


Nama: Mohammad Afif R. Lingkeh NIM: 065002400002	 Praktikum Statistika	MODUL 2 Nama Dosen: Drs. Joko Riyono, M.Si.
Hari/Tanggal: Jum'at, 14 Maret 2025		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:

```
[1]: 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)

      [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:

```
[6]: #MATRIKS
import numpy as np
cells = [3, 15, -27, 38]
r_afif = ["R1", "R2"]
c_afif = ["C1", "C2"]
afif_matrix = np.matrix(cells).reshape(2, 2)
print(afif_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:

```
[8]: import pandas as pd
import numpy as np

afif1 = [1, 2, 3, 4]
afif2 = ["red", "white", "red", np.nan] # Menggunakan np.nan untuk merepresentasikan NA
afif3 = [True, True, True, False]

data_afif = pd.DataFrame({'ID': afif1, 'Color': afif2, 'Passed': afif3})
print(data_afif)
```

	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)
```

Output:

```
[9]: import pandas as pd

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

	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

Klik **start** (Apache dan Mysql), lalu klik **Admin**

The screenshot shows the XAMPP Control Panel v3.3.0 and the phpMyAdmin interface. The XAMPP Control Panel is at the top, showing the status of various services. The phpMyAdmin interface is below it, showing the 'Basis data' (Databases) section.

XAMPP Control Panel v3.3.0

Module	Port(s)	Actions
Apache	80, 443	Start, Stop, Admin, Config, Logs
MySQL	3306	Start, Stop, Admin, Config, Logs
FileZilla		Start, Stop, Admin, Config, Logs
Mercury		Start, Stop, Admin, Config, Logs
Tomcat		Start, Stop, Admin, Config, Logs

phpMyAdmin

Basis data

Buat basis data

Houseprices utf8mb4_general_ci [Buat]

Pilih Semua Hapus

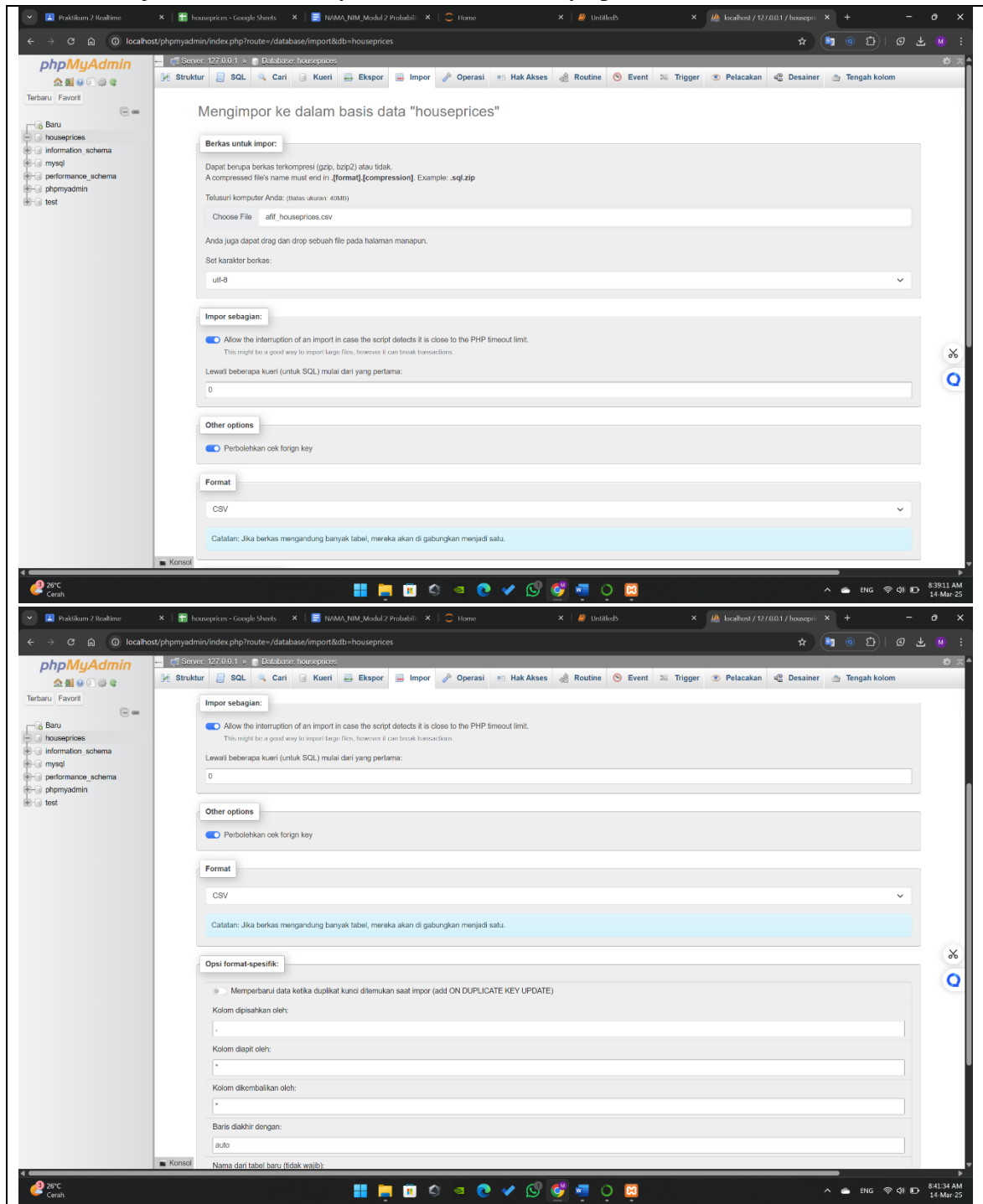
Basis data	Penyortiran	Tindakan
information_schema	utf8_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
test	latin1_swedish_ci	Check privileges

