


Nama: Kahfi Alam Ferry NIM: 064102400015	 Praktikum Statistika	MODUL 3 Nama Dosen: Dr. Dedy Sugiarto, S.Si, M.Kom
Hari/Tanggal: Senin, 17 Maret 2025		Nama Asisten Labratorium: 1. Michael Briant (064002300004) 2. Monica Sicilia Simanjuntak (065002300030)

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 Clipboard dengan cara mengcopy isi csv.

```
import pandas as pd

data_nama = pd.read_clipboard()

# Menampilkan data print(data_nama)
```

Output:

```
[12]: import pandas as pd

data_Kahfi = pd.read_clipboard()

# Menampilkan data
print(data_Kahfi)
```

	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:

```
[20]: Kahfi = data_Kahfi[data_Kahfi['Bedrooms'] == 2]

# Menampilkan nama
print(Kahfi)
```

	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

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:

```
[22]: Kahfi['Bathrooms'] = pd.to_numeric(Kahfi['Bathrooms'])

import numpy as np

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

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

	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

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

```
[24]: import numpy as np

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

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

	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

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:

```
[26]: # Menambahkan kolom baru 'newvariable'
Kahfi['newvariable'] = Kahfi['Price']/Kahfi['SqFt']

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

	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

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:

```

[28]: Kahfi = Kahfi.drop(columns=['newvariable'])

# Menampilkan DataFrame 'nama'
print(Kahfi)

```

	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

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

```

kolom1 dan2 = data_nama.iloc[:, 0:2]

```

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

Output:

```
kolom1dan2 = data_Kahfi.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:

```
# Memilih kolom 1 dan 2 dari DataFrame data_Kahfi
kolom3dan4 = data_Kahfi.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:

```
# 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

```
#import pandas library
import pandas as pd

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

```
data_Kahfi_sort = data_Kahfi.sort_values(by='Price')
print(data_Kahfi_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

```
[21]: import pandas as pd

data_Kahfi = pd.read_clipboard()

# Menampilkan data
print(data_Kahfi)
```

Output:

```
[21]: import pandas as pd

data_Kahfi = pd.read_clipboard()

# Menampilkan data
print(data_Kahfi)
```

	Nama	Gender	Angkatan	Tinggi Badan	Waktu Perjalanan	Wilayah	Tinggal
0	Lebron	L	2020	200	25	Bekasi	
1	Curry	L	2020	201	40	Bekasi	
2	Doncic	L	2020	202	35	Bekasi	
3	Gita	P	2020	165	20	Ciputat	
4	Adel	P	2020	167	15	Bogor	
5	Dila	P	2020	168	25	Ciputat	
6	Diva	P	2020	164	45	Tangsel	
7	Erine	P	2020	166	25	Jaktim	
8	Didi	L	2020	167	25	Jakut	
9	Zufar	L	2020	167	45	Jakpus	
10	Gusty	L	2020	166	35	Jakbar	
11	Satria	L	2020	167	30	Tangsel	
12	Raziq	L	2020	167	35	Bekasi	
13	Agnar	L	2020	167	30	Bekasi	
14	Adit	L	2020	176	40	Grogol	
15	Joy	P	2020	176	45	Jawa	
16	Yudha	L	2020	166	45	Cipondoh	

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


```
[175]: import pandas as pd

# Membaca data dari clipboard
df = pd.read_clipboard()

# Menghapus spasi tersembunyi pada nama kolom
df.columns = df.columns.str.strip()

# Mengambil subset hanya dengan kolom "Nama" dan "Tinggi Badan"
subset_tinggi_badan = df[["Nama", "Tinggi Badan"]]

# Salin hasil subset ke clipboard agar bisa langsung dipaste di Excel / Notepad
subset_tinggi_badan.to_clipboard(index=False)

# Menampilkan hasil
print(subset_tinggi_badan)
```

Output:

	Nama	Tinggi Badan
0	Lebron	200
1	Curry	201
2	Doncic	202
3	Gita	165
4	Adel	167
5	Dila	168
6	Diva	164
7	Erine	166
8	Didi	167
9	Zufar	167
10	Gusty	166
11	Satria	167
12	Raziq	167
13	Agnar	167
14	Adit	176
15	Joy	176
16	Yudha	166
17	Arif	167
18	Dimas	176
19	Fadhil	167
20	Franky	198
21	Zoro	197
22	Luffy	198
23	Sanji	178
24	Nami	178
25	Usop	180
26	Robin	187
27	Jimbei	204
28	Chopper	134
29	Brook	198

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

```
[65]: import pandas as pd

df = pd.read_clipboard()

# Ubah nilai kolom "Tinggi_Badan" menjadi kategori
df['Kategori_Tinggi'] = df['Tinggi_Badan'].apply(lambda x: 'Tinggi' if x > 160 else 'Pendek')

df.to_clipboard(index=False)

print(df)
```

Output:

```
[65]: import pandas as pd

df = pd.read_clipboard()

# Ubah nilai kolom "Tinggi_Badan" menjadi kategori
df['Kategori_Tinggi'] = df['Tinggi_Badan'].apply(lambda x: 'Tinggi' if x > 160 else 'Pendek')

df.to_clipboard(index=False)

print(df)
```

	Nama	Tinggi_Badan	Kategori_Tinggi
0	Lebron	200	Tinggi
1	Curry	201	Tinggi
2	Doncic	202	Tinggi
3	Gita	165	Tinggi
4	Adel	159	Pendek
5	Dila	157	Pendek
6	Diva	147	Pendek
7	Erine	173	Tinggi
8	Didi	146	Pendek
9	Zufar	154	Pendek
10	Gusty	143	Pendek
11	Satria	153	Pendek
12	Raziq	145	Pendek
13	Agnar	178	Tinggi
14	Adit	160	Pendek
15	Joy	134	Pendek
16	Yudha	123	Pendek

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”

```
[85]: import pandas as pd

df = pd.read_clipboard()

df['Jurusan'] = "Infor20"

df['Fakultas'] = "FTI"

print(df.head(30))
```

Output:

```
[85]: import pandas as pd

df = pd.read_clipboard()

df['Jurusan'] = "Infor20"

df['Fakultas'] = "FTI"

print(df.head(30))
```

26	Robin	P	2020	187	54	Bekasi
27	Jimbei	L	2020	204	54	Pesing
28	Chopper	L	2020	134	30	Pesing
29	Brook	L	2020	198	10	Jawa

	Jurusan	Fakultas
0	Infor20	FTI
1	Infor20	FTI
2	Infor20	FTI
3	Infor20	FTI
4	Infor20	FTI
5	Infor20	FTI
6	Infor20	FTI
7	Infor20	FTI
8	Infor20	FTI
9	Infor20	FTI
10	Infor20	FTI
11	Infor20	FTI
12	Infor20	FTI

5. Delete Variabel □ Hapus kolom Fakultas

```
•[87]: df = df.drop(columns=['Fakultas'])  
  
# Menampilkan hasil setelah kolom dihapus  
print(df.head(30))
```

Output:

```
•[87]: df = df.drop(columns=['Fakultas'])  
  
# Menampilkan hasil setelah kolom dihapus  
print(df.head(30))
```

25	Osop	L	2020	180	43	Bekasi
26	Robin	P	2020	187	54	Bekasi
27	Jimbei	L	2020	204	54	Pesing
28	Chopper	L	2020	134	30	Pesing
29	Brook	L	2020	198	10	Jawa

