


Nama: Mohammad Afif R. Linkeh NIM: 065002400002	 Praktikum Statistika	MODUL 3 Nama Dosen: Drs. Joko Riyono, M.Si.
Hari/Tanggal: Jumat, 21 Maret 2025		Nama Asisten Labratorium: 1. Kharisma Maulida Saara 064002200024 2. Tarum Widyasti Pertiwi 064002200027

Pengelolaan Data pada Data Frame

1. Teori Singkat

Pengelolaan data pada DataFrame merupakan konsep penting dalam analisis data menggunakan Python, terutama dengan library seperti pandas. Berikut adalah beberapa teori terkait pengelolaan data pada DataFrame:

- **DataFrame:** DataFrame adalah struktur data dua dimensi yang digunakan untuk menyimpan dan mengelola data dalam bentuk tabel yang terorganisir. Setiap kolom dalam DataFrame mewakili sebuah variabel, sedangkan setiap baris mewakili sebuah catatan atau observasi.
- **Kolom dan Baris:** Kolom dalam DataFrame mewakili variabel, sementara baris mewakili catatan atau observasi. Kolom dan baris dapat diakses menggunakan label atau indeks numerik.
- **Pengindeksan:** Pengindeksan memungkinkan akses ke bagian tertentu dari DataFrame. Anda dapat mengakses kolom menggunakan nama kolom, sedangkan baris dapat diakses menggunakan indeks numerik atau label baris.
- **Manipulasi Data:** Anda dapat melakukan berbagai operasi manipulasi data pada DataFrame, seperti menambah atau menghapus kolom dan baris, mengubah nilai, menggabungkan DataFrame, mengurutkan data, serta melakukan operasi agregasi seperti penghitungan rata-rata atau jumlah.

- **Pembersihan Data:** Pembersihan data adalah proses untuk menghapus atau memperbaiki data yang tidak valid, hilang, atau tidak lengkap. Ini melibatkan deteksi dan penanganan nilai yang hilang, duplikat, atau outlier.
- **Pengindeksan dan Pemotongan:** Pengindeksan dan pemotongan memungkinkan Anda untuk memilih subset dari DataFrame berdasarkan kriteria tertentu. Anda dapat menggunakan metode seperti `.loc[]`, `.iloc[]`, dan metode pemotongan untuk melakukan ini.
- **Pengurutan Data:** Pengurutan data memungkinkan Anda untuk mengurutkan DataFrame berdasarkan nilai pada satu atau beberapa kolom. Hal ini mempermudah analisis dan pemrosesan data yang memerlukan data yang diurutkan.
- **Operasi Statistik:** Anda dapat melakukan berbagai operasi statistik pada DataFrame, seperti penghitungan rata-rata, median, standar deviasi, korelasi, dan lainnya. Library seperti pandas menyediakan fungsi bawaan untuk melakukan operasi ini dengan mudah.
- **Menggabungkan DataFrame:** Anda dapat menggabungkan dua atau lebih DataFrame berdasarkan kunci tertentu menggunakan metode seperti `.merge()` atau `.concat()`. Ini berguna saat Anda perlu menggabungkan data dari berbagai sumber.
- **Ekspor dan Impor Data:** Anda dapat mengimpor data dari berbagai sumber seperti file CSV, Excel, SQL database, atau web API ke dalam DataFrame. Anda juga dapat mengekspor DataFrame ke berbagai format file untuk berbagi atau menyimpan data.

Pengelolaan data pada DataFrame sangat penting dalam analisis data menggunakan Python, karena memungkinkan Anda untuk menyusun, membersihkan, memanipulasi, dan menganalisis data dengan efisien. Dengan menggunakan library seperti pandas, Anda dapat dengan mudah mengelola data dalam format tabular dan menjalankan berbagai analisis data dengan Python.

2. Alat dan Bahan

Hardware : Laptop/PC
Software: Jupyter Notebook

3. Elemen Kompetensi

a. Latihan pertama – Merge Data

1. Jalankan source code berikut. Ganti nama variable (seperti `data_nama`) menjadi variable dengan nama kalian masing - masing. Data yang digunakan adalah data `houseprice.csv`. Melakukan Read CSV dengan cara menginput data `houseprice`, sesuaikan dengan lokasi direktori dimana kalian menyimpan file csvnya.

```
import pandas as pd

data_nama = pd.read_clipboard()

# Menampilkan data
print(data_nama)
```

Output:

#JANGANLUPAMENGGANTIOUTPUTMENJADIHASILSENDIRI

```
[1]: import pandas as pd

data_afif = pd.read_clipboard()

# Menampilkan data
print(data_afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	2	2	No	East
1	114200	2030	4	2	3	No	East
2	114800	1740	3	2	1	No	East
3	94700	1980	3	2	3	No	East
4	119800	2130	3	3	3	No	East
..
123	119700	1900	3	3	3	Yes	East
124	147900	2160	4	3	3	Yes	East
125	113500	2070	2	2	2	No	North
126	149900	2020	3	3	1	No	West
127	124600	2250	3	3	4	No	North

[128 rows x 7 columns]

2. Lakukan subset data dengan cara sub set data frame khusus yang memiliki nilai variabel Bedrooms =2

