


Nama: Kevin Rizky Pradana NIM: 065002300026	 Praktikum Statistika	MODUL 2 Nama Dosen: Dedy Sugiarto
Hari/Tanggal: Rabu, 13 Maret 2024		Nama Asisten Labratorium: 1. Tarum Widyasti 064002200027 2. Kharisma Maulida 064002200024

Tipe Data, Filter Data & Koneksi ke Database

1. Teori Singkat

Python memiliki beberapa tipe data dasar, di antaranya:

- Integer (int): Representasi bilangan bulat, misalnya: 5, -3, 100.
- Float (float): Representasi bilangan pecahan, misalnya: 3.14, 2.718.
- String (str): Urutan karakter, misalnya: 'hello', "world".
- Boolean (bool): Representasi nilai kebenaran, yaitu True atau False.
- List (list): Kumpulan elemen yang terurut dan dapat diubah, misalnya: [1, 2, 3, 4], ['apple', 'banana', 'cherry'].
- Tuple (tuple): Kumpulan elemen yang terurut dan tidak dapat diubah, misalnya: (1, 2, 3), ('red', 'green', 'blue').
- Dictionary (dict): Kumpulan pasangan kunci-nilai yang tidak terurut, misalnya: {'name': 'John', 'age': 30}.
- Set (set): Kumpulan elemen yang unik dan tidak terurut, misalnya: {1, 2, 3, 4}.

Filter Data dalam Python:

Untuk melakukan filter data dalam Python, Anda dapat menggunakan berbagai cara, tergantung pada struktur data yang Anda gunakan. Dalam konteks DataFrame, seperti yang digunakan dalam Pandas, Anda dapat menggunakan metode query() atau pengindeksan boolean.

2. Alat dan Bahan

Hardware : Laptop/PC

Software : R Studio

3. Elemen Kompetensi

Terdapat beberapa tipe data di Jupyter antara lain vektor, matriks dan data frame. Cantumkan setiap output yang dihasilkan dari console Jupyter, ke kolom yang sudah disediakan.

a. Latihan pertama – Vektor

1. Tuliskan Perintah berikut ini di jupyter notebook

```
a = [1, 2, -5, 0.3, 6, -2, 4] # numeric vector
b = ["one", "two", "three"] # character vector
c = [True, True, True, False, True] # logical vector
print(a)
print(b)
print(c)
```

Output:

```
In [1]: a = [1, 2, -5, 0.3, 6, -2, 4]
        b = ["one", "two", "three"]
        c = [True, True, True, False, True]
        print(a)
        print(b)
        print(c)

[1, 2, -5, 0.3, 6, -2, 4]
['one', 'two', 'three']
[True, True, True, False, True]
```

b. Latihan Kedua – Matriks

1. Seluruh kolom dalam sebuah matriks harus memiliki tipe yang sama (numerik semua, karakter semua, dll) dan memiliki panjang yang sama.

**gunakan nama variable dengan nama anda masing-masing*

```
#MATRIKS
import numpy as np
cells = [3, 15, -27, 38]
r_nama = ["R1", "R2"]
c_nama = ["C1", "C2"]
nama_matrix = np.matrix(cells).reshape(2, 2)
print(nama_matrix)
```

Output:

```
In [3]: #MATRIKS
import numpy as np
cells = [3, 15, -27, 38]
r_Kevin = ["R1", "R2"]
c_Kevin = ["C1", "C2"]
nama_matrix = np.matrix(cells).reshape(2, 2)
print(nama_matrix)
```

```
[[ 3 15]
 [-27 38]]
```

c. Latihan Ketiga – Data Frame

1. Mengubah data input menjadi data frame

*gunakan nama variable dengan nama anda masing-masing

```
import pandas as pd
import numpy as np

nama1 = [1, 2, 3, 4]
nama2 = ["red", "white", "red", np.nan] # Menggunakan np.nan untuk
merepresentasikan NA
nama3 = [True, True, True, False]

dataku = pd.DataFrame({'ID': nama1, 'Color': nama2, 'Passed': nama3})
print(dataku)
```

