

<p><b>Nama:</b> <b>Muhammad Dony</b> <b>Fatahillah</b> <b>As-Sya'bani</b></p> <p><b>NIM:</b> <b>065002300034</b></p>	 <p><b>Praktikum Statistika</b></p>	<h2>MODUL 2</h2> <p><b>Nama Dosen:</b> <b>Dedy Sugiarto</b></p> <p><b>Nama Asisten Labratorium:</b></p> <ol style="list-style-type: none"><li><b>Tarum Widyasti</b> <b>064002200027</b></li><li><b>Kharisma Maulida</b> <b>064002200024</b></li></ol>
<p><b>Hari/Tanggal:</b> <b>Rabu, 13 Maret 2024</b></p>		

## 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:

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

[[ 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 [3]: import pandas as pd
import numpy as np

dony1 = [1, 2, 3, 4]
dony2 = ["red", "white", "red", np.nan] # Menggunakan np.nan untuk merepresentasikan NA
dony3 = [True, True, True, False]

datadony = pd.DataFrame({'ID': dony1, 'Color': dony2, 'Passed': dony3})
print(datadony)
```

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

2. Selanjutnya ketikkan perintah dibawah ini

```
import pandas as pd

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

Output:

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

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

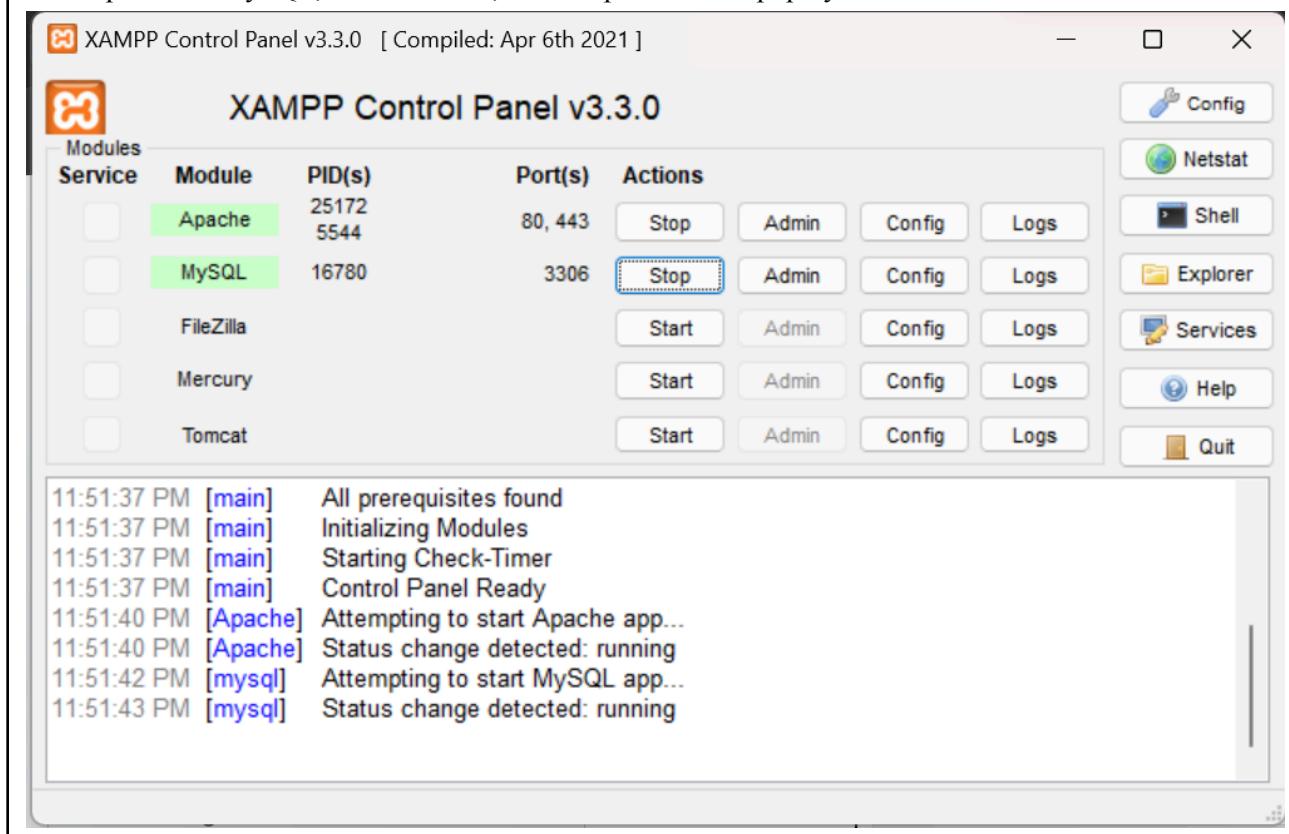
    id  x  y
0  a  1  11
1  b  2  12
2  c  3  13
3  d  4  14
4  e  5  15
5  f  6  16
6  g  7  17
7  h  8  18
8  i  9  19
9  j  10  20
```

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

d. Latihan Keempat – Koneksi ke Database

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