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

6. Merge Data Frame

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

```
[105]: import pandas as pd

# Membaca data dari clipboard
df = pd.read_clipboard()

# Bersihkan nama kolom dari spasi tersembunyi
df.columns = df.columns.str.strip()

# Gabungkan kolom Nama dan Gender
df['Nama Gender'] = df['Nama'] + ' - ' + df['Gender']

# Gabungkan kolom Angkatan dan Tinggi Badan
df['Angkatan_Tinggi'] = df['Angkatan'].astype(str) + ' | ' + df['Tinggi Badan'].astype(str)

# Gabungkan dua hasil sebelumnya
df['Final_Gabungan'] = df['Nama_Gender'] + ' || ' + df['Angkatan_Tinggi']

df.to_clipboard(index=False)

# Menampilkan hasil
print(df[['Nama_Gender', 'Angkatan_Tinggi', 'Final_Gabungan']])
```

Output:

	Nama_Gender	Angkatan_Tinggi	Final_Gabungan
0	Lebron - L	2020 200	Lebron - L 2020 200
1	Curry - L	2020 201	Curry - L 2020 201
2	Doncic - L	2020 202	Doncic - L 2020 202
3	Gita - P	2020 165	Gita - P 2020 165
4	Adel - P	2020 167	Adel - P 2020 167
5	Dila - P	2020 168	Dila - P 2020 168
6	Diva - P	2020 164	Diva - P 2020 164
7	Erine - P	2020 166	Erine - P 2020 166
8	Didi - L	2020 167	Didi - L 2020 167
9	Zufar - L	2020 167	Zufar - L 2020 167
10	Gusty - L	2020 166	Gusty - L 2020 166
11	Satria - L	2020 167	Satria - L 2020 167
12	Raziq - L	2020 167	Raziq - L 2020 167
13	Agnar - L	2020 167	Agnar - L 2020 167
14	Adit - L	2020 176	Adit - L 2020 176
15	Joy - P	2020 176	Joy - P 2020 176
16	Yudha - L	2020 166	Yudha - L 2020 166

7. Merge Baris

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

```
[155]: import pandas as pd

# Membaca data dari clipboard (pastikan data sudah disalin sebelumnya)
df = pd.read_clipboard()

# Cek jumlah total baris
print(f"Jumlah total baris dalam dataframe: {len(df)}")

# Mengambil baris 1-5 dan 25-30 dengan indeks yang benar
merged_1_5 = df.iloc[0:5]
merged_25_30 = df.iloc[24:30] # Menggunakan indeks 24-30

# Gabungkan kedua variabel di atas
final_merge = pd.concat([merged_1_5, merged_25_30], axis=0)

# Reset index agar rapi
final_merge.reset_index(drop=True, inplace=True)

# Salin hasil ke clipboard agar bisa langsung dipaste di Excel / Notepad
final_merge.to_clipboard(index=False)

# Menampilkan hasil
print(final_merge)
```

Output:

	Nama	Gender	Angkatan	Tinggi Badan	Waktu Perjalanan	Wilayah Tinggal
0	Lebron	L	2020	200	25	Bekasi
1	Curry	L	2020	201	40	Bekasi
2	Doncic	L	2020	202	35	Bekasi
3	Gita	P	2020	165	20	Ciputat
4	Adel	P	2020	167	15	Bogor
5	Nami	P	2020	178	45	Bekasi
6	Usop	L	2020	180	43	Bekasi
7	Robin	P	2020	187	54	Bekasi
8	Jimbei	L	2020	204	54	Pesing
9	Chopper	L	2020	134	30	Pesing