```
nama = data_nama[data_nama['Bedrooms'] == 2]

# Menampilkan nama
print(nama)
```

Output:

```
[5]: afif = data_afif[data_afif['Bedrooms'] == 2]

# Menampilkan nama
print(afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	2	2	No	East
11	123000	1870	2	2	2	Yes	East
18	111400	1700	2	2	1	Yes	East
28	69100	1600	2	2	3	No	North
31	112300	1930	2	2	2	Yes	North
34	117800	2000	2	2	3	No	North
36	117500	1880	2	2	2	No	North
40	106600	1560	2	2	1	No	East
42	105600	1990	2	2	3	No	East
46	129800	1990	2	3	2	No	North
48	115900	1980	2	2	2	No	East
51	91100	1860	2	2	3	No	North
52	117400	2150	2	3	4	No	North
55	125700	1720	2	2	2	Yes	East
61	100900	1610	2	2	2	No	North
63	120500	1910	2	3	2	No	East
65	111100	1450	2	2	1	Yes	North
75	106900	1900	2	2	2	No	North
82	164800	2050	2	2	1	Yes	West
84	90500	1520	2	2	3	No	North
89	97800	2010	2	2	4	No	North
95	152500	1970	2	2	1	Yes	West
97	126800	2000	2	2	1	Yes	East
110	111600	1710	2	2	1	No	North
111	114900	1740	2	2	2	No	North
112	123600	1940	2	2	2	Yes	East
119	109700	1920	2	2	4	No	North
120	110400	1930	2	3	3	No	North
122	144800	2060	2	2	1	Yes	East
125	113500	2070	2	2	2	No	North

3. Selanjutnya ialah mengubah nama variabel. Berikut adalah tahapan untuk mengubah nilai dalam kolom Bathrooms dalam kondisi jika Bathrooms >2 , maka akan diganti dengan nilai large, jika tidak maka small.

```
nama['Bathrooms'] = pd.to_numeric(nama['Bathrooms'])

import numpy as np

nama['Bathrooms'] = nama['Bathrooms'].apply(lambda x: 'large' if x > 2 else 'small')

# Menampilkan DataFrame setelah modifikasi
print(nama)
```

Output:

```
[6]: afif['Bathrooms'] = pd.to_numeric(afif['Bathrooms'])

import numpy as np

afif['Bathrooms'] = afif['Bathrooms'].apply(lambda x: 'large' if x > 2 else 'small')

# Menampilkan DataFrame setelah modifikasi
print(afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	small	2	No	East
11	123000	1870	2	small	2	Yes	East
18	111400	1700	2	small	1	Yes	East
28	69100	1600	2	small	3	No	North
31	112300	1930	2	small	2	Yes	North
34	117800	2000	2	small	3	No	North
36	117500	1880	2	small	2	No	North
40	106600	1560	2	small	1	No	East
42	105600	1990	2	small	3	No	East
46	129800	1990	2	large	2	No	North
48	115900	1980	2	small	2	No	East
51	91100	1860	2	small	3	No	North
52	117400	2150	2	large	4	No	North
55	125700	1720	2	small	2	Yes	East
61	100900	1610	2	small	2	No	North
63	120500	1910	2	large	2	No	East
65	111100	1450	2	small	1	Yes	North
75	106900	1900	2	small	2	No	North
82	164800	2050	2	small	1	Yes	West
84	90500	1520	2	small	3	No	North
89	97800	2010	2	small	4	No	North
95	152500	1970	2	small	1	Yes	West
97	126800	2000	2	small	1	Yes	East
110	111600	1710	2	small	1	No	North
111	114900	1740	2	small	2	No	North
112	123600	1940	2	small	2	Yes	East
119	109700	1920	2	small	4	No	North
120	110400	1930	2	large	3	No	North
122	144800	2060	2	small	1	Yes	East
125	113500	2070	2	small	2	No	North

4. Lalu langkah selanjutnya adalah menambah variabel. Tahap 1 □ Membuat variable baru dari dataku1 dengan nilai sesuai kondisi ifelse yang ditentukan. Sesuaikan nama variable baru dengan nama praktikan.