Total: 5

Perhatian: Pengaktifan statistik basis data dapat mengakibatkan lalu lintas besar antara webserver dan server MySQL.

Aktifkan Statistik

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.



The screenshot displays the phpMyAdmin interface for a database named 'houseprices'. The 'Import' tab is active, showing the 'Format' section set to 'CSV'. A note indicates that if the file contains multiple tables, they will be combined. The 'Options' section includes checkboxes for 'Memperbarui data ketika duplikat kunci ditemukan saat impor (add ON DUPLICATE KEY UPDATE)', 'Kolon dipisahkan oleh:' (set to comma), 'Kolon diisi oleh:' (set to empty string), 'Kolon dikembalikan oleh:' (set to empty string), 'Baris diakhiri dengan:' (set to 'auto'), 'Nama dari tabel baru (tidak wajib):', and 'Impor seberapa banyak baris ini (optional):'. The 'Import' button is at the bottom.

Below the import screen, the console shows the results of the import process:

- Import telah selesai, 2 kueri dilakukan.
- 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 struktur dengan mengklik link "Struktur".
- houseprices (Costi)
 - afif_houseprices (Struktur) (Costi)
- (afif_houseprices.csv)
- MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0.0027 detik.)
- CREATE TABLE IF NOT EXISTS 'houseprices'. 'afif_houseprices' ('Price' int(6), 'Sqft' int(4), 'Bedrooms' int(1), 'Bathrooms' int(1), 'Offers' int(1), 'Brick' varchar(3), 'Neighborhood' varchar(5)) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
- [Edit dikotak] [Ubah] [Buat kode PHP]
- 128 baris ditambahkan. (Pencarian dilakukan dalam 0.0004 detik.)
- INSERT INTO 'houseprices'. 'afif_houseprices' ('Price', 'Sqft', 'Bedrooms', 'Bathrooms', 'Offers', 'Brick', 'Neighborhood') VALUES (114300, 1790, 2, 2, 2, 'No', 'East'), (114200, 2030, 4, 2, 3, 'No', 'East'), (114800, 1740, 3, 2, 1, 'No', 'East'), (94700, 1900, 3, 2, 3, 'No', 'East'), (119800, 2130, 3, 3, 3, 'No', 'North'), (114600, 1700, 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'), (176500, 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'), (167200, 1920, 3, 3, 2, 'Yes', 'West'), (116200, 1700, 3, 2, 3, 'No', 'East'), (113800, 2000, 3, 2, [...])
- [Ubah]

