

Python for Data Science

Ch. 3 Pandas Library

Ahmad Rio Adriansyah

Pandas

- Pandas adalah library python yang banyak berkaitan dengan :
 - Series (array 1 dimensi yang homogen)
 - Data Frames (array 2 dimensi yang heterogen)
 - Panel (array 3 dimensi yang umum)
- Bawaan di Anaconda (otomatis terpasang)
- Selain dengan Anaconda, dapat diinstall melalui
\$ pip install pandas

Series

- Array 1 dimensi yang mengandung data sejenis (homogen)
- Label pada axisnya disebut sebagai index

```
In [1]: import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: seril = pd.Series(np.random.randn(10))
```


```
In [3]: print(seril)
```

```
0    -0.790227
1     1.895110
2    -0.975352
3    -0.988712
4     0.801603
5    -1.634308
6     0.043845
7     0.673665
8     1.093062
9     0.323069
dtype: float64
```

Label pada Series

- Kita bisa memberi label pada series yang dibuat dengan menambahkan argumen index dan memberikan nilai berupa list.
- Defaultnya index diberikan berupa bilangan bulat dari 0 hingga banyaknya data pada series -1.

0 ~ n-1

```
In [4]:  seri2 = pd.Series(np.random.randn(5), index = ['Ahad', 'Senin', 'Selasa', 'Rabu', 'Kamis'])  
print(seri2)
```

```
Ahad      0.306429  
Senin     -1.731769  
Selasa    -0.301920  
Rabu       0.395867  
Kamis     -0.889314  
dtype: float64
```

Contoh Series

- Nilai tukar mata uang terhadap waktu
- Daftar skor/nilai yang didapat
- Luas ruangan
- Banyak peserta dalam suatu rangkaian acara
- dll

Membuat Series dari Dictionary

- Jika dictionary digunakan sebagai inputan pada series, maka secara otomatis key akan menjadi indexnya.
- Index dapat diurutkan dengan memasukkan argumen tambahan index.

```
In [5]: data = {"Amir":170, "Beni":168, "Candra":175, "Dilan":169, "Elang":170, "Fahri":168}
        seri3 = pd.Series(data)
        print(seri3)
```

Amir	170
Beni	168
Candra	175
Dilan	169
Elang	170
Fahri	168
dtype:	int64

Membuat Series

- Selain dengan dictionary, series juga dapat dibuat dari list, numpy array, ataupun skalar

```
data = np.array([14238,14187,14187,14185,14276,14264,14192,14176])
seri4 = pd.Series(data)
print(seri4)
```

```
0    14238
1    14187
2    14187
3    14185
4    14276
5    14264
6    14192
7    14176
dtype: int32
```

```
seri5 = pd.Series(100,index=range(5), name="Skor")
print(seri5)
```

```
0    100
1    100
2    100
3    100
4    100
Name: Skor, dtype: int64
```

Pemanggilan Elemen

- Elemen dari series dipanggil seperti memanggil elemen list atau dictionary. Yaitu dengan menggunakan index atau key nya.

```
In [8]: seri3[0]
```

```
Out[8]: 170
```

```
In [9]: seri3['Dilan']
```

```
Out[9]: 169
```


Pemanggilan Elemen

- Dapat juga menggunakan slicing

```
In [10]: ► seri3[:4]
```

```
Out[10]: Amir      170  
        Beni       168  
        Candra     175  
        Dilan      169  
        dtype: int64
```

```
In [11]: ► seri3[-3:]
```

```
Out[11]: Dilan      169  
        Elang      170  
        Fahri      168  
        dtype: int64
```

```
In [12]: ► seri3["Beni":"Elang"]
```

```
Out[12]: Beni       168  
        Candra     175  
        Dilan      169  
        Elang      170  
        dtype: int64
```

Pemanggilan Elemen

- Untuk memilih elemen yang tidak berurutan, dapat dipanggil dengan menuliskan list berisi indexnya.
- Jadi kurung sikunya dobel (`[[]]`)

```
In [13]: ► seri3[[0,2,5]]
```

```
Out[13]: Amir      170  
        Candra     175  
        Fahri      168  
        dtype: int64
```

```
In [14]: ► seri3[["Beni","Elang","Fahri"]]
```

```
Out[14]: Beni       168  
        Elang       170  
        Fahri       168  
        dtype: int64
```