```
import numpy as np

nama['newvariable'] = np.where(nama['Offers'] > 2, 'large', 'small')

# Menampilkan DataFrame 'nama' setelah penambahan kolom baru
print(nama)
```

Output

```
[8]: import numpy as np

afif['newvariable'] = np.where(afif['Offers'] > 2, 'large', 'small')

# Menampilkan DataFrame 'nama' setelah penambahan kolom baru
print(afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood	newvariable
0	114300	1790	2	small	2	No	East	small
11	123000	1870	2	small	2	Yes	East	small
18	111400	1700	2	small	1	Yes	East	small
28	69100	1600	2	small	3	No	North	large
31	112300	1930	2	small	2	Yes	North	small
34	117800	2000	2	small	3	No	North	large
36	117500	1880	2	small	2	No	North	small
40	106600	1560	2	small	1	No	East	small
42	105600	1990	2	small	3	No	East	large
46	129800	1990	2	large	2	No	North	small
48	115900	1980	2	small	2	No	East	small
51	91100	1860	2	small	3	No	North	large
52	117400	2150	2	large	4	No	North	large
55	125700	1720	2	small	2	Yes	East	small
61	100900	1610	2	small	2	No	North	small
63	120500	1910	2	large	2	No	East	small
65	111100	1450	2	small	1	Yes	North	small
75	106900	1900	2	small	2	No	North	small
82	164800	2050	2	small	1	Yes	West	small
84	90500	1520	2	small	3	No	North	large
89	97800	2010	2	small	4	No	North	large
95	152500	1970	2	small	1	Yes	West	small
97	126800	2000	2	small	1	Yes	East	small
110	111600	1710	2	small	1	No	North	small
111	114900	1740	2	small	2	No	North	small
112	123600	1940	2	small	2	Yes	East	small
119	109700	1920	2	small	4	No	North	large
120	110400	1930	2	large	3	No	North	large
122	144800	2060	2	small	1	Yes	East	small
125	113500	2070	2	small	2	No	North	small

Tahap 2 □ Mengubah isi baris data dari kolom baru yang telah dibuat

```
# Menambahkan kolom baru 'newvariable'
nama['newvariable'] = nama['Price'] / nama['SqFt']

# Menampilkan DataFrame 'nama' setelah penambahan kolom baru
print(nama)
```

Output:

```
[9]: # Menambahkan kolom baru 'newvariable'
      afif['newvariable'] = afif['Price'] / afif['SqFt']

      # Menampilkan DataFrame 'nama' setelah penambahan kolom baru
      print(afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood	newvariable
0	114300	1790	2	small	2	No	East	63.854749
11	123000	1870	2	small	2	Yes	East	65.775401
18	111400	1700	2	small	1	Yes	East	65.529412
28	69100	1600	2	small	3	No	North	43.187500
31	112300	1930	2	small	2	Yes	North	58.186528
34	117800	2000	2	small	3	No	North	58.900000
36	117500	1880	2	small	2	No	North	62.500000
40	106600	1560	2	small	1	No	East	68.333333
42	105600	1990	2	small	3	No	East	53.065327
46	129800	1990	2	large	2	No	North	65.226131
48	115900	1980	2	small	2	No	East	58.535354
51	91100	1860	2	small	3	No	North	48.978495
52	117400	2150	2	large	4	No	North	54.604651
55	125700	1720	2	small	2	Yes	East	73.081395
61	100900	1610	2	small	2	No	North	62.670807
63	120500	1910	2	large	2	No	East	63.089005
65	111100	1450	2	small	1	Yes	North	76.620690
75	106900	1900	2	small	2	No	North	56.263158
82	164800	2050	2	small	1	Yes	West	80.390244
84	90500	1520	2	small	3	No	North	59.539474
89	97800	2010	2	small	4	No	North	48.656716
95	152500	1970	2	small	1	Yes	West	77.411168
97	126800	2000	2	small	1	Yes	East	63.400000
110	111600	1710	2	small	1	No	North	65.263158
111	114900	1740	2	small	2	No	North	66.034483
112	123600	1940	2	small	2	Yes	East	63.711340
119	109700	1920	2	small	4	No	North	57.135417
120	110400	1930	2	large	3	No	North	57.202073
122	144800	2060	2	small	1	Yes	East	70.291262
125	113500	2070	2	small	2	No	North	54.830918

5. Delete Variabel. Selain bisa menambah, kita juga bisa menghapus variable. Dalam percobaan ini kita akan menghapus variable yang baru saja kita buat.

```
nama = nama.drop(columns=['newvariable'])

# Menampilkan DataFrame 'nama'
print(nama)
```

Output:

```
[10]: afif = afif.drop(columns=['newvariable'])

# Menampilkan DataFrame 'nama'
print(afif)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	small	2	No	East
11	123000	1870	2	small	2	Yes	East
18	111400	1700	2	small	1	Yes	East
28	69100	1600	2	small	3	No	North
31	112300	1930	2	small	2	Yes	North
34	117800	2000	2	small	3	No	North
36	117500	1880	2	small	2	No	North
40	106600	1560	2	small	1	No	East
42	106600	1990	2	small	3	No	East
46	129800	1990	2	large	2	No	North
48	115900	1980	2	small	2	No	East
51	91100	1860	2	small	3	No	North
52	117400	2150	2	large	4	No	North
55	125700	1720	2	small	2	Yes	East
61	100900	1610	2	small	2	No	North
63	120500	1910	2	large	2	No	East
65	111100	1450	2	small	1	Yes	North
75	106900	1900	2	small	2	No	North
82	164800	2050	2	small	1	Yes	West
84	90500	1520	2	small	3	No	North
89	97800	2010	2	small	4	No	North
95	152500	1970	2	small	1	Yes	West
97	126800	2000	2	small	1	Yes	East
110	111600	1710	2	small	1	No	North
111	114900	1740	2	small	2	No	North
112	123600	1940	2	small	2	Yes	East
119	109700	1920	2	small	4	No	North
120	110400	1930	2	large	3	No	North
122	144800	2060	2	small	1	Yes	East
125	113500	2070	2	small	2	No	North

