Nama: **NUR IMAM** **NIM:** **065002300008**		**MODUL 2** **Nama Dosen:** **Dedy Sugiarto**
Hari/Tanggal: **Rabu, 13 MARET 2024**	**Praktikum Statistika**	**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**

```
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)
```

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

Output:

```
In [3]: 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_imam = ["R1","R2"]
c_imam = ["C1", "C2"]
imam_matrix = np.matrix(cells).reshape(2,2)
print(imam_matrix)
```

Output:

```
In [5]: #MATRIKS
        import numpy as np
        cells = [3,15,-27,38]
        r_imam = ["R1","R2"]
        c_imam = ["C1", "C2"]
        imam_matrix = np.matrix(cells).reshape(2,2)
        print(imam_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

imam1 = [1, 2, 3, 4]
imam2 = ["red", "white", "red", np.nan]  # menggunakan np.nan untuk merepresentasikan NA
imam3 = [True, True, True, False]

dataku = pd.DataFrame({'ID': imam1, 'Color': imam2, 'Passed': imam3})
print(dataku)
```

Output:

```
In [11]: import pandas as pd
         import numpy as np

         imam1 = [1, 2, 3, 4]
         imam2 = ["red", "white", "red", np.nan]  # menggunakan np.nan untuk merepresentasikan NA
         imam3 = [True, True, True, False]

         dataku = pd.DataFrame({'ID': imam1, 'Color': imam2, 'Passed': imam3})
         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_imam = pd.DataFrame ({'id':list('abcdefghij'),'x':list(range(1,11)),'y':list(range(11,21))})
print(data_imam)
```

Output:

In [12]:
```
import pandas as pd
data_imam = pd.DataFrame ({'id':list('abcdefghij'),'x':list(range(1,11)),'y':list(range(11,21))})
print(data_imam)
```

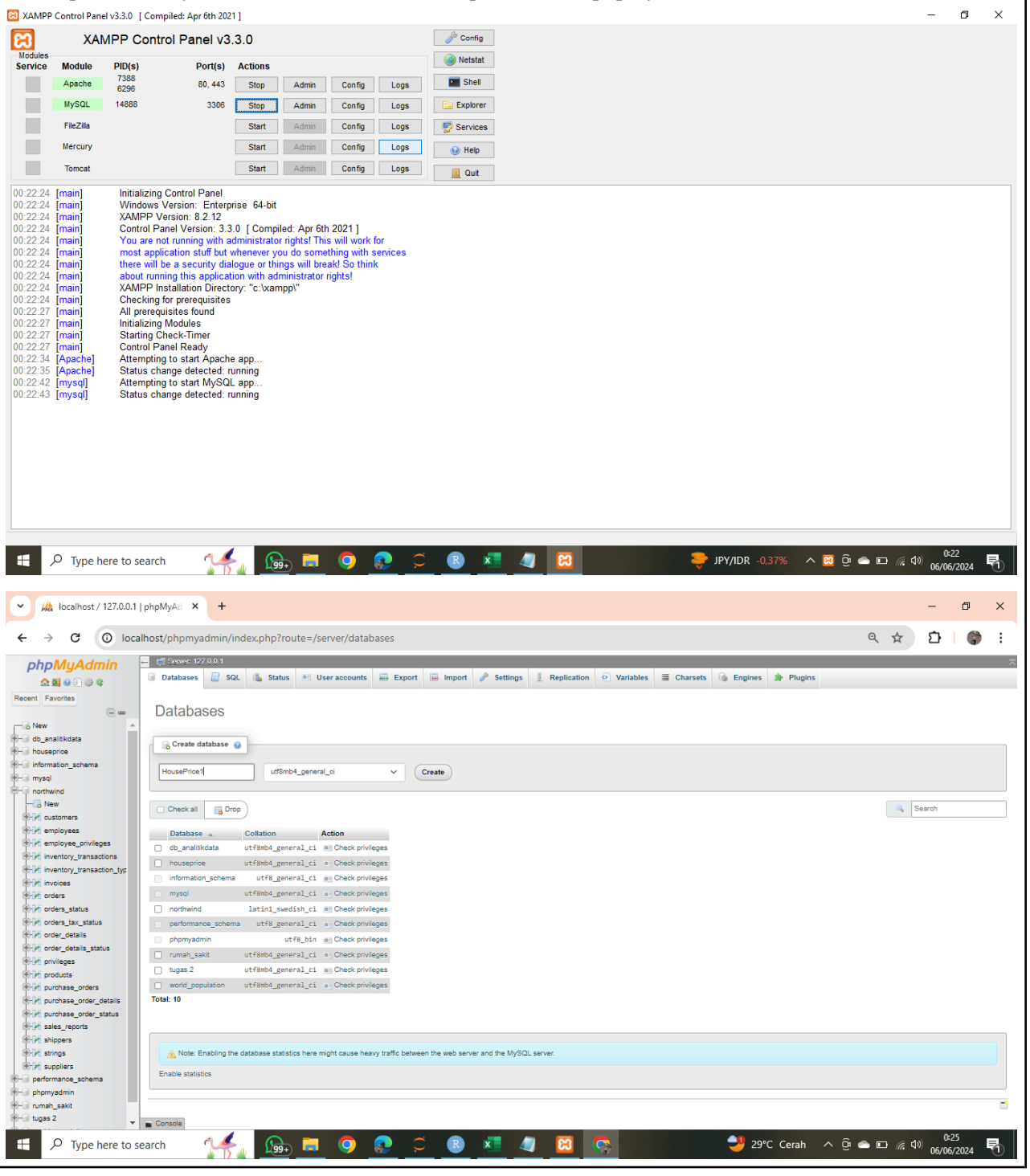
```
  id   x   y
0  a   1  11
1  b   2  12
2  c   3  13
3  d   4  14
4  e   5  15
5  f   6  16
6  g   7  17
7  h   8  18
8  i   9  19
9  j  10  20
```

