIQ.SE707	2018.09	1031.0	1030
150	CPIQ.SE707	2018.12	1029.0	1028
151	CPIQ.SE707	2019.03	999.0	997
152	CPIQ.SE707	2019.06	1010.0	1009
153	CPIQ.SE707	2019.09	1012.0	1011

Pada perintah ***print(iris.tail())*** ini digunakan untuk menampilkan data lima terbawah dari data tabular tersebut. Sehingga hasilnya akan muncul seperti yang telah tertera diatas. Karena pada data tabular biasanya memuat ratusan bahkan ribuan data dan supaya waktu kerja lebih efektif, maka digunakan perintah “***head()***” atau “***tail()***” untuk menampilkan lima data teratas atau lima data terbawah sesuai dengan kebutuhan penggunaanya.

```
print(iris.info())
```

Hasil python:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 4 columns):
Series reference      154 non-null object
Period                154 non-null float64
Initially published    154 non-null float64
Revised               154 non-null int64
dtypes: float64(2), int64(1), object(1)
memory usage: 4.9+ KB
None
```

Selanjutnya adalah perintah ***print(iris.info())*** yang digunakan untuk menampilkan informasi mengenai data tabular yang digunakan. Sebagai contoh, data tabular yang saya pakai tersebut membutuhkan penyimpanan sebesar 4.9 KB. Hal ini dapat dilihat dari hasil running script yang tertera pada kalimat “***memory usage: 4.9+ KB***”. Pada data tabular tersebut terdapat 154 data yang ada dengan 4 kolom. Informasi ini dapat ditemukan pada kalimat “***RangeIndex: 154 entries, 0 to 153***” dan “***Data columns (total 4 columns) :***”.

```
print(iris.describe())
```

Hasil python:

	Period	Initially published	Revised
count	154.000000	154.000000	154.000000
mean	2017.980000	1026.555439	1024.629870
std	0.790097	38.244334	37.961789
min	2017.030000	942.000000	941.000000
25%	2017.090000	1000.000000	1000.000000
50%	2018.060000	1020.000000	1019.500000
75%	2019.030000	1038.500000	1035.750000
max	2019.090000	1170.192308	1172.000000

Syntax tersebut digunakan untuk menampilkan deskripsi dari data tabular yang digunakan. Hasilnya, terdapat informasi mengenai jumlah data, rata-rata nilai pada data, nilai minimal, nilai maksimal, nilai pada kuartil 1 (25%), nilai pada kuartil 2 (50%), dan nilai pada kuartil 3 (75%) .

```
print(iris.groupby('Period').mean())
```

Hasil python:

	Initially published	Revised
Period		
2017.03	1047.802620	1048.071429
2017.06	1000.000000	1000.000000
2017.09	998.235781	993.500000
2017.12	1039.785714	1041.000000
2018.03	1034.500000	1032.642857
2018.06	1006.785714	1002.928571
2018.09	1036.785714	1032.857143
2018.12	1060.714286	1062.214286
2019.03	1030.428571	1024.714286
2019.06	1001.000000	1001.214286
2019.09	1036.071429	1031.785714

Dari syntax tersebut, dapat diketahui bahwa **“groupby”** digunakan untuk mengelompokkan data berdasarkan ketentuan yang telah dibuat oleh pengguna. Pada materi ini, saya mengelompokkan data berdasarkan periodenya. Sehingga hasilnya akan muncul seperti pada tabel diatas, dimana data dikelompokkan sesuai periodenya, yakni

mulai dari bulan Maret tahun 2017 hingga bulan September tahun 2019. Pengelompokan data ini tentunya sangat bermanfaat bagi pekerjaan manusia yang membutuhkan data khusus untuk suatu keperluan. Sehingga dengan syntax ini pengguna akan lebih mudah untuk mencari dan mendapatkan data sesuai dengan yang dibutuhkan.