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

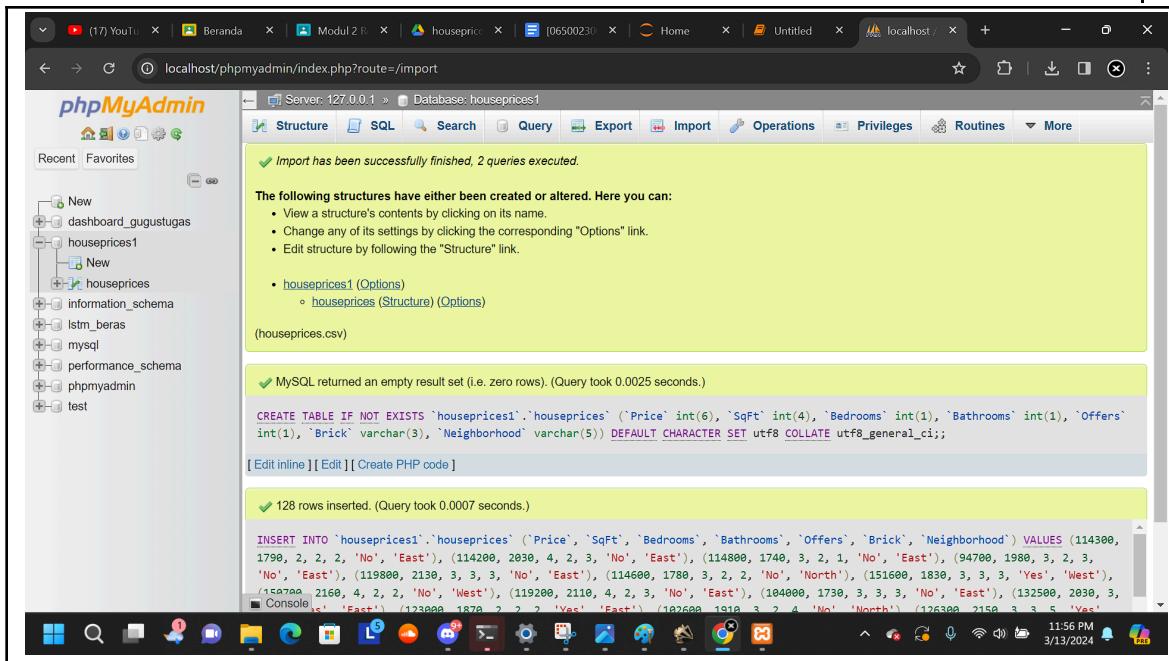


The screenshot shows the 'Databases' section of the phpMyAdmin interface. A new database named 'Houseprices1' is being created with 'utf8mb4\_general\_ci' collation. The list of databases includes 'dashboard\_gugustugas', 'information\_schema', 'lstm\_beras', 'mysql', 'performance\_schema', 'phpmyadmin', and 'test'. The status bar shows the date and time as 3/13/2024 11:52 PM.

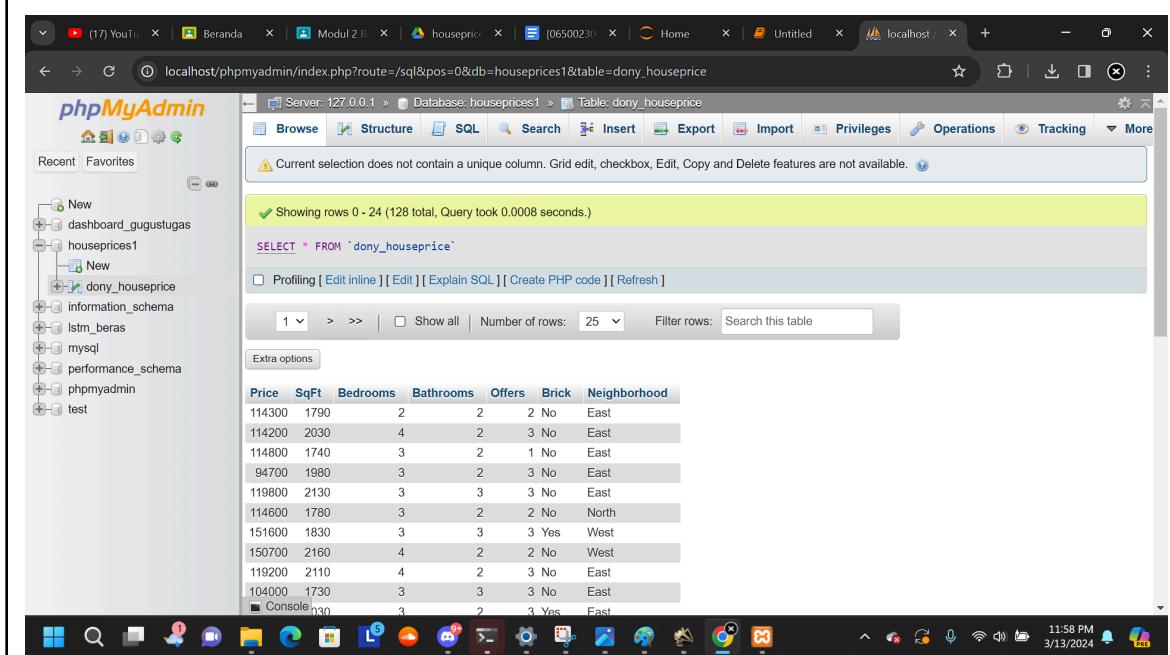
2. Pilih file yang ingin di import ke database (untuk file nama\_excel.csv), Ceklis the first line of the file contains the table column name untuk membuat baris pertama pada file excel tersebut menjadi nama atributnya atau nama kolomnya pada database.

The screenshot shows the 'Import' page for the 'houseprices1' database. The 'File to import:' section is active, showing options for compressed files and a 'Choose File' button. The 'Character set of the file:' dropdown is set to 'utf-8'. The 'Partial import:' section has a checked checkbox for allowing interruptions. The status bar shows the date and time as 3/13/2024 11:53 PM.

The screenshot shows two identical pages from the phpMyAdmin interface, both titled "localhost/phpmyadmin/index.php?route=/database/import&db=houseprices1". The top page has the "Format" dropdown set to "CSV" and the "Console" checkbox checked. The bottom page has the "Format" dropdown set to "Text" and the "Console" checkbox unchecked. Both