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

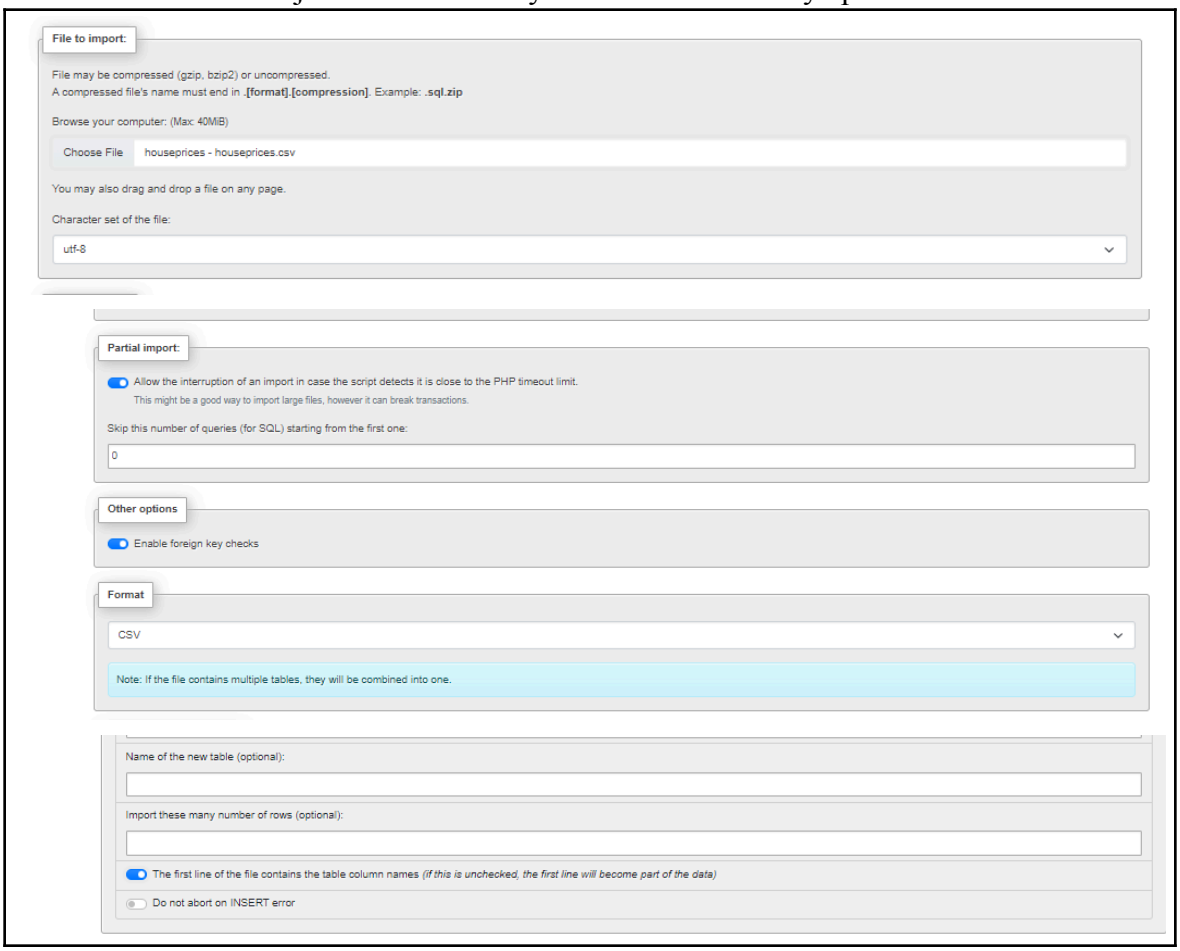
d. Latihan Keempat – Koneksi ke Database