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

The image consists of two screenshots of the phpMyAdmin web interface. The top screenshot shows the 'Operations' tab for a table named 'aff_houseprices'. The 'Move table to (database.table)' section is active, showing the table being moved from 'houseprices' to 'aff_houseprices'. The 'Pilihan untuk tabel' (Options for table) section is also visible, showing the table name 'aff_houseprices' and the storage engine 'InnoDB'. The bottom screenshot shows the 'SQL' tab with a query executed: 'SELECT * FROM `aff_houseprices`'. The results are displayed in a table with columns: Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, and Neighborhood. The table contains 24 rows of data.

Unurutkan tabel berdasarkan

Price ☒ Menaik ☐ Menurun

Kirim

Move table to (database.table)

houseprices aff_houseprices

☒ Tambahkan nilai AUTO_INCREMENT

☒ Adjust privileges

Kirim

Pilihan untuk tabel

Ubah nama tabel menjadi aff_houseprices ☒ Adjust privileges

Komentar tabel

Storage engine InnoDB

Penyortiran utf8_general_ci ☐ Change all column collations

ROW_FORMAT DYNAMIC

Kirim

Copy table to (database.table)

houseprices aff_houseprices

Kirim

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Menampilkan baris 0 - 24 (total 128, Pencarian dilakukan dalam 0,0010 detik.)

SELECT * FROM `aff_houseprices`

☐ Profi ☐ Edit kotak ☐ Ubah ☐ Jalankan SQL ☐ Buat kode PHP ☐ Segarkan

1 > >> ☐ Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini

extra options

Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
69100	1600	2	2	3	No	North
81300	1650	3	2	3	No	North
82300	1910	3	2	4	No	East
83800	1990	3	3	4	No	North
90300	2050	3	2	6	No	North
90500	1520	2	2	3	No	North
91100	1860	2	2	3	No	North
91700	1690	3	2	3	No	North
93600	2140	3	2	4	No	North
94700	1980	3	2	3	No	East
97800	2010	2	2	4	No	North
99300	1700	3	2	2	No	East
100900	1610	2	2	2	No	North
102500	1900	3	3	3	No	North
102600	1910	3	2	4	No	North
102700	1900	4	2	4	No	North
102800	1810	3	2	3	No	East
103200	2010	3	2	5	No	North
104000	1730	3	3	3	No	East
105600	1990	2	2	3	No	East
105600	1930	3	3	3	No	East
106100	1820	3	2	3	Yes	North
106600	1560	2	2	1	No	East
106900	1900	2	2	2	No	North
107300	1650	3	2	3	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