```
import matplotlib.pyplot as plt

#create a figure and axis

fig, ax = plt.subplots()

# scatter the Period against the Initially published

ax.scatter(iris['Period'], iris['Initially published'])

# set a title and labels

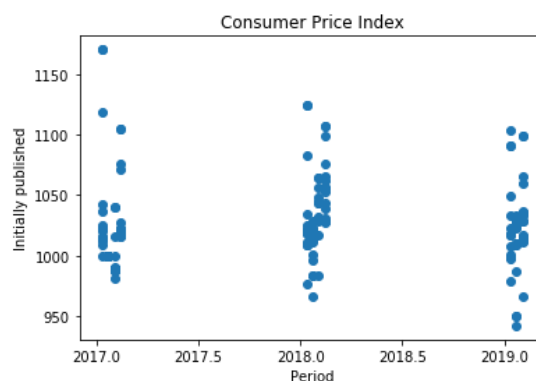
ax.set_title('Consumer Price Index')

ax.set_xlabel('Period')

ax.set_ylabel('Initially published')
```

Hasil python:

```
Text(0, 0.5, 'Initially published')
```



Pada perintah *import matplotlib.pyplot as plt* digunakan untuk mengimpor matplotlib kedalam bentuk perintah plt. Kemudian perintah berikutnya digunakan untuk membuat gambar grafik menggunakan plot. Selanjutnya, maksud dari *ax_scatter(iris['Period'], iris['Initially published'])* adalah untuk membuat plot

menyebar berdasarkan data pada kolom *“Period”* dan kolom *“Initially published”*. Adapun logika dari tiga perintah selanjutnya ialah untuk membuat judul grafik, memberi nama pada sumbu x, serta untuk memberi nama pada sumbu y.

```
# create color dictionary

colors = {'CPIQ.SE907303':'r', 'CPIQ.SE9073':'g', 'CPIQ.SE907':'b'}

# create a figure and axis

fig, ax = plt.subplots()

# plot each data-point

for i in range(len(iris['Period'])):

    ax.scatter(iris['Period'][i],    iris['Initially    published'][i],color=colors[iris['Series
    reference'][i]])

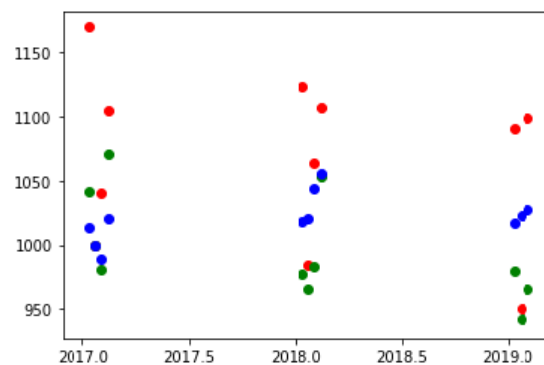
# set a title and labels

ax.set_title('Consumer Price Index')

ax.set_xlabel('Period')

ax.set_ylabel('Initially published')
```

Hasil python:



Script ini dilakukan untuk membuat plot berwarna sesuai dengan ketentuan yang dibuat oleh pengguna. Logika dari perintah `colors = {'CPIQ.SE907303':'r', 'CPIQ.SE9073':'g', 'CPIQ.SE907':'b'}` adalah jika ada data yang memuat kalimat `"CPIQ.SE907303"` plotnya berwarna merah, kemudian jika ada data yang memuat kalimat `"CPIQ.SE9073"` plotnya akan berwarna hijau, dan jika ada data yang memuat kalimat `"CPIQ.SE907"` maka plotnya berwarna biru. Pada perintah selanjutnya dimaksudkan untuk membuat gambar grafik dengan menggunakan plot. Sebenarnya plotting pada perintah ini sama dengan perintah plotting pada script sebelumnya, bedanya adalah pada script ini plot nya ada tiga warna.

Adapun logika pada dua perintah selanjutnya adalah untuk membuat plot menyebar berdasarkan data pada kolom `"Period"` dan `"Initially published"` dengan increment `i`. Increment disini maksudnya untuk memberi pengulangan sebanyak `i`, dimana untuk variabel periode increment nya sebesar 0.5 dan untuk variabel publikasi inisialnya sebesar 50, hal ini dapat dilihat dari grafik tersebut. Selain itu maksud dari `color=colors[iris['Series reference']][i])` adalah untuk memberi warna pada plot berdasarkan data dari variabel seri referensi, untuk script pewarnaan plotnya telah ditulis pada perintah colors diatas. Selanjutnya, maksud dari tiga perintah berikutnya digunakan untuk memberi judul pada grafik, pemberian label pada sumbu x, serta pemberian label pada sumbu y.