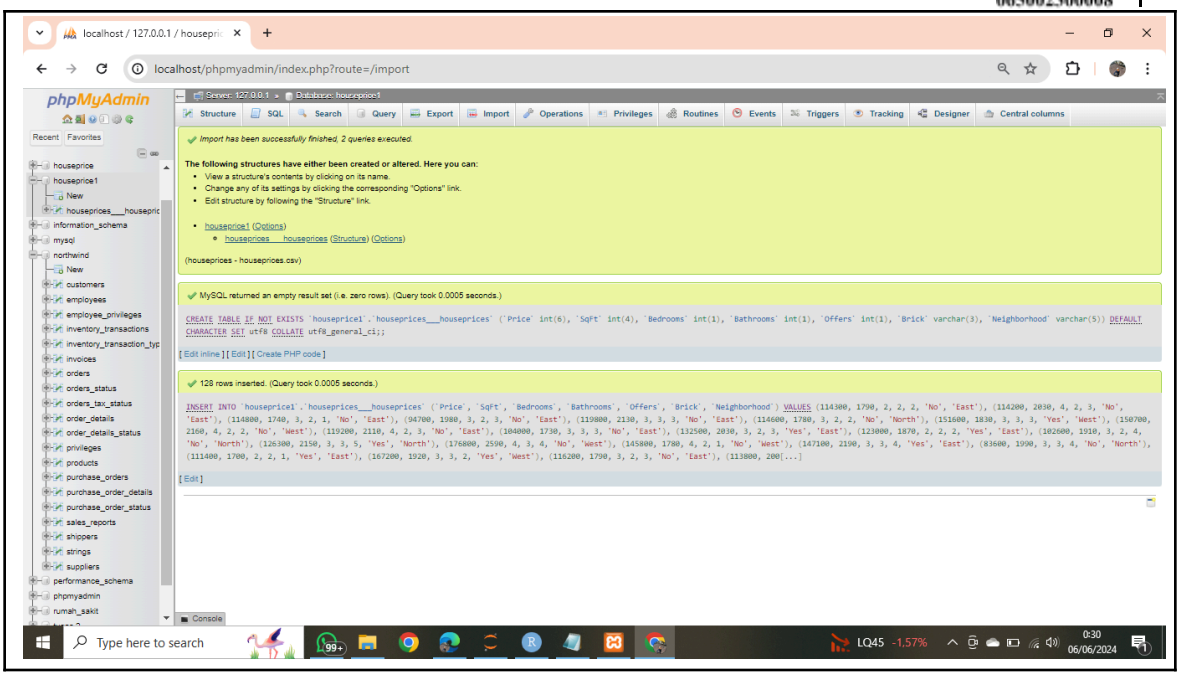
1. Buat sebuah nama database terlebih dahulu dengan nama houseprices di phpmyadmin, Lalu klik menu import