Output:

```
In [4]: import pandas as pd
import numpy as np

nama1 = [1, 2, 3, 4]
nama2 = ["red", "white", "red", np.nan] # Menggunakan np.nan untuk merepresentasikan NA
nama3 = [True, True, True, False]

dataku = pd.DataFrame({'ID': nama1, 'Color': nama2, 'Passed': nama3})
print(dataku)
```

	ID	Color	Passed
0	1	red	True
1	2	white	True
2	3	red	True
3	4	NaN	False

2. Selanjutnya ketikkan perintah dibawah ini

```
import pandas as pd

data_nama = pd.DataFrame({'id': list('abcdefghij'), 'x': list(range(1, 11)), 'y': list(range(11, 21))})
print(data_nama)
```

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

Output:

```
[5]: import pandas as pd

data_nama = pd.DataFrame({'id': list('abcdefghij'), 'x': list(range(1, 11)), 'y': list(range(11, 21))})
print(data_nama)
```

	id	x	y
0	a	1	11
1	b	2	12
2	c	3	13
3	d	4	14
4	e	5	15
5	f	6	16
6	g	7	17
7	h	8	18
8	i	9	19
9	j	10	20

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

d. Latihan Keempat – Koneksi ke Database

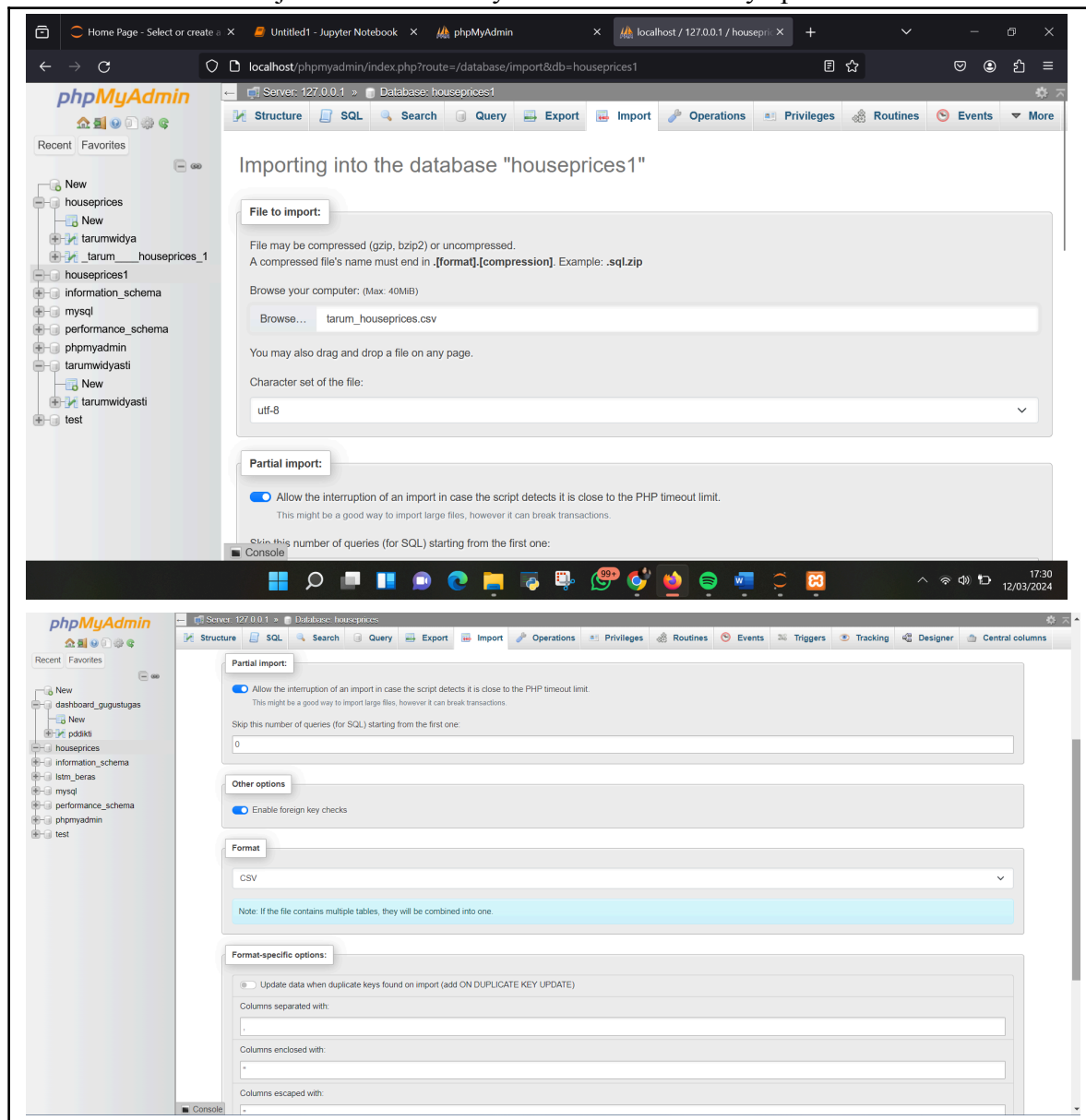
1. Buat sebuah nama database terlebih dahulu dengan nama houseprices di phpmyadmin, Lalu klik menu import