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

Collecting mysql-connector-python
  Downloading mysql_connector_python-9.2.0-cp312-cp312-win_amd64.whl.metadata (6.2 kB)
  Downloading mysql_connector_python-9.2.0-cp312-cp312-win_amd64.whl (16.1 MB)
----- 0.0/16.1 MB ? eta ----
----- 0.3/16.1 MB ? eta ----
----- 0.5/16.1 MB 882.6 kB/s eta 0:00:18
----- 1.3/16.1 MB 2.2 MB/s eta 0:00:07
----- 2.1/16.1 MB 2.7 MB/s eta 0:00:06
----- 2.9/16.1 MB 3.0 MB/s eta 0:00:05
----- 3.9/16.1 MB 3.3 MB/s eta 0:00:04
----- 5.0/16.1 MB 3.6 MB/s eta 0:00:04
----- 6.3/16.1 MB 3.9 MB/s eta 0:00:03
----- 7.3/16.1 MB 4.1 MB/s eta 0:00:03
----- 8.4/16.1 MB 4.2 MB/s eta 0:00:02
----- 9.2/16.1 MB 4.2 MB/s eta 0:00:02
----- 10.0/16.1 MB 4.1 MB/s eta 0:00:02
----- 10.5/16.1 MB 4.0 MB/s eta 0:00:02
----- 11.3/16.1 MB 3.9 MB/s eta 0:00:02
----- 11.8/16.1 MB 3.9 MB/s eta 0:00:02
----- 12.6/16.1 MB 3.9 MB/s eta 0:00:01
----- 14.2/16.1 MB 4.1 MB/s eta 0:00:01
----- 15.5/16.1 MB 4.2 MB/s eta 0:00:01
----- 16.1/16.1 MB 4.2 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.2.0
Note: you may need to restart the kernel to use updated packages.
```

```
[ ]:
```

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:

```
[1]: 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 afif_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:
(69100, 1600, 2, 2, 3, 'No', 'North')
(81300, 1650, 3, 2, 3, 'No', 'North')
(82300, 1910, 3, 2, 4, 'No', 'East')
(83600, 1990, 3, 3, 4, 'No', 'North')
(90300, 2050, 3, 2, 6, 'No', 'North')
(90500, 1520, 2, 2, 3, 'No', 'North')
(91100, 1860, 2, 2, 3, 'No', 'North')
(91700, 1690, 3, 2, 3, 'No', 'North')
(93600, 2140, 3, 2, 4, 'No', 'North')
(94700, 1980, 3, 2, 3, 'No', 'East')
(97800, 2010, 2, 2, 4, 'No', 'North')
(99300, 1700, 3, 2, 2, 'No', 'East')
(100900, 1610, 2, 2, 2, 'No', 'North')
(102500, 1900, 3, 3, 3, 'No', 'North')
(102600, 1910, 3, 2, 4, 'No', 'North')
(102700, 1900, 4, 2, 4, 'No', 'North')
```

```
[ ]:
```

6. Jalankan perintah dibawah ini:

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

Output:

```
[16]: import pandas as pd

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

df_filtered = df[df['Brick'] == 'No']

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

Hasil Filter:

	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	69100	1600	2	2	3	No	North
1	81300	1650	3	2	3	No	North
2	82300	1910	3	2	4	No	East
3	83600	1990	3	3	4	No	North
4	90300	2050	3	2	6	No	North
...
109	157600	2160	4	2	1	No	West
111	161300	2220	4	3	2	No	West
113	165600	2080	4	3	3	No	West
120	176800	2590	4	3	4	No	West
121	180900	2200	3	3	1	No	West

[86 rows x 7 columns]

7. Jalankan perintah dibawah ini:

*Perintah ini akan menampilkan 105 baris data hasil filter.