6. Merge Data Frame. Tahap 1 □ Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

```
kolom1dan2 = data_nama.iloc[:, 0:2]

# Menampilkan DataFrame kolom1dan2
print(kolom1dan2)
```

Output:

```
[11]: kolom1dan2 = data_afif.iloc[:, 0:2]

# Menampilkan DataFrame kolom1dan2
print(kolom1dan2)
```

	Price	SqFt
0	114300	1790
1	114200	2030
2	114800	1740
3	94700	1980
4	119800	2130
..
123	119700	1900
124	147900	2160
125	113500	2070
126	149900	2020
127	124600	2250

```
[128 rows x 2 columns]
```

Tahap 2 □ Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

```
# Memilih kolom 1 dan 2 dari DataFrame data_nama  
kolom3dan4 = data_nama.iloc[:, 2:4]  
  
# Menampilkan DataFrame kolom3dan4  
print(kolom3dan4)
```

Output:

```
[12]: # Memilih kolom 1 dan 2 dari DataFrame data_nama  
kolom3dan4 = data_afif.iloc[:, 2:4]  
  
# Menampilkan DataFrame kolom3dan4  
print(kolom3dan4)
```

	Bedrooms	Bathrooms
0	2	2
1	4	2
2	3	2
3	3	2
4	3	3
..
123	3	3
124	4	3
125	2	2
126	3	3
127	3	3

[128 rows x 2 columns]

Tahap 3 □ Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

```
# Menggabungkan dua DataFrame
kolom1sd4 = pd.concat([kolom1dan2, kolom3dan4], axis=1)

# Menampilkan DataFrame kolom1sd4
print(kolom1sd4)
```

Output:

```
[13]: # Menggabungkan dua DataFrame
kolom1sd4 = pd.concat([kolom1dan2, kolom3dan4], axis=1)

# Menampilkan DataFrame kolom1sd4
print(kolom1sd4)
```

	Price	SqFt	Bedrooms	Bathrooms
0	114300	1790	2	2
1	114200	2030	4	2
2	114800	1740	3	2
3	94700	1980	3	2
4	119800	2130	3	3
..
123	119700	1900	3	3
124	147900	2160	4	3
125	113500	2070	2	2
126	149900	2020	3	3
127	124600	2250	3	3

[128 rows x 4 columns]

7. Merge Baris artinya memisahkan data sesuai baris yang diinginkan dengan menggunakan range baris. Lalu kemudian digabungkan.

```
# Menggabungkan baris dari dua DataFrame
baris1sd3 = data_nama.iloc[0:3, :]
baris4sd6 = data_nama.iloc[3:6, :]
baris1sd6 = baris1sd3.append(baris4sd6)

# Menampilkan DataFrame baris1sd6
print(baris1sd6)
```

Output

```
[16]: import pandas as pd

# Menggabungkan baris dari dua DataFrame
baris1sd3 = data_afif.iloc[0:3, :]
baris4sd6 = data_afif.iloc[3:6, :]

# Gunakan pd.concat() untuk menggabungkan DataFrame
baris1sd6 = pd.concat([baris1sd3, baris4sd6])

# Menampilkan DataFrame baris1sd6
print(baris1sd6)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	2	2	No	East
1	114200	2030	4	2	3	No	East
2	114800	1740	3	2	1	No	East
3	94700	1980	3	2	3	No	East
4	119800	2130	3	3	3	No	East
5	114600	1780	3	2	2	No	North

8. Sort data frame. Apa yang terjadi dengan data setelah di sort?

Jawaban: ?

```
data_nama_sort = data_nama.sort_values(by='Price')  
  
print(data_nama_sort)
```

Output:

```
[17]: data_afif_sort = data_afif.sort_values(by='Price')  
  
print(data_afif_sort)
```

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
28	69100	1600	2	2	3	No	North
54	81300	1650	3	2	3	No	North
104	82300	1910	3	2	4	No	East
17	83600	1990	3	3	4	No	North
47	90300	2050	3	2	6	No	North
..
81	184300	2140	4	3	2	Yes	West
29	188000	2040	4	3	1	Yes	West
85	188300	2250	4	3	2	Yes	West
116	199500	2290	5	4	1	Yes	West
103	211200	2440	4	3	3	Yes	West

[128 rows x 7 columns]

b. Latihan Kedua – Tugas

Gunakan dataset pada tugas 1 yang telah ditambah lagi datanya dengan 10 mhs TIF/SI. Sehingga total baris data pada file tersebut berjumlah 30. Ulangi kembali perintah-perintah di atas dan sesuaikan dengan data anda. Dan lampirkan Screen Capture untuk tiap poin yang ada.

1. Read CSV

