

<p>Nama: Garda Yudha Wicaksana</p> <p>NIM: 064102400006</p>		<h1>MODUL 3</h1> <p>Nama Dosen: Dr. Dedy Sugiarto, S.Si, M.Kom</p>
<p>Hari/Tanggal: Senin, 17 Maret 2025</p>	<h2>Praktikum Statistika</h2>	<p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none">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

- 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:

#JANGANLUPAMENGGANTIOUTPUTMENJADIHASILSENDIRI

```
[2]: import pandas as pd
data_yudha = pd.read_clipboard()
print(data_yudha)

   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:

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Home, Untitled8, (466) Lakers vs Suns | Lakers High, [NIM] - [NAMA] - Modul 3 Prob...
- Toolbar:** File, Edit, View, Run, Kernel, Settings, Help.
- Code Cell 9:** Contains the Python code:

```
yudha = data_yudha[data_yudha['Bedrooms'] == 2]
print(yudha)
```

The output of this cell is a DataFrame showing 125 rows of house data, filtered to show only those with 2 bedrooms. The columns include Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, and Neighborhood.
- Code Cell 11:** Contains the Python code:

```
yudha['Bathrooms'] = pd.to_numeric(yudha['Bathrooms'])
```
- System Tray:** Shows various application icons including Spotify, Google Chrome, Microsoft Word, and Microsoft Excel.

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:

```
[11]: yudha['Bathrooms'] = pd.to_numeric(yudha['Bathrooms'])

import numpy as np

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

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

	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

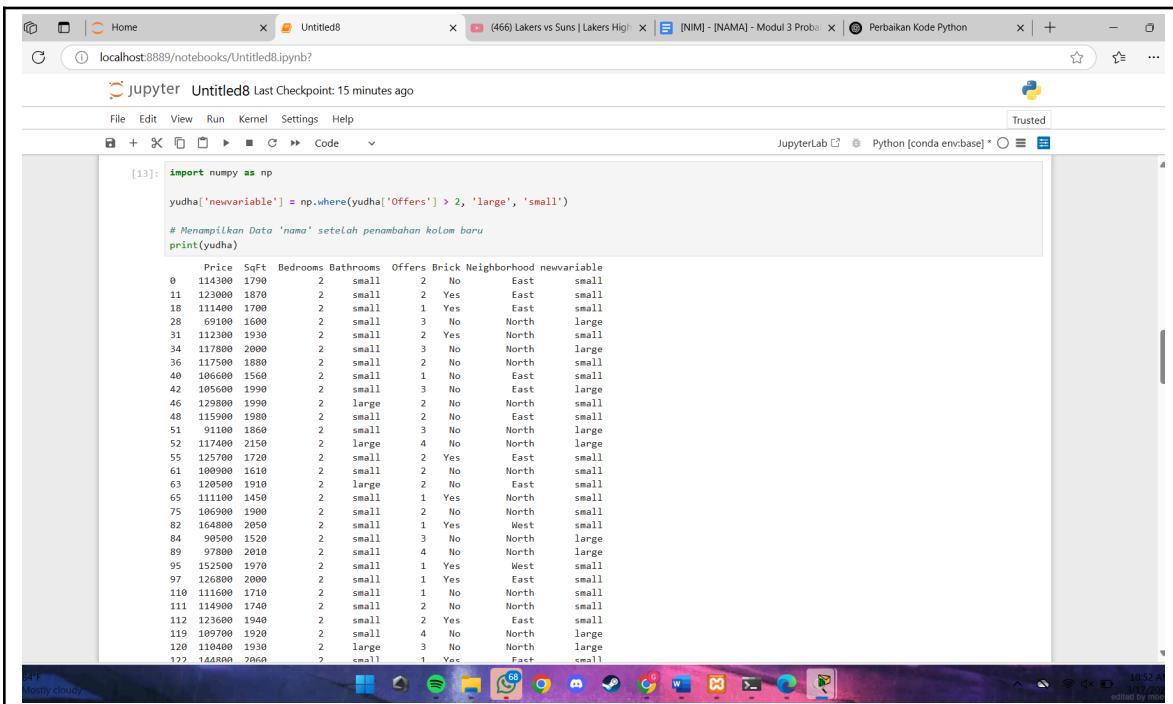
4. Lalu langkah selanjutnya adalah menambah variabel. Tahap 1 □ Membuat variable baru dari dataku1 dengan nilai sesuai kondisi ifelse yang ditentukan. Sesuaikan nama variable baru dengan nama praktikan.

```
import numpy as np

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

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

Output



