


<b>Nama:</b> NUR IMAM  <b>NIM:</b> 065002300008	  <b>Praktikum Statistika</b>	<b>MODUL 3</b>  <b>Nama Dosen:</b> Dedy Sugiarto
<b>Hari/Tanggal:</b> Hari, 12 JUNI 2024		<b>Nama Asisten Laboratorium:</b>  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**

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

data_iman = pd.read_clipboard()

# Menampilkan data
print(data_iman)
```

	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]

```
In [ ]: |
```

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:

```
In [4]: imam = data_imam[data_imam['Bedrooms'] == 2]

# Menampilkan nama
print(imam)
```

	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:

```
In [5]: imam['Bathrooms'] = pd.to_numeric(imam['Bathrooms'])

import numpy as np

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

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

	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

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

```
In [6]: import numpy as np

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

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

	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

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:

```
In [7]: # Menambahkan kolom baru 'newvariable'
        iman['newvariable'] = iman['Price'] / iman['SqFt']

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

	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

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:

```
In [8]: iman = iman.drop(columns=['newvariable'])  
  
# Menampilkan DataFrame 'nama'  
print(iman)
```

	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



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:

```
In [9]: kolom1dan2 = data_inam.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:

```
In [10]: # Memilih kolom 1 dan 2 dari DataFrame data_nama  
kolom3dan4 = data_inam.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:

```
In [11]: # 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)
```

## Output

```
n [12]: # Menggabungkan baris dari dua DataFrame
        baris1sd3 = data_imam.iloc[0:3, :]
        baris4sd6 = data_imam.iloc[3:6, :]
        baris1sd6 = baris1sd3.append(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:

```
In [13]: data_imam_sort = data_imam.sort_values(by='Price')  
         print(data_imam_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
data_imam = pd.read_clipboard()
print(data_imam)
```

```
In [14]: import pandas as pd
data_imam = pd.read_clipboard()
print(data_imam)
```

	nama	gender	angkatan	tinggi badan	waktu perjalanan	wilayah tinggal
0	risma	w	2022	165	120	jakarta
1	putri	w	2022	165	55	bandung
2	aisyah	w	2022	165	60	bekasi
3	rena	w	2022	165	70	kerawang
4	chesa	w	2022	175	60	petojo
5	sonya	w	2022	142	140	roxy
6	vania	w	2022	133	40	monas
7	aulia	w	2022	156	30	cianjur
8	andri	w	2022	176	120	bogor
9	rabih	p	2022	158	120	depok
10	risky	p	2022	188	30	cibinong
11	candy	p	2022	172	15	tambun
12	zidan	p	2022	154	120	babelan
13	wayan	p	2022	167	15	rawa belong
14	evan	p	2022	174	10	rawa kalong
15	rakha	p	2022	153	33	tangerang
16	fahmi	p	2022	167	11	cikande
17	ery	p	2022	185	23	banten
18	imam	p	2023	190	12	madura
19	fajri	p	2022	166	15	medan
20	nita	w	2023	165	12	pluit
21	rombon	p	2023	178	133	genbrong
22	akbar	p	2023	175	12	pondok indah
23	putri	w	2023	177	11	jelambar
24	april	w	2023	156	12	jelambar
25	nando	p	2023	188	11	senen
26	daud	p	2023	170	11	papua
27	refil	p	2023	177	11	pasar pagi

Output:

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

```
imam = data_imam[data_imam['tinggi badan'] > 170]
print(imam)
```

Output:

```
In [18]: imam = data_imam[data_imam['tinggi badan'] > 170]
print(imam)
```

	nama	gender	angkatan	tinggi badan	waktu perjalanan	wilayah tinggal
4	chesa	w	2022	175	60	petojo
8	andri	w	2022	176	120	bogor
10	risky	p	2022	188	30	cibinong
11	candy	p	2022	172	15	tambun
14	evan	p	2022	174	10	rawa kalong
17	ery	p	2022	185	23	banten
18	imam	p	2023	190	12	madura
21	rombon	p	2023	178	133	gembrong
22	akbar	p	2023	175	12	pondok indah
23	putri	w	2023	177	11	jelambar
25	nando	p	2023	188	11	senen
27	refil	p	2023	177	11	pasar pagi

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