Start Apache* & MySQL, Buka browser, ketik <http://localhost/phpmyadmin/>

The image shows two screenshots. The top screenshot is the XAMPP Control Panel v3.3.0. It displays a table of modules with columns: Service, Module, PID(s), Port(s), and Actions. The 'Apache' and 'MySQL' services are highlighted in green, indicating they are running. The 'Actions' column for each service includes buttons for 'Stop', 'Admin', 'Config', and 'Logs'. The 'MySQL' 'Admin' button is highlighted with a red box. Below the table, a log shows the status of the services: 'All prerequisites found', 'Initializing Modules', 'Starting Check-Timer', 'Control Panel Ready', 'Attempting to start Apache app...', 'Status change detected: running', 'Attempting to start MySQL app...', and 'Status change detected: running'.

The bottom screenshot is the phpMyAdmin interface. The 'Import' tab is selected. The 'File to Import' section shows a file named 'houseprices.csv' selected. The 'Character set of the file' is set to 'utf-8'. The 'Partial Import' section has the 'Allow the interruption of an import' checkbox checked. The 'Other options' section has the 'Enable foreign key checks' checkbox checked. The 'Format' section is empty.

2. Pilih file yang ingin di import ke database (untuk file nama_excel.csv), Ceklis the first line of the file contains the table column name untuk membuat baris pertama pada file excel tersebut menjadi nama atributnya atau nama kolomnya pada database.



phpMyAdmin

Server: 127.0.0.1 Database: houseprices

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns

Note: If the file contains multiple tables, they will be combined into one.

Format-specific options:

☐ Update data when duplicate keys found on import (add ON DUPLICATE KEY UPDATE)

Columns separated with:

Columns enclosed with:

Columns escaped with:

Lines terminated with:

auto

Name of the new table (optional):

Import these many number of rows (optional):

☐ The first line of the file contains the table column names (if this is unchecked, the first line will become part of the data)

☐ Do not abort on INSERT error

Import

Console

phpMyAdmin

Server: 127.0.0.1 Database: houseprices

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns

Import has been successfully finished. 2 queries executed.