Output:

```
[3]: import pandas as pd

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

df_filtered = df[df['Brick'] == 'No']

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

Hasil Filter:

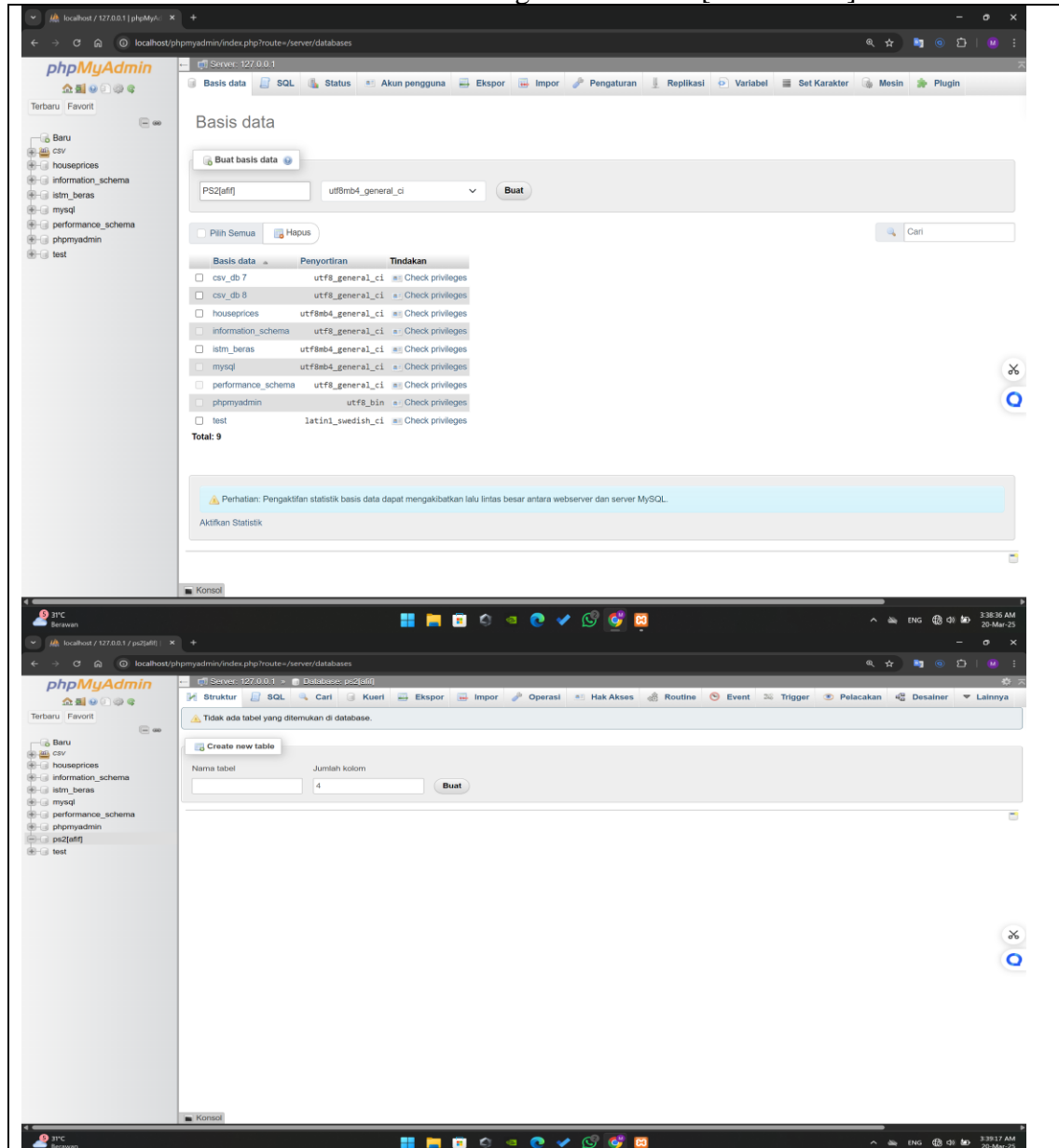
	Price	SqFt	Bedrooms	Bathrooms	Offers	Brick	Neighborhood
0	114300	1790	2	2	2	No	East
1	114200	2030	4	2	3	No	East
2	114800	1740	3	2	1	No	East
3	94700	1980	3	2	3	No	East
4	119800	2130	3	3	3	No	East
...
120	110400	1930	2	3	3	No	North