```
import pandas as pd

datapath = "C:/Users/moham/Documents/prak s2/prob/Buku2.csv"
data = pd.read_csv(datapath)
print(data)
```

Output:

```
[3]: import pandas as pd

datapath = "C:/Users/moham/Documents/prak s2/prob/Buku2.csv"
data = pd.read_csv(datapath)
print(data)
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal
0	afif	pria	24	167	7	jakbar
1	rafael	pria	24	168	10	jakbar
2	henry	pria	24	169	40	jaktim
3	bagas	pria	24	168	45	jaktim
4	faiz	pria	24	167	55	jaktim
5	stepi	pria	24	165	15	grogol
6	ridhwan	pria	24	163	90	jakut
7	rafli	pria	24	163	120	tangerang
8	shalwa	wanita	24	157	30	jakut
9	aisha	wanita	24	156	35	jaktim
10	hana	wanita	24	155	5	jakbar
11	sonia	wanita	24	157	7	jakbar
12	hasna	wanita	24	156	8	jakbar
13	charin	wanita	24	155	35	jaktim
14	dewi	wanita	24	167	60	bekasi
15	amba	pria	24	150	90	bogor
16	udin	pria	24	155	35	jakbar
17	meilin	wanita	24	150	20	jakpus
18	rusdi	pria	24	170	5	jakbar
19	amanda	wanita	24	166	40	jakut
20	rizki	pria	24	167	40	jakut
21	dedek	pria	24	170	20	jakpus
22	mawan	pria	24	165	100	jaksel
23	zowie	wanita	24	159	120	depok
24	ixa	wanita	24	155	90	tangerang
25	furqon	pria	24	154	45	jakbar
26	kevin	pria	24	167	15	jakpus
27	lam	pria	24	169	25	jakbar
28	yuni	wanita	24	170	30	bekasi
29	asgar	pria	24	177	3	grogol

2. Subset Data □ Gunakan Kolom Tinggi Badan untuk bagian ini

```
subset = data[data['tinggi'] == 155]  
print(subset)
```

Output:

```
4]: subset = data[data['tinggi'] == 155]  
print(subset)
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal
10	hana	wanita	24	155	5	jakbar
13	charin	wanita	24	155	35	jaktim
16	udin	pria	24	155	35	jakbar
24	ixa	wanita	24	155	90	tangerang

```
]:
```

3. Mengubah nilai suatu variable □ Ubah isi kolom tinggi badan untuk tinggi > 160 menjadi “Tinggi” dan jika bukan berarti “Pendek”

```
import numpy as np  
  
data['tinggi'] = pd.to_numeric(data['tinggi'])  
  
data['tinggi'] = data['tinggi'].apply(lambda x : 'Tinggi' if x>160 else 'Pendek')  
  
print(data)
```

Output:

```
[5]: import numpy as np

data['tinggi'] = pd.to_numeric(data['tinggi'])

data['tinggi'] = data['tinggi'].apply(lambda x : 'Tinggi' if x>160 else 'Pendek')

print(data)
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal
0	afif	pria	24	Tinggi	7	jakbar
1	rafael	pria	24	Tinggi	10	jakbar
2	henry	pria	24	Tinggi	40	jaktim
3	bagas	pria	24	Tinggi	45	jaktim
4	faiz	pria	24	Tinggi	55	jaktim
5	stepi	pria	24	Tinggi	15	grogol
6	ridhwan	pria	24	Tinggi	90	jakut
7	rafli	pria	24	Tinggi	120	tangerang
8	shalwa	wanita	24	Pendek	30	jakut
9	aisha	wanita	24	Pendek	35	jaktim
10	hana	wanita	24	Pendek	5	jakbar
11	sonia	wanita	24	Pendek	7	jakbar
12	hasna	wanita	24	Pendek	8	jakbar
13	charin	wanita	24	Pendek	35	jaktim
14	dewi	wanita	24	Tinggi	60	bekasi
15	amba	pria	24	Pendek	90	bogor
16	udin	pria	24	Pendek	35	jakbar
17	meilin	wanita	24	Pendek	20	jakpus
18	rusdi	pria	24	Tinggi	5	jakbar
19	amanda	wanita	24	Tinggi	40	jakut
20	rizki	pria	24	Tinggi	40	jakut
21	dedek	pria	24	Tinggi	20	jakpus
22	mawan	pria	24	Tinggi	100	jaksel
23	zowie	wanita	24	Pendek	120	depok
24	ixa	wanita	24	Pendek	90	tangerang
25	furqon	pria	24	Pendek	45	jakbar
26	kevin	pria	24	Tinggi	15	jakpus
27	lam	pria	24	Tinggi	25	jakbar
28	yuni	wanita	24	Tinggi	30	bekasi
29	asgar	pria	24	Tinggi	3	grogol

```
[ ]:
```