The following structures have either been created or altered. Here you can:

- View a structure's contents by clicking on its name.
- Change any of its settings by clicking the corresponding "Options" link.
- Edit structure by following the "Structure" link.

houseprices (Options)

houseprices (Structure) (Options)

(houseprices.csv)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

CREATE TABLE IF NOT EXISTS `houseprices`.`houseprices` (`COL 1` varchar(6), `COL 2` varchar(4), `COL 3` varchar(8), `COL 4` varchar(9), `COL 5` varchar(6), `COL 6` varchar(5), `COL 7` varchar(12))
DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

[Edit inline] [Edit] [Create PHP code]

129 rows inserted. (Query took 0.0004 seconds.)

INSERT INTO `houseprices`.`houseprices` (`COL 1`, `COL 2`, `COL 3`, `COL 4`, `COL 5`, `COL 6`, `COL 7`) VALUES ('Price', 'SqFt', 'Bedrooms', 'Bathrooms', 'Offers', 'Brick', 'Neighborhood'), ('114300', '1790', '2', '2', '2', 'No', 'East'), ('114200', '2030', '4', '2', '3', 'No', 'East'), ('114800', '1740', '3', '2', '1', 'No', 'East'), ('94700', '1980', '3', '2', '3', 'No', 'East'), ('119800', '2130', '3', '3', '3', 'No', 'East'), ('114600', '1780', '3', '2', '2', 'No', 'North'), ('151600', '1830', '3', '3', '3', 'Yes', 'West'), ('150700', '2160', '4', '2', '2', 'No', 'West'), ('119200', '2110', '4', '2', '3', 'No', 'East'), ('104000', '1730', '3', '3', '3', 'No', 'East'), ('132500', '2030', '3', '2', '3', 'Yes', 'East'), ('123000', '1870', '2', '2', '2', 'Yes', 'East'), ('102600', '1910', '3', '2', '4', 'No', 'North'), ('126300', '2150', '3', '3', '5', 'Yes', 'North'), ('176800', '2590', '4', '3', '4', 'No', 'West'), ('145800', '1780', '4', '2', '1', 'No', 'West'), ('11[...]

[Edit]

Console

3. Klik go, Rename nama tabel sesuai dengan nama anda

The first screenshot shows the phpMyAdmin interface with the 'houseprices' table selected. The 'Alter table order by' section is visible, showing 'COL 1' as the primary key. The 'Move table to (database.table)' section shows the table being moved to the 'houseprices' database. The 'Table options' section shows the table being renamed to 'houseprices', with 'Add AUTO_INCREMENT value' and 'Adjust privileges' checked. The 'Storage engine' is set to 'InnoDB' and the 'Collation' is 'utf8_general_ci'.

The second screenshot shows the phpMyAdmin interface with the 'houseprices' table selected. The 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' message is displayed. The 'Showing rows 0 - 24 (129 total. Query took 0.0001 seconds.)' message is shown. The SQL query 'SELECT * FROM `houseprices`' is displayed. The 'Extra options' section shows the table structure with columns: COL 1, COL 2, COL 3, COL 4, COL 5, COL 6, COL 7. The table data is displayed in a grid format.

COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
114300	1700	2	2	2	No	East
114200	2030	4	2	3	No	East
114800	1740	3	2	1	No	East
94700	1980	3	2	3	No	East
119800	2130	3	3	3	No	East
114600	1780	3	2	2	No	North
151600	1830	3	3	3	Yes	West
150700	2160	4	2	2	No	West
119200	2110	4	2	3	No	East
104000	1730	3	3	3	No	East
132500	2030	3	2	3	Yes	East
123000	1870	2	2	2	Yes	East
102600	1910	3	2	4	No	North
126300	2150	3	3	5	Yes	North
170800	2590	4	3	4	No	West
145800	1780	4	2	1	No	West
147100	2190	3	3	4	Yes	East
83600	1990	3	3	4	No	North
111400	1700	2	2	1	Yes	East
102600	1910	3	2	4	No	North