```
data_imam = pd.DataFrame(data)

# Mengubah kolom 'tinggi badan' menjadi numerik
data_imam['tinggi badan'] = pd.to_numeric(data_imam['tinggi badan'])

# Mengubah nilai kolom 'tinggi badan' berdasarkan kondisi
data_imam['tinggi badan'] = data_imam['tinggi badan'].apply(lambda x: 'tinggi' if x > 160
else 'pendek')

# Menampilkan DataFrame yang telah diubah
print(data_imam)
```

Output:

```
In [23]: data_imam = pd.DataFrame(data)

# Mengubah kolom 'tinggi badan' menjadi numerik
data_imam['tinggi badan'] = pd.to_numeric(data_imam['tinggi badan'])

# Mengubah nilai kolom 'tinggi badan' berdasarkan kondisi
data_imam['tinggi badan'] = data_imam['tinggi badan'].apply(lambda x: 'tinggi' if x > 160 else 'pendek')

# Menampilkan DataFrame yang telah diubah
print(data_imam)
```

	nama	gender	angkatan	tinggi badan	waktu perjalanan	wilayah tinggal
4	chesa	w	2022	175	60	petojo
8	andri	w	2022	176	120	bogor
10	risky	p	2022	188	30	cibinong
11	candy	p	2022	172	15	tambun
14	evan	p	2022	174	10	rawa kalong
17	ery	p	2022	185	23	banten
18	imam	p	2023	190	12	madura
21	rombon	p	2023	178	133	gembrong
22	akbar	p	2023	175	12	pondok indah
23	putri	w	2023	177	11	jelambar
25	nando	p	2023	188	11	senen
27	refil	p	2023	177	11	pasar pagi

#### 4. Menambah Variabel

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