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

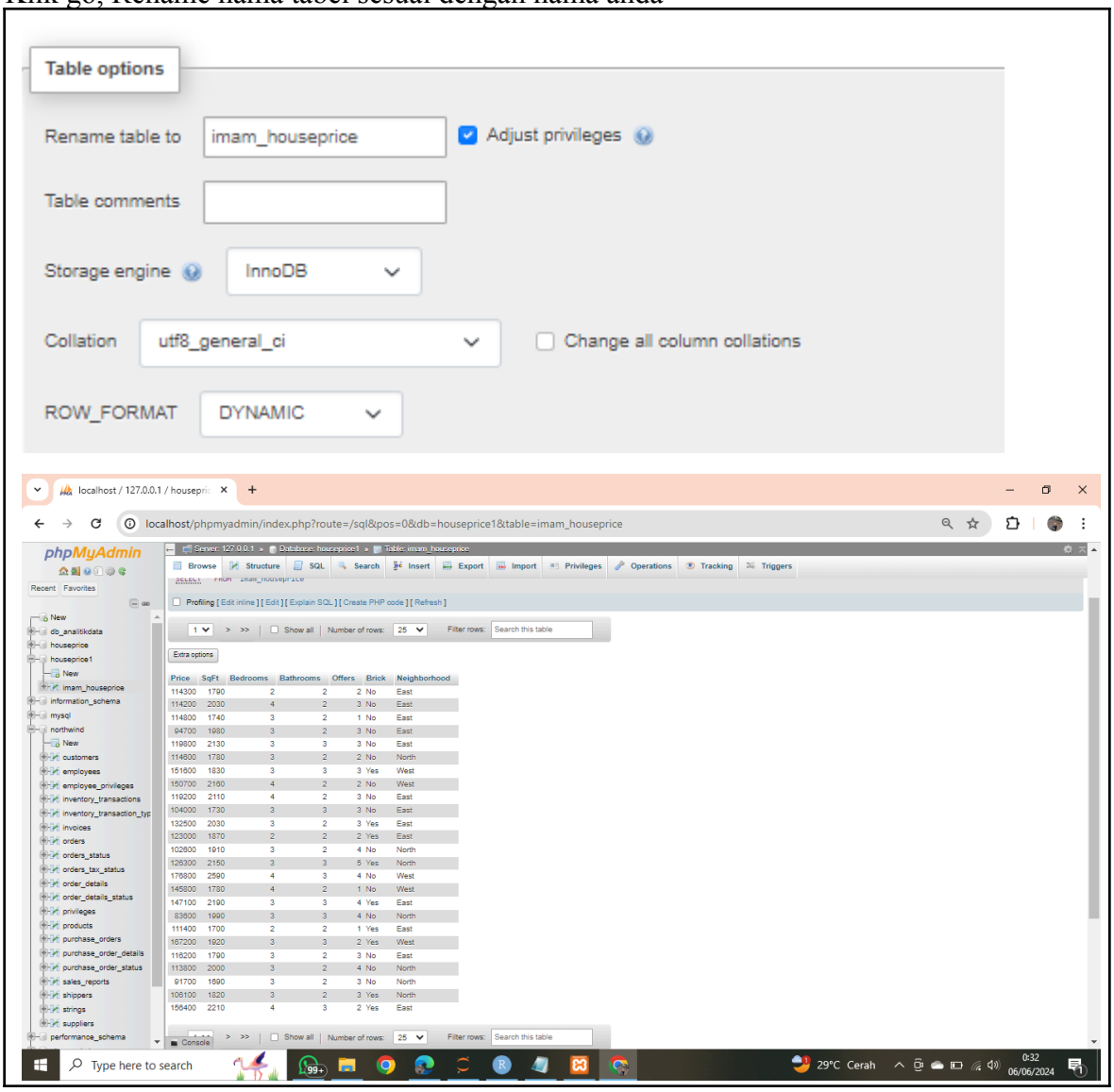


2. Pilih file yang ingin di import ke database (untuk file nama_excel.csv), Ceklis the first line of the file contains the table column name untuk membuat baris pertama pada file excel tersebut menjadi nama atributnya atau nama kolomnya pada database.




localhost / 127.0.0.1 / houseprı
localhost/phpmyadmin/index.php?route=/import
phpMyAdmin
Recent
Favorites
houseprice
houseprice1
New
houseprices___houseprid
information_schema
mysql
northwind
New
customers
employees
employee_privileges
inventory_transactions
inventory_transaction_typ
invoices
orders
orders_status
orders_tax_status
order_details
order_details_status
privileges
products
purchase_orders
purchase_order_details
purchase_order_status
sales_reports
shippers
strings
suppliers
performance_schema
phpmyadmin
rumah_sakit
Server: 127.0.0.1 » Database: houseprice1
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.
houseprice1 (Options)
houseprices___houseprices (Structure) (Options)
(houseprices - houseprices.csv)
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)
CREATE TABLE IF NOT EXISTS `houseprice1`.`houseprices___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 inline] [Edit] [Create PHP code]
128 rows inserted. (Query took 0.0005 seconds.)
INSERT INTO `houseprice1`.`houseprices___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, 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'), (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, 1790, 3, 2, 3, 'No', 'East'), (113800, 200[...]
[Edit]
Console
Type here to search
LQ45 -1.57%
0:30
06/06/2024

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



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

pip install mysql-connector-python

```
In [13]: pip install mysql-connector-python

         Requirement already satisfied: mysql-connector-python in c:\users\t470s\documents\anaconda\lib\site-packages (8.3.0)Note: you m
         ay need to restart the kernel to use updated packages.
```

```
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()
```

5. Lalu jalankan perintah dibawah ini

Output:

In [15]:
```
import mysql.connector

# membuat koneksi ke mysql
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="houseprice1"
)

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

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

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

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

except mysql.connector.Error as err:
    print(f"Error: {err}")

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

```
Hasil querynya by mamcurly:
jangan lupa follow ig:mamcurly:
(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 [28]: import pandas as pd
         # mengonversi hasil kueri ke dataframe pandas
         df = pd.DataFrame(result,columns=[desc[0] for desc in cursor.description])
         # filter data berdasarkan kolom 'brick' yang bernilai 'no'
         df_filtered = df[df['Brick'] == 'no']
         #menampilkan hasil filter
         print("\nHasil filter:")
         print(df_filtered)