121	105600	1930	3	3	3	No	East
125	113500	2070	2	2	2	No	North
126	149900	2020	3	3	1	No	West
127	124600	2250	3	3	4	No	North

[86 rows x 7 columns]

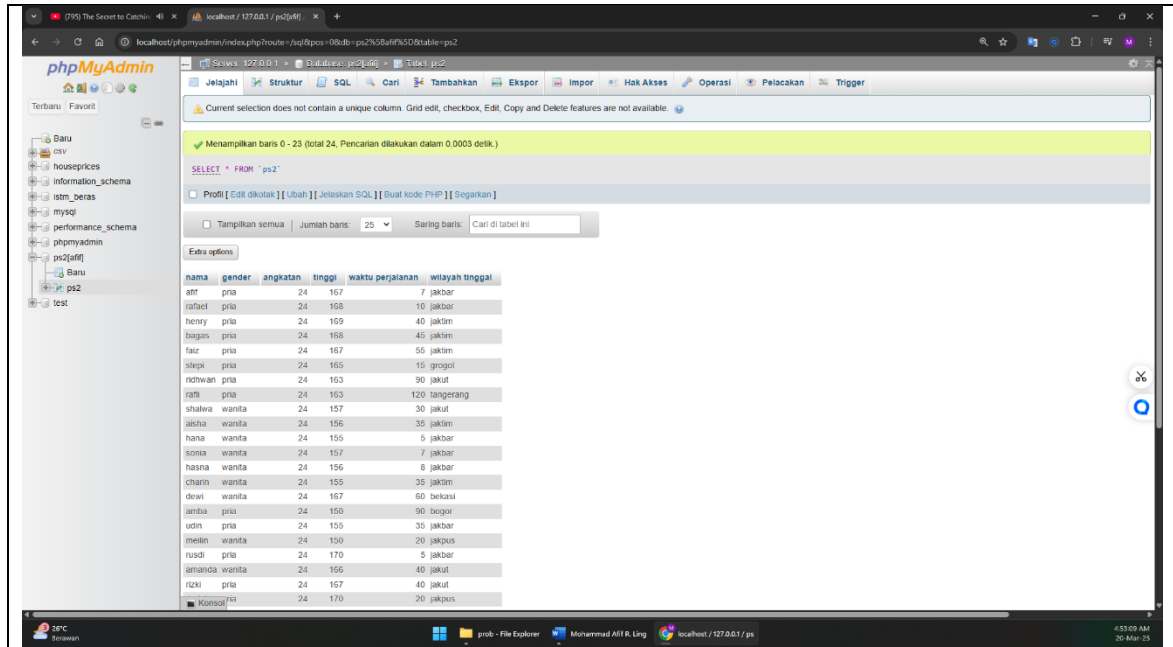
e. Latihan Keenam – Tugas

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

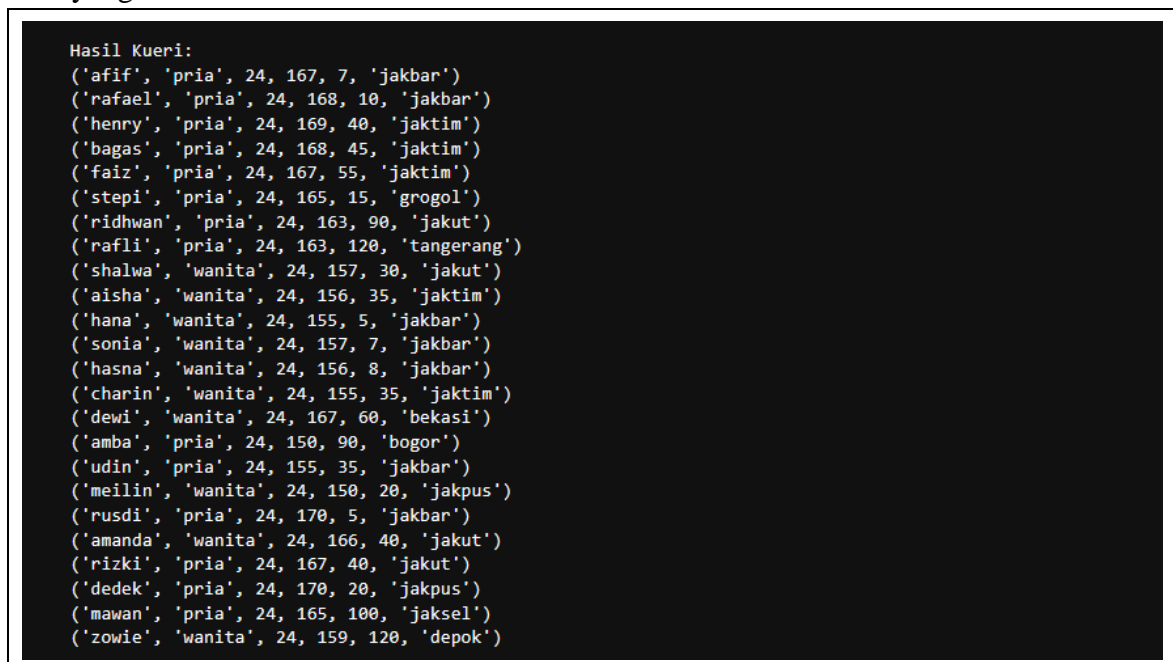
1. Buatlah terlebih dahulu Database baru dengan format PS2[NamaAnda]



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.



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