4. Kembali ke jupyter notebook, lalu instal dahulu library yang dibutuhkan pada python. Jika belum tersedia, maka lakukan instruksi:

pip install mysql-connector-python

```
[6]: pip install mysql-connector-python

Collecting mysql-connector-python
  Obtaining dependency information for mysql-connector-python from https://files.pythonhosted.org/packages/d9/91/007a0d60fee8db4f7385075dc50bf62d2d359b417b374ec06b06ce6c2d64/mysql_connector_python-8.3.0-cp311-cp311-win_amd64.whl.metadata
    Downloading mysql_connector_python-8.3.0-cp311-cp311-win_amd64.whl.metadata (2.0 kB)
    Downloading mysql_connector_python-8.3.0-cp311-cp311-win_amd64.whl (15.4 MB)
      ----- 0.0/15.4 MB ? eta :--:
      ----- 0.0/15.4 MB 660.6 kB/s eta 0:00:24
```

5. Lalu jalankan perintah dibawah ini

```
import mysql.connector

# Membuat koneksi ke MySQL
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="houseprices"
)

# Membuat objek cursor untuk mengeksekusi kueri
cursor = connection.cursor()

try:
    # Mengeksekusi kueri SQL
    my_query = "SELECT * FROM nama;"
    cursor.execute(my_query)

    # Mengambil semua hasil kueri
    result = cursor.fetchall()

    # Menampilkan hasil kueri
    print("\nHasil Kueri:")
    for row in result:
        print(row)

finally:
    # Menutup kursor dan koneksi
    cursor.close()
    connection.close()
```

Output:

```
In [8]: import mysql.connector

# Membuat koneksi ke MySQL
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="houseprices"
)

# Membuat objek cursor untuk mengeksekusi kueri
cursor = connection.cursor()

try:
    # Mengeksekusi kueri SQL
    my_query = "SELECT * FROM Kevin_houseprices;"
    cursor.execute(my_query)

    # Mengambil semua hasil kueri
    result = cursor.fetchall()

    # Menampilkan hasil kueri
    print("\nHasil Kueri:")
    for row in result:
        print(row)

finally:
    # Menutup kursor dan koneksi
    cursor.close()
    connection.close()
```

Hasil Kueri:

('Price', 'SqFt', 'Bedrooms', 'Bathrooms', 'Offers', 'Brick', 'Neighborhood')
('114300', '1790', '2', '2', '2', 'No', 'East')
('114200', '2030', '4', '2', '3', 'No', 'East')
('114800', '1740', '3', '2', '1', 'No', 'East')
('94700', '1980', '3', '2', '3', 'No', 'East')
('119800', '2130', '3', '3', '3', 'No', 'East')
('114600', '1780', '3', '2', '2', 'No', 'North')
('151600', '1830', '3', '3', '3', 'Yes', 'West')
('150700', '2160', '4', '2', '2', 'No', 'West')
('119200', '2110', '4', '2', '3', 'No', 'East')
('104000', '1730', '3', '3', '3', 'No', 'East')
('132500', '2030', '3', '2', '3', 'Yes', 'East')
('123000', '1870', '2', '2', '2', 'Yes', 'East')
('102600', '1910', '3', '2', '4', 'No', 'North')
('126300', '2150', '3', '3', '5', 'Yes', 'North')
('176800', '2590', '4', '3', '4', 'No', 'West')
('145800', '1780', '4', '2', '1', 'No', 'West')

6. Jalankan perintah dibawah ini:

***Perintah ini akan menampilkan 86 baris data hasil filter.**

Output:

```
In [19]: import pandas as pd

df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])

df_filtered = df[df['COL 6'] == 'No']

print("\nHasil Filter: ")
print(df_filtered)
```

```
Hasil Filter:
   COL 1 COL 2 COL 3 COL 4 COL 5 COL 6 COL 7
1  114300 1790    2    2    2    No  East
2  114200 2030    4    2    3    No  East
3  114800 1740    3    2    1    No  East
4   94700 1980    3    2    3    No  East
5  119800 2130    3    3    3    No  East
..  ...  ...  ...  ...  ...  ...  ...
121 110400 1930    2    3    3    No  North
122 105600 1930    3    3    3    No  East
126 113500 2070    2    2    2    No  North
127 149900 2020    3    3    1    No  West
128 124600 2250    3    3    4    No  North

[86 rows x 7 columns]
```