         Hasil kueri:
               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]

         Hasil filter:
         Empty DataFrame
         Columns: [Price, SqFt, Bedrooms, Bathrooms, Offers, Brick, Neighborhood]
         Index: []
```

7. Jalankan perintah dibawah ini:

*Perintah ini akan menampilkan 105 baris data hasil filter.

Output:

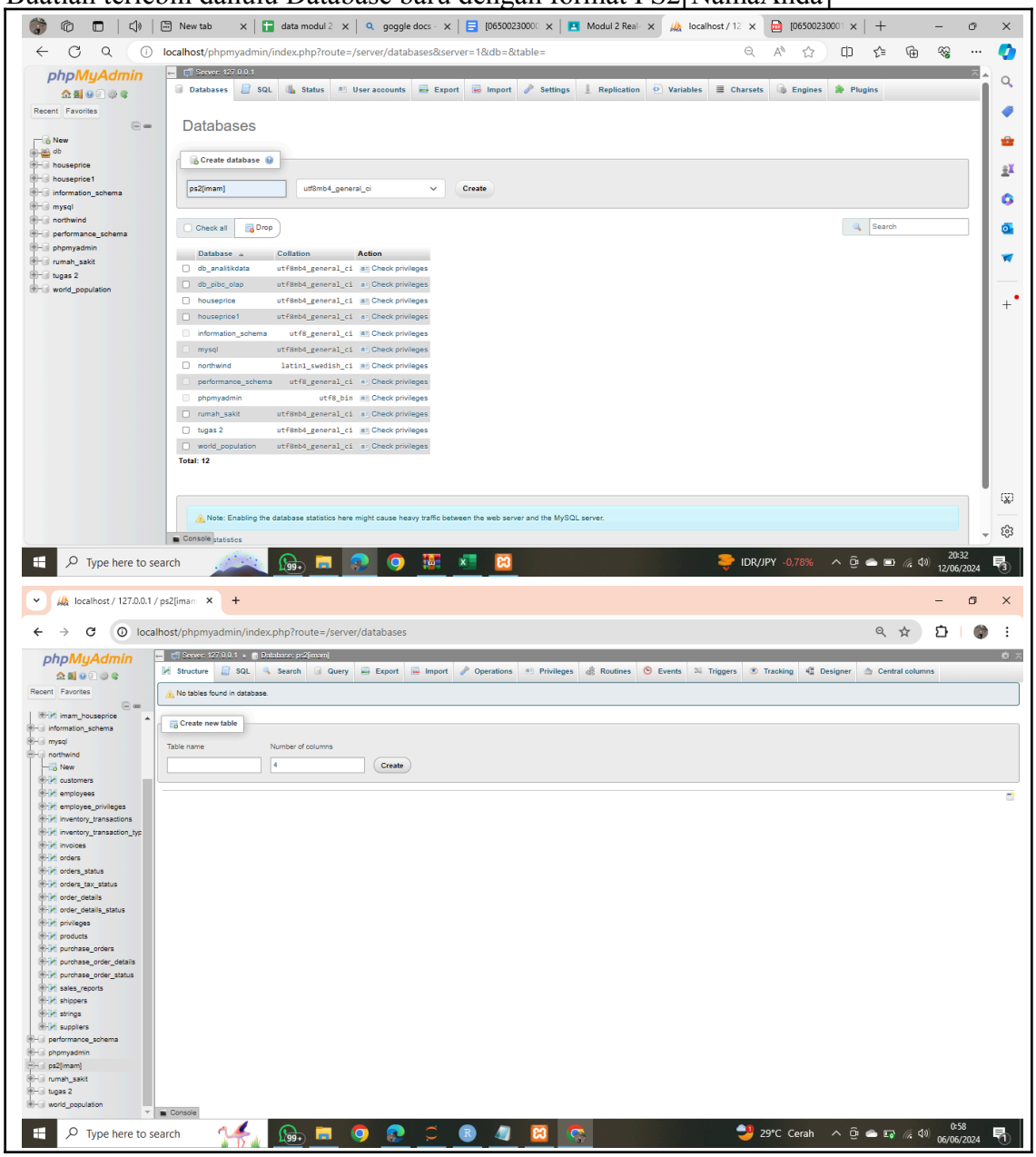
In [30]:
```
import pandas as pd