4. Menambah Variabel

- Buat variabel kolom baru dengan nama “Jurusan” dengan isi baris datanya adalah “Infor20”

```
data['jurusan'] = 'Infor24'
data['Fakultas'] = 'FTI'
```

```
# Menampilkan DataFrame 'nama' setelah penambahan kolom baru
data
```


- Buat variabel kolom baru kedua dengan nama “Fakultas” dengan isi baris datanya adalah “FTI”

Output:

```
[15]: data['jurusan'] = 'Infor24'
      data['Fakultas'] = 'FTI'

      # Menampilkan DataFrame 'nama' setelah penambahan kolom baru
      data
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal	jurusan	Fakultas
0	afif	pria	24	Tinggi	7	jakbar	Infor24	FTI
1	rafael	pria	24	Tinggi	10	jakbar	Infor24	FTI
2	henry	pria	24	Tinggi	40	jaktim	Infor24	FTI
3	bagas	pria	24	Tinggi	45	jaktim	Infor24	FTI
4	faiz	pria	24	Tinggi	55	jaktim	Infor24	FTI
5	stepi	pria	24	Tinggi	15	grogol	Infor24	FTI
6	ridhwan	pria	24	Tinggi	90	jakut	Infor24	FTI
7	raffi	pria	24	Tinggi	120	tangerang	Infor24	FTI
8	shalwa	wanita	24	Pendek	30	jakut	Infor24	FTI
9	aisha	wanita	24	Pendek	35	jaktim	Infor24	FTI
10	hana	wanita	24	Pendek	5	jakbar	Infor24	FTI
11	sonia	wanita	24	Pendek	7	jakbar	Infor24	FTI
12	hasna	wanita	24	Pendek	8	jakbar	Infor24	FTI
13	charin	wanita	24	Pendek	35	jaktim	Infor24	FTI
14	dewi	wanita	24	Tinggi	60	bekasi	Infor24	FTI
15	amba	pria	24	Pendek	90	bogor	Infor24	FTI
16	udin	pria	24	Pendek	35	jakbar	Infor24	FTI
17	meilin	wanita	24	Pendek	20	jakpus	Infor24	FTI
18	rusdi	pria	24	Tinggi	5	jakbar	Infor24	FTI
19	amanda	wanita	24	Tinggi	40	jakut	Infor24	FTI
20	rizki	pria	24	Tinggi	40	jakut	Infor24	FTI
21	dedek	pria	24	Tinggi	20	jakpus	Infor24	FTI
22	mawan	pria	24	Tinggi	100	jaksel	Infor24	FTI
23	zowie	wanita	24	Pendek	120	depok	Infor24	FTI
24	ixa	wanita	24	Pendek	90	tangerang	Infor24	FTI
25	furqon	pria	24	Pendek	45	jakbar	Infor24	FTI
26	kevin	pria	24	Tinggi	15	jakpus	Infor24	FTI
27	lam	pria	24	Tinggi	25	jakbar	Infor24	FTI
28	yuni	wanita	24	Tinggi	30	bekasi	Infor24	FTI
29	asgar	pria	24	Tinggi	3	grogol	Infor24	FTI

```
[ ]:
```

5. Delete Variabel □ Hapus kolom Fakultas

```
data = data.drop(columns=['Fakultas'])
```

```
data
```

Output:

```
[17]: data = data.drop(columns=['Fakultas'])
data
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal	jurusan
0	afif	pria	24	Tinggi	7	jakbar	Infor24
1	rafael	pria	24	Tinggi	10	jakbar	Infor24
2	henry	pria	24	Tinggi	40	jaktim	Infor24
3	bagas	pria	24	Tinggi	45	jaktim	Infor24
4	faiz	pria	24	Tinggi	55	jaktim	Infor24
5	stepi	pria	24	Tinggi	15	grogol	Infor24
6	ridhwan	pria	24	Tinggi	90	jakut	Infor24
7	rafii	pria	24	Tinggi	120	tangerang	Infor24
8	shalwa	wanita	24	Pendek	30	jakut	Infor24
9	aisha	wanita	24	Pendek	35	jaktim	Infor24
10	hana	wanita	24	Pendek	5	jakbar	Infor24
11	sonia	wanita	24	Pendek	7	jakbar	Infor24
12	hasna	wanita	24	Pendek	8	jakbar	Infor24
13	charin	wanita	24	Pendek	35	jaktim	Infor24
14	dewi	wanita	24	Tinggi	60	bekasi	Infor24
15	amba	pria	24	Pendek	90	bogor	Infor24
16	udin	pria	24	Pendek	35	jakbar	Infor24
17	meilin	wanita	24	Pendek	20	jakpus	Infor24
18	rusdi	pria	24	Tinggi	5	jakbar	Infor24
19	amanda	wanita	24	Tinggi	40	jakut	Infor24
20	rizki	pria	24	Tinggi	40	jakut	Infor24
21	dedek	pria	24	Tinggi	20	jakpus	Infor24
22	mawan	pria	24	Tinggi	100	jaksel	Infor24
23	zowie	wanita	24	Pendek	120	depok	Infor24
24	ixa	wanita	24	Pendek	90	tangerang	Infor24
25	furqon	pria	24	Pendek	45	jakbar	Infor24
26	kevin	pria	24	Tinggi	15	jakpus	Infor24
27	lam	pria	24	Tinggi	25	jakbar	Infor24
28	yuni	wanita	24	Tinggi	30	bekasi	Infor24
29	asgar	pria	24	Tinggi	3	grogol	Infor24

```
[ ]:
```