```
data_imam['jurusan'] = 'infor20'  
data_imam['fakultas'] = 'FTI'  
print(data_imam)
```

Output:

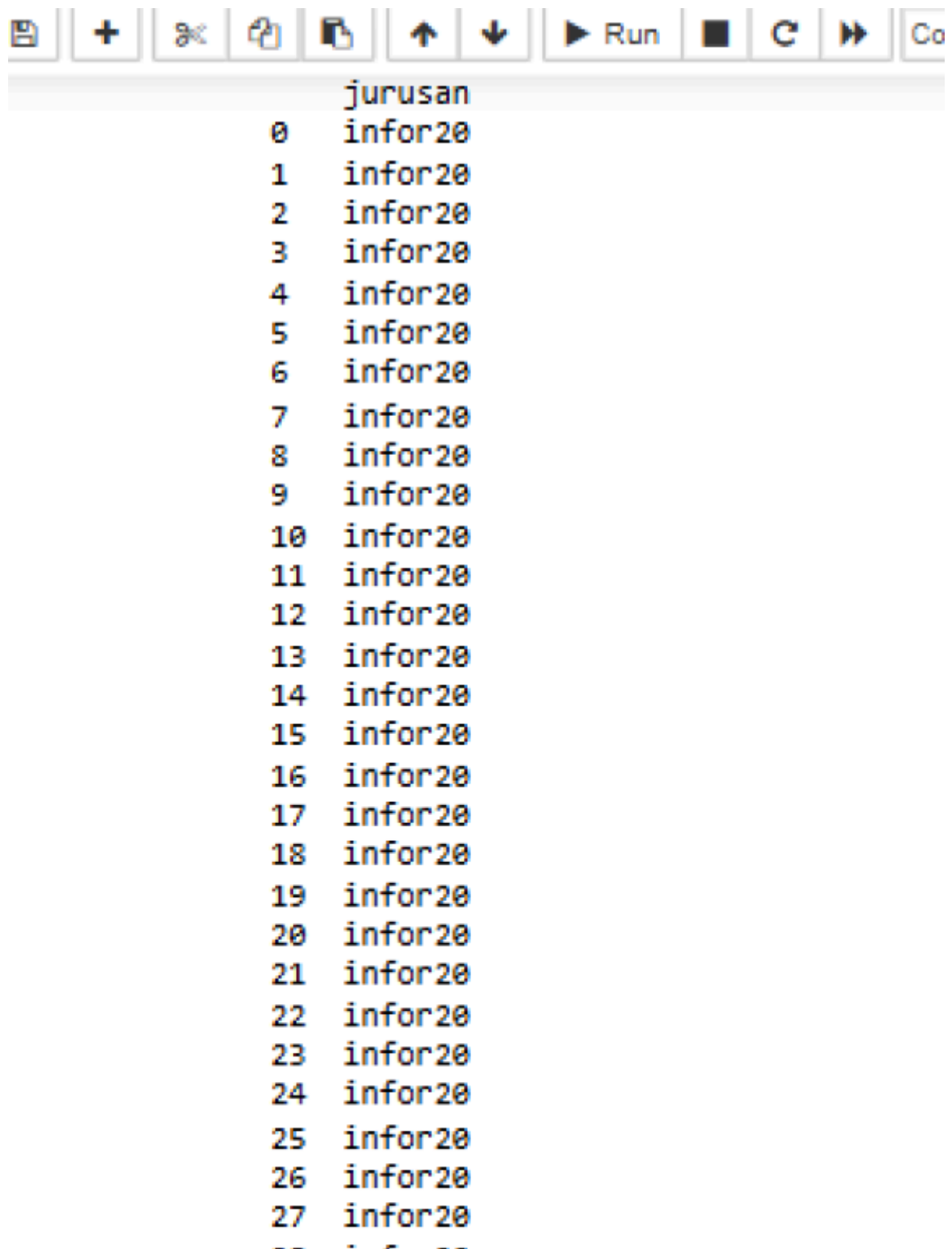
	<code>jurusan</code>	<code>fakultas</code>
<code>0</code>	<code>infor20</code>	<code>FTI</code>
<code>1</code>	<code>infor20</code>	<code>FTI</code>
<code>2</code>	<code>infor20</code>	<code>FTI</code>
<code>3</code>	<code>infor20</code>	<code>FTI</code>
<code>4</code>	<code>infor20</code>	<code>FTI</code>
<code>5</code>	<code>infor20</code>	<code>FTI</code>
<code>6</code>	<code>infor20</code>	<code>FTI</code>
<code>7</code>	<code>infor20</code>	<code>FTI</code>
<code>8</code>	<code>infor20</code>	<code>FTI</code>
<code>9</code>	<code>infor20</code>	<code>FTI</code>
<code>10</code>	<code>infor20</code>	<code>FTI</code>
<code>11</code>	<code>infor20</code>	<code>FTI</code>
<code>12</code>	<code>infor20</code>	<code>FTI</code>
<code>13</code>	<code>infor20</code>	<code>FTI</code>
<code>14</code>	<code>infor20</code>	<code>FTI</code>
<code>15</code>	<code>infor20</code>	<code>FTI</code>



##### 5. Delete Variabel □ Hapus kolom Fakultas

```
data_imam = data_imam.drop(columns=['fakultas'])  
print(data_imam)
```

Output:



	jurusan
0	infor20
1	infor20
2	infor20
3	infor20
4	infor20
5	infor20
6	infor20
7	infor20
8	infor20
9	infor20
10	infor20
11	infor20
12	infor20
13	infor20
14	infor20
15	infor20
16	infor20
17	infor20
18	infor20
19	infor20
20	infor20
21	infor20
22	infor20
23	infor20
24	infor20
25	infor20
26	infor20
27	infor20

## 6. Merge Data Frame

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

```
data_imam = pd.DataFrame(data)