# mengonversi hasil kueri ke DataFrame Pandas
df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])

# Filter data berdasarkan kondisi yang kompleks
df_filtered = df[(df['Brick'] == 'No') | (df['Neighborhood'] == 'East')]

# menampilkan hasil filter
print(df_filtered)
```

```
      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]
```

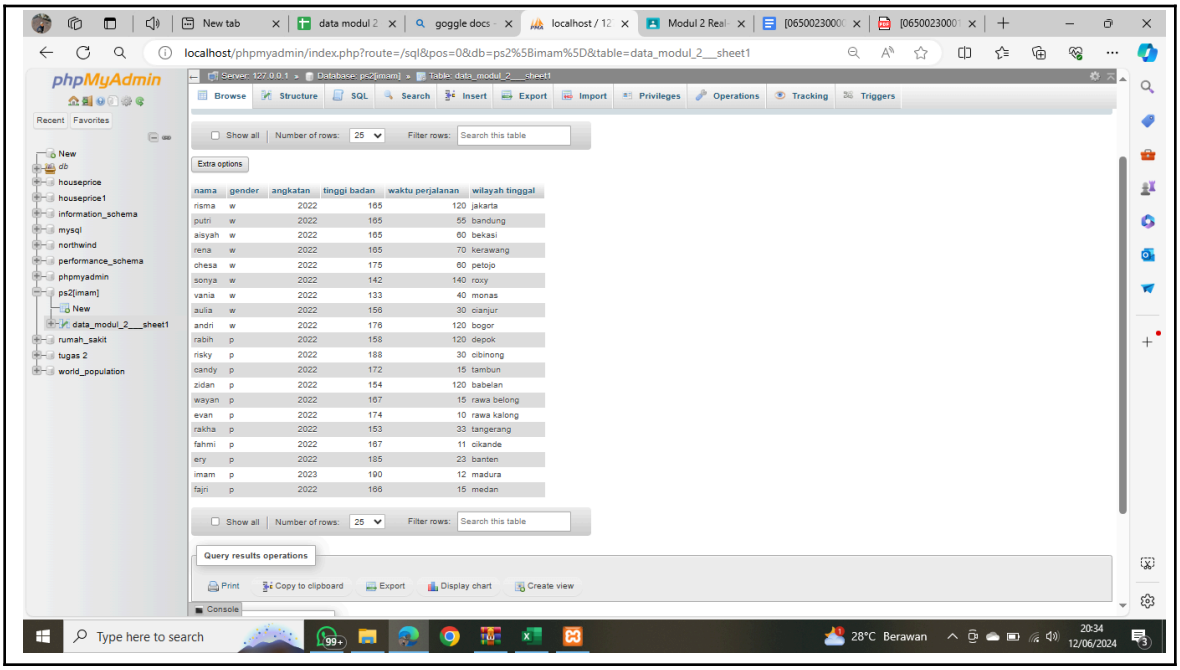
e. Latihan Keenam – Tugas

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

In [10]:
```
import mysql.connector

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

cursor = connection.cursor()

try:
    # Menjalankan query SELECT
    my_query = "SELECT * FROM data_modul_2___sheet1"
    cursor.execute(my_query)
    result = cursor.fetchall()