Boolean Operator

- Operator boolean dapat diterapkan ke dalam series untuk mendapatkan series boolean yang memenuhinya
- Untuk mendapatkan datanya, series boolean dimasukkan sebagai input pemanggilan

```
In [15]: ► seri3>170
```

```
Out[15]: Amir      False  
        Beni       False  
        Candra      True  
        Dilan       False  
        Elang       False  
        Fahri       False  
        dtype: bool
```

```
In [16]: ► seri3[seri3>170]
```

```
Out[16]: Candra     175  
        dtype: int64
```

Fungsi pada Series

- Cari fungsi apa saja yang bisa diberlakukan ke series dengan menerapkan fungsi **dir()**

```
In [ ]: ▶ dir(seri4)
```

- Beberapa contohnya :
 - **unique()**
 - **min(), max()**
 - **mean(), median(), mode()**

```
In [17]: ▶ seri4.unique()
```

```
Out[17]: array([14238, 14187, 14185, 14276, 14264, 14192, 14176], dtype=int64)
```

```
In [18]: ▶ seri4.mean()
```

```
Out[18]: 14213.125
```

Fungsi pada Series

- Beberapa atribut / metode yang sering digunakan kepada series

Attribute/Method	Description
<code>dtype</code>	data type of values in series
<code>empty</code>	True if series is empty
<code>size</code>	number of elements
<code>values</code>	Returns values as ndarray
<code>head()</code>	First n elements
<code>tail()</code>	Last n elements

Latihan

- Buat sebuah series bernama “berat badan”, simpan dalam variabel “seri”
- Isi dengan 10 buah data berat badan dengan nama orang sebagai indexnya
- Tunjukkan orang yang berat badannya lebih dari 70 kg
- Tunjukkan rerata dan nilai tengah data tersebut
- Siapa yang paling berat?

Latihan Tingkat Lanjut

- Download data nilai tukar rupiah dari <https://www.exchange-rates.org/history/IDR/USD/T>
- Masukkan ke dalam series dengan perintah

```
exc = pd.read_csv("HistoryExchangeReport.csv", index_col="Date").Rate
```

- Explore data tersebut, kapan nilai tukar rupiah tertinggi/terendah, berapa reratanya, dll.

Data Frame

- Data Frame adalah data struktur tabular 2 dimensi yang heterogen dan dengan axis (baris dan kolom) yang berlabel
- Bisa dibayangkan sebagai dictionary dari series

```
df = pd.DataFrame({"Nama":pd.Series(["Andri","Budi","Cecep","Dharma"]),  
                  "Umur":pd.Series([18,20,15,22]),  
                  "Tinggi":pd.Series([168,170,165,168])})  
print(df)
```

	Nama	Umur	Tinggi
0	Andri	18	168
1	Budi	20	170
2	Cecep	15	165
3	Dharma	22	168

Menambahkan Kolom

- Untuk menambahkan kolom, cukup menggunakan indexing karena data frame bersifat mutabel
- Note : jika index yang diberikan sudah ada, maka akan terganti datanya dengan yang baru

```
df["Berat"] = pd.Series([80,60,92,60])  
print(df)
```

	Nama	Umur	Tinggi	Berat
0	Andri	18	168	80
1	Budi	20	170	60
2	Cecep	15	165	92
3	Dharma	22	168	60

Data Frame

- Pada materi Python for Data Science ini, kita akan banyak berinteraksi dengan data frame
 - Membuat data frame
 - Manipulasi data frame
 - Mengisi data frame dari data pada file
 - Membersihkan data
 - Menganalisis data dalam data frame
 - dll

Input dari File

- Data dapat disimpan dalam berbagai macam bentuk, diantaranya csv, pdf, excel, json, xml, sql, dan lain lain.
- Yang paling umum adalah csv (comma separated variables)
- Pandas menyediakan fungsi untuk membaca file csv dengan `read_csv()` , hasilnya berupa data frame

Input dari File

- Format :

```
>>> pd.read_csv("namafile.csv")
```

- Dapat ditambahi argumen untuk :
 - Menggunakan kolom tertentu sebagai index (**index_col**)
 - Menetapkan tipe data kolom-kolomnya (**dtype**)
 - Menentukan kolom/baris yang mau diskip/diambil (**skiprows/usecols**)
 - Mengganti nilai yang kosong (missing value) dengan simbol tertentu (**na_values**)
 - dll

Contoh Dataset

- Data gaji pegawai

https://raw.githubusercontent.com/mathcoder3141/blog-data-files/master/Congress_White_House.csv

- Data kecelakaan pesawat

<https://github.com/fivethirtyeight/data/tree/master/airline-safety>

- Data cuaca

<https://github.com/fivethirtyeight/data/tree/master/us-weather-history>

Note : Sampel data yang lain bisa dicari melalui penyedia dataset seperti bps, opendata, kaggle, atau yang lainnya.