8. Sort Data Frame □ Lakukan sort berdasarkan waktu perjalanan

```
[163]: import pandas as pd

# Membaca data dari clipboard (pastikan data sudah disalin sebelumnya)
df = pd.read_clipboard()

# Cek jumlah total baris
print(f"Jumlah total baris dalam dataframe: {len(df)}")

# Mengambil baris 1-5 dan 25-30 dengan indeks yang benar
merged_1_5 = df.iloc[0:5]
merged_25_30 = df.iloc[24:30]

# Gabungkan kedua variabel di atas
final_merge = pd.concat([merged_1_5, merged_25_30], axis=0)

# Reset index agar rapi
final_merge.reset_index(drop=True, inplace=True)

# **Sorting berdasarkan waktu perjalanan**
final_merge = final_merge.sort_values(by="Waktu Perjalanan", ascending=True)

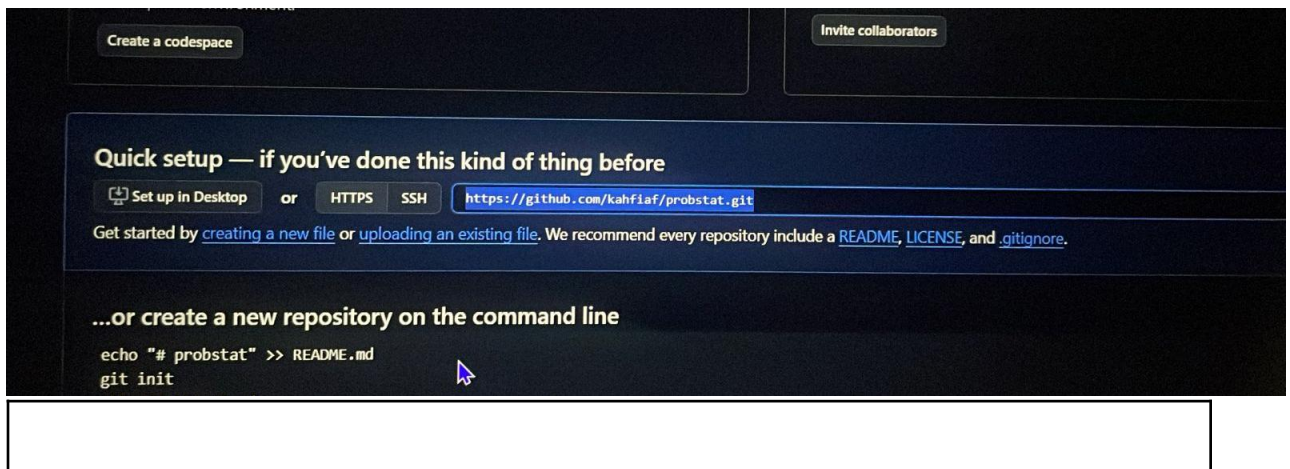
# Salin hasil ke clipboard agar bisa langsung dipaste di Excel / Notepad
final_merge.to_clipboard(index=False)

# Menampilkan hasil
print(final_merge)
```


Output:

```
Jumlah total baris dalam dataframe: 29
  Nama      Gender Angkatan  Tinggi Badan  Waktu Perjalanan Wilayah Tinggal
4   Adel     P      2020      167          15          Bogor
3   Gita     P      2020      165          20          Ciputat
0  Lebron    L      2020      200          25          Bekasi
2  Doncic    L      2020      202          35          Bekasi
1   Curry    L      2020      201          40          Bekasi
5   Nami     P      2020      178          45          Bekasi
6   Usop     L      2020      180          45          Bekasi
8  Jimbei    L      2020      204          54          Pesing
7   Robin    P      2020      187          65          Bekasi
9  Chopper   L      2020      134          75          Pesing
```

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. Membaca dan Menyimpan Data

Mengimpor data dari berbagai format seperti CSV, Excel, JSON, SQL, Clipboard, dll.

Menyimpan hasil pemrosesan dalam berbagai format untuk digunakan lebih lanjut.

2. Data kosong, Data duplikat, Data Tidak Relevan dengan Analisis.

6. Kesimpulan

- a. Dalam pengerjaan praktikum Statistika, pengelolaan data sangat penting untuk memastikan bahwa data yang digunakan bersih, relevan, dan dapat diandalkan dalam analisis.
- b. Kita juga dapat mengetahui bagaimana cara mengolah data dengan lebih efisien, termasuk dalam menghapus data yang tidak diperlukan, mengidentifikasi outlier, serta mengelola duplikasi dan inkonsistensi data.

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	140 Menit	Baik

2.	Latihan Kedua	140 Menit	Baik
----	---------------	-----------	------

Keterangan:

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