7. Jalankan perintah dibawah ini:

***Perintah ini akan menampilkan 105 baris data hasil filter.**

Output:

```
In [20]: import pandas as pd

df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])

df_filtered = df[(df['COL 6'] == 'No') | (df['COL 7'] == 'East')]

print("\nHasil Filter: ")
print(df_filtered)
```

Hasil Filter:

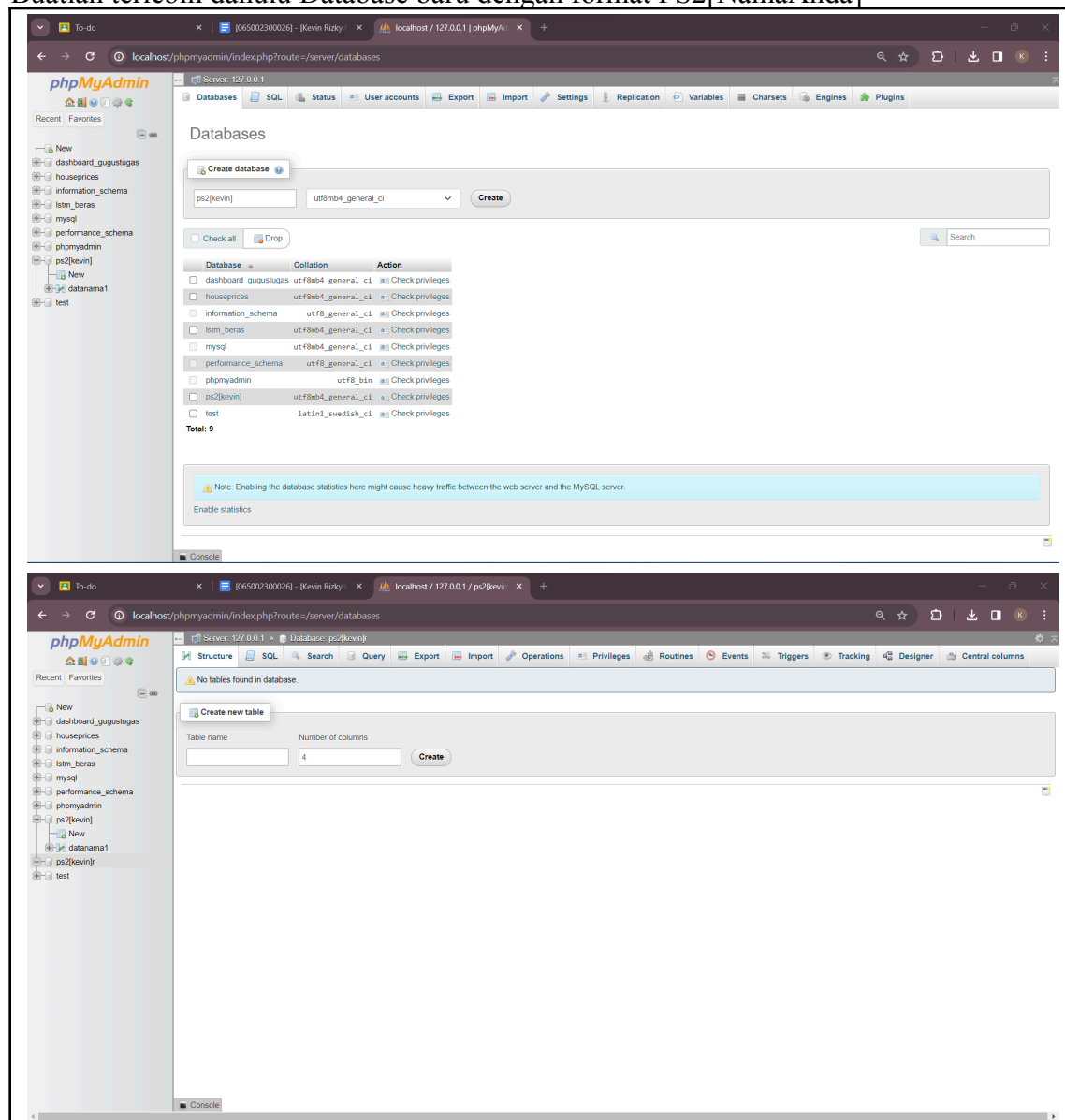
	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
1	114300	1790	2	2	2	No	East
2	114200	2030	4	2	3	No	East
3	114800	1740	3	2	1	No	East
4	94700	1980	3	2	3	No	East
5	119800	2130	3	3	3	No	East
..
124	119700	1900	3	3	3	Yes	East
125	147900	2160	4	3	3	Yes	East
126	113500	2070	2	2	2	No	North
127	149900	2020	3	3	1	No	West
128	124600	2250	3	3	4	No	North