pages include sections for "Other options" (with "Allow the interruption of an import in case the script detects it is close to the PHP timeout limit" checked), "Format-specific options" (with "Update data when duplicate keys found on import (add ON DUPLICATE KEY UPDATE)" checked), and various file import parameters like "Columns separated with:", "Columns enclosed with:", "Columns escaped with:", "Lines terminated with:", "Name of the new table (optional):", "Import these many number of rows (optional):", and checkboxes for "The first line of the file contains the table column names (if this is unchecked, the first line will become part of the data)" and "Do not abort on INSERT error".

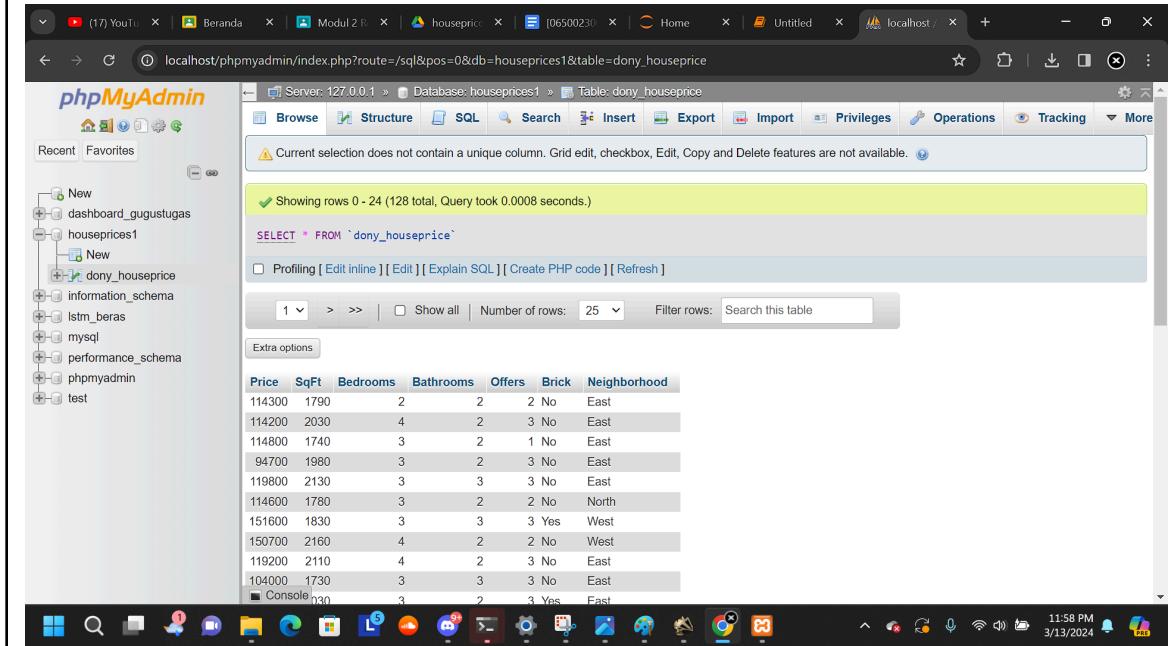


### 3. Klik go, Rename nama tabel sesuai dengan nama anda



The screenshot shows the phpMyAdmin interface for the 'houseprices1' database. The 'dony\_houseprice' table is selected. The table has columns: Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, and Neighborhood. The data shows 128 rows of house price information.

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
Console	130	3	2	3	Yes	East

The second screenshot is identical to the first one, showing the 'dony\_houseprice' table in the 'houseprices1' database. The data remains the same, indicating that the table name has not been changed yet or is still being processed.

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
Console	130	3	2	3	Yes	East

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