    print('\nHasil query:')
    for row in result:
        print(row)
except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    # Menutup cursor dan koneksi
    cursor.close()
    connection.close()
```

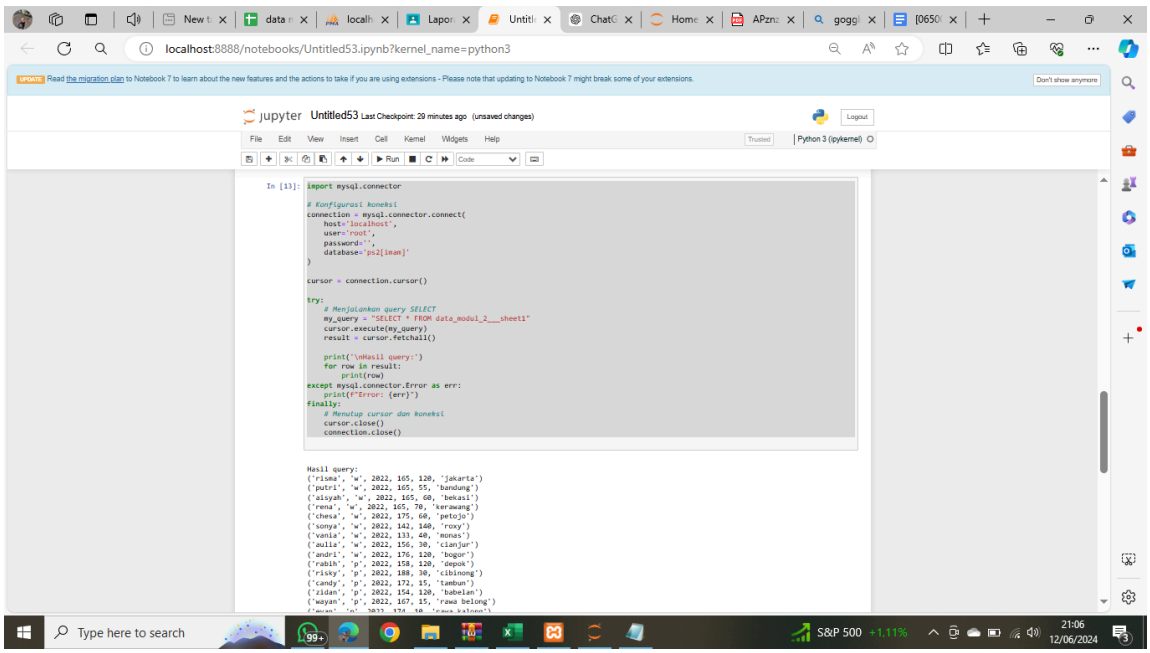
```
Hasil query:
('risma', 'w', 2022, 165, 120, 'jakarta')
('putri', 'w', 2022, 165, 55, 'bandung')
('aisyah', 'w', 2022, 165, 60, 'bekasi')
('rena', 'w', 2022, 165, 70, 'kerawang')
('chesa', 'w', 2022, 175, 60, 'petojo')
('sonya', 'w', 2022, 142, 140, 'roxy')
('vania', 'w', 2022, 133, 40, 'monas')
('aulia', 'w', 2022, 156, 30, 'cianjur')
('andri', 'w', 2022, 176, 120, 'bogor')
('rabih', 'p', 2022, 158, 120, 'depok')
```

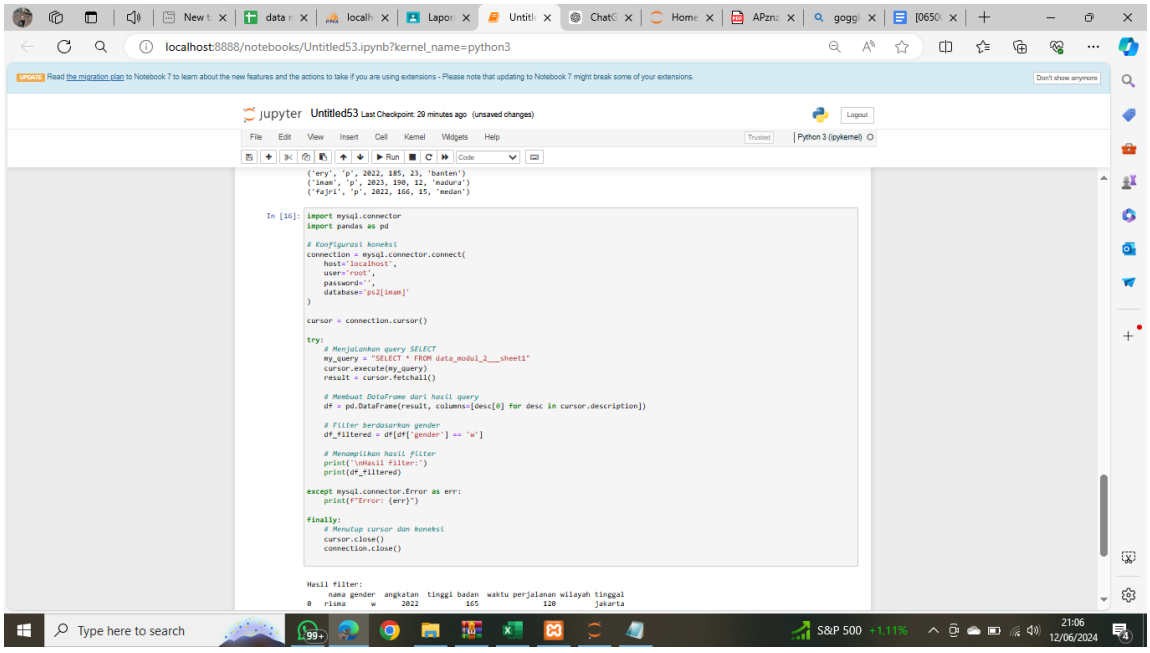
4. Lakukan filter data terhadap Kolom Gender, untuk melihat berapa baris data Pria/Wanita (Pilih salah 1).