[105 rows x 7 columns]

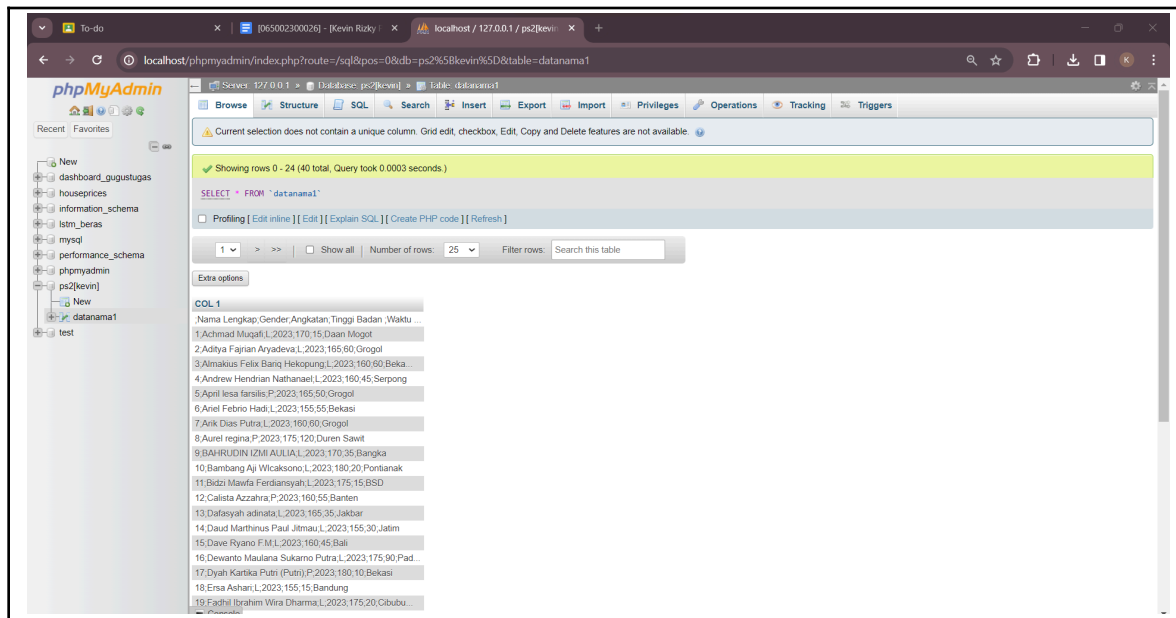
e. Latihan Keenam – Tugas

Buat sebuah database serta tabel di dalamnya (bisa gunakan data teman dipraktikum ke 1). Lakukan koneksi python ke database serta berikan beberapa filter data sesuai yang anda inginkan. Tampilkan data tersebut

1. Buatlah terlebih dahulu Database baru dengan format PS2[NamaAnda]



2. Import file berformat csv yang telah anda buat sebelumnya di Praktikum 1, dimana file tersebut berisi 6 kolom dan 20 baris data.