```
# get columns to plot

columns = iris.columns.drop(['Series reference'])

# create x data

x_data = range(0, iris.shape[0])

# create figure and axis

fig, ax = plt.subplots()

# plot each column

for column in columns:
```

```
ax.plot(x_data, iris[column], label=column)
```

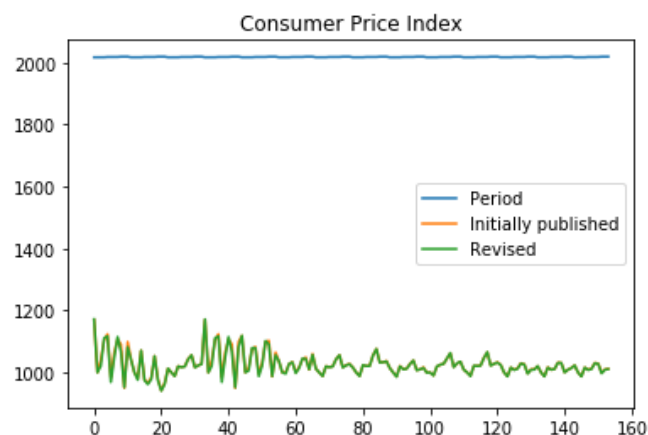
```
# set title and legend
```

```
ax.set_title('Consumer Price Index')
```

```
ax.legend()
```

Hasil python:

```
<matplotlib.legend.Legend at 0xcbf6ac8>
```



Untuk dapat menampilkan grafik yang bertumpuk pada satu gambar diperlukan script khusus, dapat disesuaikan dengan script diatas. Logika dari perintah ***columns = iris.columns.drop(['Series reference'])*** ialah digunakan untuk mengeluarkan kolom seri referensi pada grafik. Alasan saya memilih kolom seri referensi yang dihilangkan pada grafik sebab sebenarnya yang lebih dibutuhkan untuk pengolahan data selanjutnya ialah nilai-nilai pada variabel yang lain. Selain itu, jika variabel ini tidak dihilangkan maka keterangannya untuk variabel ini akan saling tumpang-tindih sehingga sulit untuk dibaca. Kemudian maksud dari perintah ***x_data = range(0, iris.shape[0])*** adalah untuk membuat data pada grafik yang dimulai dari rentang nol hingga batasan tersebut.

Logika dari perintah ***ax.plot(x_data, iris[column], label=column)*** dapat dijelaskan bahwa perintah tersebut untuk membuat plot dengan data x nya diambil dari data pada kolom di tabular data, yang dalam hal ini data kolom tersebut ialah variabel-variabel yang ada. Labelnya pun disesuaikan dengan kolom tersebut. Penjelasan untuk dua script terakhir ini ialah digunakan untuk memberi judul berupa ***“Consumer Price***

Index” dan membuat legenda pada grafik berdasarkan data variabel yang ada, warna untuk setiap variabel pun berbeda-beda.

Hal ini dimaksudkan supaya pembaca grafik lebih mudah untuk memahami isi grafik tersebut. Dari grafik hasil tersebut terlihat bahwa grafik garis yang berwarna kuning, yakni variabel “Initially published” cukup sedikit datanya bila dibandingkan dengan dua variabel lainnya. Karena itulah grafik garisnya kurang terlihat jelas, apalagi garisnya tersebut tertindih oleh garis variabel “Revised” yang lebih banyak datanya serta lebih kuat warnanya.

Alasan saya memilih data tersebut adalah sebagai konsumen perlu mengetahui juga index harga konsumen di pasaran, supaya dapat membeli barang dengan bijak dan sesuai dengan kebutuhan. Selain itu, data ini juga tergolong mudah untuk diolah. Hasil dari dari pengolahan datanya pun mudah dipahami oleh masyarakat sehingga dapat membantu memberikan informasi tambahan yang berguna untuk kehidupan manusia.