```
[13]: import numpy as np

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

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

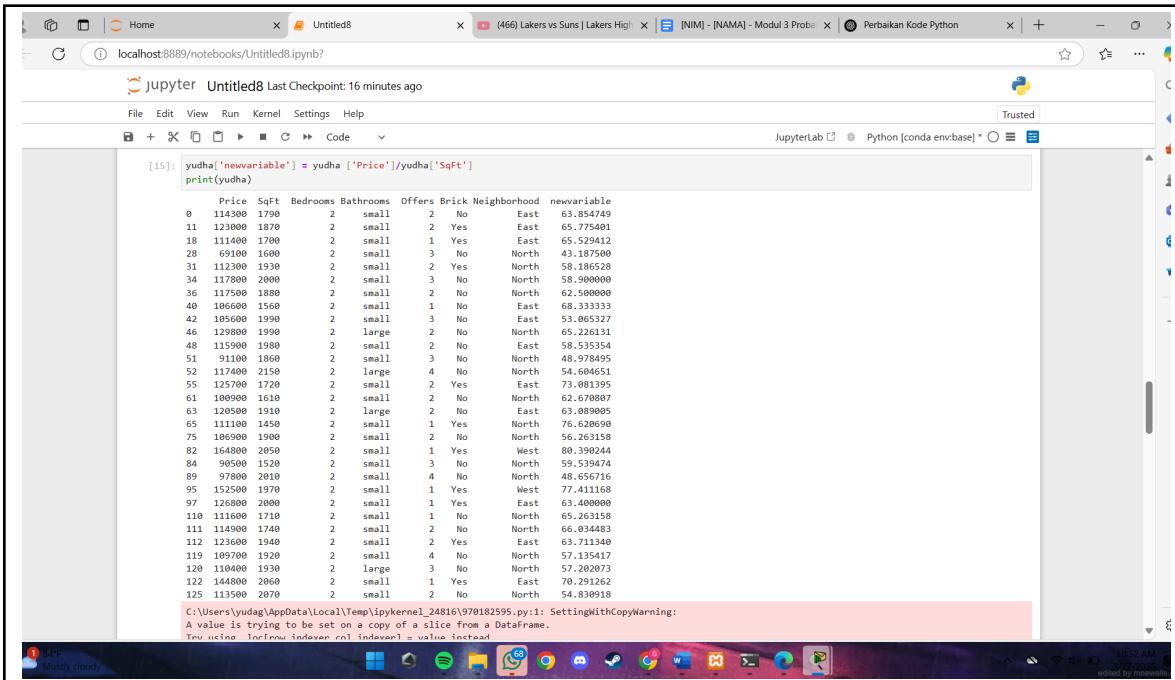
	Price	Sqft	Bedrooms	Bathrooms	Offers	Brick	Neighborhood	newvariable
0	114300	1790	2	small	2	No	East	small
11	123000	1870	2	small	2	Yes	East	small
18	111400	1700	2	small	1	Yes	East	small
28	69100	1600	2	small	3	No	North	large
31	112300	1930	2	small	2	Yes	North	small
34	117800	2000	2	small	3	No	North	large
36	117500	1880	2	small	2	No	North	small
40	106500	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	108900	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	164000	2050	2	small	1	Yes	West	small
84	90500	1520	2	small	3	No	North	large
89	97400	1610	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
113	123600	1940	2	small	2	Yes	East	small
119	109700	1920	2	small	4	No	North	large
128	110400	1930	2	large	3	No	North	large
122	144800	2060	2	small	1	Yes	East	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:



```
[15]: yudha['newvariable'] = yudha['Price']/yudha['SqFt']
print(yudha)

   Price Sqft Bedrooms Bathrooms Offers Brick Neighborhood  newvariable
0  114300  1790      2    small       2   No     East  63.854749
11 1230000  1870      2    small       2   Yes    East  65.775401
18  1114000  1700      2    small       1   Yes    East  65.529412
28  691000  1600      2    small       3   No    North  43.187500
31  1123000  1930      2    small       2   Yes    North  58.186528
34  1123000  1930      2    small       3   No    North  58.308000
39  1175000  1830      2    small       3   No    North  62.670800
49  1066000  1560      2    small       1   No    East  68.333333
42  1056000  1990      2    small       3   No    East  53.065337
46  1298000  1990      2    large       2   No    North  65.226131
48  1159000  1980      2    small       2   No    East  58.535354
51  911000  1860      2    small       3   No    North  48.978495
52  1174000  2150      2    large       4   No    North  54.604651
55  1257000  1720      2    small       2   Yes   East  73.081395
61  1009000  1610      2    small       2   No    North  62.670807
63  1205000  1910      2    large       2   No    East  63.089005
65  1111000  1450      2    small       1   Yes    North  76.620696
75  1069000  1900      2    small       2   No    North  56.263158
82  1648000  2050      2    small       1   Yes   West  80.390244
84  908000  1520      2    small       3   No    North  59.539474
89  978000  2010      2    small       4   No    North  48.604650
95  1525000  1970      2    small       1   Yes   West  77.411168
98  1260000  2000      2    small       1   Yes   East  63.409000
110 1116000  1710      2    small       1   No    North  65.263158
111 1149000  1740      2    small       2   No    North  66.034483
112 1236000  1940      2    small       2   Yes   East  63.711340
119 1097000  1920      2    small       4   No    North  57.135417
120 1104000  1930      2    large       3   No    North  57.202073
122 1448000  2060      2    small       1   Yes   East  70.291262
125 1135000  2070      2    small       2   No    North  54.830918
```

C:\Users\yuda\AppData\Local\Temp\ipykernel_24816\970182595.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead.

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

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

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

Output:

The screenshot shows a Jupyter Notebook interface with the following code in cell [17]:

```
yudha = yudha.drop(columns=['newvariable'])
print(yudha)
```

The output displays a DataFrame with 125 rows and 8 columns. The columns are labeled: Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, Neighborhood, and newvariable (which is present in the code but not in the output). The data includes various numerical values and categorical labels like 'small', 'large', 'Yes', 'No', and neighborhood names like 'East' and 'West'. In cell [19], there is another piece of code:

```
kolom1dan2 = data_yudha.iloc[:, 0:2]
print(kolom1dan2)
```

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:

The screenshot shows a Jupyter Notebook interface with a toolbar at the top. Below the toolbar, a code cell contains the following Python code:

```
!]: kolom1dan2 = data_yudha.iloc[:, 0:2]
print(kolom1dan2)
```

The output of the code cell is a DataFrame with the following structure:

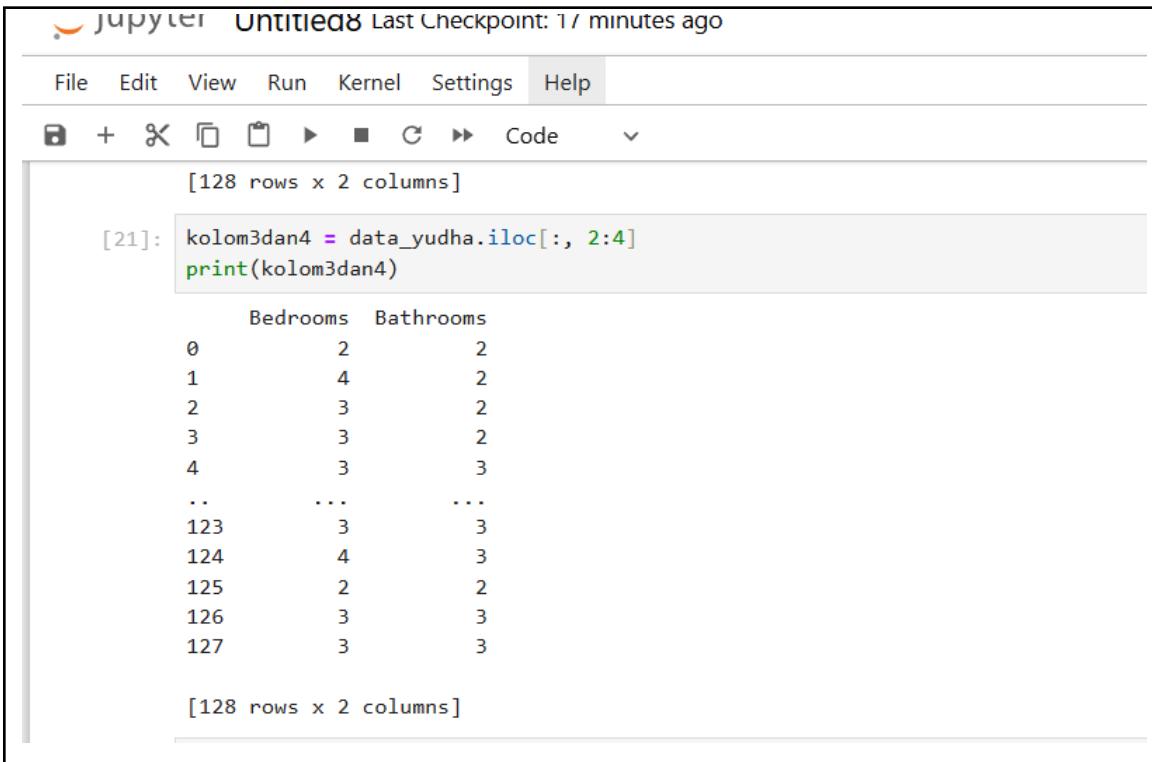
	Price	SqFt
0	114300	1790
1	114200	2030
2	114800	1740
3	94700	1980
4	119800	2130
..
123	119700	1900
124	147900	2160
125	113500	2070
126	149900	2020
127	124600	2250