# Menampilkan hanya kolom 'nama' dan 'gender'
data_imam_filtered = data_imam[['nama', 'gender']]

# Menampilkan DataFrame yang telah difilter
print(data_imam_filtered)

data_imam = pd.DataFrame(data)

# Memilih kolom ke-3 (angkatan) dan kolom ke-5 (tinggi badan)
angkatantinggi = data_imam.iloc[:, [2, 3]]

# Menampilkan DataFrame yang telah difilter
print(angkatantinggi)

data_imam = pd.DataFrame(data)

# Menggabungkan kolom 'angkatan' dan 'tinggi badan' menjadi satu kolom baru
data_imam['angkatan_tinggi'] = data_imam['angkatan'].astype(str) + '_' +
data_imam['tinggi badan'].astype(str)
```

Output:

```
In [38]: data_imam = pd.DataFrame(data)

# Menampilkan hanya kolom 'nama' dan 'gender'
data_imam_filtered = data_imam[['nama', 'gender']]

# Menampilkan DataFrame yang telah difilter
print(data_imam_filtered)
```

	nama	gender
0	risma	w
1	putri	w
2	aisyah	w
3	rena	w
4	chesa	w
5	sonya	w
6	vania	w
7	aulia	w
8	andri	w
9	rabih	p
10	risky	p
11	candy	p
12	zidan	p
13	wayan	p
14	evan	p
15	rakha	p
16	fahmi	p
17	ery	p
18	imam	p
19	fajri	p
20	akbar	d

---

	angkatan	tinggi badan
0	2022	165
1	2022	165
2	2022	165
3	2022	165
4	2022	175
5	2022	142
6	2022	133
7	2022	156
8	2022	176
9	2022	158
10	2022	188
11	2022	172
12	2022	154
13	2022	167
14	2022	174
15	2022	153
16	2022	167
17	2022	185
18	2022	190
19	2022	166
20	2022	170
21	2022	160
22	2022	168
23	2022	155
24	2022	157
25	2022	166
26	2022	169
27	2022	165
28	2022	171
29	2022	177
30	2022	160
31	2022	158
32	2022	162
33	2022	173
34	2022	159
35	2022	175
36	2022	164
37	2022	161
38	2022	163
39	2023	160

---

```
In [42]: data_imam = pd.DataFrame(data)

# Menggabungkan kolom 'angkatan' dan 'tinggi badan' menjadi satu kolom baru
data_imam['angkatan_tinggi'] = data_imam['angkatan'].astype(str) + '_' + data_imam['tinggi badan'].astype(str)

# Menampilkan DataFrame yang telah diubah
print(data_imam[['nama', 'gender', 'angkatan_tinggi']])
```

	nama	gender	angkatan_tinggi
0	risma	w	2022_165
1	putri	w	2022_165
2	aisyah	w	2022_165
3	rena	w	2022_165
4	chesa	w	2022_175
5	sonya	w	2022_142
6	vania	w	2022_133
7	aulia	w	2022_156
8	andri	w	2022_176
9	rabih	p	2022_158
10	risky	p	2022_188
11	candy	p	2022_172
12	zidan	p	2022_154
13	wayan	p	2022_167
14	evan	p	2022_174
15	rakha	p	2022_153
16	fahmi	p	2022_167
17	ery	p	2022_185
18	imam	p	2022_190
19	fajri	p	2022_166
20	akbar	p	2022_170
21	budi	p	2022_160
22	diana	w	2022_168
23	elsa	w	2022_155
24	fitri	w	2022_157
25	gina	w	2022_166
26	haris	p	2022_169
27	indra	p	2022_165

## 7. Merge Baris

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

```
In [43]: bar15 = data_imam.iloc[0:5,:]  
bar2530 = data_imam.iloc[5:30, :]  
baris = bar15.append(bar2530)  
print(baris)
```

```
In [45]: data_imam = pd.DataFrame(data)