3. Koneksikan Python ke Database tersebut sebagaimana yang telah anda lakukan pada Elemen Kompetensi 1 dimodul kedua ini dengan menyesuaikan kembali nama Database baru yang sudah dibuat.

```
# Mengambil semua hasil kueri
result = cursor.fetchall()

# Menampilkan hasil kueri
print("\nHasil Kueri:")
for row in result:
    print(row)

finally:
    # Menutup kursor dan koneksi
    cursor.close()
    connection.close()
```

Hasil Kueri:

```
(';Nama Lengkap;Gender;Angkatan;Tinggi Badan ;Waktu Perjalanan;Wilayah tinggal,')
('1;Achmad Muqafi;L;2023;170;15;Daan Mogot',)
('2;Aditya Fajrian Aryadeva;L;2023;165;60;Grogol',)
('3;Almakius Felix Bariq Hekopung;L;2023;160;60;Bekasi',)
('4;Andrew Hendrian Nathanael;L;2023;160;45;Serpong',)
('5;April lesa farsilis;P;2023;165;50;Grogol',)
('6;Ariel Febrio Hadi;L;2023;155;55;Bekasi',)
('7;Arik Dias Putra;L;2023;160;60;Grogol',)
('8;Aurel regina;P;2023;175;120;Duren Sawit',)
('9;BAHRUDIN IZMI AULIA;L;2023;170;35;Bangka',)
('10;Bambang Aji Wicaksono;L;2023;180;20;Pontianak',)
('11;Bidzi Mawfa Ferdiansyah;L;2023;175;15;BSD',)
('12;Calista Azzahra;P;2023;160;55;Banten',)
('13;Dafasyah adinata;L;2023;165;35;Jakbar',)
('14;Daud Marthinus Paul Jitmau;L;2023;155;30;Jatim',)
('15;Dave Ryano F.M;L;2023;160;45;Bali',)
('16;Dewanto Maulana Sukarno Putra;L;2023;175;90;Padang',)
('17;Dyah Kartika Putri (Putri);P;2023;180;10;Bekasi',)
('18;Ersa Ashari;L;2023;155;15;Bandung',)
('19;Fadhil Ibrahim Wira Dharma;L;2023;175;20;Cibubur',)
```

4. Lakukan filter data terhadap Kolom Gender, untuk melihat berapa baris data Pria/Wanita (Pilih salah 1).

```
In [5]: import pandas as pd

df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])

df_filtered = df[df['COL 1'] == 'No']

print("\nHasil Filter: ")
print(df_filtered)
```

Hasil Filter:

```
Empty DataFrame
Columns: [COL 1]
Index: []
```

5. Lampirkan Screenshot

☞ Kode koneksi Jupyter ke Database

☞ Kode serta hasil filter di Jupyter

```
import pandas as pd
```

```
df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
```



```
df_filtered = df[df['COL 1'] == 'No']  
  
print("\nHasil Filter: ")  
print(df_filtered)
```

4. File Praktikum

Github Repository:

5. Kesimpulan

- a. Dalam pengerjaan praktikum Statistika, ...

b. Kita juga dapat mengetahui...

6. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	...	
2.	Latihan Kedua	...	
3.	Latihan Ketiga	...	
4.	Latihan Keempat	...	
5.	Latihan Kelima	...	
6.	Latihan Keenam	...	

7. Formulir Umpan Balik

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

4.	Latihan Keempat	... Menit	...
5.	Latihan Kelima	... Menit	...
6.	Latihan Keenam	... Menit	...

Keterangan:

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