[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:



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter Untitled8 Last Checkpoint: 17 minutes ago
- Toolbar:** File Edit View Run Kernel Settings Help
- Code Cell:** [21]:
kolom3dan4 = data_yudha.iloc[:, 2:4]
print(kolom3dan4)
- Output:** [128 rows x 2 columns]
A table with columns "Bedrooms" and "Bathrooms". The data is as follows:

	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:

```
[128 rows x 2 columns]  
[9]: # 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

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Home, Untitled8, (466) Lakers vs Suns | Lakers
- Address Bar:** localhost:8889/notebooks/Untitled8.ipynb?
- Toolbar:** File, Edit, View, Run, Kernel, Settings, Help
- Code Cell:** [25]:

```
import pandas as pd
baris1sd3 = data_yudha.iloc[0:3, :]
baris4sd6 = data_yudha.iloc[3:6, :]
baris1sd6 = pd.concat([baris1sd3, baris4sd6])

print(baris1sd6)
```
- Output:** A table showing the merged DataFrame baris1sd6. The columns are Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, and Neighborhood. The data is as follows:

	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:

The screenshot shows a Jupyter Notebook cell with the following content:

```
[27]: data_yudha_sort = data_yudha.sort_values(by='Price')
print(data_yudha_sort)
```

The output of the code is a Pandas DataFrame:

	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

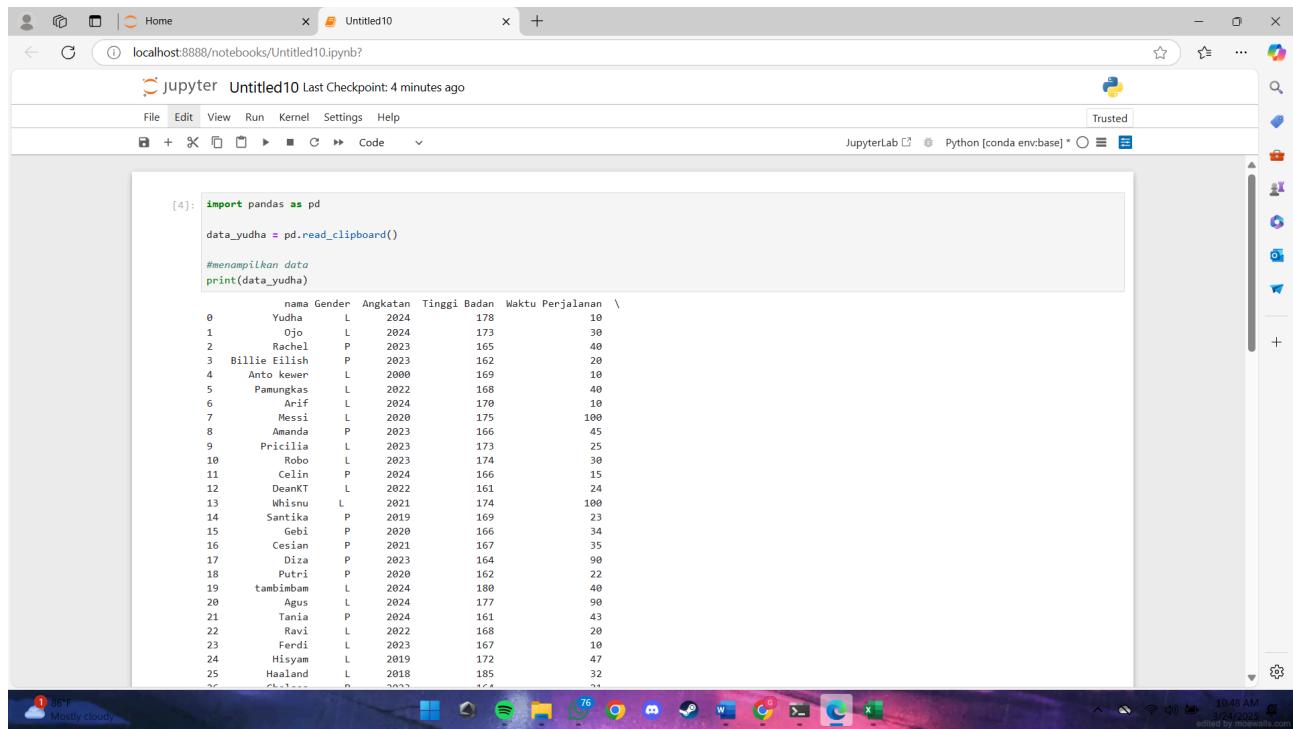
Gunakan dataset pada tugas 1 yang telah ditambah lagi datanya dengan 10 mhs TIF/SI. Sehingga total baris data pada file tersebut berjumlah 30. Ulangi kembali perintah-perintah di atas dan sesuaikan dengan data anda. Dan lampirkan Screen Capture untuk tiap poin yang ada.

1. Read CSV

```
[4]: import pandas as pd  
  
data_yudha = pd.read_clipboard()  
  
#menampilkan data  
print(data_yudha)
```



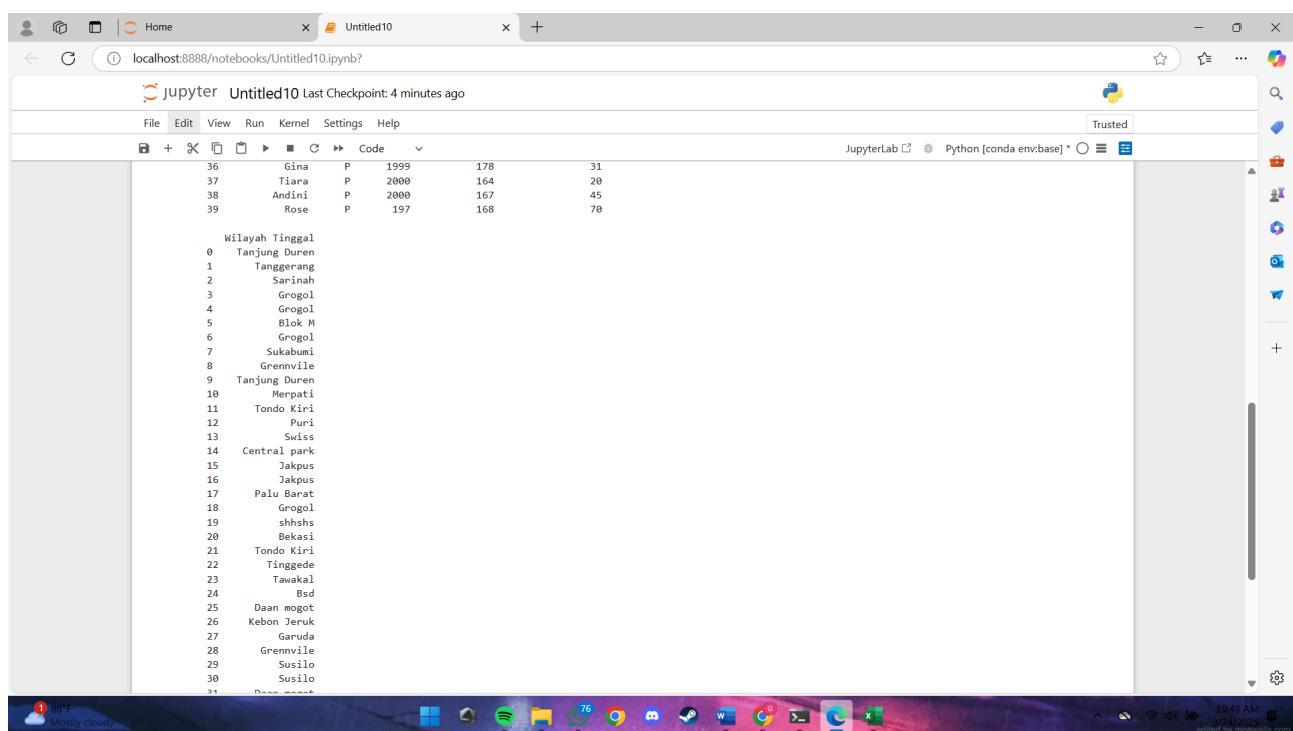
Output:



Jupyter Untitled10 Last Checkpoint: 4 minutes ago