# Mengurutkan DataFrame berdasarkan kolom 'tinggi badan'  
data_imam_sorted = data_imam.sort_values(by='tinggi badan')

# Menampilkan DataFrame yang telah diurutkan  
print(data_imam_sorted)
```

Output:

	nama	gender	angkatan	tinggi badan	waktu perjalanan	wilayah tinggal	\
0	risma	w	2022	165	120	jakarta	
1	putri	w	2022	165	55	bandung	
2	aisyah	w	2022	165	60	bekasi	
3	rena	w	2022	165	70	kerawang	
4	chesa	w	2022	175	60	petojo	
5	sonya	w	2022	142	140	roxy	
6	vania	w	2022	133	40	monas	
7	aulia	w	2022	156	30	cianjur	
8	andri	w	2022	176	120	bogor	
9	rabih	p	2022	158	120	depok	
10	risky	p	2022	188	30	cibinong	
11	candy	p	2022	172	15	tambun	
12	zidan	p	2022	154	120	babelan	
13	wayan	p	2022	167	15	rawa belong	
14	evan	p	2022	174	10	rawa kalong	
15	rakha	p	2022	153	33	tangerang	
16	fahmi	p	2022	167	11	cikande	
17	ery	p	2022	185	23	banten	
18	imam	p	2022	190	12	madura	
19	fajri	p	2022	166	15	medan	
20	akbar	p	2022	170	50	yogyakarta	
21	budi	p	2022	160	45	semarang	
22	diana	w	2022	168	55	surabaya	
23	elsa	w	2022	155	40	solo	
24	fitri	w	2022	157	65	malang	
25	gina	w	2022	166	70	bogor	
26	haris	p	2022	169	20	bekasi	
27	indra	p	2022	165	35	depok	
28	joko	p	2022	171	25	cibubur	
29	kevin	p	2022	177	15	tangerang	
	fakultas	angkatan	tinggi				
0	teknik	2022	165				
1	teknik	2022	165				
2	teknik	2022	165				
3	teknik	2022	165				
4	teknik	2022	175				
5	teknik	2022	142				
6	teknik	2022	133				
7	teknik	2022	156				
8	teknik	2022	176				
9	teknik	2022	158				
10	teknik	2022	188				
11	teknik	2022	172				
12	teknik	2022	154				

#### 4. File Praktikum

Github Repository:

#### 5. Soal Latihan

Soal:

1. Apa saja kegunaan pengelolaan data pada data frame?

2. Dalam kasus apakah data perlu dihapus?

Jawaban:

1. Pengelolaan data dalam DataFrame memungkinkan manipulasi dan analisis data secara efisien di Python. Dengan DataFrame, pengguna dapat mengimpor, menyimpan, dan mengatur data dalam format tabel dua dimensi. Operasi seperti pemilihan kolom dan baris, pengindeksan, penyaringan, penggabungan, pengurutan, agregasi, dan visualisasi data dapat dilakukan menggunakan pustaka seperti pandas, NumPy, dan Matplotlib. DataFrame juga menyediakan metode untuk membersihkan dan memanipulasi data, termasuk mengelola nilai yang hilang atau duplikat, mengubah tipe data, dan melakukan operasi statistik deskriptif. Ini menjadikan DataFrame penting untuk analisis data yang efektif dan pengambilan keputusan berbasis data dalam berbagai aplikasi.

2. Penghapusan data diperlukan ketika data tidak lagi relevan, tidak valid, rusak, mengandung informasi sensitif yang tidak lagi diperlukan, atau duplikat. Data yang tidak valid atau rusak dapat menghasilkan kesimpulan yang salah, sementara data duplikat tidak memberikan informasi tambahan dan dapat mempengaruhi akurasi analisis. Membersihkan data dari elemen yang tidak diperlukan atau tidak valid adalah langkah penting untuk analisis yang akurat dan efektif.

## 6. Kesimpulan

- Dalam pengerjaan praktikum Statistika, kita harus teliti dalam mengolah dataframe
- Kita juga dapat mengetahui cara mengelola dan memfilter dataframe

## 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	15 Menit	menarik
2.	Latihan Kedua	15 Menit	menarik

Keterangan:

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