```python
# Konfigurasi koneksi
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='',
    database='ps2[imam]'
)

cursor = connection.cursor()

try:
    # Menjalankan query SELECT
    my_query = "SELECT * FROM data_modul_2___sheet1"
    cursor.execute(my_query)
    result = cursor.fetchall()

    # Membuat DataFrame dari hasil query
    df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.descriptic

    # Filter berdasarkan gender
    df_filtered = df[df['gender'] == 'w']

    # Menampilkan hasil filter
    print('\nHasil filter:')
    print(df_filtered)

except mysql.connector.Error as err:
    print(f"Error: {err}")

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

```
Hasil filter:
     nama gender  angkatan  tinggi badan  waktu perjalanan wilayah tinggal
0   risma      w      2022           165               120         jakarta
1   putri      w      2022           165                55         bandung
2  aisyah      w      2022           165                60          bekasi
3    rena      w      2022           165                70        kerawang
4   chesa      w      2022           175                60          petojo
5   sonya      w      2022           142               140            roxy
6   vania      w      2022           133                40           monas
7   aulia      w      2022           156                30         cianjur
8   andri      w      2022           176               120           bogor
```

5. Lampirkan Screenshot
   - Kode koneksi Jupyter ke Database
   - Kode serta hasil filter di Jupyter


localhost:8888/notebooks/Untitled53.ipynb?kernel_name=python3
jupyter Untitled53 Last Checkpoint: 29 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3 (ipykernel)
In [13]:
import mysql.connector
# Konfigurasi koneksi
connection = mysql.connector.connect(
host='localhost',
user='root',
password='',
database='ps2[iman]'
)
cursor = connection.cursor()
try:
# Menjalankan query SELECT
my_query = "SELECT * FROM data_modul_2___sheet1"
cursor.execute(my_query)
result = cursor.fetchall()
print('\nHasil query:')
for row in result:
print(row)
except mysql.connector.Error as err:
print(f"Error: {err}")
finally:
# Menutup cursor dan koneksi
cursor.close()
connection.close()
Hasil query:
('risma', 'w', 2022, 165, 120, 'jakarta')
('putri', 'w', 2022, 165, 55, 'bandung')
('aisyah', 'w', 2022, 165, 60, 'bekasi')
('rena', 'w', 2022, 165, 70, 'kerawang')
('chesa', 'w', 2022, 175, 60, 'petojo')
('sonya', 'w', 2022, 142, 140, 'roxy')
('vania', 'w', 2022, 133, 40, 'monas')
('aulia', 'w', 2022, 156, 30, 'cianjur')
('andri', 'w', 2022, 176, 120, 'bogor')
('rabih', 'p', 2022, 158, 120, 'depok')
('risky', 'p', 2022, 188, 30, 'cibinong')
('candy', 'p', 2022, 172, 15, 'tambun')
('zidan', 'p', 2022, 154, 120, 'babelan')
('wayan', 'p', 2022, 167, 15, 'rawa belong')
Type here to search
S&P 500 +1.11%
21:06
12/06/2024


localhost:8888/notebooks/Untitled53.ipynb?kernel_name=python3
jupyter Untitled53 Last Checkpoint: 29 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3 (ipykernel)
('ery', 'p', 2022, 185, 23, 'banten')
('iman', 'p', 2023, 190, 12, 'madura')
('fajri', 'p', 2022, 166, 15, 'medan')
In [16]:
import mysql.connector
import pandas as pd
# Konfigurasi koneksi
connection = mysql.connector.connect(
host='localhost',
user='root',
password='',
database='ps2[iman]'
)
cursor = connection.cursor()
try:
# Menjalankan query SELECT
my_query = "SELECT * FROM data_modul_2___sheet1"
cursor.execute(my_query)
result = cursor.fetchall()
# Membuat DataFrame dari hasil query
df = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
# Filter berdasarkan gender
df_filtered = df[df['gender'] == 'w']
# Menampilkan hasil filter
print('\nHasil filter:')
print(df_filtered)
except mysql.connector.Error as err:
print(f"Error: {err}")
finally:
# Menutup cursor dan koneksi
cursor.close()
connection.close()
Hasil filter:
nama gender angkatan tinggi badan waktu perjalanan wilayah tinggal
0 risma w 2022 165 120 jakarta
Type here to search
S&P 500 +1.11%
21:06
12/06/2024

4. File Praktikum

Github Repository:

5. Kesimpulan

a. Dalam pengerjaan praktikum Statistika, Kita harus teliti dengan code yang kita input dan nama dataframe atau basis data yang kita buat, karena jika kita salah menginput data akan terjadi error

b. Kita juga dapat mengetahui cara menghubungkan basis data sql ke dalam python dan cara menyaring data yang kita impor agar dapat terlihat data yang dibutuhkan

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	10 menit	menarik
2.	Latihan Kedua	10 menit	menarik
3.	Latihan Ketiga	10 menit	menarik

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

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