6. Merge Data Frame

- Gabung kolom Nama dan Gender
- Gabung kolom Angkatan dan Tinggi Badan
- Gabungkan 2 variabel kolom gabungan diatas

```
kolom1dan2 = data.iloc[:, 0:2]

# Menampilkan DataFrame kolom1dan2
print(kolom1dan2)

kolom3dan4 = data.iloc[:, 2:4]
print(kolom3dan4)

kolomgabung = pd.concat([kolom1dan2, kolom3dan4], axis=1)

print(kolomgabung)
```

Output:

```
[18]: kolom1dan2 = data.iloc[:, 0:2]
```

```
# Menampilkan DataFrame kolom1dan2
print(kolom1dan2)
```

	nana	gender
0	afif	pria
1	rafael	pria
2	henry	pria
3	bagas	pria
4	faiz	pria
5	stepi	pria
6	ridhwan	pria
7	rafli	pria
8	shalwa	wanita
9	aisha	wanita
10	hana	wanita
11	sonia	wanita
12	hasna	wanita
13	charin	wanita
14	dewi	wanita
15	amba	pria
16	udin	pria
17	neilin	wanita
18	rusdi	pria
19	amanda	wanita
20	rizki	pria
21	dedek	pria
22	mawan	pria
23	zowie	wanita
24	ixa	wanita
25	furqon	pria
26	kevin	pria
27	lam	pria
28	yuni	wanita
29	asgar	pria

```
[21]: kolom3dan4 = data.iloc[:, 2:4]
print(kolom3dan4)
```

	angkatan	tinggi
0	24	Tinggi
1	24	Tinggi
2	24	Tinggi
3	24	Tinggi
4	24	Tinggi
5	24	Tinggi
6	24	Tinggi
7	24	Tinggi
8	24	Pendek
9	24	Pendek
10	24	Pendek
11	24	Pendek
12	24	Pendek
13	24	Pendek
14	24	Tinggi
15	24	Pendek
16	24	Pendek
17	24	Pendek
18	24	Tinggi
19	24	Tinggi
20	24	Tinggi
21	24	Tinggi
22	24	Tinggi
23	24	Pendek
24	24	Pendek
25	24	Pendek
26	24	Tinggi
27	24	Tinggi
28	24	Tinggi
29	24	Tinggi

```
[22]: kolomgabung = pd.concat([kolom1dan2, kolom3dan4], axis=1)
print(kolomgabung)
```

	nana	gender	angkatan	tinggi
0	afif	pria	24	Tinggi
1	rafael	pria	24	Tinggi
2	henry	pria	24	Tinggi
3	bagas	pria	24	Tinggi
4	faiz	pria	24	Tinggi
5	stepi	pria	24	Tinggi
6	ridhwan	pria	24	Tinggi
7	rafli	pria	24	Tinggi
8	shalwa	wanita	24	Pendek
9	aisha	wanita	24	Pendek
10	hana	wanita	24	Pendek
11	sonia	wanita	24	Pendek
12	hasna	wanita	24	Pendek
13	charin	wanita	24	Pendek
14	dewi	wanita	24	Tinggi
15	amba	pria	24	Pendek
16	udin	pria	24	Pendek
17	neilin	wanita	24	Pendek
18	rusdi	pria	24	Tinggi
19	amanda	wanita	24	Tinggi
20	rizki	pria	24	Tinggi
21	dedek	pria	24	Tinggi
22	mawan	pria	24	Tinggi
23	zowie	wanita	24	Pendek
24	ixa	wanita	24	Pendek
25	furqon	pria	24	Pendek
26	kevin	pria	24	Tinggi
27	lam	pria	24	Tinggi
28	yuni	wanita	24	Tinggi
29	asgar	pria	24	Tinggi

7. Merge Baris

- Buat gabungan baris 1-5
- Buat gabungan baris 25-30
- Gabungkan 2 variabel diatas

```
gabungan_1_5 = data.iloc[0:5]
gabungan_25_30 = data.iloc[24:30]

gabungan_final = pd.concat([gabungan_1_5, gabungan_25_30])

gabungan_final
```

Output:

```
[23]: gabungan_1_5 = data.iloc[0:5]
      gabungan_25_30 = data.iloc[24:30]

      gabungan_final = pd.concat([gabungan_1_5, gabungan_25_30])

      gabungan_final
```