```
pip install mysql-connector-python
```

```
[5]: !pip install mysql-connector-python
Collecting mysql-connector-python
  Obtaining dependency information for mysql-connector-python from https://files.pythonhosted.org/packages/d9/91/007a0d60fee8db4f7385075dc50bf62d2d359b417b374eac96b06ce5c2d64/mysql_connector_python-8.3.0-cp311-cp311-win_amd64.whl.metadata
    Downloading mysql_connector_python-8.3.0-cp311-cp311-win_amd64.whl (15.4 kB)
      0.0/15.4 MB ? eta :---:-
      0.0/15.4 MB ? eta :---:-
      0.0/15.4 MB 435.7 kB/s eta 0:00:36
      0.1/15.4 MB 656.4 kB/s eta 0:00:24
      0.1/15.4 MB 656.4 kB/s eta 0:00:24
      0.1/15.4 MB 656.4 kB/s eta 0:00:24
      0.1/15.4 MB 481.4 kB/s eta 0:00:32
      0.3/15.4 MB 874.6 kB/s eta 0:00:18
      0.5/15.4 MB 1.3 MB/s eta 0:00:12
      0.6/15.4 MB 1.6 MB/s eta 0:00:10
      0.6/15.4 MB 1.5 MB/s eta 0:00:10
      0.7/15.4 MB 1.4 MB/s eta 0:00:11
      0.9/15.4 MB 1.7 MB/s eta 0:00:09
```

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:

```
[9]: import mysql.connector

# Membuat koneksi ke MySQL
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="houseprices1"
)

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

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

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

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

Hasil Kueri:

```
(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:

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