```
[4]: import pandas as pd
data_yudha = pd.read_clipboard()
#menampilkan data
print(data_yudha)
```

	nama	Gender	Angkatan	Tinggi Badan	Waktu	Perjalanan
0	Yudha	L	2024	178	10	
1	Ojo	L	2024	173	30	
2	Rachel	P	2023	165	40	
3	Billie Eilish	P	2023	162	20	
4	Anto Kewer	L	2000	169	10	
5	Pamungkas	L	2022	168	40	
6	Arif	L	2024	170	10	
7	Messi	L	2020	175	100	
8	Amanda	P	2023	166	45	
9	Pricia	L	2023	173	25	
10	Robo	L	2023	174	30	
11	Celin	P	2024	166	15	
12	DeamKT	L	2022	161	24	
13	Whisnu	L	2021	174	100	
14	Santika	P	2019	169	23	
15	Gebi	P	2020	166	34	
16	Cesian	P	2021	167	35	
17	Diza	P	2023	164	90	
18	Putri	P	2020	162	22	
19	tambimbam	L	2024	180	40	
20	Agus	L	2024	177	90	
21	Tania	P	2024	161	43	
22	Ravi	L	2022	168	20	
23	Ferdi	L	2023	167	10	
24	Hisyam	L	2019	172	47	
25	Haaland	L	2018	185	32	

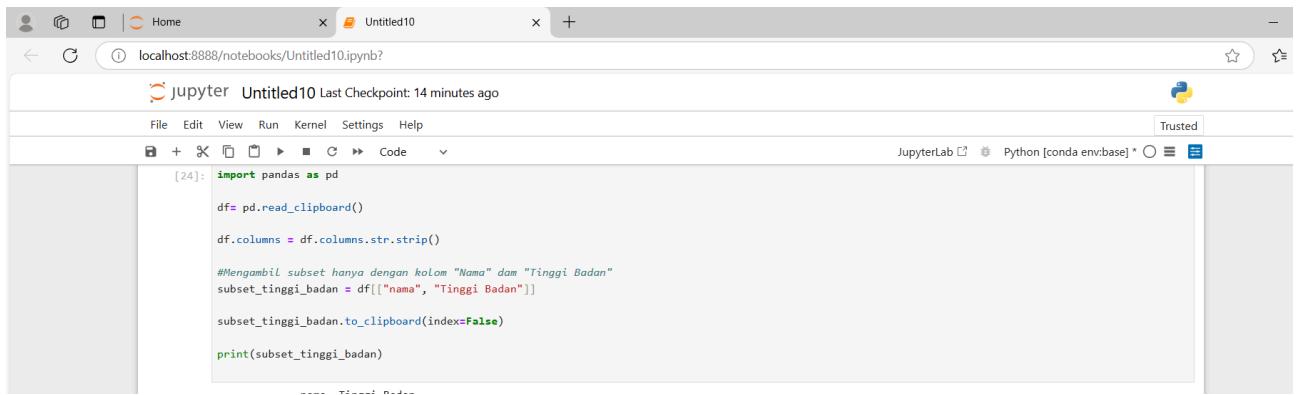


Jupyter Untitled10 Last Checkpoint: 4 minutes ago

```
[4]: import pandas as pd
data_yudha = pd.read_clipboard()
#menampilkan data
print(data_yudha)
```

	Wilayah Tinggal	Kode
0	Tanjung Duren	
1	Tangerang	
2	Sarina	
3	Grogol	
4	Grogol	
5	Blok M	
6	Grogol	
7	Sukabumi	
8	Grenville	
9	Tanjung Duren	
10	Merpati	
11	Tondo Kiri	
12	Puri	
13	Swiss	
14	Central Park	
15	Jakpus	
16	Jakpus	
17	Palu Barat	
18	Grogol	
19	shhhs	
20	Bekasi	
21	Tondo Kiri	
22	Tinggede	
23	Tawakal	
24	Bsd	
25	Daan mogot	
26	Kebon Jeruk	
27	Garuda	
28	Grenville	
29	Susilo	
30	Susilo	

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

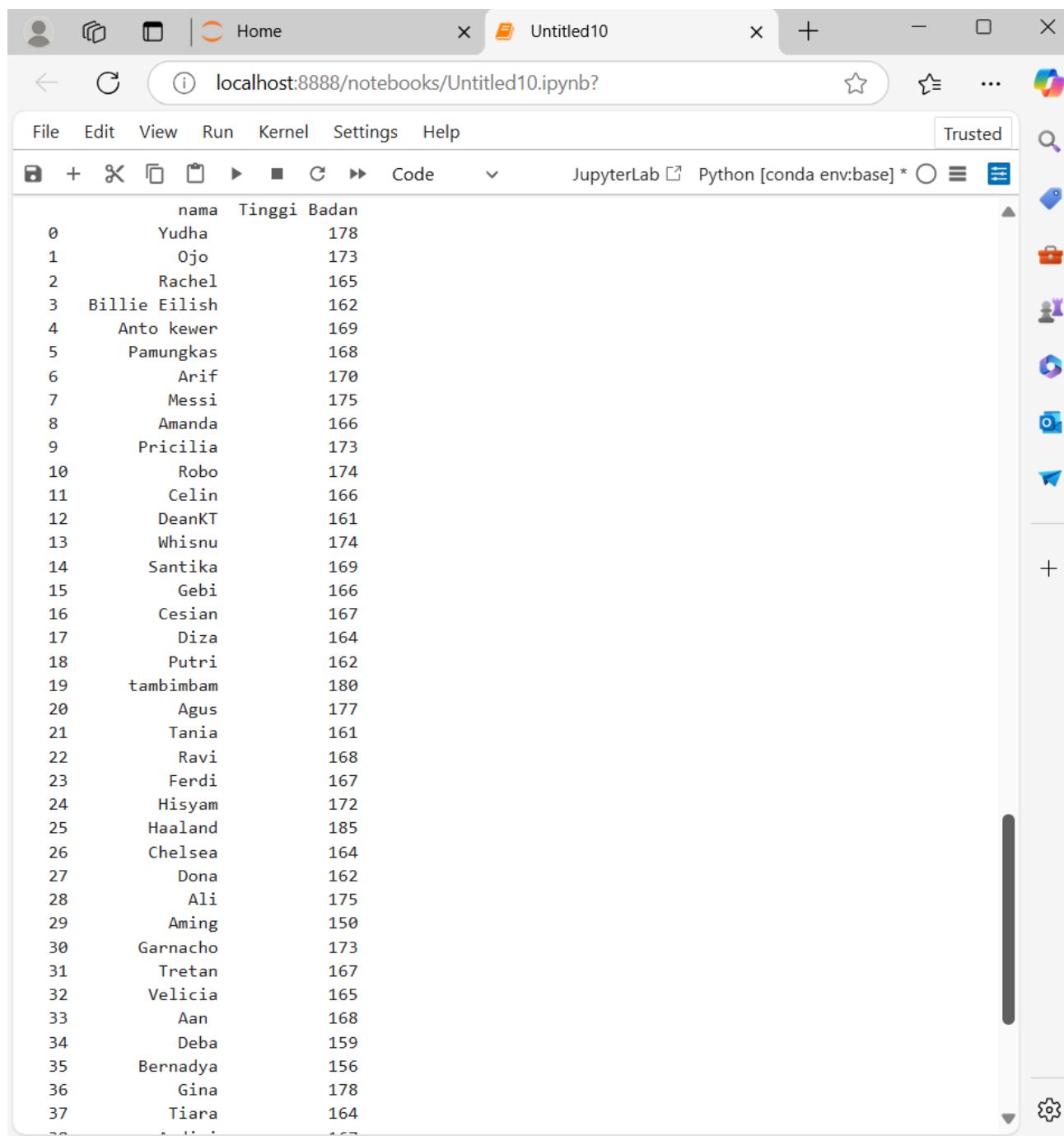


The screenshot shows a Jupyter Notebook interface with the title "Untitled10". The code cell [24] contains the following Python code:

```
[24]: import pandas as pd
df = pd.read_clipboard()
df.columns = df.columns.str.strip()
#Mengambil subset hanya dengan kolom "Nama" dan "Tinggi Badan"
subset_tinggi_badan = df[["Nama", "Tinggi Badan"]]
subset_tinggi_badan.to_clipboard(index=False)
print(subset_tinggi_badan)
```

The code imports pandas, reads data from clipboard, strips column names, creates a subset of 'Nama' and 'Tinggi Badan', and prints the subset.

Output:



The screenshot shows a Jupyter Notebook interface with the title "Untitled10". The URL in the address bar is "localhost:8888/notebooks/Untitled10.ipynb?". The notebook contains a single code cell displaying a DataFrame:

```
nama Tinggi Badan
0 Yudha 178
1 Ojo 173
2 Rachel 165
3 Billie Eilish 162
4 Anto kewer 169
5 Pamungkas 168
6 Arif 170
7 Messi 175
8 Amanda 166
9 Pricilia 173
10 Robo 174
11 Celin 166
12 DeanKT 161
13 Whisnu 174
14 Santika 169
15 Gebi 166
16 Cesian 167
17 Diza 164
18 Putri 162
19 tambimbam 180
20 Agus 177
21 Tania 161
22 Ravi 168
23 Ferdi 167
24 Hisyam 172
25 Haaland 185
26 Chelsea 164
27 Dona 162
28 Ali 175
29 Aming 150
30 Garnacho 173
31 Tretan 167
32 Velicia 165
33 Aan 168
34 Deba 159
35 Bernadya 156
36 Gina 178
37 Tiara 164
```

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

[86]:

```
import pandas as pd

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

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

# Menyimpan DataFrame kembali ke clipboard tanpa kolom index
df.to_clipboard(index=False)

# Menampilkan DataFrame yang sudah dimodifikasi
print(df)
```

nama Tinggi Badan Kategori_Tinggi Kategori Tinggi

Output:

	nama	Tinggi Badan	Kategori_Tinggi	Kategori Tinggi
0	Yudha	178	Tinggi	Tinggi
1	Ojo	173	Tinggi	Tinggi
2	Rachel	165	Tinggi	Tinggi
3	Billie Eilish	162	Tinggi	Tinggi
4	Anto Kewer	169	Tinggi	Tinggi
5	Pamungkas	168	Tinggi	Tinggi
6	Arif	170	Tinggi	Tinggi
7	Messi	175	Tinggi	Tinggi
8	Ananda	166	Tinggi	Tinggi
9	Priscilia	173	Tinggi	Tinggi
10	Robo	174	Tinggi	Tinggi
11	Colin	166	Tinggi	Tinggi
12	DeanKT	161	Tinggi	Tinggi
13	Whisnu	174	Tinggi	Tinggi
14	Santika	169	Tinggi	Tinggi
15	Gebi	166	Tinggi	Tinggi
16	Cesian	167	Tinggi	Tinggi
17	Diza	164	Tinggi	Tinggi
18	Putri	162	Tinggi	Tinggi
19	tambahan	180	Tinggi	Tinggi
20	Agus	177	Tinggi	Tinggi
21	Tarla	161	Tinggi	Tinggi
22	Ravi	168	Tinggi	Tinggi
23	Ferdii	167	Tinggi	Tinggi
24	Hisyam	172	Tinggi	Tinggi
25	Haaland	185	Tinggi	Tinggi
26	Chelsea	164	Tinggi	Tinggi
27	Dona	162	Tinggi	Tinggi
28	Ali	175	Tinggi	Tinggi
29	Aming	158	Pendek	Pendek
30	Garnacho	173	Tinggi	Tinggi
31	Tetan	167	Tinggi	Tinggi
32	Velicia	165	Tinggi	Tinggi
33	Aan	168	Tinggi	Tinggi
34	Deba	159	Pendek	Pendek
35	Bernadya	156	Pendek	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”

A screenshot of a Jupyter Notebook interface. The code cell contains the following Python code:

```
[92]: import pandas as pd
df = pd.read_clipboard()
df['Jurusan'] = "Infor20"
df['Fakultas'] = "FTI"
print(df.head(30))
```

Output:

A screenshot of a Jupyter Notebook interface showing the output of the printed DataFrame. The output displays student data with columns: nama, Gender, Angkatan, Tinggi Badan, Waktu Perjalanan, Wilayah Tinggal, and Jurusan Fakultas.

	nama	Gender	Angkatan	Tinggi Badan	Waktu Perjalanan	Wilayah Tinggal	Jurusan Fakultas
0	Yudha	L	2024	178	10	Tanjung Duren	Infor20
1	Ojo	L	2024	173	30	Tangerang	Infor20
2	Rachel	P	2023	165	40	Sarinh	Infor20
3	Billie Eilish	P	2023	162	20	Grogol	FTI
4	Anto kewer	L	2000	169	10		
5	Pamungkas	L	2022	168	40		
6	Arif	L	2024	170	10		
7	Messi	L	2020	175	100		
8	Ananda	P	2023	166	45		
9	Pricilia	L	2023	173	25		
10	Robo	L	2023	174	30		
11	Celin	P	2024	166	15		
12	DeanKT	L	2022	161	24		
13	Whisnu	L	2021	174	100		
14	Santika	P	2019	169	23		
15	Gebi	P	2020	166	34		
16	Cesian	P	2021	167	35		
17	Diza	P	2023	164	90		
18	Putri	P	2020	162	22		
19	tambimbam	L	2024	180	40		
20	Agus	L	2024	177	90		
21	Tania	P	2024	161	43		
22	Ravi	L	2022	168	20		
23	Ferdhi	L	2023	167	10		
24	Hiyam	L	2019	172	47		
25	Haaland	L	2018	185	32		
26	Chelsea	P	2023	164	21		
27	Dona	P	2021	162	23		
28	Ali	L	2017	175	10		
29	Aming	L	2000	150	5		

5. Delete Variabel Hapus kolom Fakultas

A screenshot of a Jupyter Notebook interface. The title bar shows tabs for 'Home', 'Untitled10', 'gpt - Search', and 'Kateg'. Below the title bar, the URL 'localhost:8888/notebooks/Untitled10.ipynb?' is displayed. The main area shows a Jupyter logo and the text 'Untitled10 Last Checkpoint: 1 hour ago'. A toolbar with icons for File, Edit, View, Run, Kernel, Settings, and Help is visible. Below the toolbar, a code editor window displays the following Python code:

```
29      Susilo Infor20      FTI
[94]: df = df.drop(columns=['Fakultas'])

# Menampilkan hasil setelah kolom dihapus
print(df.head(30))
```

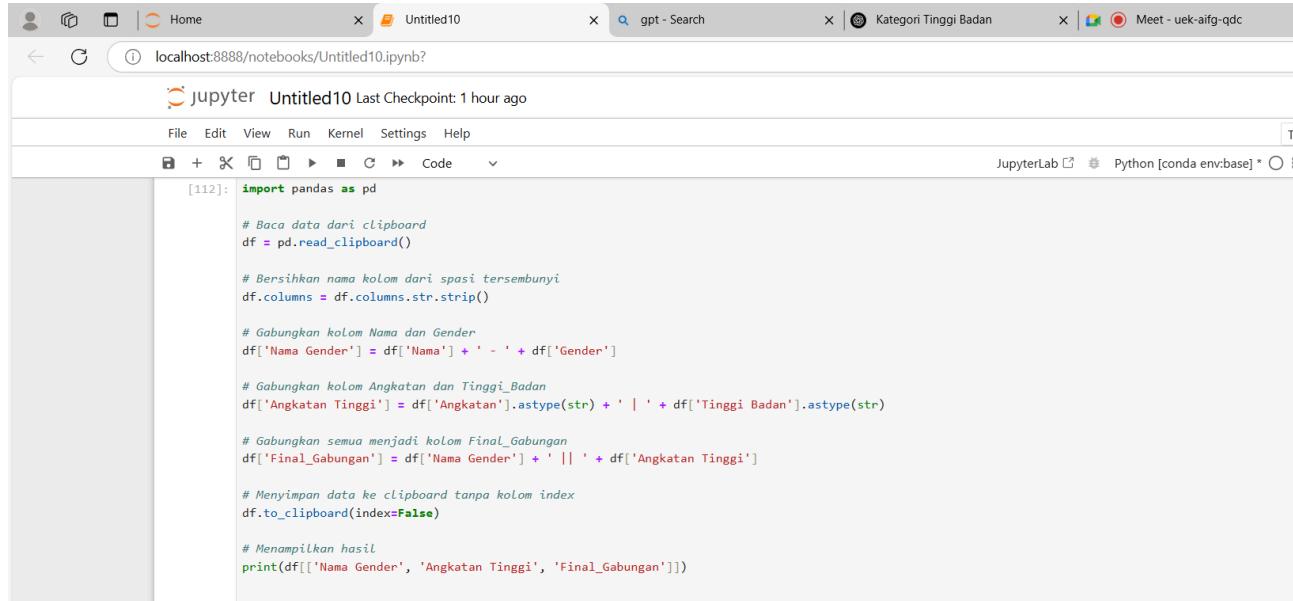
Output:

A screenshot of a Jupyter Notebook interface, identical to the one above, showing the output of the code execution. The title bar, URL, and toolbar are the same. The main area shows the Jupyter logo and the text 'Untitled10 Last Checkpoint: 1 hour ago'. The code editor window is now empty, and the output pane below it displays the following data:

	Wilayah	Tinggal	Jurusan
0	Tanjung Duren	Infor20	
1	Tangerang	Infor20	
2	Sarinah	Infor20	
3	Grogol	Infor20	
4	Grogol	Infor20	
5	Blok M	Infor20	
6	Grogol	Infor20	
7	Sukabumi	Infor20	
8	Grennvile	Infor20	
9	Tanjung Duren	Infor20	
10	Merpati	Infor20	
11	Tondo Kiri	Infor20	
12	Puri	Infor20	
13	Swiss	Infor20	
14	Central park	Infor20	
15	Jakpus	Infor20	
16	Jakpus	Infor20	
17	Palu Barat	Infor20	
18	Grogol	Infor20	
19	shhshs	Infor20	
20	Bekasi	Infor20	
21	Tondo Kiri	Infor20	
22	Tinggede	Infor20	
23	Tawakal	Infor20	
24	Bsd	Infor20	
25	Daan mogot	Infor20	
26	Kebon Jeruk	Infor20	
27	Garuda	Infor20	
28	Grennvile	Infor20	
29	Susilo	Infor20	

6. Merge Data Frame

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



```
[112]: import pandas as pd

# Baca 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']

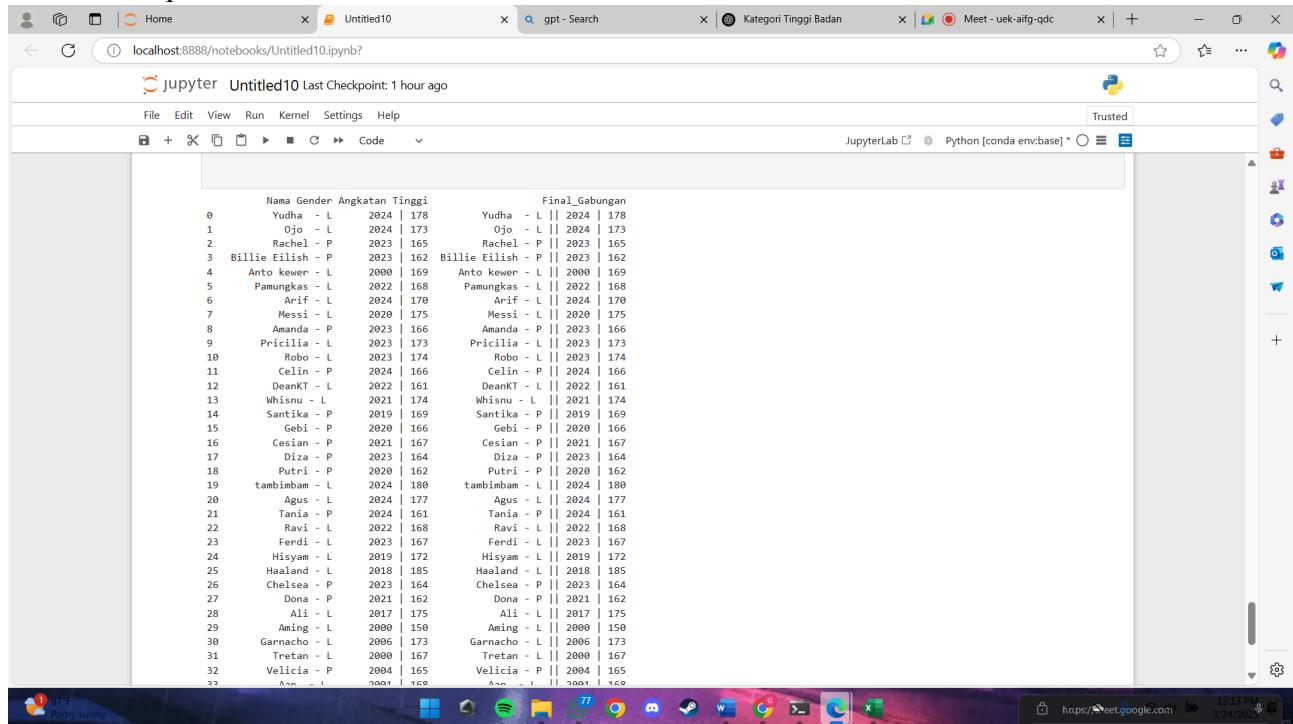
# Gabungkan kolom Angkatan dan Tinggi_Badan
df['Angkatan Tinggi'] = df['Angkatan'].astype(str) + ' | ' + df['Tinggi Badan'].astype(str)

# Gabungkan semua menjadi kolom Final_Gabungan
df['Final_Gabungan'] = df['Nama Gender'] + ' || ' + df['Angkatan Tinggi']

# Menyimpan data ke clipboard tanpa kolom index
df.to_clipboard(index=False)

# Menampilkan hasil
print(df[['Nama Gender', 'Angkatan Tinggi', 'Final_Gabungan']])
```

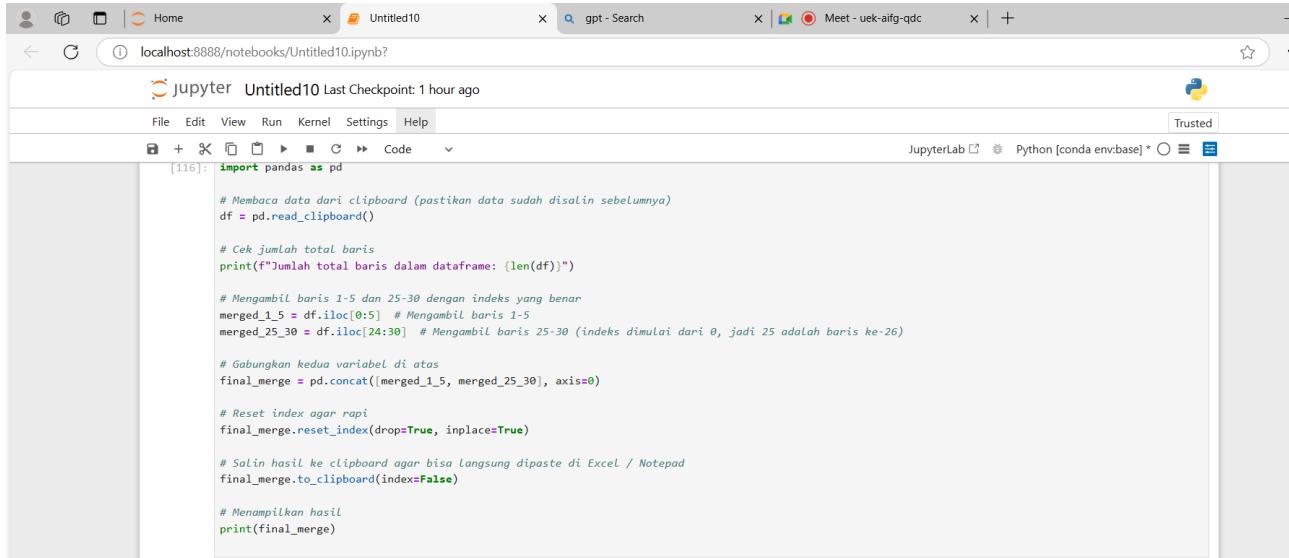
Output:



	Nama	Gender	Angkatan	Tinggi	Final_Gabungan
0	Yudha	- L	2024	178	Yudha - L 2024 178
1	Ojo	- L	2024	173	Ojo - L 2024 173
2	Rachel	- P	2023	165	Rachel - P 2023 165
3	Billie Eilish	- P	2023	162	Billie Eilish - P 2023 162
4	Anto Kewer	- L	2000	169	Anto kewer - L 2000 169
5	Pamungkas	- L	2022	168	Pamungkas - L 2022 168
6	Arif	- L	2024	170	Arif - L 2024 170
7	Messi	- L	2020	175	Messi - L 2020 175
8	Amanda	- P	2023	166	Amanda - P 2023 166
9	Pricia	- L	2023	173	Pricia - L 2023 173
10	Robo	- L	2023	174	Robo - L 2023 174
11	Celin	- P	2024	166	Celin - P 2024 166
12	DeanKT	- L	2022	161	DeanKT - L 2022 161
13	Whisnu	- L	2021	174	Whisnu - L 2021 174
14	Santika	- P	2019	169	Santika - P 2019 169
15	Gebi	- P	2020	166	Gebi - P 2020 166
16	Cesian	- P	2021	167	Cesian - P 2021 167
17	Diza	- P	2023	164	Diza - P 2023 164
18	Putri	- P	2020	162	Putri - P 2020 162
19	tambimbam	- L	2024	180	tambimbam - L 2024 180
20	Agus	- L	2024	177	Agus - L 2024 177
21	Tania	- P	2024	161	Tania - P 2024 161
22	Ravi	- L	2022	168	Ravi - L 2022 168
23	Ferd	- L	2023	167	Ferd - L 2023 167
24	Hisyam	- L	2019	172	Hisyam - L 2019 172
25	Haaland	- L	2018	185	Haaland - L 2018 185
26	Chelsea	- P	2023	164	Chelsea - P 2023 164
27	Dona	- P	2021	162	Dona - P 2021 162
28	Ali	- L	2017	175	Ali - L 2017 175
29	Aming	- L	2000	150	Aming - L 2000 150
30	Garnacho	- L	2006	173	Garnacho - L 2006 173
31	Tretan	- L	2000	167	Tretan - L 2000 167
32	Velicia	- P	2004	165	Velicia - P 2004 165
33	Ann	- L	2001	169	Ann - L 2001 169

7. Merge Baris

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