[23]:	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal	jurusan
0	afif	pria	24	Tinggi	7	jakbar	Infor24
1	rafael	pria	24	Tinggi	10	jakbar	Infor24
2	henry	pria	24	Tinggi	40	jaktim	Infor24
3	bagas	pria	24	Tinggi	45	jaktim	Infor24
4	faiz	pria	24	Tinggi	55	jaktim	Infor24
24	ixa	wanita	24	Pendek	90	tangerang	Infor24
25	furqon	pria	24	Pendek	45	jakbar	Infor24
26	kevin	pria	24	Tinggi	15	jakpus	Infor24
27	lam	pria	24	Tinggi	25	jakbar	Infor24
28	yuni	wanita	24	Tinggi	30	bekasi	Infor24
29	asgar	pria	24	Tinggi	3	grogol	Infor24

8. Sort Data Frame □ Lakukan sort berdasarkan waktu perjalanan

```
datasort = data.sort_values(by='waktu perjalanan ', ascending=True)

datasort
```

Output:

```
[26]: datasort = data.sort_values(by='waktu perjalanan ', ascending=True)
      datasort
```

```
[26]:
```

	nama	gender	angkatan	tinggi	waktu perjalanan	wilayah tinggal	jurusan
29	asgar	pria	24	Tinggi	3	grogol	Infor24
18	rusdi	pria	24	Tinggi	5	jakbar	Infor24
10	hana	wanita	24	Pendek	5	jakbar	Infor24
11	sonia	wanita	24	Pendek	7	jakbar	Infor24
0	afif	pria	24	Tinggi	7	jakbar	Infor24
12	hasna	wanita	24	Pendek	8	jakbar	Infor24
1	rafael	pria	24	Tinggi	10	jakbar	Infor24
5	stepi	pria	24	Tinggi	15	grogol	Infor24
26	kevin	pria	24	Tinggi	15	jakpus	Infor24
21	dedek	pria	24	Tinggi	20	jakpus	Infor24
17	meilin	wanita	24	Pendek	20	jakpus	Infor24
27	lam	pria	24	Tinggi	25	jakbar	Infor24
28	yuni	wanita	24	Tinggi	30	bekasi	Infor24
8	shalwa	wanita	24	Pendek	30	jakut	Infor24
9	aisha	wanita	24	Pendek	35	jaktim	Infor24
13	charin	wanita	24	Pendek	35	jaktim	Infor24
16	udin	pria	24	Pendek	35	jakbar	Infor24
2	henry	pria	24	Tinggi	40	jaktim	Infor24
19	amanda	wanita	24	Tinggi	40	jakut	Infor24
20	rizki	pria	24	Tinggi	40	jakut	Infor24
3	bagas	pria	24	Tinggi	45	jaktim	Infor24
25	furqon	pria	24	Pendek	45	jakbar	Infor24
4	faiz	pria	24	Tinggi	55	jaktim	Infor24
14	dewi	wanita	24	Tinggi	60	bekasi	Infor24
24	ixa	wanita	24	Pendek	90	tangerang	Infor24
6	ridhwan	pria	24	Tinggi	90	jakut	Infor24
15	amba	pria	24	Pendek	90	bogor	Infor24
22	mawan	pria	24	Tinggi	100	jaksel	Infor24
7	raffi	pria	24	Tinggi	120	tangerang	Infor24
23	zowie	wanita	24	Pendek	120	depok	Infor24

```
[ ]:
```

4. File Praktikum

Github Repository:

[GitHub - Afif-lingkeh/praktikum-probstat](https://github.com/Afif-lingkeh/praktikum-probstat)

5. Soal Latihan

Soal:

1. Apa saja kegunaan pengelolaan data pada data frame?
2. Dalam kasus apakah data perlu dihapus?

Jawaban:

1. Kegunaan pengelolaan data pada DataFrame :
 - Memudahkan manipulasi data seperti filtering, sorting, dan grouping.
 - Mempermudah analisis data dengan berbagai metode statistik dan visualisasi.
 - Mengubah format data seperti encoding, normalisasi, dan imputasi nilai yang hilang.
 - Memproses data untuk kebutuhan machine learning atau analisis lebih lanjut.
2. Kasus di mana data perlu dihapus :
 - Ketika terdapat banyak data yang hilang (missing values) dan tidak dapat diimputasi.
 - Jika ada data duplikat yang tidak diperlukan dalam analisis.
 - Ketika terdapat data yang tidak relevan dengan tujuan analisis.
 - Jika ada data yang mengandung outlier ekstrem yang dapat mengganggu hasil analisis.

6. Kesimpulan

- a. Dalam praktikum statistika ini, kita dapat mempelajari bagaimana mengelola data menggunakan DataFrame di Python dengan pustaka pandas. Kita belajar cara mengakses dan memanipulasi baris serta kolom dalam DataFrame, seperti melakukan subset data, mengubah nilai variabel, menambah dan menghapus kolom, serta menggabungkan data dari berbagai sumber. Kita juga dapat mengetahui...
- b. Selain itu, kita juga memahami cara mengurutkan data untuk analisis lebih lanjut, yang sangat berguna dalam proses pengolahan informasi.

7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	20 Menit	1
2.	Latihan Kedua	20 Menit	1

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang