

Introduction to Pandas

Basic Dataframe



Hello, everyone!

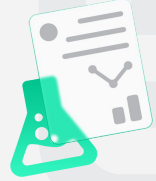


Sebelum kita memulai kelas, kita awali dengan:

1. Berdoa
2. siapkan diri



Pertemuan lalu...



Let's Recall

pertemuan sebelumnya kita sudah belajar pengenalan numpy.
Ada yang inget apa aja?



Target



1. What is Dataframe?
2. Why Dataframe?
3. Creating Dataframe
4. Selecting Dataframe
5. Read/Write Dataframe



Tools



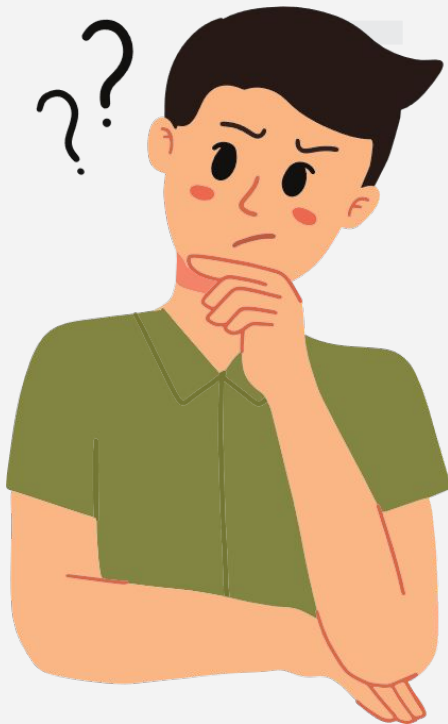
<https://colab.research.google.com/>

1. What is Dataframe?

Apaitu *Dataframe* ?

- Dataframe merupakan salah satu struktur data penting di Python
- Dataframe terdiri dari baris dan kolom seperti tabel
- Dataframe berfungsi untuk menyimpan data dalam format yang dapat diubah-ubah, dengan fleksibilitas yang sangat besar.
- Dataframe dapat dianggap seperti tabel di program spreadsheet seperti Microsoft Excel, tetapi memiliki banyak fitur yang lebih canggih dan berguna untuk melakukan manipulasi dan analisis data.
- Dataframe merupakan salah satu fitur utama dari library Pandas di Python, dan sangat berguna untuk mengelola dan menganalisis data.
- Contoh Dataset sederhana

Nama	Domisili	Pekerjaan
Tiara	Tangerang Selatan	Guru
Indah	Bekasi	Dokter
Tyo	Semarang	Dosen



Struktur Dataframe

Struktur dataframe terdiri dari 3 komponen utama:

- Indeks

Indeks pada dataframe digunakan untuk mengidentifikasi setiap baris dalam DataFrame. Setiap baris memiliki indeks yang unik. Kita bisa menggunakan indeks ini untuk mengakses atau memanipulasi data pada baris tersebut.

- Kolom

kolom pada dataframe digunakan untuk mengidentifikasi setiap kolom dalam DataFrame.

- Nilai

nilai di dataframe merupakan data sebenarnya

2. Why Dataframe?

Kenapa harus dataframe?

- Keuntungan utama dari dataframe adalah efisiensi dalam memanipulasi dan menganalisis data.
- Dengan DataFrame, kita dapat memfilter data
- Dengan DataFrame dapat melakukan operasi aritmatika, melakukan agregasi data, dan banyak lainnya.
- Mempermudah perhitungan statistika, salah satunya dengan fungsi describe()
- Menggabungkan data dari berbagai sumber menjadi satu dataframe.
- Menggunakan label pada baris dan kolom data untuk memudahkan pengindeksan dan slicing data.

3. Creating Dataframe

Dataframe Pandas

Pandas adalah library di Python yang dipakai untuk bekerja dengan DataFrame. Melalui Pandas, kita bisa lebih mudah memanipulasi, mengorganisir, dan membersihkan data.



Dataframe Pandas

Pandas menyediakan beragam kemampuan manipulasi data, termasuk memfilter, mengelompokkan, membentuk ulang, dan menggabungkan data. Ini mencakup fungsi untuk menangani data yang hilang dan bekerja dengan data deret waktu. Selain itu, Pandas menawarkan berbagai fungsi statistik, seperti mean, median, deviasi standar, dan analisis korelasi, yang memungkinkan untuk mengekstrak wawasan berharga dari data.



Dataframe Pandas

Pandas memiliki 2 struktur data:

1. Series adalah array berlabel satu dimensi yang dapat menampung data jenis apa pun, seperti bilangan bulat, string, atau objek Python lainnya.
2. Dataframe adalah struktur data berlabel dua dimensi yang terdiri dari baris dan kolom, seperti spreadsheet atau tabel SQL. Ini menyediakan cara yang nyaman dan efisien untuk menyimpan dan memanipulasi data tabular, menjadikannya pilihan populer untuk tugas analisis data.

```
10    g
11    e
12    e
13    k
14    s
dtype: object
```

Contoh
series

```
      calories  duration
0          420         50
1          380         40
2          390         45
```

Contoh
dataframe

Create a DataFrame Using Dictionary Nddarray/Lists

Untuk membuat Pandas DataFrame dari dictionary ndarray/list, semua ndarray harus memiliki panjang yang sama. Jika indeks Data dilewatkan maka indeks panjangnya harus sama dengan panjang array. Jika tidak ada indeks yang diteruskan, secara default indeks akan menjadi rentang(n) dengan n adalah panjang array

✓ Create a DataFrame Using Dictionary Nddarray/Lists

```
✓ [1] import pandas as pd  
0s user = {  
    'name' : ["Budi", "Candra", "Melati", "Diana"],  
    'domisili' : ["Tangerang Selatan", "Bekasi", "Bandung", "Solo"],  
    'pekerjaan' : ["Guru", "Dosen", "Pilot", "Data"]  
}  
  
df = pd.DataFrame(user)  
print(df)
```

	name	domisili	pekerjaan
0	Budi	Tangerang Selatan	Guru
1	Candra	Bekasi	Dosen
2	Melati	Bandung	Pilot
3	Diana	Solo	Data

Using List of Dictionaries

Kita dapat membuat Pandas DataFrame dengan berbagai cara dengan memuat kumpulan data dari penyimpanan yang ada, penyimpanan dapat berupa file Excel, file CSV, dan Database SQL. Pandas DataFrame dapat dibuat dari daftar, kamus, dan dari daftar kamus, dll

✓ Using List of Dictionaries

```
✓  
0s [3] import pandas as pd  
data = [{'a = 1', 'b = 2', 'c = 3'}, {'a = 5', 'b = 6', 'c = 7'}]  
ds = pd.DataFrame(data)  
print(ds)
```

	0	1	2
0	b = 2	c = 3	a = 1
1	a = 5	c = 7	b = 6

Using 2D List

Kita dapat membuat DataFrame menggunakan daftar 2 dimensi.

✓ Using 2D List

```
✓ [4] import pandas as pd  
0s    person_ = [['Fatimah', 28], ['Bryan', 19], ['Rike', 20]]  
      dd = pd.DataFrame(person_, columns= ['Name', 'Age'])  
      print(dd)
```

	Name	Age
0	Fatimah	28
1	Bryan	19
2	Rike	20

Using Indexes

Kita dapat membuat data dalam argumen indeks.



The image shows a Jupyter Notebook interface. On the left is a sidebar with icons for search, a variable {x}, a key (representing a cell), a folder, and a table. The main area is titled 'Using Indexes' with a dropdown arrow. Below the title is a code cell with a play button icon, a green checkmark, and '0s' indicating execution time. The code defines a pandas DataFrame with three rows and two columns. Below the code is the output, which is a table with three rows and two columns.

```
import pandas as pd
tenant_ = {'Name' : ['Fatimah', 'Bryan', 'Rike'], 'Umur' : [28, 19, 20]}
de = pd.DataFrame(tenant_, index = [1, 2, 3])
print(de)
```

	Name	Umur
1	Fatimah	28
2	Bryan	19
3	Rike	20

4. Selecting DataFrame

Pick & Choose your Data

Slide sebelumnya kita telah memuat data kita ke dalam DataFrame. Pada bagian ini kita akan mempelajari memilih data, seperti:

1. Selecting Columns
2. Selecting Rows

Selecting Columns

Memilih kolom yang dibutuhkan

{x}

✓
0s

```
[8] # 2. Use it as an index to the DataFrame  
df[columns_to_be_selected]
```



	name	device	gender
0	Budi	android	L
1	Candra	android	L
2	Melati	iOS	P
3	Diana	android	P



Selecting Rows

Berbeda dengan kolom, DataFrame kita saat ini tidak memiliki label yang dapat kita gunakan untuk merujuk data baris. Namun seperti array, DataFrame menyediakan pengindeksan numerik (0, 1, 2...) secara default. Pengindeksan bisa menggunakan `iloc`.

```
✓ [10] # 1. using numerical indexes - iloc  
0s df.iloc[0:3, :]
```

	name	domisili	pekerjaan	Age	device	gender	birthday
0	Budi	Tangerang Selatan	Guru	27	android	L	1997-01-06
1	Candra	Bekasi	Dosen	30	android	L	1993-09-08
2	Melati	Bandung	Pilot	35	iOS	P	1988-10-09