```
print("Hasil 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]

7. Jalankan perintah dibawah ini:

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

Output:

```
import pandas as pd

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

df_filtered = df[(df['Brick'] == 'No') | (df['Neighborhood'] == 'East')]

print("Hasil 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
..      ...
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

[105 rows x 7 columns]
```

e. Latihan Keenam – Tugas

Buat sebuah database serta tabel di dalamnya (bisa gunakan data teman diperlakuk 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]

The screenshot shows two instances of the phpMyAdmin interface. The top instance is on the 'Databases' page, where a new database named 'PS2[dony]' is being created with the character set 'utf8mb4\_general\_ci'. The bottom instance is on the 'Structure' page of the 'ps2(dony)' database, which currently contains no tables. A 'Create new table' dialog is open, prompting for a table name and 4 columns.

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

The screenshot shows the phpMyAdmin interface connected to a MySQL database. The left sidebar lists databases: dashboard\_gugustugas, houseprices1, information\_schema, lstim\_beras, mysql, performance\_schema, phpmyadmin, ps2(dony), New, praktikum1, test, and world\_population. The 'praktikum1' database is selected. The main area displays the 'praktikum1' table with the following data:

Nama	Gender	Angkatan	Tinggi Badan
Bambang Aji Wicaksono	L	2023	180
Aditya	L	2023	174
Rayyan	L	2023	173
Wantod	L	2023	165
Felix	L	2023	190
Vira	L	2023	200
Anggi	L	2023	181
Imam	L	2023	177

Below the table, there are buttons for 'Show all', 'Number of rows: 25', 'Filter rows: Search this table', and 'Extra options'. At the bottom, there's a 'Query results operations' section with buttons for Print, Copy to clipboard, Export, Display chart, Create view, and Console.

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.

```
In [6]: import mysql.connector

# membuat koneksi ke mysql
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='',
    database='ps2[dony]'
)

# membuat objek cursor untuk mengeksekusi query
cursor = connection.cursor()

try:
    # mengeksekusi query
    my_query = "SELECT * FROM praktikum1;"
    cursor.execute(my_query)

    # mengambil semua hasil query
    result = cursor.fetchall()

    # memanggil hasil query
    print("\nHasil Query:")
    for row in result:
        print(row)
finally:
    # menutup cursor dan koneksi
    cursor.close()
    connection.close()
```

Hasil Query:

```
('Bambang Aji Wicaksono', 'L', '2023', 180)
('Aditya', 'L', '2023', 174)
('Rayyan', 'L', '2023', 173)
('Wantod', 'L', '2023', 165)
('Felix', 'L', '2023', 190)
('Vira', 'L', '2023', 200)
('Anggi', 'L', '2023', 181)
('Imam', 'L', '2023', 177)
```

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

```
In [7]: import pandas as pd
df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
df_filtered = df[df['Gender'] == 'L']
print("\nHasil Filter")
print(df_filtered)
```

Hasil Filter

	Nama	Gender	Angkatan	Tinggi	Badan
0	Bambang Aji Wicaksono	L	2023	180	
1	Aditya	L	2023	174	
2	Rayyan	L	2023	173	
3	Wantod	L	2023	165	
4	Felix	L	2023	190	
5	Vira	L	2023	200	
6	Anggi	L	2023	181	
7	Imam	L	2023	177	

## 5. Lampirkan Screenshot

- ⌚ Kode koneksi Jupyter ke Database
- ⌚ Kode serta hasil filter di Jupyter

```
In [6]: import mysql.connector

# membuat koneksi ke mysql
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='',
    database='ps2[dony]'
)

# membuat objek cursor untuk mengeksekusi query
cursor = connection.cursor()

try:
    # mengeksekusi query
    my_query = "SELECT * FROM praktikum1;"
    cursor.execute(my_query)

    # mengambil semua hasil query
    result = cursor.fetchall()

    # memanggil hasil query
    print("\nHasil Query:")
    for row in result:
        print(row)
finally:
    # menutup cursor dan koneksi
    cursor.close()
    connection.close()
```

```
In [7]: import pandas as pd
df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
df_filtered = df[df['Gender'] == 'L']
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 Umpam 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