```
[116]: 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] # Mengambil baris 1-5
merged_25_30 = df.iloc[24:30] # Mengambil baris 25-30 (indeks dimulai dari 0, jadi 25 adalah baris ke-26)

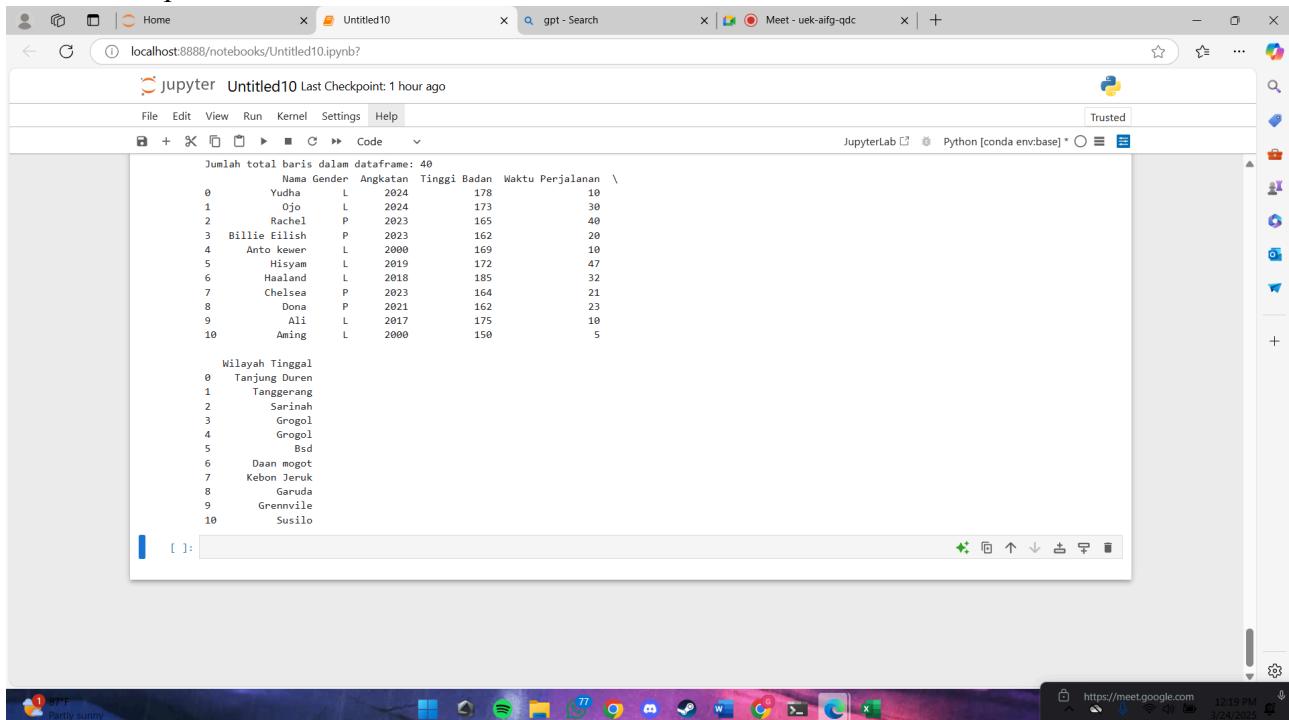
# Gabungkan kedua variabel di atas
final_merge = pd.concat([merged_1_5, merged_25_30], axis=0)

# 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
0	Yudha	L	2024	178	10
1	Ojo	L	2024	173	30
2	Rachel	P	2023	165	40
3	Billie Eilish	P	2023	162	20
4	Anto Kewer	L	2000	169	10
5	Hisyam	L	2019	172	47
6	Haaland	L	2018	185	32
7	Chelsea	P	2023	164	21
8	Dona	P	2021	162	23
9	Ali	L	2017	175	10
10	Aming	L	2000	150	5

	Wilayah Tinggal
0	Tanjung Duren
1	Tangerang
2	Sarinah
3	Grogol
4	Grogol
5	Bsd
6	Daan Mogot
7	Kebon Jeruk
8	Garuda
9	Grenville
10	Susilo

8. Sort Data Frame Lakukan sort berdasarkan waktu perjalanan

A screenshot of a Jupyter Notebook interface. The code cell [120] contains Python code for reading data from clipboard, merging rows, and printing the result. The output shows a DataFrame with columns: Nama, Gender, Angkatan, Tinggi Badan, and Waktu Perjalanan.

```
[120]: 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 (ini sudah benar)

# 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:

A screenshot of a Jupyter Notebook interface showing the output of the previous code cell. The output displays a DataFrame with 10 rows and 5 columns: Nama, Gender, Angkatan, Tinggi Badan, and Waktu Perjalanan. Below the DataFrame, a list of locations is shown under the heading "Wilayah Tinggal".

```
Jumlah total baris dalam dataframe: 40
   Nama Gender Angkatan Tinggi Badan Waktu Perjalanan \
0    Yudha      L    2024      178       10
1     Ojo       L    2024      173       30
2   Rachel      P    2023      165       40
3 Billie Eilish      P    2023      162       20
4   Anto Kewer      L    2000      169       10
5    Hisyam      L    2019      172       47
6   Haaland      L    2018      185       32
7   Chelsea      P    2023      164       21
8     Dona       P    2021      162       23
9      Ali       L    2017      175       10
10    Aming      L    2000      150        5

   Wilayah Tinggal
0   Tanjung Duren
1   Tanggerang
2    Sarinah
3     Grogol
4     Grogol
5       Bsd
6   Daan mogot
7   Kebon Jeruk
8     Garuda
9   Grenville
10   Susilo
```

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.Untuk membaca dan meyimpan data

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

6. Kesimpulan

- a. Dalam pengerjaan praktikum Statistika, kita belajar tentang Pengelolaan DataFrame yang dimana Pengelolaan DataFrame adalah salah satu aspek penting dalam analisis data menggunakan **pandas** di Python. DataFrame adalah struktur data dua dimensi yang menyerupai tabel, dengan baris dan kolom.
- b. Kita juga dapat mengetahui cara Menganalisis data dengan lebih dalam,kita juga bisa menggunakan fungsi dan metode spesifik untuk membersihkan data,ada visualisasi data dengan menggunakan matplotlib atau seaborn

7. Cek List (✓)

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

8. Formulir Umpam Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	30 Menit	Baik
2.	Latihan Kedua	30 Menit	Baik

Keterangan:

1. Menarik

- 2. Baik
- 3. Cukup
- 4. Kurang