```
[10]: import pandas as pd

df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
df_filtered = df[df['gender']=='pria']
print("\nhasil filter")
print(df_filtered)
```

```
hasil filter
   nama gender  angkatan  tinggi  waktu perjalanan  wilayah  tinggal
0   afif  pria      24      167             7      jakbar
1  rafael  pria      24      168            10      jakbar
2   henry  pria      24      169            40      jaktim
3   bagas  pria      24      168            45      jaktim
4   faiz  pria      24      167            55      jaktim
5   stepi  pria      24      165            15      grogol
6  ridhwan  pria      24      163            90      jakut
7   rafli  pria      24      163           120  tangerang
15   amba  pria      24      150            90      bogor
16   udin  pria      24      155            35      jakbar
18  rusdi  pria      24      170             5      jakbar
20  rizki  pria      24      167            40      jakut
21  dedek  pria      24      170            20      jakpus
22  mawan  pria      24      165           100      jaksel
```

5. Lampirkan Screenshot

☐ Kode koneksi Jupyter ke Database

☐ Kode serta hasil filter di Jupyter

```
import mysql.connector

# Membuat koneksi ke MySQL
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="ps2[afif]"
)

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

# Mengeksekusi kueri SQL
my_query = "SELECT * FROM ps2;"
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()
```



```
[10]: import pandas as pd

df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
df_filtered = df[df['gender']=='pria']
print("\nhasil filter")
print(df_filtered)
```

```
hasil filter
   nama gender  angkatan  tinggi  waktu perjalanan  wilayah tinggal
0    afif  pria      24     167             7      jakbar
1  rafael  pria      24     168            10      jakbar
2   henry  pria      24     169            40     jaktim
3   bagas  pria      24     168            45     jaktim
4    faiz  pria      24     167            55     jaktim
5   stepi  pria      24     165            15    grogol
6  ridhwan  pria      24     163            90     jakut
7   rafli  pria      24     163           120  tangerang
15   amba  pria      24     150            90    bogor
16   udin  pria      24     155            35     jakbar
18  rusdi  pria      24     170             5     jakbar
20   rizki  pria      24     167            40     jakut
21  dedek  pria      24     170            20    jakpus
22  mawan  pria      24     165           100    jaksel
```

4. Praktikum

Github Repository:

<https://github.com/Afif-lingkeh/praktikum-probstat>

5. Kesimpulan

- Dalam pengerjaan praktikum Statistika, kita dapat mengetahui cara mengakses database menggunakan library python dan melakukan pemfilteran terhadap database
- Kita juga dapat mengetahui beberapa tipe data di jupyter antara lain vektor, matriks dan data frame, memfilter Data dan juga dapat mengkoneksi jupyter ke Database

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	2 Menit	Menarik
2.	Latihan Kedua	2 Menit	Menarik
3.	Latihan Ketiga	2 Menit	Menarik
4.	Latihan Keempat	2 Menit	Menarik
5.	Latihan Kelima	10 Menit	Menarik
6.	Latihan Keenam	10 Menit	Menarik

Keterangan:

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