- Modul **sklearn**, **seaborn**, **statsmodels** juga memiliki dataset bawaan yang bisa digunakan

SKLearn's Toy Datasets

```
▶ from sklearn import datasets
iris = datasets.load_iris()
# https://scikit-learn.org/stable/datasets/index.html
```

They can be loaded using the following functions:

<code>load_boston</code> ([return_X_y])	Load and return the boston house-prices dataset (regression).
<code>load_iris</code> ([return_X_y])	Load and return the iris dataset (classification).
<code>load_diabetes</code> ([return_X_y])	Load and return the diabetes dataset (regression).
<code>load_digits</code> ([n_class, return_X_y])	Load and return the digits dataset (classification).
<code>load_linnerud</code> ([return_X_y])	Load and return the linnerud dataset (multivariate regression).
<code>load_wine</code> ([return_X_y])	Load and return the wine dataset (classification).
<code>load_breast_cancer</code> ([return_X_y])	Load and return the breast cancer wisconsin dataset (classification).

Another Toy Datasets

```
▶ import seaborn as sns
iris = sns.load_dataset('iris')
# https://github.com/mwaskom/seaborn-data
```

```
▶ import statsmodels.api as sm
iris = sm.datasets.get_rdataset('iris').data
# https://github.com/vincentarelbundock/Rdatasets/tree/master/csv/datasets

data = sm.datasets.longley.load_pandas()
# https://www.statsmodels.org/dev/datasets/index.html
```

- Daftar datasetnya dapat dilihat pada tautan berikut :
 - <https://scikit-learn.org/stable/datasets/index.html>
 - <https://github.com/mwaskom/seaborn-data>
 - <https://github.com/vincentarelbundock/Rdatasets/tree/master/csv/datasets>
 - <https://www.statsmodels.org/dev/datasets/index.html>

Data Frame

- Load salah satu dataset (contoh : mtcars)
- Periksa daftar nama kolom
>>> mt.columns
- Tampilkan beberapa baris data
>>> mt.head()
>>> mt.tail()
>>> mt.sample()
- Periksa tipe data dari kolom tertentu
>>> mt["qsec"].dtype
>>> mt.dtypes

Atribut Data Frame

df.attribute	description
dtypes	list the types of the columns
columns	list the column names
axes	list the row labels and column names
ndim	number of dimensions
size	number of elements
shape	return a tuple representing the dimensionality
values	numpy representation of the data

Metode Data Frame

df.method()	description
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values

Pemanggilan Elemen

- Elemen data frame adalah berupa series yang dituliskan per kolom
- Untuk memanggil kolom tertentu, dapat dilakukan dengan menuliskan nama kolomnya sebagai index
- Atau dengan menggunakan nama kolomnya sebagai atribut

```
In [55]: ▶ mt["model"].head()
```

```
Out[55]: 0      Mazda RX4
         1      Mazda RX4 Wag
         2      Datsun 710
         3      Hornet 4 Drive
         4      Hornet Sportabout
         Name: model, dtype: object
```

```
In [56]: ▶ mt.model.head()
```

```
Out[56]: 0      Mazda RX4
         1      Mazda RX4 Wag
         2      Datsun 710
         3      Hornet 4 Drive
         4      Hornet Sportabout
         Name: model, dtype: object
```

Pemanggilan Elemen

- Selain menggunakan nama indexnya, user juga dapat memilih sel/baris/kolom dengan menggunakan fungsi **loc** atau **iloc**
- Format : **df.loc**[<baris>, <kolom>] , atau **df.iloc**[<baris>, <kolom>]
- Loc digunakan dengan **nama** kolom/barisnya
- Iloc digunakan dengan **index** kolom/barisnya