Selecting Rows

Pengindeksan bisa menggunakan loc

✓
0s

```
# 2. using labels as index - loc  
row_index_to_select = [2,3]  
df.loc[row_index_to_select]
```



	name	domisili	pekerjaan	Age	device	gender	birthday
2	Melati	Bandung	Pilot	35	iOS	P	1988-10-09
3	Diana	Solo	Data	43	android	P	1980-12-01



|



0s

completed at 13:24

Selecting Rows

Jika kita ingin mengambil kondisi tertentu pada suatu tabel.
`df.loc[df['column name'] condition]`

✓
0s



```
# specified conditions  
df.loc[df['device'] == "iOS"]
```



	name	domisili	pekerjaan	Age	device	gender	birthday
2	Melati	Bandung	Pilot	35	iOS	P	1988-10-09



Selecting Rows

Jika kita ingin mengambil kondisi tertentu pada suatu tabel, seperti sama atau lebih besar atau lebih kecil.

`df.loc[df['column name'] condition]`

```
0s # specified conditions
df.loc[df['device'] == "iOS"]
```

	name	domisili	pekerjaan	Age	device	gender	birthday
2	Melati	Bandung	Pilot	35	iOS	P	1988-10-09

```
0s # specified conditions
df.loc[(df['device'] == "android") & (df['gender'] == 'P')]
```

	name	domisili	pekerjaan	Age	device	gender	birthday
3	Diana	Solo	Data	43	android	P	1980-12-01

```
0s [14] # specified conditions
df.loc[df['Age'] <= 30]
```

	name	domisili	pekerjaan	Age	device	gender	birthday
0	Budi	Tangerang Selatan	Guru	27	android	L	1997-01-06
1	Candra	Bekasi	Dosen	30	android	L	1993-09-08

```
0s [16] # specified conditions
df.loc[(df['Age'] <= 30) | (df['Age'] > 40)]
```

	name	domisili	pekerjaan	Age	device	gender	birthday
0	Budi	Tangerang Selatan	Guru	27	android	L	1997-01-06
1	Candra	Bekasi	Dosen	30	android	L	1993-09-08
3	Diana	Solo	Data	43	android	P	1980-12-01

```
0s [18] # specified conditions
df.loc[df['gender'] != 'L']
```

	name	domisili	pekerjaan	Age	device	gender	birthday
2	Melati	Bandung	Pilot	35	iOS	P	1988-10-09
3	Diana	Solo	Data	43	android	P	1980-12-01

Selecting Column and Rows

Jika kita ingin mengambil kondisi tertentu pada suatu tabel yaitu row dan kolom khusus bisa menggunakan loc dan memilih kolom tertentu

✓
1s



```
df.loc[df['Age'] > 40, ["name", "Age"]]
```

	name	Age
3	Diana	43



Summary

1. Saat memilih subkumpulan data, tanda kurung siku [] digunakan.
2. Di dalam brackets tanda kurung ini, Anda dapat menggunakan label kolom/baris tunggal, daftar label kolom/baris, potongan label, ekspresi kondisional, atau titik dua.
3. Pilih baris dan/atau kolom tertentu menggunakan loc saat menggunakan nama baris dan kolom.
4. Pilih baris dan/atau kolom tertentu menggunakan iloc saat menggunakan posisi dalam tabel.

4. Read/Write DataFrame

Write Dataframe

Kita dapat write data file CSV

Sintaks untuk menulis ke file akan tetap sama tetapi kita akan menggunakan mode tulis

✓ write

```
✓ [24] import csv
0s      with open('user.csv', 'w') as user_:
          writer = csv.writer(user_, delimiter=',')
          writer.writerow(['Name', 'Domisili', 'Pekerjaan', 'Age', 'Device', 'Gender', 'Birthday'])
          writer.writerow(['Diana', 'Tangerang Selatan', 'Dokter', 30, 'android', 'P', '1993-09-01'])
          writer.writerow(['Karel', 'Bekasi', 'Dosen', 28, 'iOS', 'L', '1995-08-01'])

          writer.writerow(['Putri', 'Bandung', 'Dokter', 39, 'iOS', 'P', '1984-10-08'])
          writer.writerow(['Dina', 'Solo', 'Perawat', 25, 'iOS', 'P', '1998-07-05'])
```

Read Data

Pandas menyediakan fungsi membaca dataset yaitu `read_*` seperti (csv, excel, sql, json,dst..).

✓ read

✓
0s

```
[25] import pandas as pd  
      df = pd.read_csv('user.csv')  
      df
```

	Name	Domisili	Pekerjaan	Age	Device	Gender	Birthday
0	Diana	Tangerang Selatan	Dokter	30	android	P	1993-09-01
1	Karel	Bekasi	Dosen	28	iOS	L	1995-08-01
2	Putri	Bandung	Dokter	39	iOS	P	1984-10-08
3	Dina	Solo	Perawat	25	iOS	P	1998-07-05



NEXT!!

Lanjutan dataframe



Terima Kasih

