

MODUL 4

PYTHON (FOR DATA SCIENCE)

VISUALISASI DATA - PANDAS DATAFRAME

A. Data Science

Adalah ilmu yang mempelajari tentang data terutama data kuantitatif atau data numerik. Secara umum data science juga bisa disebut sebagai proses mengekstrak data agar dapat difilter serta didapatkan data yang benar untuk menghasilkan produk data yang sebenar-benarnya.

B. Pandas DataFrame

1. Pandas

Pandas adalah library open source pada Python yang sering digunakan untuk memproses data yang meliputi pembersihan data, manipulasi data, hingga melakukan analisis data. Dimana pandas mendukung data multi-dimensi.

- a) Series merupakan struktur data dasar dalam Pandas yang dapat berisi tipe data seperti integer, string, dan lain-lain, yang mendukung tipe data sama atau campuran. Contoh Membuat Series

The screenshot shows a VS Code interface with the following details:

- File Explorer (Left):** Shows a folder structure with files like `Modulus.py`, `ModulusSpy`, `py2_modul7.py`, `data2.csv`, `ModulusSpy`, and `py2_modul8Spy`.
- Terminal (Bottom):** Displays the output of running `Activate.ps1` and then executing the Python script `py2_modul6.py`. The output shows the execution of a function `hensusit Series` which prints the numbers 1, 2, 3, and 4.

- b) Ubah Nama Index. Contoh

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like `Modul4.py`, `Modul5.py`, `py2_modul7.py`, `data2.csv`, `Modul6.py`, `py2_modul8.py`, and `data.csv`.
- Terminal:** Displays the command `PS C:\Python\Python36\& c:/Python/Env/Scripts/activate.ps1` followed by the output of a Python script:

```
(env) PS C:\Python\Python36\& c:/Python/Env/Scripts/python.exe c:/Python/py2_modul6.py
Resultat: Series
a    1
b    2
c    3
d    4
dtype: int64
```
- Status Bar:** Shows "Line 8, Col 17" and "Python 3.11.2 (main, Mar 28 2023, 16:45:46) [GCC 11.2.0]".

c) DataFrame

Merupakan array dua dimensi dengan baris dan kolom. DataFrame merupakan tabel/data tabular. Setiap kolom pada DataFrame merupakan objek dari Series, dan baris terdiri dari elemen yang ada pada Series. Contoh DataFrame Menggunakan List

The screenshot shows the Visual Studio Code interface with the Python extension installed. The Explorer sidebar on the left lists files in a directory structure under 'PYTHON'. The 'py2_modul16.py' file is open in the editor, containing a script that imports pandas and prints its version. The terminal at the bottom shows the command being run and the output 'Python 3.8.5'. The status bar at the bottom right indicates the file is 1112 lines long.

```
File Edit Selection View Go Run ... > < > Python8

... + py2_modul16.py py2_modul17.py data2.csv + py2_modul8.py x data.csv

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Python8> & c:/Python8/ems/Scripts/Activate.ps1
● (env) PS C:\Python8> & c:/Python8/ems/Scripts/python.exe c:/Python8/py2_modul16.py
Python 3.8.5
```

d) Contoh DataFrame Menggunakan Dictionary

e) Contoh DataFrame Menggunakan List dengan Tipe Data Campuran

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (left):** Shows a folder structure with files like `Modul5.py`, `Modul5Spy`, `py2_modul7.py`, `data2.csv`, `Modul5.py`, `py2_modul5Spy`, and `data.csv`.
- Code Editor (center):** Displays a Python script (`py2_modul7.py`) containing code to read CSV data and calculate average marks.
- Terminal (bottom):** Shows the output of running the script, displaying the calculated average marks for each student.

```
PS C:\Python&env\Scripts> python py2_modul7.py
  0 Aida 80 85 90 98.5
  1 Caca 80 85 90 98.6
  2 Lala 70 75 80 78.5
  
```

- f) Contoh DataFrame Menggunakan List & Dictionary dengan Tipe Data Campuran

The screenshot shows a Visual Studio Code interface with the following details:

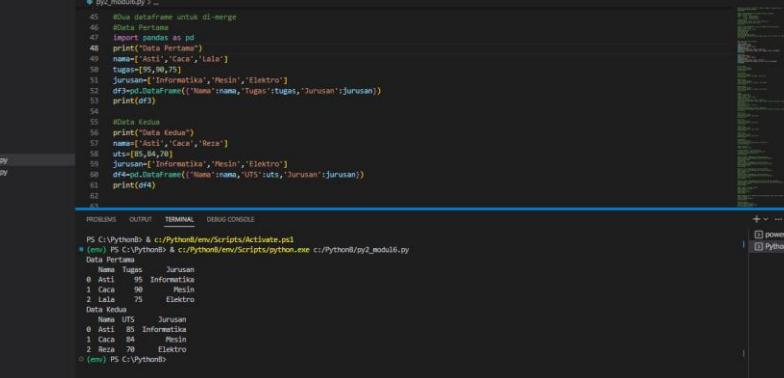
- File Explorer:** Shows a folder structure under "PYTHON".
 - pyModule1.py
 - pyModule2.py
 - pyModule3.py
 - pyModule4.py
 - pyModule5.py
 - pyModule6.py
 - pyModule7.py
 - pyModule8.py
 - pyModule9.py
 - pyModule10.py
 - pyModule11.py
 - pyModule12.py
 - pyModule13.py
 - pyModule14.py
 - pyModule15.py
 - pyModule16.py
- Code Editor:** Displays a Python script named "pyModule7.py". The code uses pandas to read CSV files, merge them, and calculate averages.

```
>_pyache_ ... pyModule5.py pyModule6.py pyModule7.py data2.csv pyModule8.py pyModule9.py
  pyModule7.py
    34 #Buat frame menggunakan list & dic dengan tipe data campuran
    35 import pandas as pd
    36 df1 = pd.read_csv('C:/.../data1.csv')
    37 tugas=[90,80,70]
    38 uts=[85,85,75]
    39 uas=[95,90,80]
    40 df2=pd.DataFrame({'Name':names,'Tugas':tugas,'UTS':uts,'UAS':uas,'Rate':rate})
    41 df2=df2[['Name','Tugas','UTS','UAS','Rate']]
    42 print(df2)
    43
    44
```
- Terminal:** Shows the command line output for running "pyModule16.py". The output displays student names and their average scores.

2. Merge, Join, & Concatenate DataFrame

Fungsi marge, join, dan concatenate merupakan operasi digunakan untuk penggabungan.

- a) Merge adalah operasi penggabungan antara DataFrame
Contoh Persiapkan 2 Data Frame Untuk di Marge



The screenshot shows the PyCharm IDE interface with the following details:

- File Structure (EXPLORER):** Shows the project structure with files like `Modul1.py`, `Modul2.py`, `Modul3.py`, `Modul4.py`, `py2_modul1.py`, `py2_modul2.py`, `py2_modul3.py`, `py2_modul4.py`, `data.csv`, and `data2.csv`.
- Code Editor:** Displays Python code for reading CSV files and creating DataFrames.

```
... Modul1.py Modul2.py py2_modul1.py ...  
# py2_modul2.py ...  
# py2_modul3.py ...  
# py2_modul4.py ...  
  
# data.csv  
# data2.csv  
# Modul1.py  
# Modul2.py  
# Modul3.py  
# Modul4.py  
# data.csv  
# Modul1.py  
# Modul2.py  
# Modul3.py  
# Modul4.py  
# py2_modul1.py  
# py2_modul2.py  
# py2_modul3.py  
# py2_modul4.py  
  
#!/usr/bin/python  
# coding: utf-8  
  
import pandas as pd  
from pandas import DataFrame  
  
# Data Kedua  
print("Data Kedua")  
name = ["Asti", "Caca", "Alia"]  
tugas = [95, 90, 75]  
uts = [85, 84, 78]  
jurusan = ["Informatika", "Mein", "Elektro"]  
  
dta2 = DataFrame({'Name': name, 'Tugas': tugas, 'Uts': uts, 'Jurusan': jurusan})  
print(dta2)  
  
# Data Pertama  
print("Data Pertama")  
name = ["Asti", "Caca", "Alia"]  
tugas = [95, 90, 75]  
uts = [85, 84, 78]  
jurusan = ["Informatika", "Mein", "Elektro"]  
  
dta1 = DataFrame({'Name': name, 'Tugas': tugas, 'Uts': uts, 'Jurusan': jurusan})  
print(dta1)
```
- Terminal:** Shows the command line output of running the script.

```
PS C:\Python2 & c:/Python/env/Scripts/Activate.ps1  
$env:Path += "c:/Python/env/Scripts/python.exe" c:/Python/py2_modul2.py  
Data Pertama  
Name Tugas Jurusan  
0 Asti 95 Informatika  
1 Caca 84 Mein  
2 Alia 75 Elektro  
Data Kedua  
Name Uts Jurusan  
0 asti 85 Informatika  
1 Caca 84 Mein  
2 Alia 75 Elektro  
$ (cmd) PS C:\Python2
```
- Bottom Status Bar:** Shows the current status: Line 46, Col 15, Status 2, UFT-8, CR/LF, 14 Python 3.11.2 (env), and IP.

- b) Contoh Inner Merge.

The screenshot shows the Visual Studio Code interface with the Python extension installed. The Explorer sidebar on the left lists files like Modul4.py, Modul5.py, py2_modul7.py, data2.csv, Modul6.py, py2_modul6.py, and data.csv. The terminal tab at the bottom is active, displaying the command PS C:\Python3 & c:/Python3/Scripts/Activate.ps1 followed by the output of a script named py2_modul6.py. The output shows a table with columns Name, NRP, Jurusan, and ITS. The table has two rows: one for Aktif (NRP 0) and one for Keluar (NRP 1). The data is as follows:

	NRP	Jurusan	ITS
0	0	Aktif	Informatika 85
1	1	Keluar	Mesin 84

c) Contoh Left Merge

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, Modul3.py, Modul4.py, Modul5.py, Modul6.py, py2_modul5.py, and py2_modul7.py. The terminal window shows the command PS C:\Python38 & c:/Python38/Scripts/activate.ps1 being run. The code in the editor is:

```
PS C:\Python38 & c:/Python38/Scripts/activate.ps1
PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Left Merge
    df1 = pd.read_csv('data1.csv')
    df2 = pd.read_csv('data2.csv')
    print("Left Merge")
    left=df3.merge(df4, on='Name', how='left')
    print(left)
    df3
```

The output in the terminal shows the merged data frame:

Name	F1	F2	F3	F4	F5
Astl	95.0	Informatica	85.0	Informatica	
Caca	90.0	Mesin	84.0	Mesin	
Lala	75.0	Elektro	Nan	Nan	Elektro
Risa	Nan	Nan	70.0	Nan	Elektro

d) Contoh Right Merge

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, Modul3.py, Modul4.py, Modul5.py, Modul6.py, py2_modul5.py, and py2_modul7.py. The terminal window shows the command PS C:\Python38 & c:/Python38/Scripts/activate.ps1 being run. The code in the editor is:

```
PS C:\Python38 & c:/Python38/Scripts/activate.ps1
PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Right Merge
    right=df3.merge(df4, on='Name', how='right')
    print(right)
    df3
```

The output in the terminal shows the merged data frame:

Name	F1	F2	F3	F4	F5
Astl	95.0	Informatica	85.0	Informatica	
Caca	90.0	Mesin	84.0	Mesin	
Lala	75.0	Elektro	Nan	Nan	Elektro
Risa	Nan	Nan	70.0	Nan	Elektro

e) Contoh Outer Merge

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, Modul3.py, Modul4.py, Modul5.py, Modul6.py, py2_modul5.py, and py2_modul7.py. The terminal window shows the command PS C:\Python38 & c:/Python38/Scripts/activate.ps1 being run. The code in the editor is:

```
PS C:\Python38 & c:/Python38/Scripts/activate.ps1
PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Outer Merge
    outer=df3.merge(df4, on='Name', how='outer')
    print(outer)
    df3
```

The output in the terminal shows the merged data frame:

Name	F1	F2	F3	F4	F5
Astl	95.0	Informatica	85.0	Informatica	
Caca	90.0	Mesin	84.0	Mesin	
Lala	75.0	Elektro	Nan	Nan	Elektro
Risa	Nan	Nan	70.0	Nan	Elektro

3. Join

Adalah operasi penggabungan dengan menggunakan index. Siapkan Dua DataFrame untuk di-Join .

- a) Data join sebagai berikut :

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `Modul1.py`, `Modul1say.py`, `py2_modul7.py`, `data.csv`, `Modul6.py`, `py2_modul6.py`, and `data.csv`.
- Code Editor:** Displays Python code for joining data frames. The code includes imports for `pandas` and `dataframe`, and uses `pd.read_csv` to load data from `data.csv`. It then performs two joins: `Pertama Join` and `Kedua Join`, printing the results to the terminal.
- Terminal:** Shows the command prompt (`PS C:\Python36\`) and the output of the executed script. The output includes:
 - Output of `Modul1say`:
 - Data Pertama Join
 - Prints "Data Pertama Join"
 - Imports `pandas` as `pd`
 - Creates a `data` frame with columns `Nama`, `Jurusan`, and `Tugas`.
 - Prints the frame.
 - Output of `py2_modul6.py`:
 - Data Kedua Join
 - Prints "Data Kedua Join"
 - Imports `pandas` as `pd`
 - Creates a `data` frame with columns `Nama`, `Jurusan`, and `Bobot`.
 - Prints the frame.
 - Final joined output:
 - Prints "Data Pertama Join"
 - Prints "Data Kedua Join"
 - Prints the final joined frame with columns `Nama`, `UTS`, `IPTS`, `Jurusan`, and `Bobot`.

- ### b) Contoh Inner Join

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows files like `Modul8.py`, `Modul8.py`, `py2_modul7.py`, `data.csv`, `Modul8.py`, `py2_modul6.py`, and `data.csv`. It also lists `__pycache__`, `.vscode`, `Marketplace`, `data.csv`, and `py2_modul7.py`.
- Terminal:** Displays the command `Ps C:\Python36\ & c:\Python36\env\Scripts\Activate.ps1` followed by the output of a script named `py2_modul6.py`. The output shows student data with columns: Nama, Tugas, Jurusan, Nama B, UT斯, Jurusan B. It lists four students: L2 Caca (98, Mesin), Asti (85, Informatika), L3 (79, Elektro), and Caca (84, Mesin). A prompt at the end indicates the environment is activated.
- Status Bar:** Shows "Ln 105, Col 1" and "Source 2 / 109-8".

- c) Contoh Left Join

The screenshot shows a dark-themed instance of VS Code. The top menu bar includes File, Edit, Selection, View, Go, Run, and a separator. Below the menu is a search bar with the text "PythonB". The Explorer sidebar on the left lists files: ModulA.py, ModulS.py, py2_modul7.py, data.csv, ModulS.py, py2_modulS.py, and data.csv. The Problems, Output, Terminal, and Debug Console tabs are visible at the bottom of the editor area. The Terminal tab is active, displaying the command PS C:\PythonB & c:\PythonB\env\Scripts\Activate.ps1 followed by the output of a script named py2_modul6.py. This output is a table with columns: Nama, Tugas, Jurusan, Nama B, UT斯, Jurusan B. It contains five rows of data. The status bar at the bottom shows "Ln 110 Col 1" and "Python 3.11.2 (env) ::env".

Nama	Tugas	Jurusan	Nama B	UT斯	Jurusan B
L3 Asti	95	Informatika	Nah	Nah	Nah
L3 Asti	88	Informatika	Ayah	89	Informasi
L3 Risa	75	Elektro	Caca	8.8	Micin
L3 Risa	75	Elektro	Caca	8.8	Micin

d) Contoh Right Join

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py through Modul7.py, py2_modul4.py, py2_modul5.py, py2_modul7.py, data2.csv, and Modul6.py. The terminal window displays the following code and its execution:

```
112 #Right Join
113 print("Right Join")
114 right=df3.join(df4, how='right')
115 print(right)
116
117
118 print("Outer Join")
119 outer=df3.join(df4, how='outer')
120 print(outer)
```

```
(env) PS C:\PythonB> & c:/PythonB/env/Scripts/Activate.ps1
PS C:\PythonB> & c:/PythonB/env/Scripts/python.exe c:/PythonB/py2_modul6.py
Name Tugas Jurusan Nama_B UTS Jurusan_B
L1 Caca 95.0 Informatika Nama_A Asti 85.0 Informatika
L2 Caca 98.0 Mesin Asti 84.0 Mesin
L3 Reza 75.0 Elektro Caca 84.0 Mesin
L4 Nabi Nabi Nabi Bella 70.0 Metalurgi
```

e) Contoh Outer Join

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py through Modul7.py, py2_modul4.py, py2_modul5.py, py2_modul7.py, data2.csv, and Modul6.py. The terminal window displays the following code and its execution:

```
117 #Outer Join
118 print("Outer Join")
119 outer=df3.join(df4, how='outer')
120 print(outer)
```

```
(env) PS C:\PythonB> & c:/PythonB/env/Scripts/Activate.ps1
PS C:\PythonB> & c:/PythonB/env/Scripts/python.exe c:/PythonB/py2_modul6.py
Outer Join
Name Tugas Jurusan Nama_B UTS Jurusan_B
L1 Asti 95.0 Informatika Nama_A Nabi 85.0 Informatika
L2 Caca 98.0 Mesin Asti 84.0 Mesin
L3 Reza 75.0 Elektro Caca 84.0 Mesin
L4 Nabi Nabi Nabi Bella 70.0 Metalurgi
```

f) Concatenate adalah operasi penggabungan objek DataFrame secara vertical. Contoh

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py through Modul7.py, py2_modul4.py, py2_modul5.py, py2_modul7.py, data2.csv, and Modul6.py. The terminal window displays the following code and its execution:

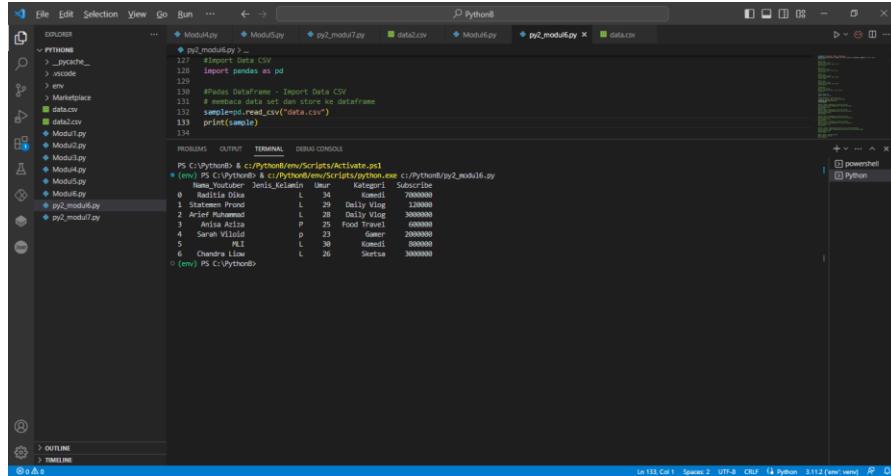
```
122 #CONCATENATE
123 print("concatenate")
124 dfc=pd.concat([df3,df4], sort=False)
125 print(dfc)
```

```
(env) PS C:\PythonB> & c:/PythonB/env/Scripts/Activate.ps1
PS C:\PythonB> & c:/PythonB/env/Scripts/python.exe c:/PythonB/py2_modul6.py
Jurusan_B Jurusan_B Nama_Nama_B Tugas UTS
L1 Informatika Nabi Asti Nama_A 95.0 Nabi
L2 Mesin Reza Caca 98.0 Nabi
L3 Elektro Nabi Reza Nama_B 75.0 Nabi
L2 Nabi Informatika Nabi Asti Nama_A 85.0
L3 Reza Mesin Caca 84.0 Nabi
L4 Nabi Metalurgi Nabi Bella 70.0
```

4. Pandas DataFrame - Import Data CSV

Pada tahap ini akan dibahas mengenai cara import data CSV ke dalam Panda DataFrame dan mengolah datanya.

a) Import Data CSV

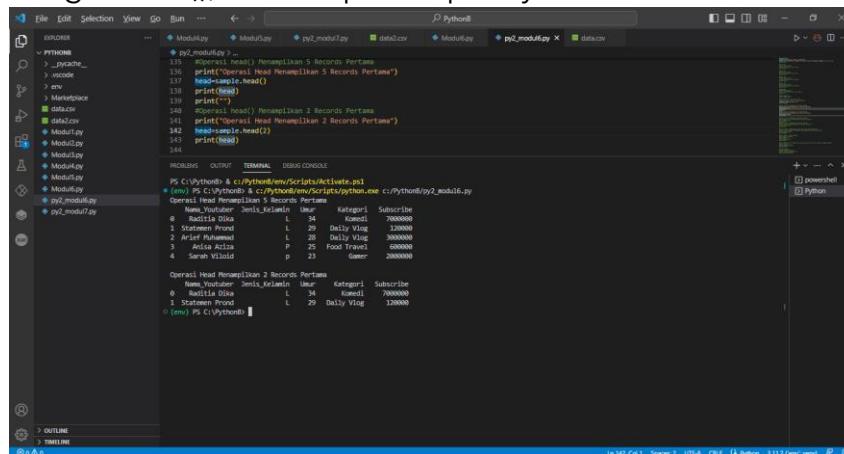


```
File Edit Selection View Go Run ... <- > 🔍 PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
C:\Python38\&c:\Python\env\Scripts\activate.ps1
# Import Data CSV
import pandas as pd
# Membaca data set dan store ke dataframe
sample1.read_csv("data.csv")
print(sample1)
0 Radita Dila L 34 Komedi 700000
1 Siti Nurhaliza L 29 Daily Vlog 1200000
2 Arif Muhammad L 28 Daily Vlog 3000000
3 Anisa Aziza P 25 Food Travel 600000
4 Sarah Vilid P 22 Game 2000000
5 M.I L 30 Komedi 800000
6 Chandra Liow L 26 Sketsa 3000000
(�) PS C:\PythonB>
```

b) Macam - Macam Operasi

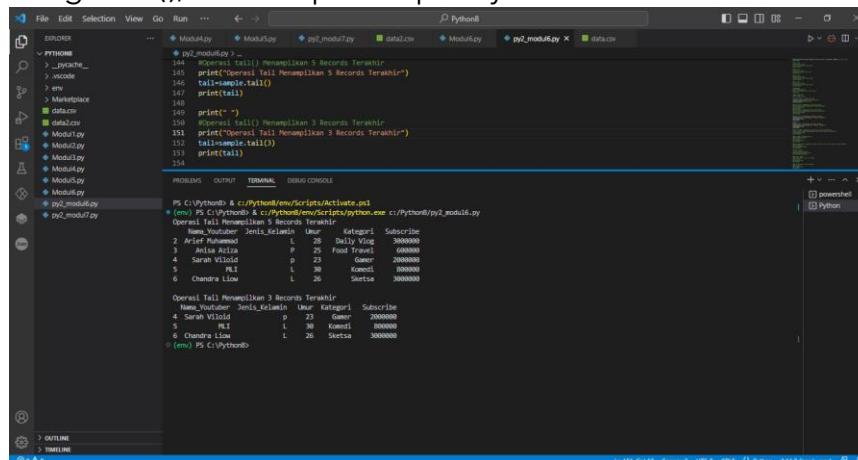
Fungsi-fungsi yang dapat dilakukan oleh Pandas DataFrame adalah sebagai berikut.

1) Fungsi head(), Contoh penerapannya



```
File Edit Selection View Go Run ... <- > 🔍 PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
C:\Python38\&c:\Python\env\Scripts\activate.ps1
# OpenHead Head Mengambil 5 Records Pertama
Operasi Head Mengambil 5 Records Pertama
0 Radita Dila Jenis_Kelamin L Umur Kategori Subscribe
1 Siti Nurhaliza Jenis_Kelamin L Umur Kategori Subscribe
2 Arif Muhammad Jenis_Kelamin L Umur Kategori Subscribe
3 Anisa Aziza Jenis_Kelamin P Umur Kategori Subscribe
4 Sarah Vilid Jenis_Kelamin P Umur Kategori Subscribe
Operasi Head Mengambil 2 Records Pertama
Operasi Head Mengambil 2 Records Pertama
0 Radita Dila Jenis_Kelamin L Umur Kategori Subscribe
1 Siti Nurhaliza Jenis_Kelamin L Umur Kategori Subscribe
Operasi Head Mengambil 2 Records Pertama
Operasi Head Mengambil 2 Records Pertama
0 Radita Dila Jenis_Kelamin L Umur Kategori Subscribe
1 Siti Nurhaliza Jenis_Kelamin L Umur Kategori Subscribe
2 Arif Muhammad Jenis_Kelamin L Umur Kategori Subscribe
3 Anisa Aziza Jenis_Kelamin P Umur Kategori Subscribe
4 Sarah Vilid Jenis_Kelamin P Umur Kategori Subscribe
(�) PS C:\PythonB>
```

2) Fungsi tail(), Contoh penerapannya



```
File Edit Selection View Go Run ... <- > 🔍 PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
C:\Python38\&c:\Python\env\Scripts\activate.ps1
# OpenTail Tail Mengambil 5 Records Terakhir
Operasi Tail Mengambil 5 Records Terakhir
0 Radita Dila Jenis_Kelamin L Umur Kategori Subscribe
1 Siti Nurhaliza Jenis_Kelamin L Umur Kategori Subscribe
2 Arif Muhammad Jenis_Kelamin L Umur Kategori Subscribe
3 Anisa Aziza Jenis_Kelamin P Umur Kategori Subscribe
4 Sarah Vilid Jenis_Kelamin P Umur Kategori Subscribe
Operasi Tail Mengambil 3 Records Terakhir
Operasi Tail Mengambil 3 Records Terakhir
0 Radita Dila Jenis_Kelamin L Umur Kategori Subscribe
1 Siti Nurhaliza Jenis_Kelamin L Umur Kategori Subscribe
2 Arif Muhammad Jenis_Kelamin L Umur Kategori Subscribe
3 Anisa Aziza Jenis_Kelamin P Umur Kategori Subscribe
4 Sarah Vilid Jenis_Kelamin P Umur Kategori Subscribe
5 M.I L 30 Komedi 800000
6 Chandra Liow L 26 Sketsa 3000000
(�) PS C:\PythonB>
```

3) Fungsi shape, contoh penerapannya

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, and py2_modul6.py. The terminal window displays the following Python script and its execution:

```
PS C:\Python> & c:\Python\env\Scripts\activate.ps1
* (env) PS C:\Python> & c:\Python\env\Scripts\python.exe c:/Python/py2_modul6.py
Operasi Shape() Menampilkan Jumlah Baris dan Kolom DataFrame
152 #Operasi shape() Menampilkan Jumlah Baris dan Kolom DataFrame
153 print("Operasi Shape() Menampilkan Jumlah Baris dan Kolom DataFrame")
154 print(df.shape)
155 print("Shape")
156 print(shape)
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
```

The output shows the execution of the script, printing the shape of the DataFrame.

4) Contoh Mean, Median, Standar Deviasi

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, and py2_modul6.py. The terminal window displays the following Python script and its execution:

```
PS C:\Python> & c:\Python\env\Scripts\activate.ps1
* (env) PS C:\Python> & c:\Python\env\Scripts\python.exe c:/Python/py2_modul6.py
Mean, Median, Standar Deviasi Mean adalah rata-rata
158 #Mean, Median, Standar Deviasi Mean adalah rata-rata
159 print("Mean")
160 print("Median")
161 print("Standar Deviasi")
162 print("")
163 print("Menunjukkan nilai tengah dari data yang telah diurut dari terkecil hingga terbesar")
164 median=sample.median()
165 print("Median")
166 print(median)
167 print("")
168 #Standar Deviasi
169 print("Standar Deviasi")
170 print("Standar Deviasi = rata-rata - Mean")
171 print("Standar Deviasi")
172 print(Standar_Deviasi)
```

The output shows the execution of the script, calculating the mean, median, and standard deviation of the sample data.

5) Contoh Max, Min, Count

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modul1.py, Modul2.py, and py2_modul6.py. The terminal window displays the following Python script and its execution:

```
PS C:\Python> & c:\Python\env\Scripts\activate.ps1
* (env) PS C:\Python> & c:\Python\env\Scripts\python.exe c:/Python/py2_modul6.py
MAX untuk mencari nilai tertinggi :
174 MAX, MIN, COUNT, MAX untuk mencari nilai tertinggi
175 print("MAX untuk mencari nilai tertinggi : ")
176 print(MAX)
177 print("MAX untuk mencari nilai tertinggi : ")
178 print(MAX)
179 print("MIN untuk mencari nilai terendah")
180 print("MIN untuk mencari nilai terendah : ")
181 print(MIN)
182 print(MIN)
183 print("COUNT untuk perhitungan non null record pada setiap kolom : ")
184 print("Count untuk perhitungan non null record pada setiap kolom : ")
185 COUNT=sample.count()
186 print(COUNT)

MAX untuk mencari nilai tertinggi :
Name_Youtubeer  Stemon Prod
Jenis_Kelamin  J
Umur  34
Kategori  Sketsa
Subscribe  7000000
dtype: object

MIN untuk mencari nilai terendah :
Name_Youtubeer  Anisa Alina
Jenis_Kelamin  L
Umur  23
Kategori  Daily Vlog
Subscribe  100000
dtype: object

Count untuk perhitungan non null record pada setiap kolom :
Name_Youtubeer  7
Jenis_Kelamin  7
Umur  7
Kategori  7
Subscribe  7
dtype: int64
```

The output shows the execution of the script, demonstrating the use of max(), min(), and count() functions on a DataFrame.

6) Contoh describe()

The screenshot shows the Visual Studio Code interface with the Python extension installed. The Explorer sidebar on the left lists files like `data.csv`, `data2.csv`, and Python modules. The terminal at the bottom displays the output of running `py2_modul16.py`, which prints statistical data for the `statistic_data` variable. The status bar at the bottom right shows the current file is `py2_modul16.py`.

```
... Modul4.py Modul5.py py2_modul7.py data2.csv Modul6.py py2_modul8.py data.csv

188 #DISCRIBE ringkas statistik data
189 print("DISCRIBE")
190 DESCRIBE=sample.describe()
191 print(DESCRIBE)
192

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\PythonDB & c:/Python/env/Scripts/activate.ps1
* (env) PS C:\PythonDB & c:/Python/env/Scripts/python.exe c:/Python/py2_modul16.py
DISCRIBE
      Umur      Subcribe
count    7.000000e+00
mean   27.857143 2.306000e+00
std    11.500000 1.200000e+00
min    23.000000 1.200000e+00
25K    25.500000 7.000000e+00
50K    30.000000 7.000000e+00
75K    39.500000 3.000000e+00
max   34.000000 7.000000e+00

* (env) PS C:\PythonDB>
```

7) Contoh Rename dan Drop Kolom

The screenshot shows a Jupyter Notebook interface on PythonAnywhere. The code cell contains a script to read CSV files, drop columns, and print specific rows. The output cell shows the resulting DataFrame with columns: 'Name_Youtuber', 'Genre', 'Age', 'Subscribe', and 'Category'. The 'Genre' column has been renamed to 'Kategori'.

```
RENAME
  0 Raditza Dika   L  34 Komedi 7000000
  1 Staten Prod    L  20 Daily Vlog 1200000
  2 Arista Pradi   P  25 Food Vlog 2000000
  3 Sarah Alisid   P  23 Game 2000000
  4 Sarah Alisid   P  30 Komedi 8000000
  5 Chandra Llow   L  26 Sketsa 3000000
```

8) Contoh iloc

9) Contoh loc

The screenshot shows the PyCharm IDE interface with the following details:

- File**, **Edit**, **Selection**, **View**, **Go**, **Run**, **...** menu bar.
- PythonB** tab in the top right corner.
- Project Structure** sidebar on the left showing packages like `src`, `__pycache__`, `py2_modulay`, and `py2_modulay.py`.
- Code Editor** pane displaying Python code for `py2_modulay.py`. The code uses list slicing to extract specific records from a list of dictionaries based on the `name_youtuber` key.
- Terminal** pane at the bottom showing the output of running the script:

 - Execution command: `PS C:\Python&c:\PycharmProjects\py2_modulay> python c:\Python\py2_modulay.py`
 - Output:
 - Loc menampilkan record 1 kolom, dengan results nama kolanya name_youtuber
 - 1
 - Statement Pround
 - 2
 - Artika Artika
 - 3
 - Luthfi Luthfi
 - 4
 - Chandra Chandra
 - Name: Name_Youtuber, type: object
 - Loc menampilkan record dari indeks ke-4 sampai ke-3 dari kolom name_youtuber
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
 - 11
 - 12
 - 13
 - 14
 - 15
 - 16
 - 17
 - 18
 - 19
 - 20
 - 21
 - 22
 - 23
 - 24
 - 25
 - 26
 - 27
 - 28
 - 29
 - 30
 - 31
 - 32
 - 33
 - 34
 - 35
 - 36
 - 37
 - 38
 - 39
 - 40
 - 41
 - 42
 - 43
 - 44
 - 45
 - 46
 - 47
 - 48
 - 49
 - 50
 - 51
 - 52
 - 53
 - 54
 - 55
 - 56
 - 57
 - 58
 - 59
 - 60
 - 61
 - 62
 - 63
 - 64
 - 65
 - 66
 - 67
 - 68
 - 69
 - 70
 - 71
 - 72
 - 73
 - 74
 - 75
 - 76
 - 77
 - 78
 - 79
 - 80
 - 81
 - 82
 - 83
 - 84
 - 85
 - 86
 - 87
 - 88
 - 89
 - 90
 - 91
 - 92
 - 93
 - 94
 - 95
 - 96
 - 97
 - 98
 - 99
 - 100
 - 101
 - 102
 - 103
 - 104
 - 105
 - 106
 - 107
 - 108
 - 109
 - 110
 - 111
 - 112
 - 113
 - 114
 - 115
 - 116
 - 117
 - 118
 - 119
 - 120
 - 121
 - 122
 - 123
 - 124
 - 125
 - 126
 - 127
 - 128
 - 129
 - 130
 - 131
 - 132
 - 133
 - 134
 - 135
 - 136
 - 137
 - 138
 - 139
 - 140
 - 141
 - 142
 - 143
 - 144
 - 145
 - 146
 - 147
 - 148
 - 149
 - 150
 - 151
 - 152
 - 153
 - 154
 - 155
 - 156
 - 157
 - 158
 - 159
 - 160
 - 161
 - 162
 - 163
 - 164
 - 165
 - 166
 - 167
 - 168
 - 169
 - 170
 - 171
 - 172
 - 173
 - 174
 - 175
 - 176
 - 177
 - 178
 - 179
 - 180
 - 181
 - 182
 - 183
 - 184
 - 185
 - 186
 - 187
 - 188
 - 189
 - 190
 - 191
 - 192
 - 193
 - 194
 - 195
 - 196
 - 197
 - 198
 - 199
 - 200
 - 201
 - 202
 - 203
 - 204
 - 205
 - 206
 - 207
 - 208
 - 209
 - 210
 - 211
 - 212
 - 213
 - 214
 - 215
 - 216
 - 217
 - 218
 - 219
 - 220
 - 221
 - 222
 - 223
 - 224
 - 225
 - 226
 - 227
 - 228
 - 229
 - 230
 - 231
 - 232
 - 233
 - 234
 - 235
 - 236
 - 237
 - 238
 - 239
 - 240
 - 241
 - 242
 - 243
 - 244
 - 245
 - 246
 - 247
 - 248
 - 249
 - 250
 - 251
 - 252
 - 253
 - 254
 - 255
 - 256
 - 257
 - 258
 - 259
 - 260
 - 261
 - 262
 - 263
 - 264
 - 265
 - 266
 - 267
 - 268
 - 269
 - 270
 - 271
 - 272
 - 273
 - 274
 - 275
 - 276
 - 277
 - 278
 - 279
 - 280
 - 281
 - 282
 - 283
 - 284
 - 285
 - 286
 - 287
 - 288
 - 289
 - 290
 - 291
 - 292
 - 293
 - 294
 - 295
 - 296
 - 297
 - 298
 - 299
 - 300
 - 301
 - 302
 - 303
 - 304
 - 305
 - 306
 - 307
 - 308
 - 309
 - 310
 - 311
 - 312
 - 313
 - 314
 - 315
 - 316
 - 317
 - 318
 - 319
 - 320
 - 321
 - 322
 - 323
 - 324
 - 325
 - 326
 - 327
 - 328
 - 329
 - 330
 - 331
 - 332
 - 333
 - 334
 - 335
 - 336
 - 337
 - 338
 - 339
 - 340
 - 341
 - 342
 - 343
 - 344
 - 345
 - 346
 - 347
 - 348
 - 349
 - 350
 - 351
 - 352
 - 353
 - 354
 - 355
 - 356
 - 357
 - 358
 - 359
 - 360
 - 361
 - 362
 - 363
 - 364
 - 365
 - 366
 - 367
 - 368
 - 369
 - 370
 - 371
 - 372
 - 373
 - 374
 - 375
 - 376
 - 377
 - 378
 - 379
 - 380
 - 381
 - 382
 - 383
 - 384
 - 385
 - 386
 - 387
 - 388
 - 389
 - 390
 - 391
 - 392
 - 393
 - 394
 - 395
 - 396
 - 397
 - 398
 - 399
 - 400
 - 401
 - 402
 - 403
 - 404
 - 405
 - 406
 - 407
 - 408
 - 409
 - 410
 - 411
 - 412
 - 413
 - 414
 - 415
 - 416
 - 417
 - 418
 - 419
 - 420
 - 421
 - 422
 - 423
 - 424
 - 425
 - 426
 - 427
 - 428
 - 429
 - 430
 - 431
 - 432
 - 433
 - 434
 - 435
 - 436
 - 437
 - 438
 - 439
 - 440
 - 441
 - 442
 - 443
 - 444
 - 445
 - 446
 - 447
 - 448
 - 449
 - 450
 - 451
 - 452
 - 453
 - 454
 - 455
 - 456
 - 457
 - 458
 - 459
 - 460
 - 461
 - 462
 - 463
 - 464
 - 465
 - 466
 - 467
 - 468
 - 469
 - 470
 - 471
 - 472
 - 473
 - 474
 - 475
 - 476
 - 477
 - 478
 - 479
 - 480
 - 481
 - 482
 - 483
 - 484
 - 485
 - 486
 - 487
 - 488
 - 489
 - 490
 - 491
 - 492
 - 493
 - 494
 - 495
 - 496
 - 497
 - 498
 - 499
 - 500
 - 501
 - 502
 - 503
 - 504
 - 505
 - 506
 - 507
 - 508
 - 509
 - 510
 - 511
 - 512
 - 513
 - 514
 - 515
 - 516
 - 517
 - 518
 - 519
 - 520
 - 521
 - 522
 - 523
 - 524
 - 525
 - 526
 - 527
 - 528
 - 529
 - 530
 - 531
 - 532
 - 533
 - 534
 - 535
 - 536
 - 537
 - 538
 - 539
 - 540
 - 541
 - 542
 - 543
 - 544
 - 545
 - 546
 - 547
 - 548
 - 549
 - 550
 - 551
 - 552
 - 553
 - 554
 - 555
 - 556
 - 557
 - 558
 - 559
 - 560
 - 561
 - 562
 - 563
 - 564
 - 565
 - 566
 - 567
 - 568
 - 569
 - 570
 - 571
 - 572
 - 573
 - 574
 - 575
 - 576
 - 577
 - 578
 - 579
 - 580
 - 581
 - 582
 - 583
 - 584
 - 585
 - 586
 - 587
 - 588
 - 589
 - 590
 - 591
 - 592
 - 593
 - 594
 - 595
 - 596
 - 597
 - 598
 - 599
 - 600
 - 601
 - 602
 - 603
 - 604
 - 605
 - 606
 - 607
 - 608
 - 609
 - 610
 - 611
 - 612
 - 613
 - 614
 - 615
 - 616
 - 617
 - 618
 - 619
 - 620
 - 621
 - 622
 - 623
 - 624
 - 625
 - 626
 - 627
 - 628
 - 629
 - 630
 - 631
 - 632
 - 633
 - 634
 - 635
 - 636
 - 637
 - 638
 - 639
 - 640
 - 641
 - 642
 - 643
 - 644
 - 645
 - 646
 - 647
 - 648
 - 649
 - 650
 - 651
 - 652
 - 653
 - 654
 - 655
 - 656
 - 657
 - 658
 - 659
 - 660
 - 661
 - 662
 - 663
 - 664
 - 665
 - 666
 - 667
 - 668
 - 669
 - 670
 - 671
 - 672
 - 673
 - 674
 - 675
 - 676
 - 677
 - 678
 - 679
 - 680
 - 681
 - 682
 - 683
 - 684
 - 685
 - 686
 - 687
 - 688
 - 689
 - 690
 - 691
 - 692
 - 693
 - 694
 - 695
 - 696
 - 697
 - 698
 - 699
 - 700
 - 701
 - 702
 - 703
 - 704
 - 705
 - 706
 - 707
 - 708
 - 709
 - 710
 - 711
 - 712
 - 713
 - 714
 - 715
 - 716
 - 717
 - 718
 - 719
 - 720
 - 721
 - 722
 - 723
 - 724
 - 725
 - 726
 - 727
 - 728
 - 729
 - 730
 - 731
 - 732
 - 733
 - 734
 - 735
 - 736
 - 737
 - 738
 - 739
 - 740
 - 741
 - 742
 - 743
 - 744
 - 745
 - 746
 - 747
 - 748
 - 749
 - 750
 - 751
 - 752
 - 753
 - 754
 - 755
 - 756
 - 757
 - 758
 - 759
 - 760
 - 761
 - 762
 - 763
 - 764
 - 765
 - 766
 - 767
 - 768
 - 769
 - 770
 - 771
 - 772
 - 773
 - 774
 - 775
 - 776
 - 777
 - 778
 - 779
 - 780
 - 781
 - 782
 - 783
 - 784
 - 785
 - 786
 - 787
 - 788
 - 789
 - 790
 - 791
 - 792
 - 793
 - 794
 - 795
 - 796
 - 797
 - 798
 - 799
 - 800
 - 801
 - 802
 - 803
 - 804
 - 805
 - 806
 - 807
 - 808
 - 809
 - 810
 - 811
 - 812
 - 813
 - 814
 - 815
 - 816
 - 817
 - 818
 - 819
 - 820
 - 821
 - 822
 - 823
 - 824
 - 825
 - 826
 - 827
 - 828
 - 829
 - 830
 - 831
 - 832
 - 833
 - 834
 - 835
 - 836
 - 837
 - 838
 - 839
 - 840
 - 841
 - 842
 - 843
 - 844
 - 845
 - 846
 - 847
 - 848
 - 849
 - 850
 - 851
 - 852
 - 853
 - 854
 - 855
 - 856
 - 857
 - 858
 - 859
 - 860
 - 861
 - 862
 - 863
 - 864
 - 865
 - 866
 - 867
 - 868
 - 869
 - 870
 - 871
 - 872
 - 873
 - 874
 - 875
 - 876
 - 877
 - 878
 - 879
 - 880
 - 881
 - 882
 - 883
 - 884
 - 885
 - 886
 - 887
 - 888
 - 889
 - 890
 - 891
 - 892
 - 893
 - 894
 - 895
 - 896
 - 897
 - 898
 - 899
 - 900
 - 901
 - 902
 - 903
 - 904
 - 905
 - 906
 - 907
 - 908
 - 909
 - 910
 - 911
 - 912
 - 913
 - 914
 - 915
 - 916
 - 917
 - 918
 - 919
 - 920
 - 921
 - 922
 - 923
 - 924
 - 925
 - 926
 - 927
 - 928
 - 929
 - 930
 - 931
 - 932
 - 933
 - 934
 - 935
 - 936
 - 937
 - 938
 - 939
 - 940
 - 941
 - 942
 - 943
 - 944
 - 945
 - 946
 - 947
 - 948
 - 949
 - 950
 - 951
 - 952
 - 953
 - 954
 - 955
 - 956
 - 957
 - 958
 - 959
 - 960
 - 961
 - 962
 - 963
 - 964
 - 965
 - 966
 - 967
 - 968
 - 969
 - 970
 - 971
 - 972
 - 973
 - 974
 - 975
 - 976
 - 977
 - 978
 - 979
 - 980
 - 981
 - 982
 - 983
 - 984
 - 985
 - 986
 - 987
 - 988
 - 989
 - 990
 - 991
 - 992
 - 993
 - 994
 - 995
 - 996
 - 997
 - 998
 - 999
 - 1000
 - 1001
 - 1002
 - 1003
 - 1004
 - 1005
 - 1006
 - 1007
 - 1008
 - 1009
 - 1010
 - 1011
 - 1012
 - 1013
 - 1014
 - 1015
 - 1016
 - 1017
 - 1018
 - 1019
 - 1020
 - 1021
 - 1022
 - 1023
 - 1024
 - 1025
 - 1026
 - 1027
 - 1028
 - 1029
 - 1030
 - 1031
 - 1032
 - 1033
 - 1034
 - 1035
 - 1036
 - 1037
 - 1038
 - 1039
 - 1040
 - 1041
 - 1042
 - 1043
 - 1044
 - 1045
 - 1046
 - 1047
 - 1048
 - 1049
 - 1050
 - 1051
 - 1052
 - 1053
 - 1054
 - 1055
 - 1056
 - 1057
 - 1058
 - 1059
 - 1060
 - 1061
 - 1062
 - 1063
 - 1064
 - 1065
 - 1066
 - 1067
 - 1068
 - 1069
 - 1070
 - 1071
 - 1072
 - 1073
 - 1074
 - 1075
 - 1076
 - 1077
 - 1078
 - 1079
 - 1080
 - 1081
 - 1082
 - 1083
 - 1084
 - 1085
 - 1086
 - 1087
 - 1088
 - 1089
 - 1090
 - 1091
 - 1092
 - 1093
 - 1094
 - 1095
 - 1096
 - 1097
 - 1098
 - 1099
 - 1100
 - 1101
 - 1102
 - 1103
 - 1104
 - 1105
 - 1106
 - 1107
 - 1108
 - 1109
 - 1110
 - 1111
 - 1112
 - 1113
 - 1114
 - 1115
 - 1116
 - 1117
 - 1118
 - 1119
 - 1120
 - 1121
 - 1122
 - 1123
 - 1124
 - 1125
 - 1126
 - 1127
 - 1128
 - 1129
 - 1130
 - 1131
 - 1132
 - 1133
 - 1134
 - 1135
 - 1136
 - 1137
 - 1138
 - 1139
 - 1140
 - 1141
 - 1142
 - 1143
 - 1144
 - 1145
 - 1146
 - 1147
 - 1148
 - 1149
 - 1150
 - 1151
 - 1152
 - 1153
 - 1154
 - 1155
 - 1156
 - 1157
 - 1158
 - 1159
 - 1160
 - 1161
 - 1162
 - 1163
 - 1164
 - 1165
 - 1166
 - 1167
 - 1168
 - 1169
 - 1170
 - 1171
 - 1172
 - 1173
 - 1174
 - 1175
 - 1176
 - 1177
 - 1178
 - 1179
 - 1180
 - 1181
 - 1182
 - 1183
 - 1184
 - 1185
 - 1186
 - 1187
 - 1188
 - 1189
 - 1190
 - 1191
 - 1192
 - 1193
 - 1194
 - 1195
 - 1196
 - 1197
 - 1198
 - 1199
 - 1200
 - 1201
 - 1202
 - 1203
 - 1204
 - 1205
 - 1206
 - 1207
 - 1208
 - 1209
 - 1210
 - 1211
 - 1212
 - 1213
 - 1214
 - 1215
 - 1216
 - 1217
 - 1218
 - 1219
 - 1220
 - 1221
 - 1222
 - 1223
 - 1224
 - 1225
 - 1226
 - 1227
 - 1228
 - 1229
 - 1230
 - 1231
 - 1232
 - 1233
 - 1234
 - 1235
 - 1236
 - 1237
 - 1238
 - 1239
 - 1240
 - 1241
 - 1242
 - 1243
 - 1244
 - 1245
 - 1246
 - 1247
 - 1248
 - 1249
 - 1250
 - 1251
 - 1252
 - 1253
 - 1254
 - 1255
 - 1256
 - 1257
 - 1258
 - 1259
 - 1260
 - 1261
 - 1262
 - 1263
 - 1264
 - 1265
 - 1266
 - 1267
 - 1268
 - 1269
 - 1270
 - 1271
 - 1272
 - 1273
 - 1274
 - 1275
 - 1276
 - 1277
 - 1278
 - 1279
 - 1280
 - 1281
 - 1282
 - 1283
 - 1284
 - 1285
 - 1286
 - 1287
 - 1288
 - 1289
 - 1290
 - 1291
 - 1292
 - 1293
 - 1294
 - 1295
 - 1296
 - 1297
 - 1298
 - 1299
 - 1300
 - 1301
 - 1302
 - 1303
 - 1304
 - 1305
 - 1306
 - 1307
 - 1308
 - 1309
 - 1310
 - 1311
 - 1312
 - 1313
 - 1314
 - 1315
 - 1316
 - 1317
 - 1318
 - 1319
 - 1320
 - 1321
 - 1322
 - 1323
 - 1324
 - 1325
 - 1326
 - 1327
 - 1328
 - 1329
 - 1330
 - 1331
 - 1332
 - 1333
 - 1334
 - 1335
 - 1336
 - 1337
 - 1338
 - 1339
 - 1340
 - 1341
 - 1342
 - 1343
 - 1344
 - 1345
 - 1346
 - 1347
 - 1348
 - 1349
 - 1350
 - 1351
 - 1352
 - 1353
 - 1354
 - 1355
 - 1356
 - 1357
 - 1358
 - 1359
 - 1360
 - 1361
 - 1362
 - 1363
 - 1364
 - 1365
 - 1366
 - 1367
 - 1368
 - 1369
 - 1370
 - 1371
 - 1372
 - 1373
 - 1374
 - 1375
 - 1376
 - 1377
 - 1378
 - 1379
 - 1380
 - 1381
 - 1382
 - 1383
 - 1384
 - 1385
 - 1386
 - 1387
 - 1388
 - 1389
 - 1390
 - 1391
 - 1392
 - 1393
 - 1394
 - 1395
 - 1396
 - 1397
 - 1398
 - 1399
 - 1400
 - 1401
 - 1402
 - 1403
 - 1404
 - 1405
 - 1406
 - 1407
 - 1408
 - 1409
 - 1410
 - 1411
 - 1412
 - 1413
 - 1414
 - 1415
 - 1416
 - 1417
 - 1418
 - 1419
 - 1420
 - 1421
 - 1422
 - 1423
 - 1424
 - 1425
 - 1426
 - 1427
 - 1428
 - 1429
 - 1430
 - 1431
 - 1432
 - 1433
 - 1434
 - 1435
 - 1436
 - 1437
 - 1438
 - 1439
 - 1440
 - 1441
 - 1442
 - 1443
 - 1444
 - 1445
 - 1446
 - 1447
 - 1448
 - 1449
 - 1450
 - 1451
 - 145

10) Contoh Mengisi Nilai Sama Untuk 1 Kolom

The screenshot shows a Windows 10 desktop with the Visual Studio Code application open. The title bar reads "Python8". The left sidebar has sections for EXPLORER, PYTHON, and TERMINAL. The EXPLORER section shows files like Modul4.py, Modul5.py, py2_modul7.py, data1.csv, Modul6.py, py2_modul6.py, and data2.csv. The PYTHON section shows a file named "py2_modul6.py" with the following code:

```
235
236     Mengisi nilai sama untuk 1 kolom
237     print("Mengisi nilai sama untuk 1 kolom")
238     sample["subscribe"] = 1
239     print(sample)
240
241
```

The TERMINAL tab shows the command PS C:\Python8 & cd\Python\env\Scripts\activate.ps1 followed by PS C:\Python8 & cd\python\env\Scripts\python.exe c:/Python8/py2_modul6.py. The output of the script is displayed below:

```
Mengisi nilai sama untuk 1 kolom
Name          Value   Jenis_Kelamin Umur Kategori1 Subscribes subscribe
0  Ratu Dibia    L      34   Komedi 7000000 1
1  Sistemem Prond L      29   Daily Vlog 1200000 1
2  Arif Muhammad L      28   Daily Vlog 3000000 1
3  Anisa Atilza P      25   Food Travel 6000000 1
4  Sarah Villard P      22   Food Travel 2000000 1
5  MLI          L      30   Komedi 8000000 1
6  Chandra Low  L      26   Sketsa 3000000 1
```

The status bar at the bottom indicates the code has 239 lines, 1 space, and 112 characters. It also shows tabs for Python 3.11.2 (env) and env.

11) Contoh Sorting

12) Contoh Filter

The screenshot shows a Python script named `py2_modul8.py` in the VS Code editor. The code uses the `pandas` library to filter a dataset. It prints the original records, then filters for users aged 28 or older, and finally filters for users aged 27 and in the 'Daily Vlog' category. The terminal below shows the execution of the script and its output.

```
File Edit Selection View Go Run ... PythonB
EXPLORER > PYTHON > py2_modul8.py
Modul1.py Modul2.py py2_modul8.py data2.csv Modul8.py py2_modul8.py data.csv
249 #Filter record yang umurnya lebih dari 28 dan akan menampilkan status true atau false
250 print("FILTER record yang umurnya lebih dari 28 dan akan menampilkan status true atau false ")
251 ans=sample["Umur">>28
252 print(ans)
253 print()
254 print("filter memungkinkan record")
255 print("Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )")
256 filter=sample["Umur"]>28
257 filter=filter[filter["Kategori"]=="Daily Vlog"]
258 print(filter)
259 print("Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )")
260 print("FILTER dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )")
261 filter=sample["Umur"]==27 & sample["Kategori"]=="Daily Vlog"
262 print(filter)
263 filter=filter[filter["Umur"]==27]
264 print(filter)
265 print(filter)
266
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Python\ & c:/Python/env/Scripts/activate.ps1
(env) PS C:\Python\ & c:/Python/env/Scripts/python.exe c:/Python/py2_modul8.py
FILTER record yang umurnya lebih dari 28 dan akan menampilkan status true atau false
0 True
1 True
2 False
3 False
4 False
5 True
6 True
Name: Umur, dtype: bool

Filter memungkinkan record
   Nama_Vtuber Jenis_Kelamin Umur Kategori Subscribe
0  Headline Rock     L    29 Daily Vlog 1500000
1  Statement Prod   L    29 Daily Vlog 1500000
2  Arfiana Putri    P    28 Daily Vlog 3000000
3  Antika Astria    P    25 Daily Vlog 3000000
4  Sarah Viladi    P    23 Gamer 2000000
5  Rizki ARI       L    30 Gamer 2000000
6  Chandra Low     L    26 Skater 3000000

Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )
   Nama_Vtuber Jenis_Kelamin Umur Kategori Subscribe
0  Headline Rock     L    29 Daily Vlog 1500000
1  Statement Prod   L    29 Daily Vlog 1500000
2  Arfiana Putri    P    28 Daily Vlog 3000000
3  Antika Astria    P    25 Daily Vlog 3000000
4  Sarah Viladi    P    23 Gamer 2000000
5  Rizki ARI       L    30 Gamer 2000000
6  Chandra Low     L    26 Skater 3000000
0 (env) PS C:\Vpython>
```

MODUL 5

PYTHON (FOR DATA SCIENCE)

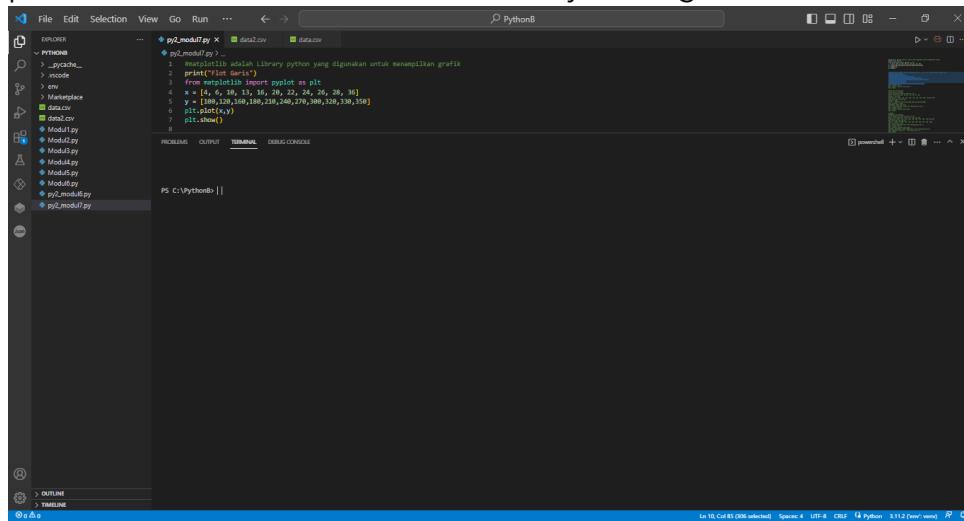
VISUALISASI DATA - MATPLOTLIB

A. Matplotlib

Merupakan library Python 2D yang dapat menghasilkan plot dengan kualitas tinggi dalam berbagai format yang dapat digunakan di banyak platform. Matplotlib dapat digunakan sebagai pembuat grafik dalam berbagai platform, seperti Python dan Jupyter. Grafik yang dapat dibuat beragam, seperti grafik garis, batang, lingkaran, histogram. Terdapat berbagai macam jenis plot/grafik yang dapat dibuat oleh Matplotlib sebagai berikut :

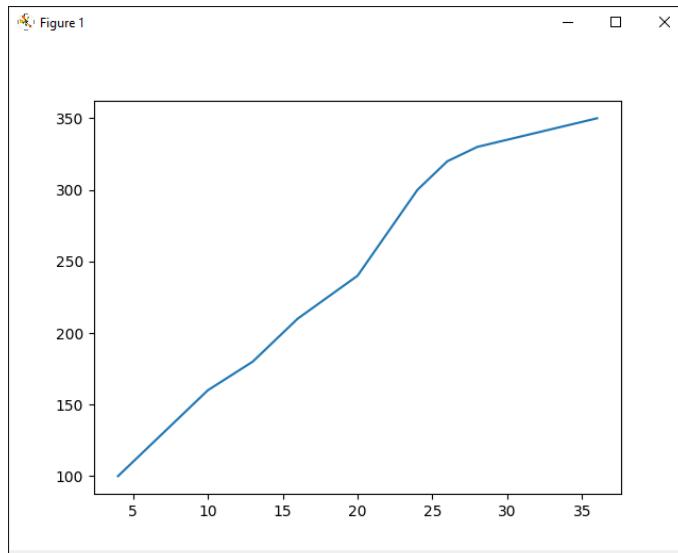
1. Jenis Plot Garis

Merupakan representasi berupa garis yang menghubungkan antar posisi koordinat data. Cara membuatnya sebagai berikut.

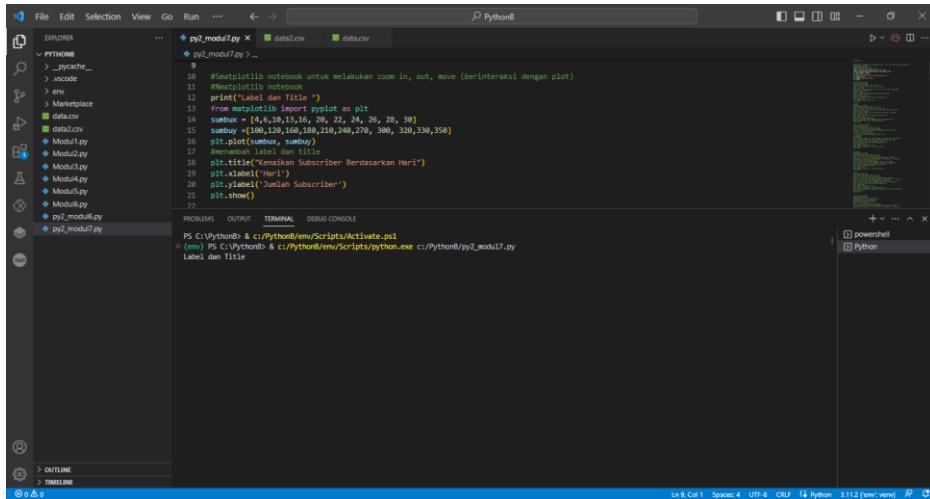


The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists several Python files and CSV files. The code editor window displays a Python script named 'py2_modul5.py'. The code itself is as follows:

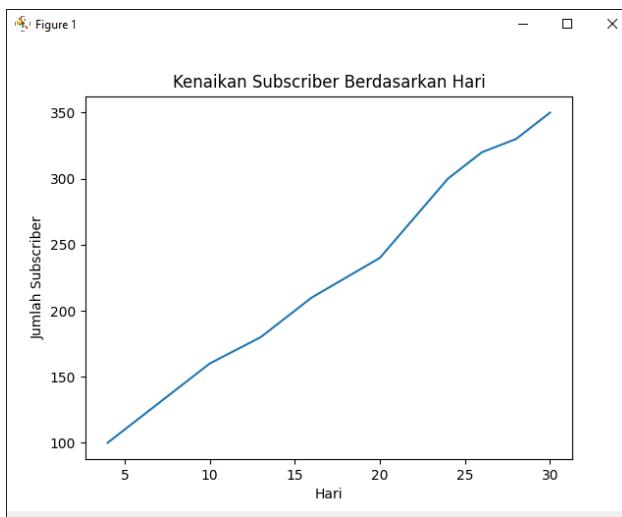
```
py2_modul5.py
...
#matplotlib adalah library python yang digunakan untuk menampilkan grafik
1 print("Matplotlib")
2 from matplotlib import pyplot as plt
3 x = [4, 6, 8, 10, 12, 16, 18, 20, 22, 24, 26, 28, 30]
4 y = [100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350]
5 plt.plot(x,y)
6 plt.show()
```



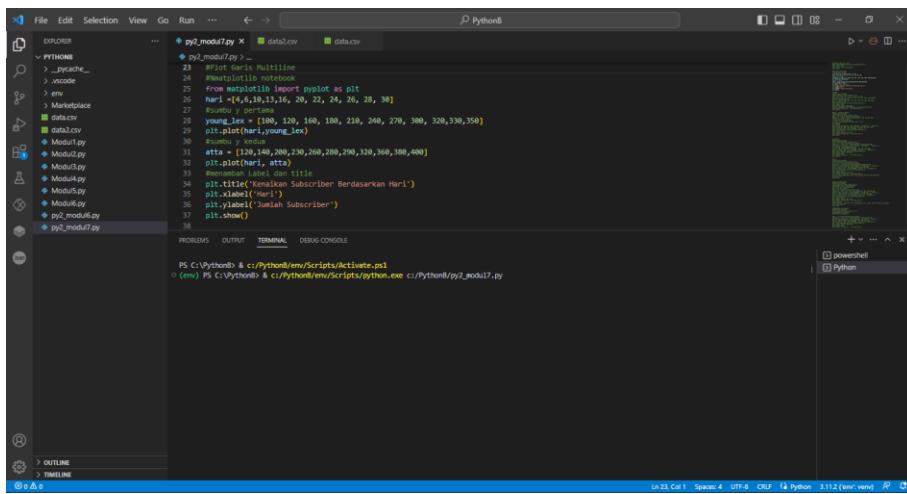
2. Contoh Menambah Label dan Title



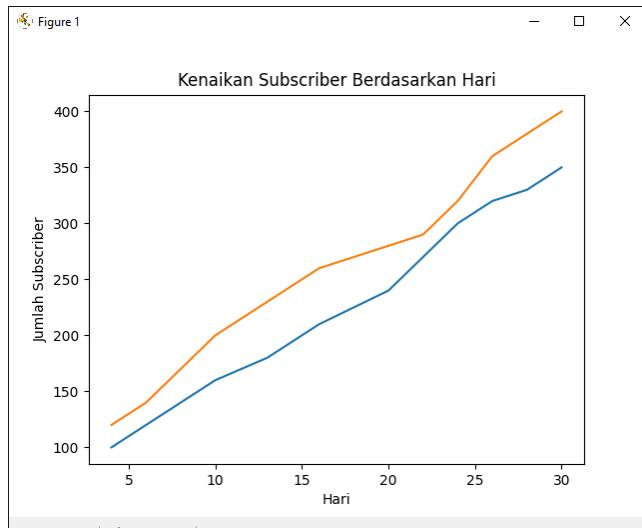
```
#Seaborn library notebook untuk melakukan zoom in, out, move (berinteraksi dengan plot)
#Matplotlib notebook
print("Label dan Title")
hari = [4,6,10,13,16,20,22,24,26,28,30]
sumbuy = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(sumbuy, sumbuy)
plt.title("Kenaikan Subscriber Berdasarkan Hari")
plt.xlabel("Hari")
plt.ylabel("Jumlah Subscriber")
plt.show()
```



3. Contoh Plot Garis Multiline



```
#grid library
#Matplotlib notebook
from matplotlib import pyplot as plt
hari = [4,6,10,13,16,20,22,24,26,28,30]
sumbuy1 = [100,120,160,180,210,240,270,300,320,330,350]
sumbuy2 = [200,220,240,260,280,290,310,330,350,380,400]
plt.plot(hari,sumbuy1)
plt.plot(hari,sumbuy2)
plt.title("Kenaikan Subscriber Berdasarkan Hari")
plt.xlabel("Hari")
plt.ylabel("Jumlah Subscriber")
plt.show()
```

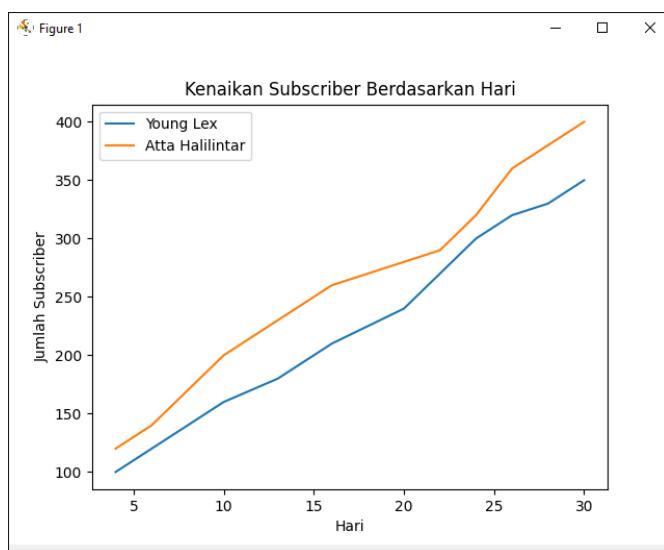


4. Contoh Legend

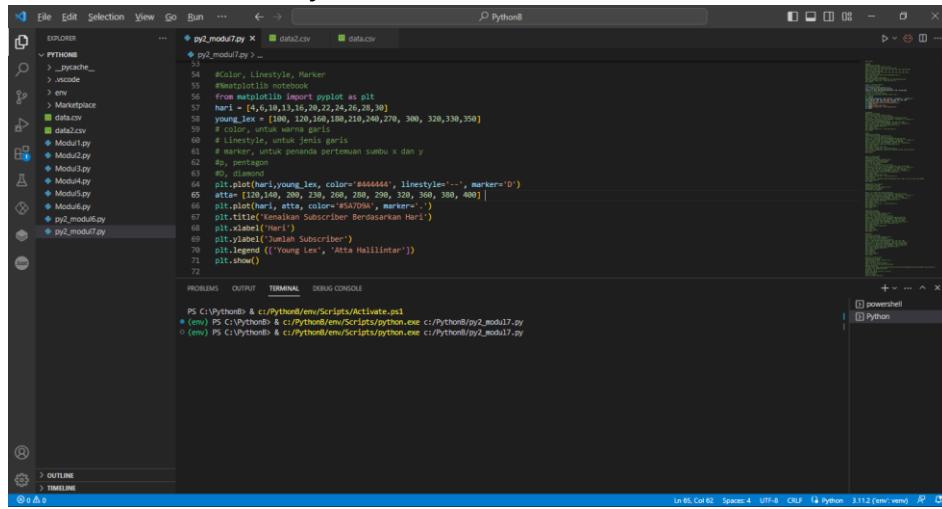
```

File Edit Selection View Go Run ... < > PythonB
EXPLORER ... py2_modul7.py data2.csv data.csv
> _pycache_ ...
> vscode ...
> env ...
> Marketplace ...
data.csv
Modul1.py
Modul2.py
Modul3.py
Modul4.py
Modul5.py
Modul6.py
py2_modul6.py
py2_modul7.py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> & c:\PythonB\env\Scripts\Activate.ps1
● (env) PS C:\PythonB> & c:\PythonB\env\Scripts\python.exe c:/PythonB/py2_modul7.py
● (env) PS C:\PythonB>

```

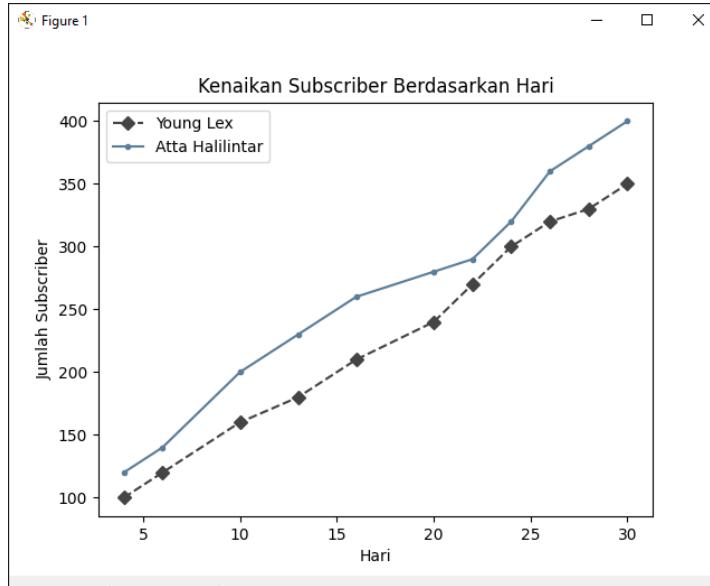


5. Contoh Color, Linestyle, Marker

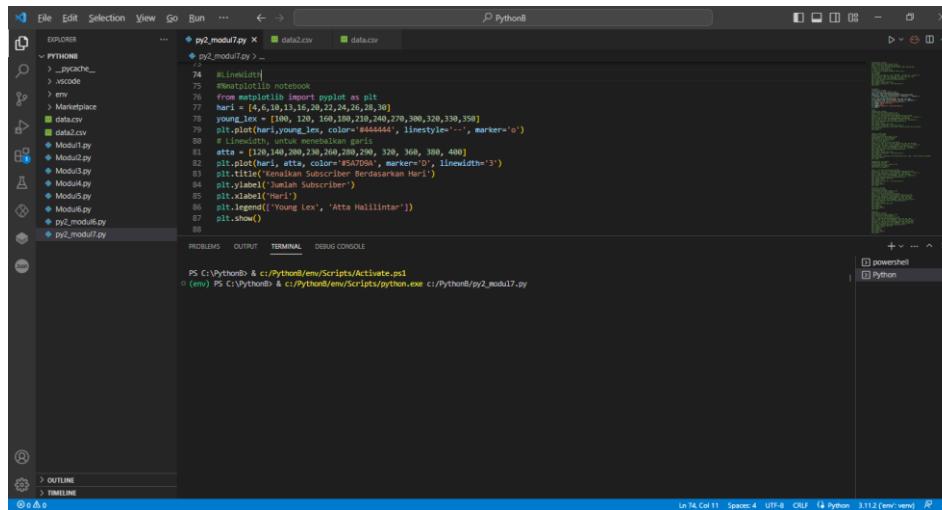


```
File Edit Selection View Go Run ... < - > PythonB

EXPLORER ... py2_modul7.py data2.csv data.csv
py2_modul7.py ...
1 #Color, Linestyle, Marker
2 #matplotlib notebook
3 from matplotlib import pyplot as plt
4 hari = [4,6,10,13,16,20,22,24,26,28,30]
5 young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
6 # color, untuk warna garis
7 # marker, untuk penanda pertemuan sumbu x dan y
8 # Linestyle, untuk jenis garis
9 atta = [120,140,200,230,260,280,290,300,320,330,400]
10 atta_label = [4,6,10,13,16,20,22,24,26,28,30]
11 atta_color = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 400]
12 atta_label_color = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 400]
13 plt.title('Kenaikan Subscriber Berdasarkan Hari')
14 plt.xlabel('Hari')
15 plt.ylabel('Jumlah Subscriber')
16 plt.legend(['Young Lex', 'Atta Halilintar'])
17 plt.show()
18
```

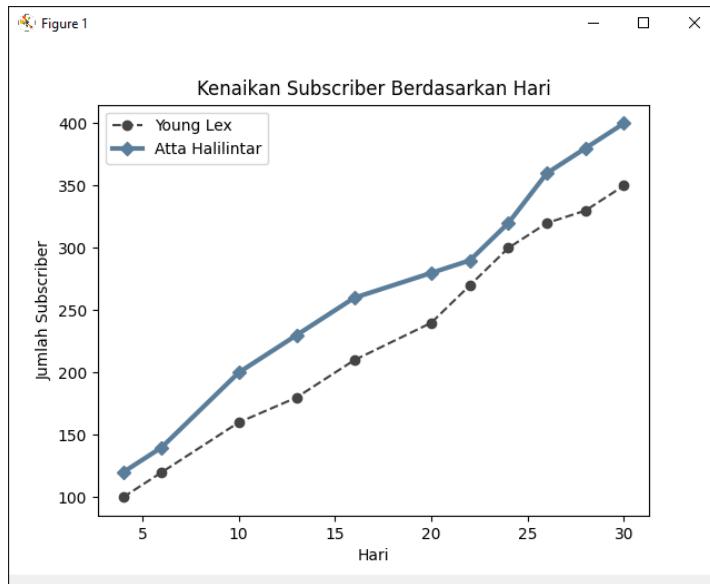


6. Contoh Linewidth



```
File Edit Selection View Go Run ... < - > PythonB

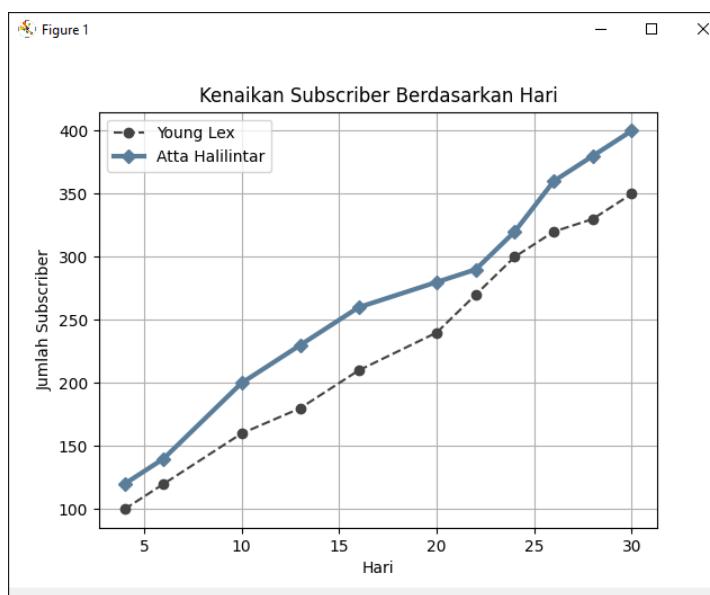
EXPLORER ... py2_modul7.py data2.csv data.csv
py2_modul7.py ...
1 #Linewidth
2 #matplotlib notebook
3 from matplotlib import pyplot as plt
4 hari = [4,6,10,13,16,20,22,24,26,28,30]
5 young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
6 plt.plot(hari,young_lex,color="#444444",linestyle='--',marker='o')
7 atta = [120,140,200,230,260,280,290,300,320,330,400]
8 plt.plot(hari,atta,color="#55708A",marker='o',linewidth=3)
9 plt.title('Kenaikan Subscriber Berdasarkan Hari')
10 plt.xlabel('Hari')
11 plt.ylabel('Jumlah Subscriber')
12 plt.legend(['Young Lex', 'Atta Halilintar'])
13 plt.show()
14
```



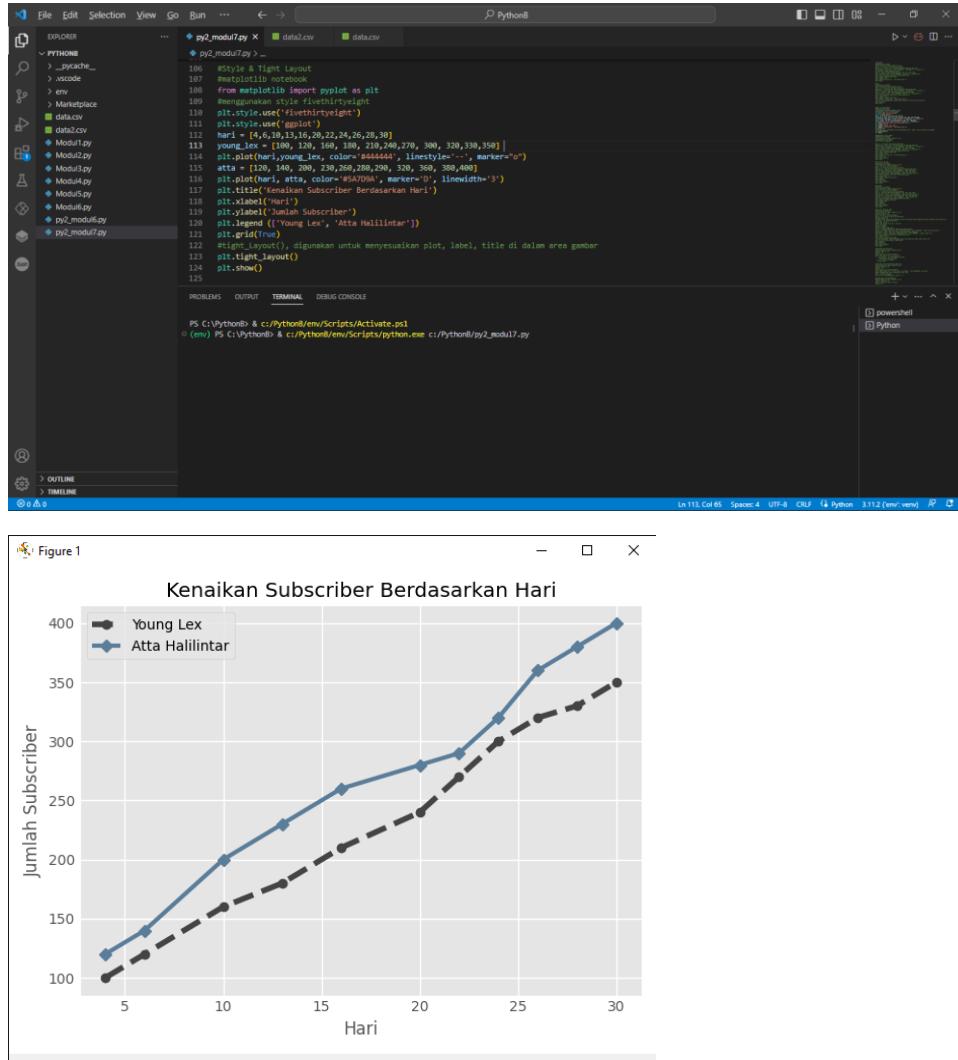
7. Contoh Grid

The screenshot shows a Python development environment with the following details:

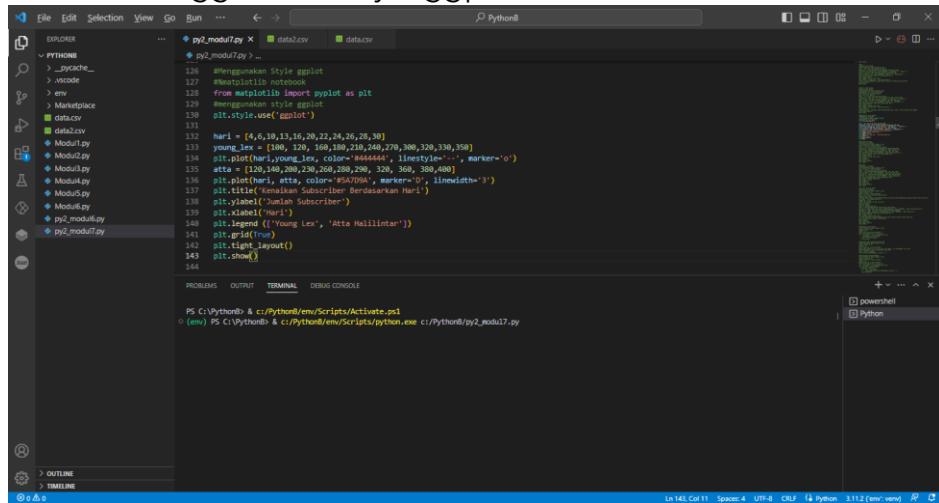
- File Explorer:** Shows files like `py2_modul7.py`, `data.csv`, and `data.py`.
- Code Editor:** Displays the content of `py2_modul7.py`. The code uses Matplotlib to plot data from `data.csv`, creating a scatter plot with horizontal grid lines.
- Terminal:** Shows the command `PS C:\Python\env\Scripts\Activitas.ps1` followed by the output of running `py2_modul7.py`.
- Output:** Shows the generated plot image.
- Debug Console:** Not visible in the screenshot.

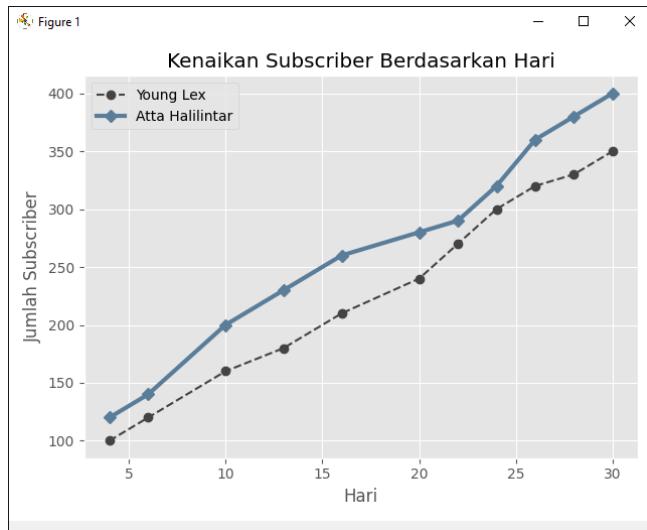


8. Contoh Menggunakan Style (fivethirtyeight) dan Tight Layout



9. Contoh Menggunakan Style ggplot





10. Contoh Plot Garis dan Bar

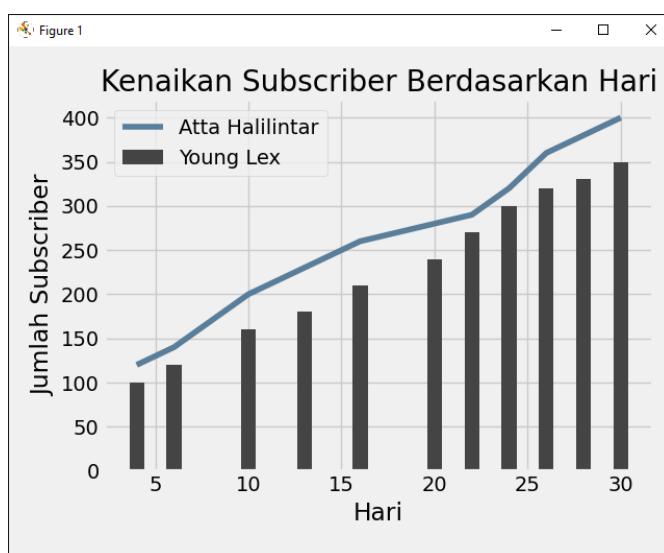
The screenshot shows the PyCharm IDE interface with the following details:

- File Structure:** The left sidebar shows a project named "DPROSES" with files like "py2_modul7Py", "py2_modul7Py.py", "data2.csv", and "data.csv".
- Code Editor:** The main window displays Python code for plotting subscriber data. The code uses pandas to read CSV files and matplotlib to create a bar chart.

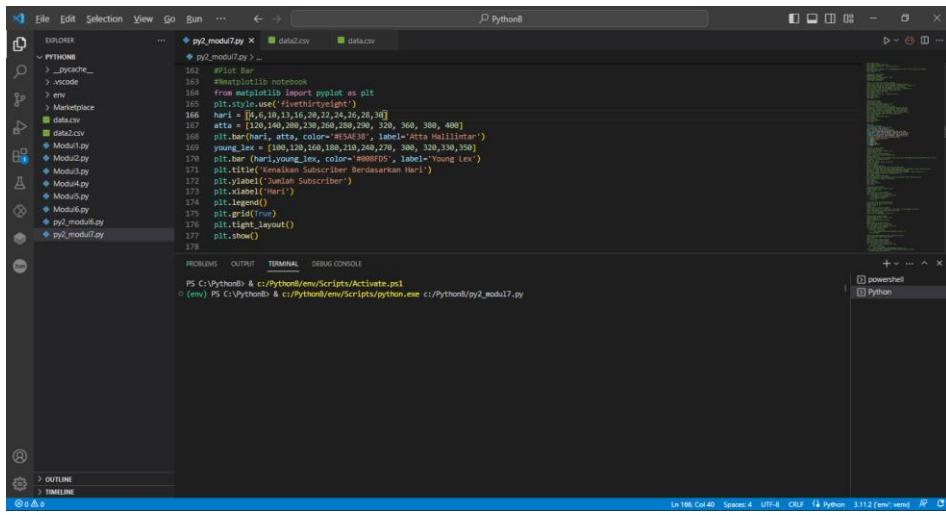
```
143 #!/usr/bin/python
144 # Berikut ini Bar chart dan Bar
145 # scatterplot menggunakan
146 # matplotlib
147 from matplotlib import pyplot as plt
148 plt.style.use('fivethirtyeight')
149 hari = [6,19,13,16,20,24,25,26,28,30]
150 hari_bar = [160, 180, 190, 200, 210, 220, 230, 240, 250, 260]
151 plt.bar(hari, hari_bar, color="#4444AA", label="Young Lek")
152 atta = [120,140,200,230,260,280,320,360,380,400]
153 plt.plot(hari,atta, color="#A7070A", label="Atta Hallilintar")
154 plt.title("Kemajuan Subscriber Berdasarkan Hari")
155 plt.xlabel("Hari")
156 plt.ylabel("Jumlah Subscriber")
157 plt.legend()
158 plt.grid(True)
159 plt.tight_layout()
160 plt.show()
```

- Terminal:** The bottom terminal window shows the command to activate the Python environment and run the script.

```
PS: C:\Python2\ & c:/Python3/env/Scripts/Activate.ps1
(venv) PS: C:\Python2\ & c:/Python3/env/Scripts/python.exe c:/Python2/py2_modul7.py
```



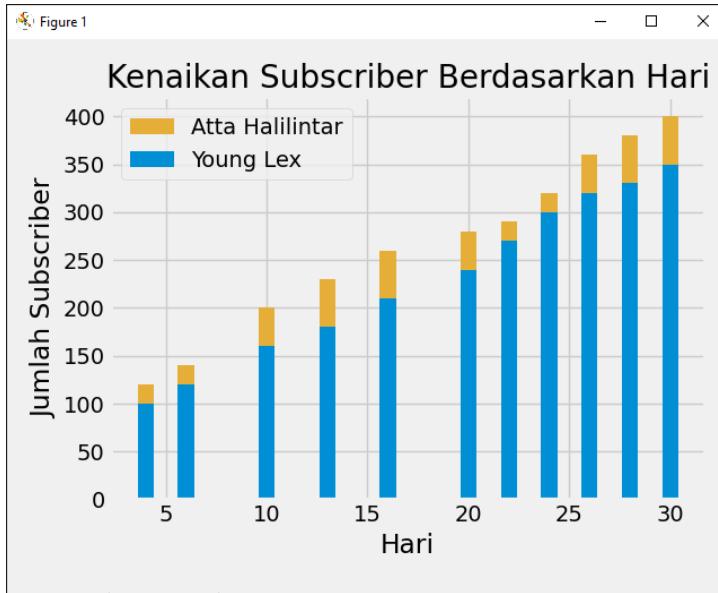
11. Contoh Plot Bar



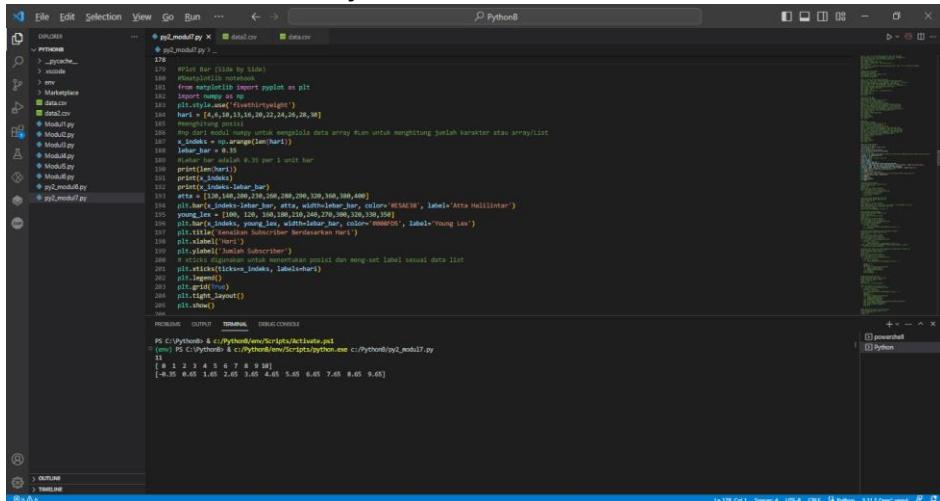
The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like `py2_modul7.py`, `data.csv`, and `data.csv`. The Editor tab contains the following Python code:

```
#py2_modul7.py
#matplotlib notebook
from matplotlib import pyplot as plt
plt.style.use('fivethirtyeight')
hari = [6,8,10,13,15,16,20,22,24,26,28,30]
atta = [120,140,200,230,260,280,290,320,360,380,400]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.bar(hari, atta, color='#E5A3B', label='Atta Halilintar')
plt.bar(hari, young_lex, bottom=atta, color='#0072BD', label='Young Lex')
plt.legend()
plt.tight_layout()
plt.show()
```

The terminal shows the command to run the script: `PS C:\Python38\& c:/Python38/Scripts/Activate.ps1` and `(env) PS C:\Python38\& c:/Python38/Scripts/python.exe c:/Python38/py2_modul7.py`.



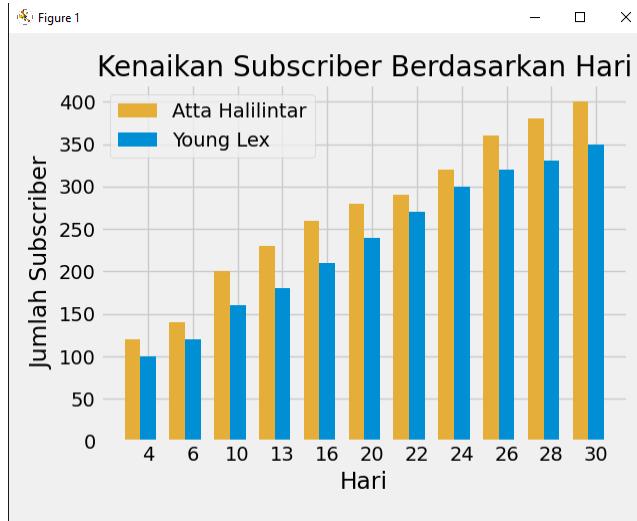
12. Contoh Plot Bar (Side by Side)



The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like `py2_modul7.py`, `data.csv`, and `data.csv`. The Editor tab contains the following Python code:

```
#py2_modul7.py
#matplotlib notebook
from matplotlib import pyplot as plt
plt.style.use('fivethirtyeight')
hari = [6,8,10,13,15,16,20,22,24,26,28,30]
x_index = np.arange(len(hari))
label_hari = []
for i in range(0, len(hari)):
    label_hari.append(str(hari[i]) + ' hari')
print(x_index)
print(label_hari)
plt.bar(x_index, atta, width=0.5, color='#E5A3B', label='Atta Halilintar')
plt.bar(x_index, young_lex, width=0.5, color='#0072BD', label='Young Lex')
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.xticks(x_index, label_hari)
plt.grid(True)
plt.tight_layout()
plt.show()
```

The terminal shows the command to run the script: `PS C:\Python38\& c:/Python38/Scripts/Activate.ps1` and `(env) PS C:\Python38\& c:/Python38/Scripts/python.exe c:/Python38/py2_modul7.py`.



B. Matplotlib - Import Data CSV

Berikut praktikum cara menampilkan plot dengan menggunakan file CSV. Dengan Plot Bar Ikuti langkah-langkah dibawah untuk menampilkan data dari CSV menjadi plot bar.

1. Tampilkan Key dan Value

2. Hitung Semua Jumlah Pengguna

3. Tampilkan Jumlah Pengguna Teratas dan Batas Tertentu

```
File Edit Selection View Go Run ... ⏪ ⏴ PythonB

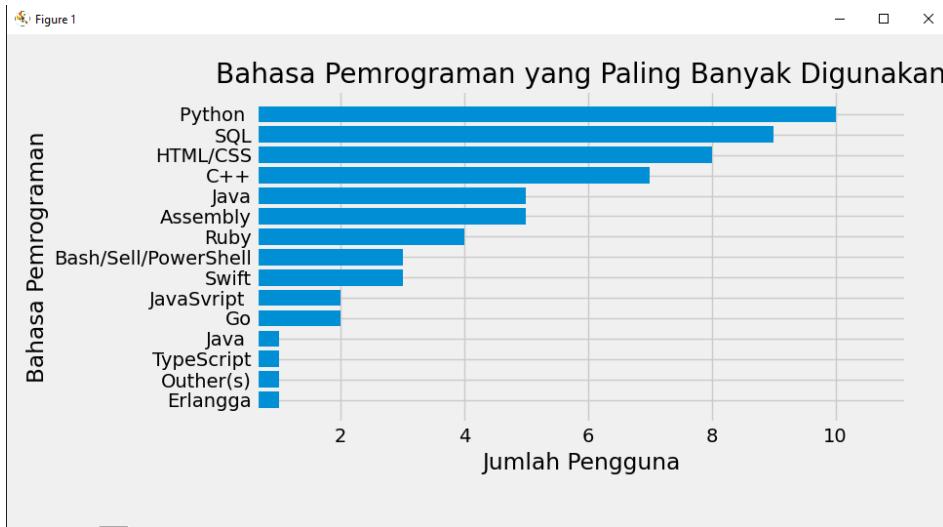
... + py2_modul7.py x data.csv data.csv
  + py2_modul7.py
    242 #Tampilkan Jumlah Pengguna Teratas dan Batas Tertentu
    243 from matplotlib import pyplot as plt
    244 from collections import Counter
    245 plt.style.use('fivethirtyeight')
    246 with open('data.csv') as csv_file:
    247     csv_reader = csv.DictReader(csv_file)
    248     hitung = Counter()
    249     for row in csv_reader:
    250         hitung.update(row['jumlahPengguna'].split(','))
    251     most_common_15 = hitung.most_common(15)
    252     print(hitung.most_common(15))
    253
    254
    255
    256
    257
    258
    259
    260
    261
    262
    263
    264
    265
    266
    267
    268
    269
    270
    271
    272
    273
    274
    275
    276
    277
    278
    279
    280
    281
    282
    283
    284
    285
    286
    287
    288
    289
    290
    291
    292
    293
    294
    295
    296
    297
    298
    299
    300
    301
    302
    303
    304
    305
    306
    307
    308
    309
    310
    311
    312
    313
    314
    315
    316
    317
    318
    319
    320
    321
    322
    323
    324
    325
    326
    327
    328
    329
    330
    331
    332
    333
    334
    335
    336
    337
    338
    339
    340
    341
    342
    343
    344
    345
    346
    347
    348
    349
    350
    351
    352
    353
    354
    355
    356
    357
    358
    359
    360
    361
    362
    363
    364
    365
    366
    367
    368
    369
    370
    371
    372
    373
    374
    375
    376
    377
    378
    379
    380
    381
    382
    383
    384
    385
    386
    387
    388
    389
    390
    391
    392
    393
    394
    395
    396
    397
    398
    399
    400
    401
    402
    403
    404
    405
    406
    407
    408
    409
    410
    411
    412
    413
    414
    415
    416
    417
    418
    419
    420
    421
    422
    423
    424
    425
    426
    427
    428
    429
    430
    431
    432
    433
    434
    435
    436
    437
    438
    439
    440
    441
    442
    443
    444
    445
    446
    447
    448
    449
    450
    451
    452
    453
    454
    455
    456
    457
    458
    459
    460
    461
    462
    463
    464
    465
    466
    467
    468
    469
    470
    471
    472
    473
    474
    475
    476
    477
    478
    479
    480
    481
    482
    483
    484
    485
    486
    487
    488
    489
    490
    491
    492
    493
    494
    495
    496
    497
    498
    499
    500
    501
    502
    503
    504
    505
    506
    507
    508
    509
    510
    511
    512
    513
    514
    515
    516
    517
    518
    519
    520
    521
    522
    523
    524
    525
    526
    527
    528
    529
    530
    531
    532
    533
    534
    535
    536
    537
    538
    539
    540
    541
    542
    543
    544
    545
    546
    547
    548
    549
    550
    551
    552
    553
    554
    555
    556
    557
    558
    559
    560
    561
    562
    563
    564
    565
    566
    567
    568
    569
    570
    571
    572
    573
    574
    575
    576
    577
    578
    579
    580
    581
    582
    583
    584
    585
    586
    587
    588
    589
    590
    591
    592
    593
    594
    595
    596
    597
    598
    599
    600
    601
    602
    603
    604
    605
    606
    607
    608
    609
    610
    611
    612
    613
    614
    615
    616
    617
    618
    619
    620
    621
    622
    623
    624
    625
    626
    627
    628
    629
    630
    631
    632
    633
    634
    635
    636
    637
    638
    639
    640
    641
    642
    643
    644
    645
    646
    647
    648
    649
    650
    651
    652
    653
    654
    655
    656
    657
    658
    659
    660
    661
    662
    663
    664
    665
    666
    667
    668
    669
    670
    671
    672
    673
    674
    675
    676
    677
    678
    679
    680
    681
    682
    683
    684
    685
    686
    687
    688
    689
    690
    691
    692
    693
    694
    695
    696
    697
    698
    699
    700
    701
    702
    703
    704
    705
    706
    707
    708
    709
    710
    711
    712
    713
    714
    715
    716
    717
    718
    719
    720
    721
    722
    723
    724
    725
    726
    727
    728
    729
    730
    731
    732
    733
    734
    735
    736
    737
    738
    739
    740
    741
    742
    743
    744
    745
    746
    747
    748
    749
    750
    751
    752
    753
    754
    755
    756
    757
    758
    759
    760
    761
    762
    763
    764
    765
    766
    767
    768
    769
    770
    771
    772
    773
    774
    775
    776
    777
    778
    779
    780
    781
    782
    783
    784
    785
    786
    787
    788
    789
    790
    791
    792
    793
    794
    795
    796
    797
    798
    799
    800
    801
    802
    803
    804
    805
    806
    807
    808
    809
    810
    811
    812
    813
    814
    815
    816
    817
    818
    819
    820
    821
    822
    823
    824
    825
    826
    827
    828
    829
    830
    831
    832
    833
    834
    835
    836
    837
    838
    839
    840
    841
    842
    843
    844
    845
    846
    847
    848
    849
    850
    851
    852
    853
    854
    855
    856
    857
    858
    859
    860
    861
    862
    863
    864
    865
    866
    867
    868
    869
    870
    871
    872
    873
    874
    875
    876
    877
    878
    879
    880
    881
    882
    883
    884
    885
    886
    887
    888
    889
    890
    891
    892
    893
    894
    895
    896
    897
    898
    899
    900
    901
    902
    903
    904
    905
    906
    907
    908
    909
    910
    911
    912
    913
    914
    915
    916
    917
    918
    919
    920
    921
    922
    923
    924
    925
    926
    927
    928
    929
    930
    931
    932
    933
    934
    935
    936
    937
    938
    939
    940
    941
    942
    943
    944
    945
    946
    947
    948
    949
    950
    951
    952
    953
    954
    955
    956
    957
    958
    959
    960
    961
    962
    963
    964
    965
    966
    967
    968
    969
    970
    971
    972
    973
    974
    975
    976
    977
    978
    979
    980
    981
    982
    983
    984
    985
    986
    987
    988
    989
    990
    991
    992
    993
    994
    995
    996
    997
    998
    999
    1000
    1001
    1002
    1003
    1004
    1005
    1006
    1007
    1008
    1009
    1010
    1011
    1012
    1013
    1014
    1015
    1016
    1017
    1018
    1019
    1020
    1021
    1022
    1023
    1024
    1025
    1026
    1027
    1028
    1029
    1030
    1031
    1032
    1033
    1034
    1035
    1036
    1037
    1038
    1039
    1040
    1041
    1042
    1043
    1044
    1045
    1046
    1047
    1048
    1049
    1050
    1051
    1052
    1053
    1054
    1055
    1056
    1057
    1058
    1059
    1060
    1061
    1062
    1063
    1064
    1065
    1066
    1067
    1068
    1069
    1070
    1071
    1072
    1073
    1074
    1075
    1076
    1077
    1078
    1079
    1080
    1081
    1082
    1083
    1084
    1085
    1086
    1087
    1088
    1089
    1090
    1091
    1092
    1093
    1094
    1095
    1096
    1097
    1098
    1099
    1100
    1101
    1102
    1103
    1104
    1105
    1106
    1107
    1108
    1109
    1110
    1111
    1112
    1113
    1114
    1115
    1116
    1117
    1118
    1119
    1120
    1121
    1122
    1123
    1124
    1125
    1126
    1127
    1128
    1129
    1130
    1131
    1132
    1133
    1134
    1135
    1136
    1137
    1138
    1139
    1140
    1141
    1142
    1143
    1144
    1145
    1146
    1147
    1148
    1149
    1150
    1151
    1152
    1153
    1154
    1155
    1156
    1157
    1158
    1159
    1160
    1161
    1162
    1163
    1164
    1165
    1166
    1167
    1168
    1169
    1170
    1171
    1172
    1173
    1174
    1175
    1176
    1177
    1178
    1179
    1180
    1181
    1182
    1183
    1184
    1185
    1186
    1187
    1188
    1189
    1190
    1191
    1192
    1193
    1194
    1195
    1196
    1197
    1198
    1199
    1200
    1201
    1202
    1203
    1204
    1205
    1206
    1207
    1208
    1209
    1210
    1211
    1212
    1213
    1214
    1215
    1216
    1217
    1218
    1219
    1220
    1221
    1222
    1223
    1224
    1225
    1226
    1227
    1228
    1229
    1230
    1231
    1232
    1233
    1234
    1235
    1236
    1237
    1238
    1239
    1240
    1241
    1242
    1243
    1244
    1245
    1246
    1247
    1248
    1249
    1250
    1251
    1252
    1253
    1254
    1255
    1256
    1257
    1258
    1259
    1260
    1261
    1262
    1263
    1264
    1265
    1266
    1267
    1268
    1269
    1270
    1271
    1272
    1273
    1274
    1275
    1276
    1277
    1278
    1279
    1280
    1281
    1282
    1283
    1284
    1285
    1286
    1287
    1288
    1289
    1290
    1291
    1292
    1293
    1294
    1295
    1296
    1297
    1298
    1299
    1300
    1301
    1302
    1303
    1304
    1305
    1306
    1307
    1308
    1309
    1310
    1311
    1312
    1313
    1314
    1315
    1316
    1317
    1318
    1319
    1320
    1321
    1322
    1323
    1324
    1325
    1326
    1327
    1328
    1329
    1330
    1331
    1332
    1333
    1334
    1335
    1336
    1337
    1338
    1339
    1340
    1341
    1342
    1343
    1344
    1345
    1346
    1347
    1348
    1349
    1350
    1351
    1352
    1353
    1354
    1355
    1356
    1357
    1358
    1359
    1360
    1361
    1362
    1363
    1364
    1365
    1366
    1367
    1368
    1369
    1370
    1371
    1372
    1373
    1374
    1375
    1376
    1377
    1378
    1379
    1380
    1381
    1382
    1383
    1384
    1385
    1386
    1387
    1388
    1389
    1390
    1391
    1392
    1393
    1394
    1395
    1396
    1397
    1398
    1399
    1400
    1401
    1402
    1403
    1404
    1405
    1406
    1407
    1408
    1409
    1410
    1411
    1412
    1413
    1414
    1415
    1416
    1417
    1418
    1419
    1420
    1421
    1422
    1423
    1424
    1425
    1426
    1427
    1428
    1429
    1430
    1431
    1432
    1433
    1434
    1435
    1436
    1437
    1438
    1439
    1440
    1441
    1442
    1443
    1444
    1445
    1446
    1447
    1448
    1449
    1450
    1451
    1452
    1453
    1454
    1455
    1456
    1457
    1458
    1459
    1460
    1461
    1462
    1463
    1464
    1465
    1466
    1467
    1468
    1469
    1470
    1471
    1472
    1473
    1474
    1475
    1476
    1477
    1478
    1479
    1480
    1481
    1482
    1483
    1484
    1485
    1486
    1487
    1488
    1489
    1490
    1491
    1492
    1493
    1494
    1495
    1496
    1497
    1498
    1499
    1500
    1501
    1502
    1503
    1504
    1505
    1506
    1507
    1508
    1509
    1510
    1511
    1512
    1513
    1514
    1515
    1516
    1517
    1518
    1519
    1520
    1521
    1522
    1523
    1524
    1525
    1526
    1527
    1528
    1529
    1530
    1531
    1532
    1533
    1534
    1535
    1536
    1537
    1538
    1539
    1540
    1541
    1542
    1543
    1544
    1545
    1546
    1547
    1548
    1549
    1550
    1551
    1552
    1553
    1554
    1555
    1556
    1557
    1558
    1559
    1560
    1561
    1562
    1563
    1564
    1565
    1566
    1567
    1568
    1569
    1570
    1571
    1572
    1573
    1574
    1575
    1576
    1577
    1578
    1579
    1580
    1581
    1582
    1583
    1584
    1585
    1586
    1587
    1588
    1589
    1590
    1591
    1592
    1593
    1594
    1595
    1596
    1597
    1598
    1599
    1600
    1601
    1602
    1603
    1604
    1605
    1606
    1607
    1608
    1609
    1610
    1611
    1612
    1613
    1614
    1615
    1616
    1617
    1618
    1619
    1620
    1621
    1622
    1623
    1624
    1625
    1626
    1627
    1628
    1629
    1630
    1631
    1632
    1633
    1634
    1635
    1636
    1637
    1638
    1639
    1640
    1641
    1642
    1643
    1644
    1645
    1646
    1647
    1648
    1649
    1650
    1651
    1652
    1653
    1654
    1655
    1656
    1657
    1658
    1659
    1660
    1661
    1662
    1663
    1664
    1665
    1666
    1667
    1668
    1669
    1670
    1671
    1672
    1673
    1674
    1675
    1676
    1677
    1678
    1679
    1680
    1681
    1682
    1683
    1684
    1685
    1686
    1687
    1688
    1689
    1690
    1691
    1692
    1693
    1694
    1695
    1696
    1697
    1698
    1699
    1700
    1701
    1702
    1703
    1704
    1705
    1706
    1707
    1708
    1709
    1710
    1711
    1712
    1713
    1714
    1715
    1716
    1717
    1718
    1719
    1720
    1721
    1722
    1723
    1724
    1725
    1726
    1727
    1728
    1729
    1730
    1731
    1732
    1733
    1734
    1735
    1736
    1737
    1738
    1739
    1740
    1741
    1742
    1743
    1744
    1745
    1746
    1747
    1748
    1749
    1750
    1751
    1752
    1753
    1754
    1755
    1756
    1757
    1758
    1759
    1760
    1761
    1762
    1763
    1764
    1765
    1766
    1767
    1768
    1769
    1770
    1771
    1772
    1773
    1774
    1775
    1776
    1777
    1778
    1779
    1780
    1781
    1782
    1783
    1784
    1785
    1786
    1787
    1788
    1789
    1790
    1791
    1792
    1793
    1794
    1795
    1796
    1797
    1798
    1799
    1800
    1801
    1802
    1803
    1804
    1805
    1806
    1807
    1808
    1809
    1810
    1811
    1812
    1813
    1814
    1815
    1816
    1817
    1818
    1819
    1820
    1821
    1822
    1823
    1824
    1825
    1826
    1827
    1828
    1829
    1830
    1831
    1832
    1833
    1834
    1835
    1836
    1837
    1838
    1839
    1840
    1841
    1842
    1843
    1844
    1845
    1846
    1847
    1848
    1849
    1850
    1851
    1852
    1853
    1854
    1855
    1856
    1857
    1858
    1859
    1860
    1861
    1862
    1863
    1864
    1865
    1866
    1867
    1868
    1869
    1870
    1871
    1872
    1873
    1874
    1875
    1876
    1877
    1878
    1879
    1880
    1881
    1882
    1883
    1884
    1885
    1886
    1887
    1888
    1889
    1890
    1891
    1892
    1893
    1894
    1895
    1896
    1897
    1898
    1899
    1900
    1901
    1902
    1903
    1904
    1905
    1906
    1907
    1908
    1909
    1910
    1911
    1
```



6. Contoh Ubah Data Menjadi Horizontal (Jika Diperlukan)

```

File Edit Selection View Go Run ... ← → ⌘ Python
PROJECT OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Python\ & c:\Python\env\Scripts\python.exe c:/Python/py2_mod17.py
[1m[2m[3m[4m[5m[6m[7m[8m[9m[10m[11m[12m[13m[14m[15m[16m[17m[18m[19m[20m[21m[22m[23m[24m[25m[26m[27m[28m[29m[30m[31m[32m[33m[34m[35m[36m[37m[38m[39m[40m[41m[42m[43m[44m[45m[46m[47m[48m[49m[50m[51m[52m[53m[54m[55m[56m[57m[58m[59m[60m[61m[62m[63m[64m[65m[66m[67m[68m[69m[70m[71m[72m[73m[74m[75m[76m[77m[78m[79m[80m[81m[82m[83m[84m[85m[86m[87m[88m[89m[90m[91m[92m[93m[94m[95m[96m[97m[98m[99m[100m[101m[102m[103m[104m[105m[106m[107m[108m[109m[110m[111m[112m[113m[114m[115m[116m[117m[118m[119m[120m[121m[122m[123m[124m[125m[126m[127m[128m[129m[130m[131m[132m[133m[134m[135m[136m[137m[138m[139m[140m[141m[142m[143m[144m[145m[146m[147m[148m[149m[150m[151m[152m[153m[154m[155m[156m[157m[158m[159m[160m[161m[162m[163m[164m[165m[166m[167m[168m[169m[170m[171m[172m[173m[174m[175m[176m[177m[178m[179m[180m[181m[182m[183m[184m[185m[186m[187m[188m[189m[190m[191m[192m[193m[194m[195m[196m[197m[198m[199m[200m[201m[202m[203m[204m[205m[206m[207m[208m[209m[210m[211m[212m[213m[214m[215m[216m[217m[218m[219m[220m[221m[222m[223m[224m[225m[226m[227m[228m[229m[230m[231m[232m[233m[234m[235m[236m[237m[238m[239m[240m[241m[242m[243m[244m[245m[246m[247m[248m[249m[250m[251m[252m[253m[254m[255m[256m[257m[258m[259m[260m[261m[262m[263m[264m[265m[266m[267m[268m[269m[270m[271m[272m[273m[274m[275m[276m[277m[278m[279m[280m[281m[282m[283m[284m[285m[286m[287m[288m[289m[290m[291m[292m[293m[294m[295m[296m[297m[298m[299m[300m[301m[302m[303m[304m[305m[306m[307m[308m[309m[310m[311m[312m[313m[314m[315m[316m[317m[318m[319m[320m[321m[322m[323m[324m[325m[326m[327m[328m[329m[330m[331m[332m[333m[334m[335m[336m[337m[338m[339m[340m[341m[342m[343m[344m[345m[346m[347m[348m[349m[350m[351m[352m[353m[354m[355m[356m[357m[358m[359m[360m[361m[362m[363m[364m[365m[366m[367m[368m[369m[370m[371m[372m[373m[374m[375m[376m[377m[378m[379m[380m[381m[382m[383m[384m[385m[386m[387m[388m[389m[390m[391m[392m[393m[394m[395m[396m[397m[398m[399m[400m[401m[402m[403m[404m[405m[406m[407m[408m[409m[410m[411m[412m[413m[414m[415m[416m[417m[418m[419m[420m[421m[422m[423m[424m[425m[426m[427m[428m[429m[430m[431m[432m[433m[434m[435m[436m[437m[438m[439m[440m[441m[442m[443m[444m[445m[446m[447m[448m[449m[450m[451m[452m[453m[454m[455m[456m[457m[458m[459m[460m[461m[462m[463m[464m[465m[466m[467m[468m[469m[470m[471m[472m[473m[474m[475m[476m[477m[478m[479m[480m[481m[482m[483m[484m[485m[486m[487m[488m[489m[490m[491m[492m[493m[494m[495m[496m[497m[498m[499m[500m[501m[502m[503m[504m[505m[506m[507m[508m[509m[510m[511m[512m[513m[514m[515m[516m[517m[518m[519m[520m[521m[522m[523m[524m[525m[526m[527m[528m[529m[530m[531m[532m[533m[534m[535m[536m[537m[538m[539m[540m[541m[542m[543m[544m[545m[546m[547m[548m[549m[550m[551m[552m[553m[554m[555m[556m[557m[558m[559m[559m[560m[561m[562m[563m[564m[565m[566m[567m[568m[569m[569m[570m[571m[572m[573m[574m[575m[576m[577m[578m[579m[579m[580m[581m[582m[583m[584m[585m[586m[587m[588m[589m[589m[590m[591m[592m[593m[594m[595m[596m[597m[598m[599m[599m[600m[601m[602m[603m[604m[605m[606m[607m[608m[609m[609m[610m[611m[612m[613m[614m[615m[616m[617m[618m[619m[619m[620m[621m[622m[623m[624m[625m[626m[627m[628m[629m[629m[630m[631m[632m[633m[634m[635m[636m[637m[638m[639m[639m[640m[641m[642m[643m[644m[645m[646m[647m[648m[649m[649m[650m[651m[652m[653m[654m[655m[656m[657m[658m[659m[659m[660m[661m[662m[663m[664m[665m[666m[667m[668m[669m[669m[670m[671m[672m[673m[674m[675m[676m[677m[678m[679m[679m[680m[681m[682m[683m[684m[685m[686m[687m[688m[689m[689m[690m[691m[692m[693m[694m[695m[696m[697m[698m[699m[699m[700m[701m[702m[703m[704m[705m[706m[707m[708m[709m[709m[710m[711m[712m[713m[714m[715m[716m[717m[718m[719m[719m[720m[721m[722m[723m[724m[725m[726m[727m[728m[729m[729m[730m[731m[732m[733m[734m[735m[736m[737m[738m[739m[739m[740m[741m[742m[743m[744m[745m[746m[747m[748m[749m[749m[750m[751m[752m[753m[754m[755m[756m[757m[758m[759m[759m[760m[761m[762m[763m[764m[765m[766m[767m[768m[769m[769m[770m[771m[772m[773m[774m[775m[776m[777m[778m[779m[779m[780m[781m[782m[783m[784m[785m[786m[787m[788m[789m[789m[790m[791m[792m[793m[794m[795m[796m[797m[798m[799m[799m[800m[801m[802m[803m[804m[805m[806m[807m[808m[809m[809m[810m[811m[812m[813m[814m[815m[816m[817m[818m[819m[819m[820m[821m[822m[823m[824m[825m[826m[827m[828m[829m[829m[830m[831m[832m[833m[834m[835m[836m[837m[838m[839m[839m[840m[841m[842m[843m[844m[845m[846m[847m[848m[849m[849m[850m[851m[852m[853m[854m[855m[856m[857m[858m[859m[859m[860m[861m[862m[863m[864m[865m[866m[867m[868m[869m[869m[870m[871m[872m[873m[874m[875m[876m[877m[878m[879m[879m[880m[881m[882m[883m[884m[885m[886m[887m[888m[889m[889m[890m[891m[892m[893m[894m[895m[896m[897m[898m[899m[899m[900m[901m[902m[903m[904m[905m[906m[907m[908m[909m[909m[910m[911m[912m[913m[914m[915m[916m[917m[918m[919m[919m[920m[921m[922m[923m[924m[925m[926m[927m[928m[929m[929m[930m[931m[932m[933m[934m[935m[936m[937m[938m[939m[939m[940m[941m[942m[943m[944m[945m[946m[947m[948m[949m[949m[950m[951m[952m[953m[954m[955m[956m[957m[958m[959m[959m[960m[961m[962m[963m[964m[965m[966m[967m[968m[969m[969m[970m[971m[972m[973m[974m[975m[976m[977m[978m[979m[979m[980m[981m[982m[983m[984m[985m[986m[987m[988m[989m[989m[990m[991m[992m[993m[994m[995m[996m[997m[998m[999m[999m[1000m[1001m[1002m[1003m[1004m[1005m[1006m[1007m[1008m[1009m[1009m[1010m[1011m[1012m[1013m[1014m[1015m[1016m[1017m[1018m[1019m[1019m[1020m[1021m[1022m[1023m[1024m[1025m[1026m[1027m[1028m[1029m[1029m[1030m[1031m[1032m[1033m[1034m[1035m[1036m[1037m[1038m[1039m[1039m[1040m[1041m[1042m[1043m[1044m[1045m[1046m[1047m[1048m[1049m[1049m[1050m[1051m[1052m[1053m[1054m[1055m[1056m[1057m[1058m[1059m[1059m[1060m[1061m[1062m[1063m[1064m[1065m[1066m[1067m[1068m[1069m[1069m[1070m[1071m[1072m[1073m[1074m[1075m[1076m[1077m[1078m[1079m[1079m[1080m[1081m[1082m[1083m[1084m[1085m[1086m[1087m[1088m[1089m[1089m[1090m[1091m[1092m[1093m[1094m[1095m[1096m[1097m[1098m[1098m[1099m[1099m[1100m[1101m[1102m[1103m[1104m[1105m[1106m[1107m[1108m[1109m[1109m[1110m[1111m[1112m[1113m[1114m[1115m[1116m[1117m[1118m[1119m[1119m[1120m[1121m[1122m[1123m[1124m[1125m[1126m[1127m[1128m[1129m[1129m[1130m[1131m[1132m[1133m[1134m[1135m[1136m[1137m[1138m[1139m[1139m[1140m[1141m[1142m[1143m[1144m[1145m[1146m[1147m[1148m[1149m[1149m[1150m[1151m[1152m[1153m[1154m[1155m[1156m[1157m[1158m[1159m[1159m[1160m[1161m[1162m[1163m[1164m[1165m[1166m[1167m[1168m[1169m[1169m[1170m[1171m[1172m[1173m[1174m[1175m[1176m[1177m[1178m[1179m[1179m[1180m[1181m[1182m[1183m[1184m[1185m[1186m[1187m[1188m[1189m[1189m[1190m[1191m[1192m[1193m[1194m[1195m[1196m[1197m[1198m[1198m[1199m[1199m[1200m[1201m[1202m[1203m[1204m[1205m[1206m[1207m[1208m[1209m[1209m[1210m[1211m[1212m[1213m[1214m[1215m[1216m[1217m[1218m[1219m[1219m[1220m[1221m[1222m[1223m[1224m[1225m[1226m[1227m[1228m[1229m[1229m[1230m[1231m[1232m[1233m[1234m[1235m[1236m[1237m[1238m[1239m[1239m[1240m[1241m[1242m[1243m[1244m[1245m[1246m[1247m[1248m[1249m[1249m[1250m[1251m[1252m[1253m[1254m[1255m[1256m[1257m[1258m[1259m[1259m[1260m[1261m[1262m[1263m[1264m[1265m[1266m[1267m[1268m[1269m[1269m[1270m[1271m[1272m[1273m[1274m[1275m[1276m[1277m[1278m[1279m[1279m[1280m[1281m[1282m[1283m[1284m[1285m[1286m[1287m[1288m[1289m[1289m[1290m[1291m[1292m[1293m[1294m[1295m[1296m[1297m[1298m[1298m[1299m[1299m[1300m[1301m[1302m[1303m[1304m[1305m[1306m[1307m[1308m[1309m[1309m[1310m[1311m[1312m[1313m[1314m[1315m[1316m[1317m[1318m[1319m[1319m[1320m[1321m[1322m[1323m[1324m[1325m[1326m[1327m[1328m[1329m[1329m[1330m[1331m[1332m[1333m[1334m[1335m[1336m[1337m[1338m[1339m[1339m[1340m[1341m[1342m[1343m[1344m[1345m[1346m[1347m[1348m[1349m[1349m[1350m[1351m[1352m[1353m[1354m[1355m[1356m[1357m[1358m[1359m[1359m[1360m[1361m[1362m[1363m[1364m[1365m[1366m[1367m[1368m[1369m[1369m[1370m[1371m[1372m[1373m[1374m[1375m[1376m[1377m[1378m[1379m[1379m[1380m[1381m[1382m[1383m[1384m[1385m[1386m[1387m[1388m[1389m[1389m[1390m[1391m[1392m[1393m[1394m[1395m[1396m[1397m[1398m[1398m[1399m[1399m[1400m[1401m[1402m[1403m[1404m[1405m[1406m[1407m[1408m[1409m[1409m[1410m[1411m[1412m[1413m[1414m[1415m[1416m[1417m[1418m[1419m[1419m[1420m[1421m[1422m[1423m[1424m[1425m[1426m[1427m[1428m[1429m[1429m[1430m[1431m[1432m[1433m[1434m[1435m[1436m[1437m[1438m[1439m[1439m[1440m[1441m[1442m[1443m[1444m[1445m[1446m[1447m[1448m[1449m[1449m[1450m[1451m[1452m[1453m[1454m[1455m[1456m[1457m[1458m[1459m[1459m[1460m[1461m[1462m[1463m[1464m[1465m[1466m[1467m[1468m[1469m[1469m[1470m[1471m[1472m[1473m[1474m[1475m[1476m[1477m[1478m[1479m[1479m[1480m[1481m[1482m[1483m[1484m[1485m[1486m[1487m[1488m[1489m[1489m[1490m[1491m[1492m[1493m[1494m[1495m[1496m[1497m[1498m[1498m[1499m[1499m[1500m[1501m[1502m[1503m[1504m[1505m[1506m[1507m[1508m[1509m[1509m[1510m[1511m[1512m[1513m[1514m[1515m[1516m[1517m[1518m[1519m[1519m[1520m[1521m[1522m[1523m[1524m[1525m[1526m[1527m[1528m[1529m[1529m[1530m[1531m[1532m[1533m[1534m[1535m[1536m[1537m[1538m[1539m[1539m[1540m[1541m[1542m[1543m[1544m[1545m[1546m[1547m[1548m[1549m[1549m[1550m[1551m[1552m[1553m[1554m[1555m[1556m[1557m[1558m[1559m[1559m[1560m[1561m[1562m[1563m[1564m[1565m[1566m[1567m[1568m[1569m[1569m[1570m[1571m[1572m[1573m[1574m[1575m[1576m[1577m[1578m[1579m[1579m[1580m[1581m[1582m[1583m[1584m[1585m[1586m[1587m[1588m[1589m[1589m[1590m[1591m[1592m[1593m[1594m[1595m[1596m[1597m[1598m[1598m[1599m[1599m[1600m[1601m[1602m[1603m[1604m[1605m[1606m[1607m[1608m[1609m[1609m[1610m[1611m[1612m[1613m[1614m[1615m[1616m[1617m[1618m[1619m[1619m[1620m[1621m[1622m[1623m[1624m[1625m[1626m[1627m[1628m[1629m[1629m[1630m[1631m[1632m[1633m[1634m[1635m[1636m[1637m[1638m[1639m[1639m[1640m[1641m[1642m[1643m[1644m[1645m[1646m[1647m[1648m[1649m[1649m[1650m[1651m[1652m[1653m[1654m[1655m[1656m[1657m[1658m[1659m[1659m[1660m[1661m[1662m[1663m[1664m[1665m[1666m[1667m[1668m[1669m[1669m[1670m[1671m[1672m[1673m[1674m[1675m[1676m[1677m[1678m[1679m[1679m[1680m[1681m[1682m[1683m[1684m[1685m[1686m[1687m[1688m[1689m[1689m[1690m[1691m[1692m[1693m[1694m[1695m[1696m[1697m[1698m[1698m[1699m[1699m[1700m[1701m[1702m[1703m[1704m[1705m[1706m[1707m[1708m[1709m[1709m[1710m[1711m[1712m[1713m[1714m[1715m[1716m[1717m[1718m[1719m[1719m[1720m[1721m[1722m[1723m[1724m[1725m[1726m[1727m[1728m[1729m[1729m[1730m[1731m[1732m[1733m[1734m[1735m[1736m[1737m[1738m[1739m[1739m[1740m[1741m[1742m[1743m[1744m[1745m[1746m[1747m[1748m[1749m[1749m[1750m[1751m[1752m[1753m[1754m[1755m[1756m[1757m[1758m[1759m[1759m[1760m[1761m[1762m[1763m[1764m[1765m[1766m[1767m[1768m[1769m[1769m[1770m[1771m[1772m[1773m[1774m[1775m[1776m[1777m[1778m[1779m[1779m[1780m[1781m[1782m[1783m[1784m[1785m[1786m[1787m[1788m[1789m[1789m[1790m[1791m[1792m[1793m[1794m[1795m[1796m[1797m[1798m[1798m[1799m[1799m[1800m[1801m[1802m[1803m[1804m[1805m[1806m[1807m[1808m[1809m[1809m[1810m[1811m[1812m[1813m[1814m[1815m[1816m[1817m[1818m[1819m[1819m[1820m[1821m[1822m[1823m[1824m[1825m[1826m[1827m[1828m[1829m[1829m[1830m[1831m[1832m[1833m[1834m[1835m[1836m[1837m[1838m[1839m[1839m[1840m[1841m[1842m[1843m[1844m[1845m[1846m[1847m[1848m[1849m[1849m[1850m[1851m[185
```



C. Plot Pie Chart

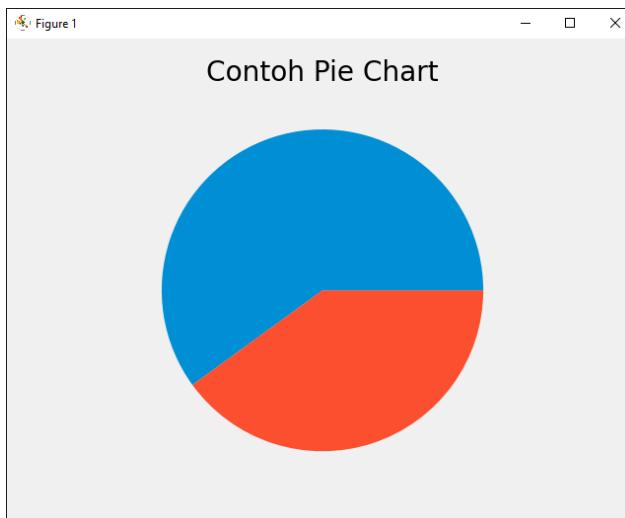
Berikut cara menampilkan data menjadi Pie Chart.

1. Contoh Slices

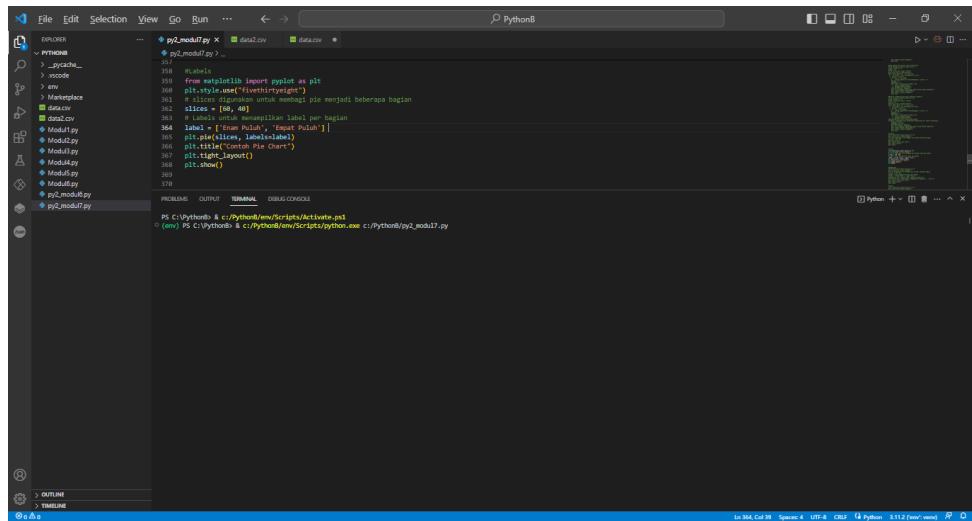
```

File Edit Selection View Go Run ... ← → ⌘ PythonB6
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB6\> & PythonEnv\Scripts\Activate.ps1
PS C:\PythonB6> c:\PythonB6\env\Scripts\python.exe c:/PythonB6/py2_modul7.py
346
347 # Slices
348 # Slices
349 plt.style.use('fivethirtyeight')
350 # slices merupakan untuk membagi pie menjadi beberapa bagian
351 slices = [60, 40]
352 plt.pie(slices)
353 plt.title('Contoh Pie Chart')
354 plt.axis('equal')
355 plt.show()
356

```

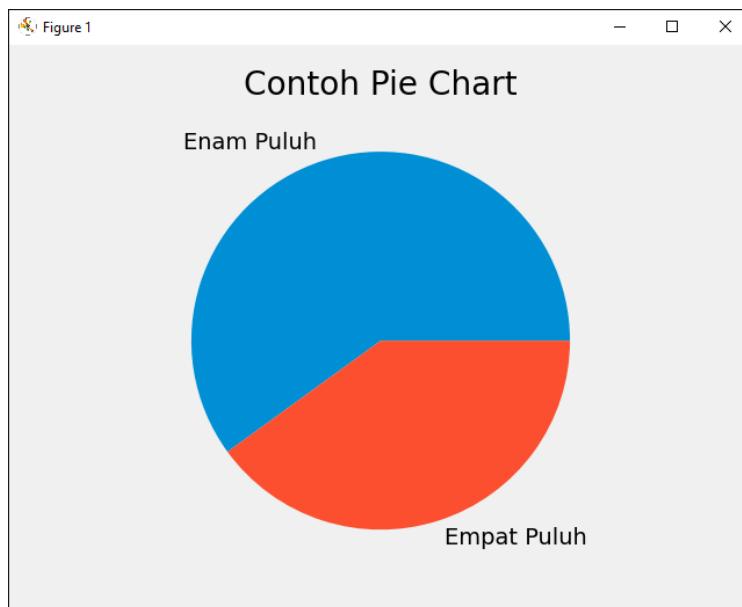


2. Contoh Labels

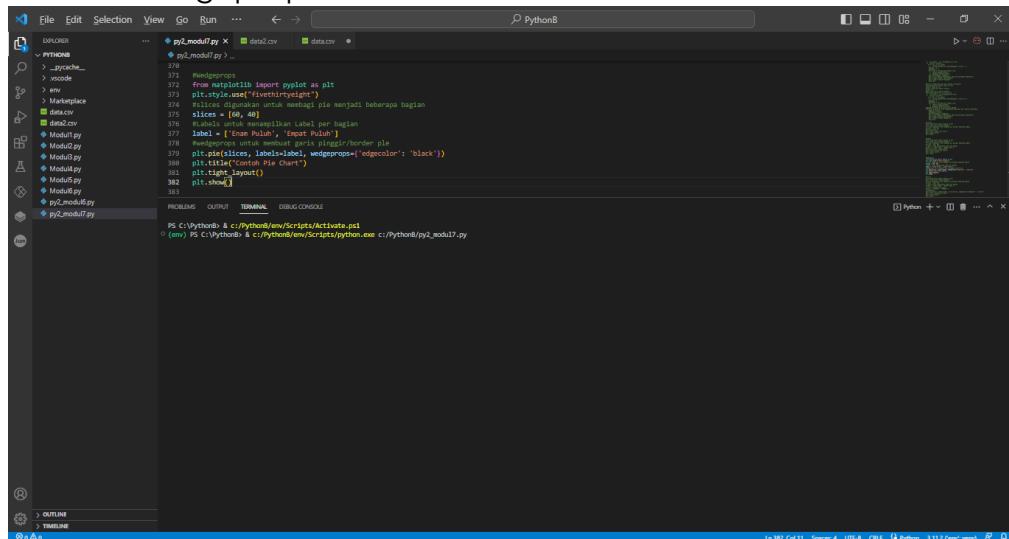


A screenshot of a Python code editor (PyCharm) showing a script named `py2_modul17.py`. The code uses the `matplotlib.pyplot` module to create a pie chart titled "Contoh Pie Chart". The chart has two segments: one blue segment labeled "Enam Puluh" and one orange segment labeled "Empat Puluh". The code includes imports for `matplotlib.pyplot`, defines labels for the slices, and sets the title and layout.

```
py2_modul17.py
...
358 import matplotlib.pyplot as plt
359 plt.style.use('fivethirtyeight')
360 # slices digunakan untuk membagi pie menjadi beberapa bagian
361 slices = [60, 40]
362 # labels untuk menampilkan label per bagian
363 labels = ['Enam Puluh', 'Empat Puluh']
364 plt.title("Contoh Pie Chart")
365 plt.pie(slices, labels=labels)
366 plt.tight_layout()
367 plt.show()
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
```

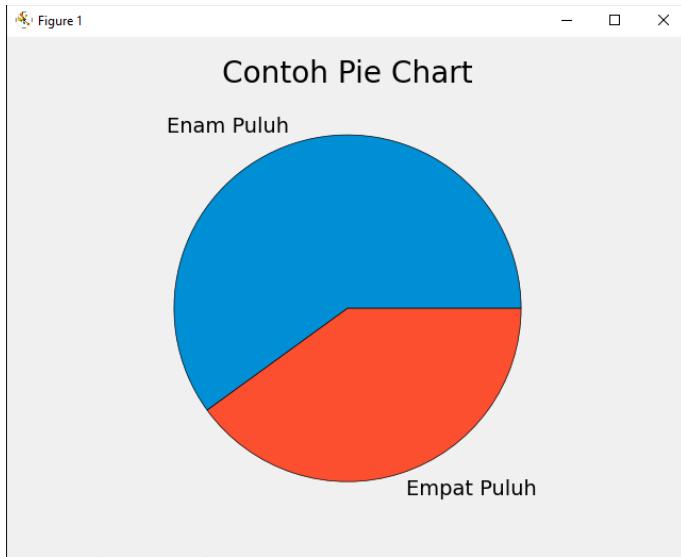


3. Contoh Wedgeprops



A screenshot of a Python code editor (PyCharm) showing a script named `py2_modul17.py`. The code uses the `matplotlib.pyplot` module to create a pie chart titled "Contoh Pie Chart". The chart has two segments: one blue segment and one orange segment. The blue segment is labeled "Enam Puluh" and the orange segment is labeled "Empat Puluh". The code includes imports for `matplotlib.pyplot`, defines labels for the slices, and sets the title and layout. It also uses the `wedgeprops` parameter to set the border color of the slices to black.

```
py2_modul17.py
...
371 #wedgeprops
372 from matplotlib import pyplot as plt
373 plt.style.use('fivethirtyeight')
374 #slices digunakan untuk membagi pie menjadi beberapa bagian
375 slices = [60, 40]
376 #labels untuk menampilkan label per bagian
377 labels = ['Enam Puluh', 'Empat Puluh']
378 #wedgeprops untuk membuat garis pinggir/border pie
379 plt.pie(slices, labels=labels, wedgeprops={'edgecolor': 'black'})
380 plt.title("Contoh Pie Chart")
381 plt.tight_layout()
382 plt.show()
383
```



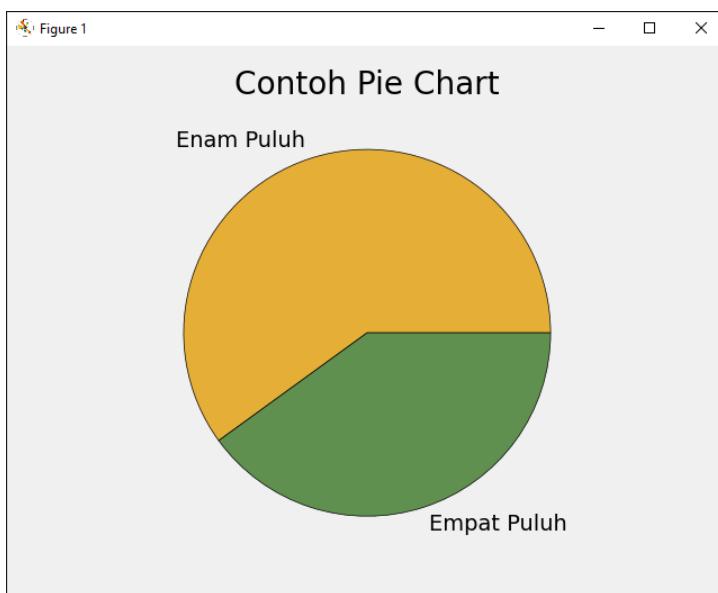
4. Contoh Colors

```

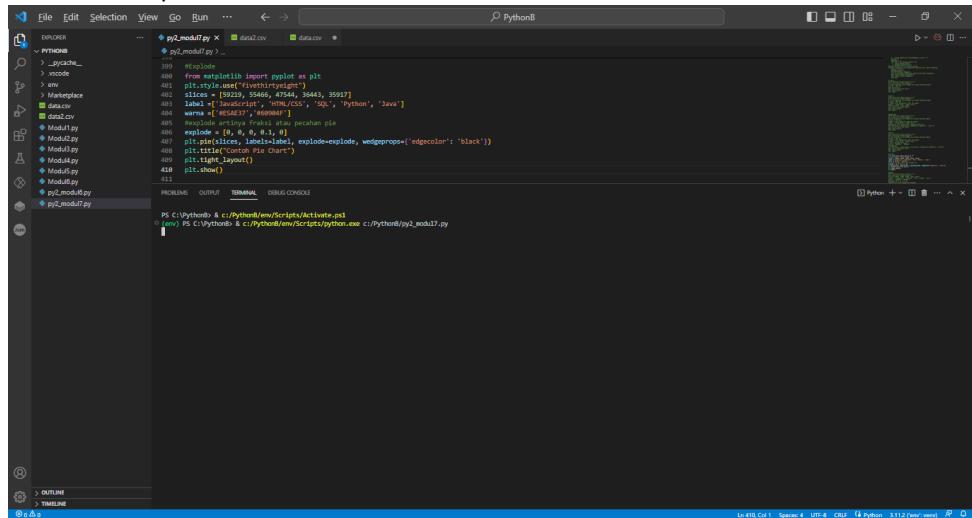
#PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Python36\& C:\Python\env\Scripts\activate.ps1
(env) PS C:\Python\env\Scripts\python.exe c:/Python/py_00017.py
  
```

```

#Colors
from matplotlib import pyplot as plt
plt.style.use("fivethirtyeight") #pita menjadi berwarna bagian
alias = {0, 40}
label = ["Enam Puluh", "Empat Puluh"]
warna = ["#E69138", "#FF0000"]
# Menghasilkan label dan warna
label, warna, alias
plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
  
```

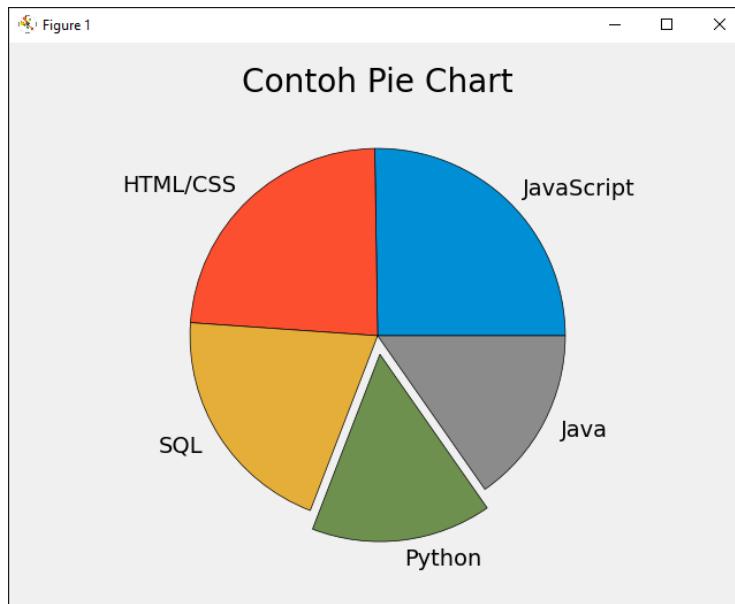


5. Contoh Explode

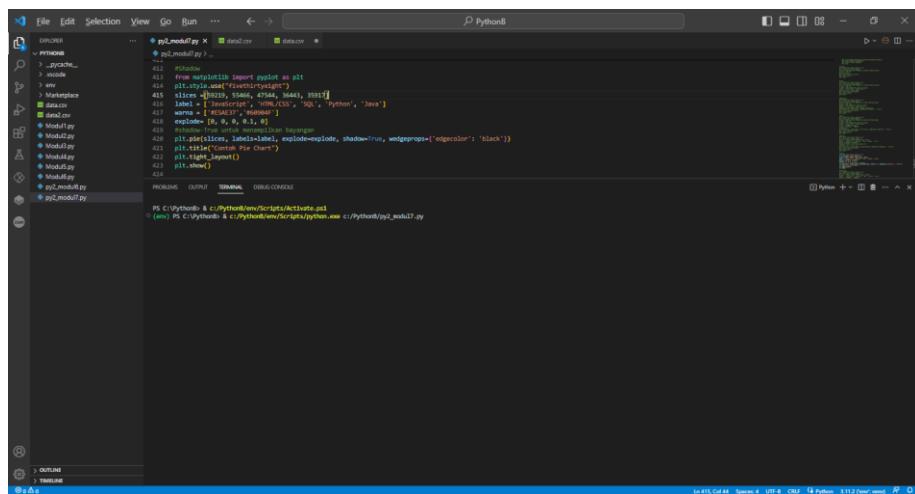


```
File Edit Selection View Go Run ... ← → ⌘ PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB & c:/PythonB/env/Scripts/activate.ps1
(env) PS C:\PythonB & c:/PythonB/env/Scripts/python.exe c:/PythonB/py2_modul17.py

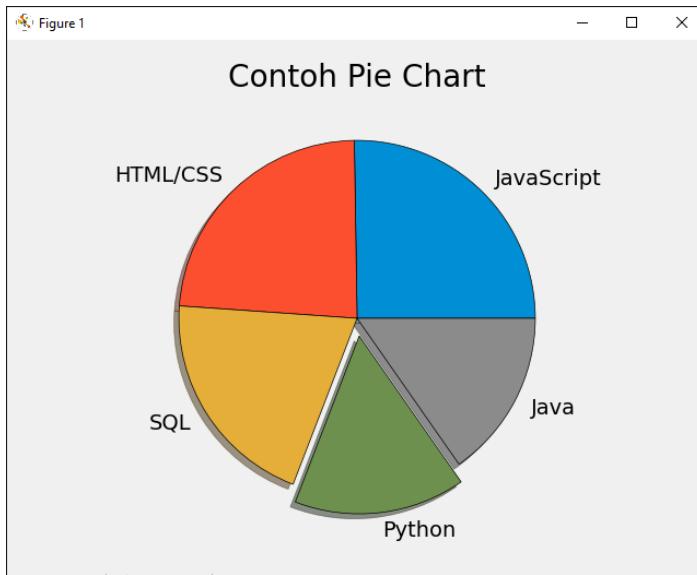
400 from matplotlib import pyplot as plt
401 plt.style.use('fivethirtyeight')
402 slices = [50219, 55665, 47544, 36442, 30917]
403 label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
404 colors = ['#E31A1C', '#F08080', '#FFDAB9', '#80E6AA', '#808080']
405 warn(['#E31A1C', '#F08080'])
406 explode = [0, 0, 0, 0.1, 0]
407 plt.pie(slices, labels=label, explode=explode, wedgeprops={'edgecolor': 'black'})
408 plt.title("Contoh Pie Chart")
409 plt.tight_layout()
410 plt.show()
411
```



6. Contoh Shadow



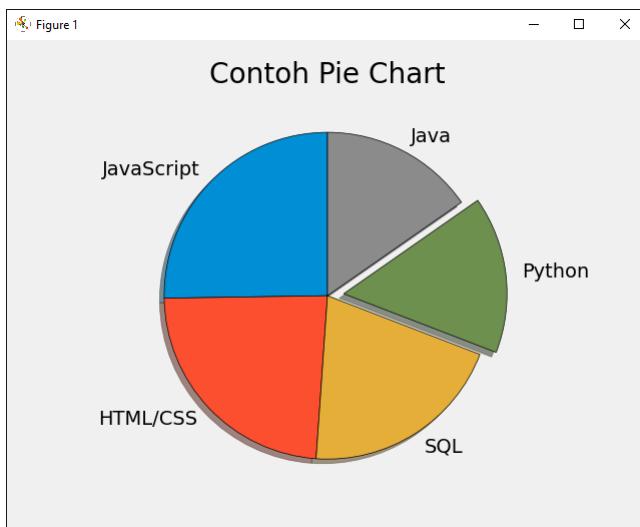
```
File Edit Selection View Go Run ... ← → ⌘ PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB & c:/PythonB/env/Scripts/activate.ps1
(env) PS C:\PythonB & c:/PythonB/env/Scripts/python.exe c:/PythonB/py2_modul27.py
```



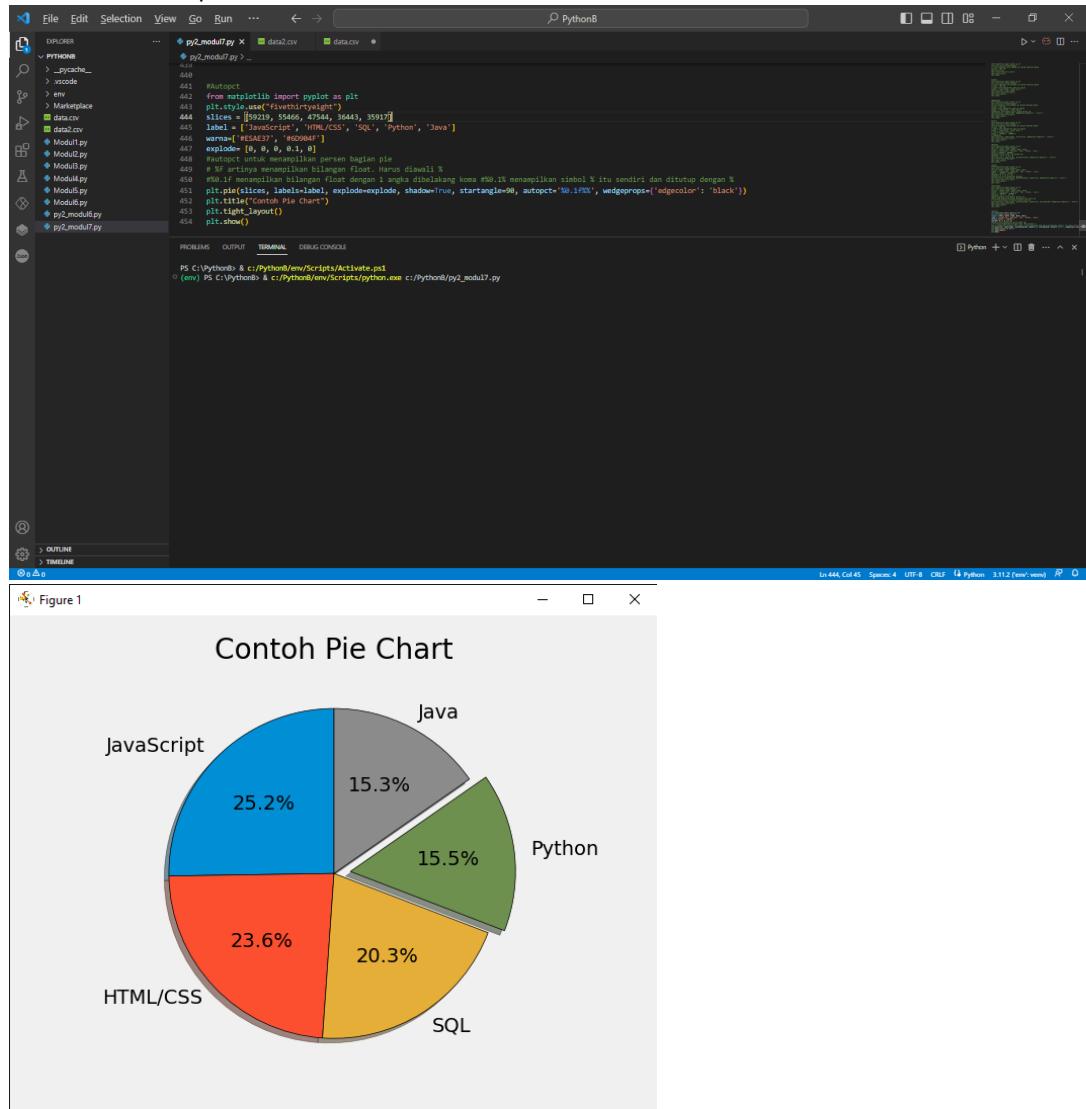
7. Contoh Startangle

The screenshot shows the PyCharm IDE interface with the following details:

- File Structure:** The left sidebar shows a project structure with files like `py2_modul7.py`, `py3_modul7.py`, `data1.csv`, and `data2.csv`.
- Code Editor:** The main editor contains Python code for generating a 3D bar chart. The code uses `matplotlib` to create a 3D bar plot with labels for Java, Python, SQL, and HTML/CSS.
- Terminal:** Below the editor, the terminal window shows the command `python c:/Python38/python.exe c:/Python38/py2_modul7.py` being run.
- Status Bar:** The bottom status bar indicates the current file is `py2_modul7.py`, with line 448, column 110, and other status information.



8. Contoh Autopct



MODUL 6

PYTHON (DATA SCIENCE)

VISUALISASI DATA - MATPLOTLIB IMPORT CSV

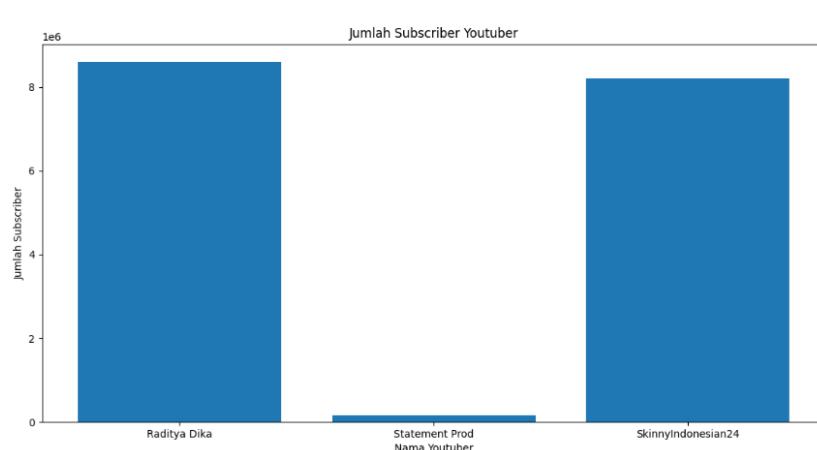
A. Membuat Plot

1. Membuat tipe data variabel sumbu x dan y menjadi list, ditandai dengan[] atau kurung besar. Misalnya b=[1,2,3], atau jika nilai String b=["Data","Baru","Dibuat"]
2. Mulai membuat plot, seperti pada contoh gambar dibawah terdapat plot atau grafik jumlah subscribe youtuber dengan nama Raditya Dika

The screenshot shows the Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar with a tree view of files and folders. The main area displays a Python script titled 'modul6_datascience.py'. The code uses the Matplotlib library to create a bar chart titled 'Jumlah Subscriber Youtuber' comparing three youtubers: Raditya Dika, Statement Prod Nama Youtuber, and SkinnyIndonesian24. The y-axis is labeled 'Jumlah Subscriber' and ranges from 0 to 1e6 (1,000,000). The x-axis lists the youtubers. The chart shows Raditya Dika has approximately 9,000,000 subscribers, while the other two have significantly fewer.

```
#Plot dasar dengan inisialisasi nilai pada variabel
import numpy as np
import matplotlib.pyplot as plt
#inisialisasi Nilai
name_youtuber=['Raditya Dika','Statement Prod', 'SkinnyIndonesian24']
jumlah_subscriber=[8600000, 156000, 6200000]
plt.bar(name_youtuber, jumlah_subscriber)
plt.title("Jumlah Subscriber Youtuber")
plt.xlabel("Nama Youtuber")
plt.ylabel("Jumlah Subscriber")
plt.show()
```

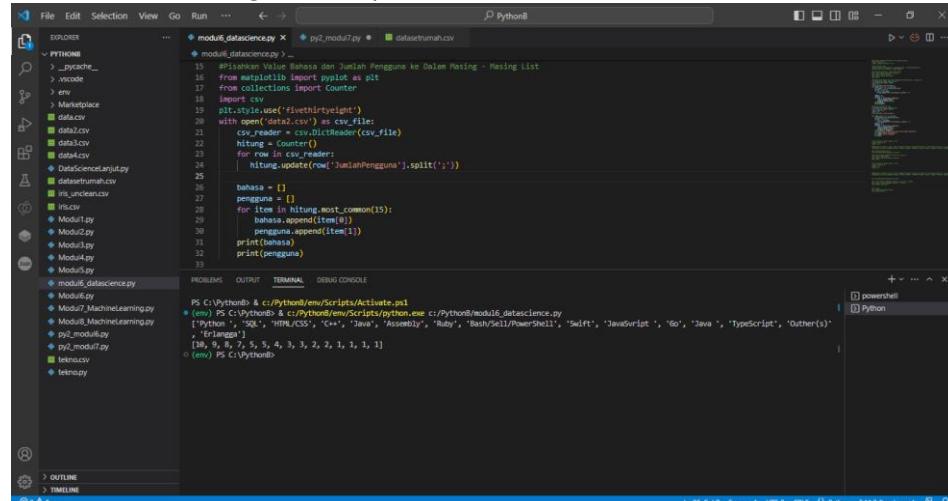
Output :



Import CSV

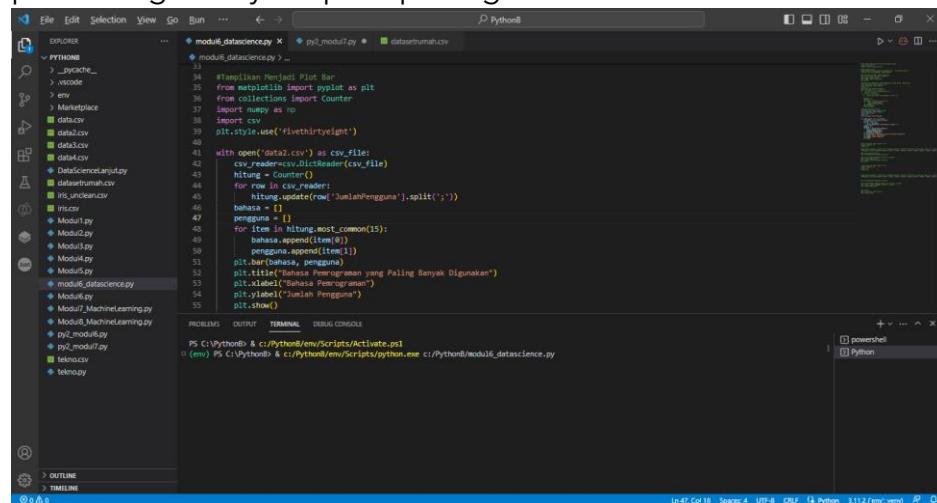
- a. Contoh import csv menghitung jumlah pengguna bahasa pemrograman dengan menggunakan data csv.
 - Data dihitung dengan kelas Counter seperti `hitung=Counter()`
 - Kemudian membuat variabel dengan tipe data list seperti `bahasa = []`
`pengguna = []`
 - Selanjutnya untuk memasukkan nilai kedalam variabel dengan menggunakan
`bahasa.append(item[0])`
`pengguna.append(item[1])`
 - Dan untuk menampilkan hasil dengan menggunakan `print(bahasa)`
`print(pengguna)`

Untuk Code Program Dapat Dilihat Pada Gambar



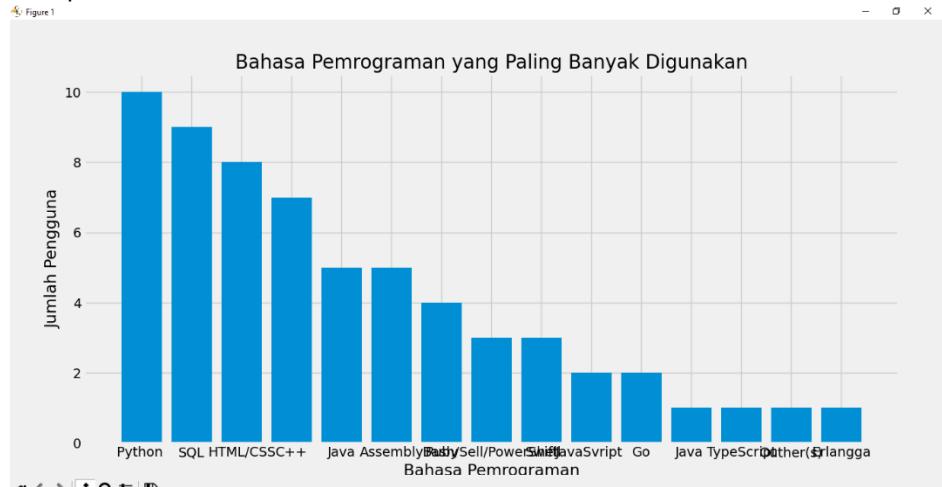
```
#import Counter from collections
#open dataset1.csv as csv_file
#read dataset1.csv into csv_reader
#initialize Counter object
#loop through csv_reader
#update Counter object with language and count
#initialize empty lists for language and user count
#loop through Counter object
#append language and user count to respective lists
#return lists
```

Jika nilai-nilai pada data sudah masuk kedalam variabel bahasa dan pengguna dengan tipe data list, maka selanjutnya kita akan membuat plot atau grafiknya seperti pada gambar dibawa :



```
#import Counter from collections
#open dataset1.csv as csv_file
#read dataset1.csv into csv_reader
#initialize Counter object
#loop through csv_reader
#update Counter object with language and count
#initialize empty lists for language and user count
#loop through Counter object
#append language and user count to respective lists
#return lists
#function to plot
#call function with bahasa and pengguna lists
#title of plot
#y-axis label
#x-axis label
#show plot
```

Output :



b. Contoh import csv data sportify

Data csv sportify yang sudah di import

The CSV file 'data3.csv' contains the following data:

Rank	Name Band	Tahun	Pendengar per Bulan
1	Nirvana	1993	6603189
2	Slipknot	1995	17418362
3	Linkin Park	1996	17418352
4	Avgent	1999	1880758
5	Trivium	2009	5410274
6	Bring Me the Horizon	2004	4428374
7	Green Day	1987	15692462
8	Muse	1994	8528417
9	As I Lay Dying	2008	396422
10	Trivium	1999	1880758
11	Limp Bizkit	1994	5910689
12	Skid Row	1986	2492894
13	Bon Jovi	1983	15581843

Pandas DataFrame untuk menampilkan data csv

```
import pandas as pd
pd.read_csv("data3.csv")
print(pd)
```

The output in the terminal shows the same data as the CSV file:

Rank	Name Band	Tahun	Pendengar per Bulan
1	Nirvana	1993	6603189
2	Slipknot	1995	17418362
3	Linkin Park	1996	17418352
4	Avgent	1999	1880758
5	Trivium	2009	5410274
6	Bring Me the Horizon	2004	4428374
7	Green Day	1987	15692462
8	Muse	1994	8528417
9	As I Lay Dying	2008	396422
10	Trivium	1999	1880758
11	Limp Bizkit	1994	5910689
12	Skid Row	1986	2492894
13	Bon Jovi	1983	15581843

Membuat nama band menjadi sebagai sumbu x dan pendengar per bulan menjadi sumbu y dengan variabel tipe data list, dengan code program sebagai berikut :

The screenshot shows the PythonB IDE interface. The left sidebar displays a file tree with several Python files and CSV files. The main area shows a code editor with a script named `modulis_datascience.py`. The terminal below shows command-line output related to the script's execution.

```
PS C:\PythonB> & c:/Python/env/Scripts/activate.ps1
PS C:\PythonB> & c:/Python/env/Scripts/python.exe c:/Python/modulis_datascience.py
['SLipknot', 'Linkin Park', 'Avenged Sevenfold', 'Bring Me The Horizon', 'Green Day', 'Muse', 'As I Lay Dying', 'Trivium', 'Limp Bizkit', 'Skid Row', 'Bon Jovi']
[6681389, 17418362, 5410274, 442874, 15692462, 8528417, 996422, 1880758, 5910669, 2492044, 15561843]
```

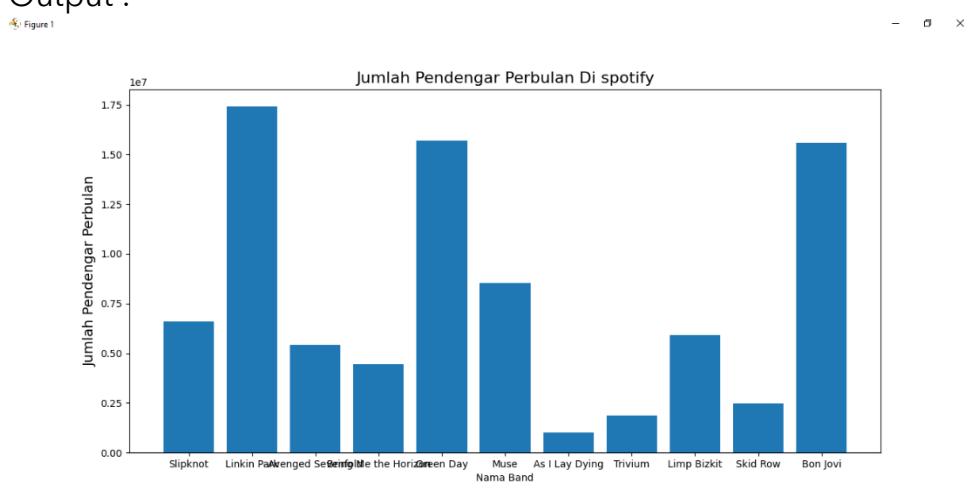
Membuat plot dengan menggunakan data pada csv dengan code program sebagai berikut :

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it lists several Python files and CSV files. The file `modul6_dataScience.py` is currently selected.
- Code Editor:** The main area displays the following Python code:

```
79
80     from matplotlib import pyplot as plt
81     import numpy as np
82     import os
83     import sys
84
85     Nama_Band=(Nama_Band)
86     Pendengar_per_Bulan=(pb)
87
88     plt.bar(Nama_Band,Pendengar_per_Bulan)
89
90     plt.title("Jumlah Pendengar Perbulan Di spotify", size=16)
91     plt.xlabel("Jumlah Pendengar Perbulan", size=14)
92     plt.ylabel("Nama Band")
93
94     plt.show()
95     plt.savefig(sys.stdout.buffer)
96     sys.stdout.flush()
97
```
- Terminal:** At the bottom, the terminal window shows the command `PS C:\PythonB& c:\Python\env\Scripts\Activate.ps1` followed by a prompt `(env)`.
- Bottom Status Bar:** Shows standard VS Code status icons like file save, search, and zoom.

Output :



Plot dengan bentuk berbeda

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

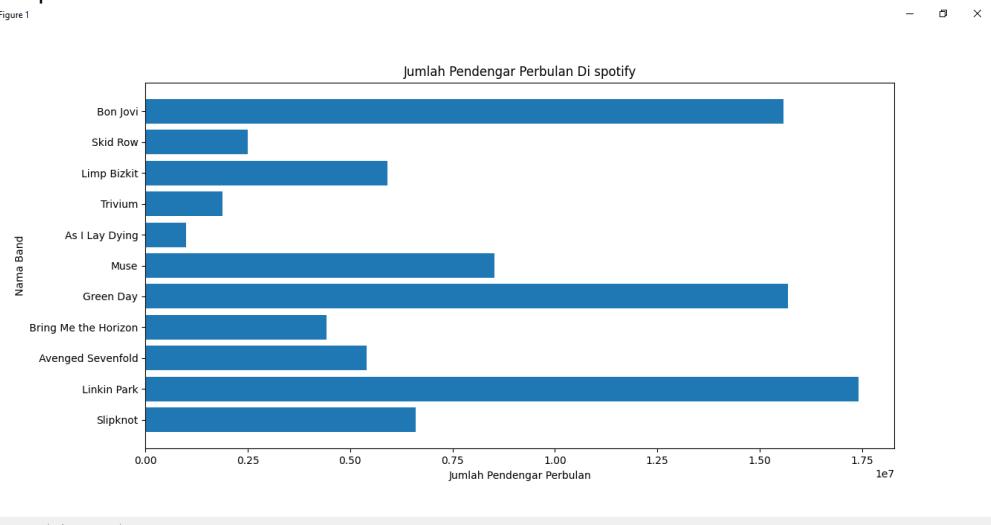
- File Explorer:** Shows files in the current directory, including `modul6_datascience.py`, `data3.csv`, `py2_modul7.py`, and `dataseumrah.csv`.
- Code Cell:** Displays Python code for reading a CSV file and creating a bar chart.

```
from matplotlib import pyplot as plt
import numpy as np
import csv
import pandas as pd
df = pd.read_csv("data3.csv")
Nama_Band=list(pd.read_csv("Nama_Band"))
ppb=list(pd.read_csv("Pendengar_per_Bulan"))
print(Nama_Band)
print(ppb)

Nama_Band=(Nama_Band)
Pendengar_per_Bulan=(ppb)

plt.figure(figsize=(12,2))
plt.barh(Nama_Band,Pendengar_per_Bulan)
plt.title("Jumlah Pendengar Perbulan Di spotify")
plt.xlabel("Jumlah Pendengar Perbulan")
plt.ylabel("Nama Band")
plt.show()
```
- Output Cell:** Shows the command run and its output.
 - PS C:\Python8 & c:/Python/env/Scripts/Activate.ps1
 - (env) PS C:\Python8 & c:/Python/env/Scripts/python.exe c:/Python8/modul6_datascience.py
 - (env) PS C:\Python8 []
- Terminal:** Shows the command run and its output.
- Python Kernel:** Shows the Python version and path.
- Bottom Status Bar:** Shows the current line (Ln 73), column (Col 1), and other system information.

Output :



MODUL 7

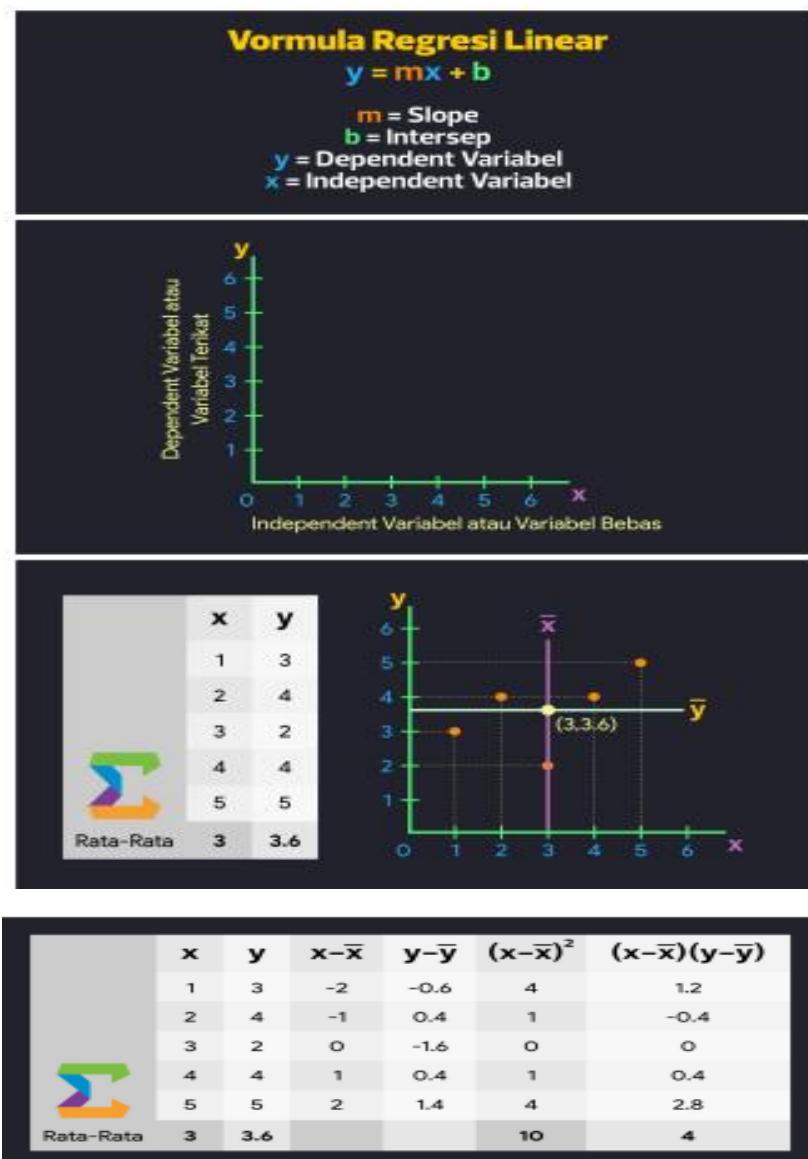
MACHINE LEARNING PREDIKSI HARGA RUMAH

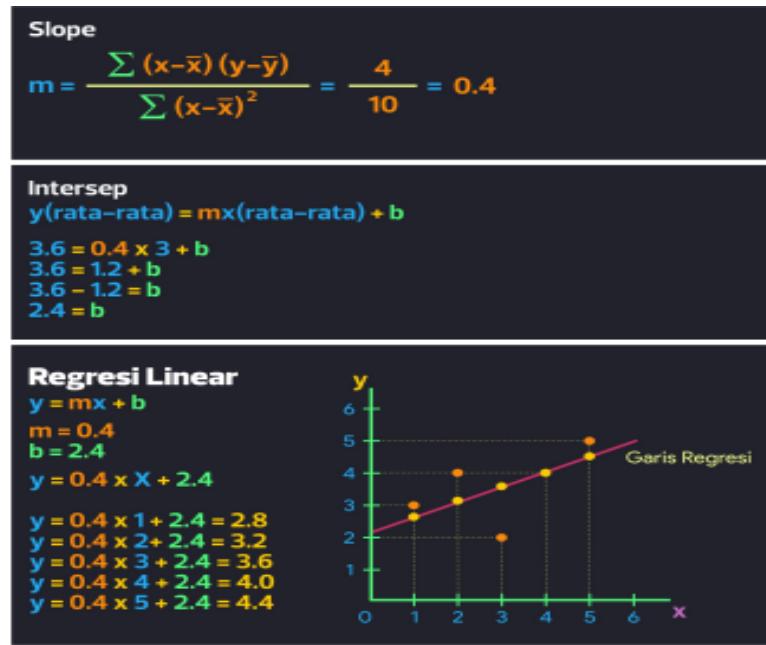
DENGAN REGRESI LINIER (SATU VARIABEL)

A. Pendahuluan

Machine learning merupakan cabang dari kecerdasan buatan yang berfokus pada pengembangan suatu sistem yang dapat belajar sendiri tanpa berulang-ulang dilakukan pemrograman.

Regresi Linier merupakan metode statistika yang digunakan untuk membentuk model hubungan antara variabel terkait/dependen y dengan satu atau lebih variabel bebas/independen





Berikut Pengimplementasian dengan python

1. Import Data Rumah

```
File Edit Selection View Go Run ... ⟲ ⟳ Python
-- modul6_dascience.py data3.csv py2_modul7.py datasetrumah.csv
PYTHOND
> _pycache_...
> vscode...
> venv...
> vsc...
Marketplace
data.csv
data2.csv
data4.csv
data5.csv
data6.csv
data7.csv
data8.csv
data9.csv
DataScienceTutorial.py
datasetrumah.csv
f6_undangan...
Modul1.py
Modul2.py
Modul3.py
Modul4.py
Modul5.py
Modul6_dascience.py
Modul7.py
Modul_MachineLearning.py
Modul_MachineLearning.py
my2_modul7.py
py2_modul7.py
teknologi...
teknologi.py

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python38 & c:/Python38/Scripts/Activate.ps1
○ (env) PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python/modul6_dascience.py
```

2. Mengimplementasikan

Code program dibawah digunakan untuk menampilkan dan membaca isi dari file csv data rumah

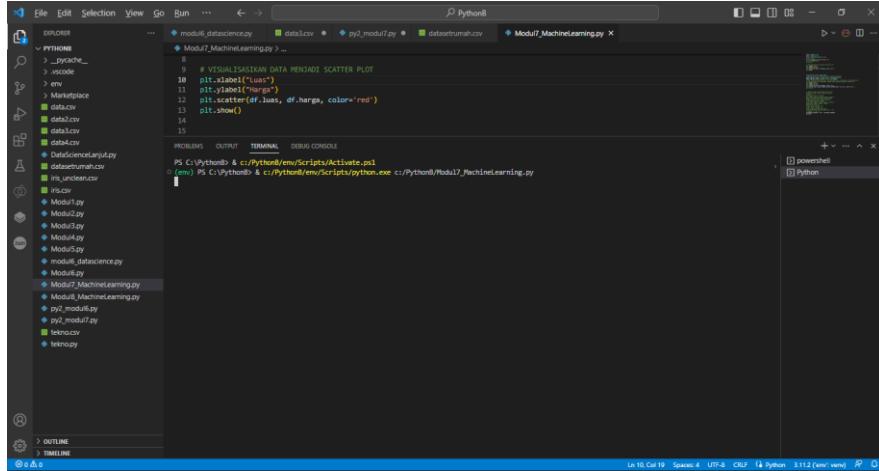
```
File Edit Selection View Go Run ... ⟲ ⟳ Python
-- modul6_dascience.py ... data3.csv py2_modul7.py datasetrumah.csv Modul7_MachineLearning.py
PYTHOND
> _pycache_...
> vscode...
> venv...
> vsc...
Marketplace
data.csv
data2.csv
data4.csv
data5.csv
data6.csv
data7.csv
data8.csv
data9.csv
DataScienceTutorial.py
datasetrumah.csv
f6_undangan...
Modul1.py
Modul2.py
Modul3.py
Modul4.py
Modul5.py
Modul6_dascience.py
Modul7_MachineLearning.py
Modul_MachineLearning.py
my2_modul7.py
py2_modul7.py
teknologi...
teknologi.py

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python38 & c:/Python38/Scripts/Activate.ps1
○ (env) PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python/Modul7_MachineLearning.py
In[1]: import pandas as pd
In[2]: df = pd.read_csv('datasetrumah.csv')
In[3]: print(df)
Out[3]:
   0    120  13000000
   1    130  15000000
   2    140  16000000
   3    150  22000000
   4    170  23000000
   5    180  25000000
   6    200  26000000
   7    220  30000000
   8    230  32000000
   9    250  35000000
   10   270  36000000
   11   280  38000000
   12   290  40000000
   13   300  42000000
   14   320  45000000
   15   340  48000000
   16   360  50000000
   17   380  52000000
   18   400  55000000
   19   420  58000000
   20   440  60000000
   21   460  62000000
   22   480  65000000
   23   500  68000000
   24   520  70000000
   25   540  72000000
   26   560  75000000
   27   580  78000000
   28   600  80000000
   29   620  82000000
   30   640  85000000
   31   660  88000000
   32   680  90000000
   33   700  92000000
   34   720  95000000
   35   740  98000000
   36   760  100000000
   37   780  102000000
   38   800  105000000
   39   820  108000000
   40   840  110000000
   41   860  112000000
   42   880  115000000
   43   900  118000000
   44   920  120000000
   45   940  122000000
   46   960  125000000
   47   980  128000000
   48   1000 130000000
   49   1020 132000000
   50   1040 135000000
   51   1060 138000000
   52   1080 140000000
   53   1100 142000000
   54   1120 145000000
   55   1140 148000000
   56   1160 150000000
   57   1180 152000000
   58   1200 155000000
   59   1220 158000000
   60   1240 160000000
   61   1260 162000000
   62   1280 165000000
   63   1300 168000000
   64   1320 170000000
   65   1340 172000000
   66   1360 175000000
   67   1380 178000000
   68   1400 180000000
   69   1420 182000000
   70   1440 185000000
   71   1460 188000000
   72   1480 190000000
   73   1500 192000000
   74   1520 195000000
   75   1540 198000000
   76   1560 200000000
   77   1580 202000000
   78   1600 205000000
   79   1620 208000000
   80   1640 210000000
   81   1660 212000000
   82   1680 215000000
   83   1700 218000000
   84   1720 220000000
   85   1740 222000000
   86   1760 225000000
   87   1780 228000000
   88   1800 230000000
   89   1820 232000000
   90   1840 235000000
   91   1860 238000000
   92   1880 240000000
   93   1900 242000000
   94   1920 245000000
   95   1940 248000000
   96   1960 250000000
   97   1980 252000000
   98   2000 255000000
   99   2020 258000000
   100  2040 260000000
   101  2060 262000000
   102  2080 265000000
   103  2100 268000000
   104  2120 270000000
   105  2140 272000000
   106  2160 275000000
   107  2180 278000000
   108  2200 280000000
   109  2220 282000000
   110  2240 285000000
   111  2260 288000000
   112  2280 290000000
   113  2300 292000000
   114  2320 295000000
   115  2340 298000000
   116  2360 300000000
   117  2380 302000000
   118  2400 305000000
   119  2420 308000000
   120  2440 310000000
   121  2460 312000000
   122  2480 315000000
   123  2500 318000000
   124  2520 320000000
   125  2540 322000000
   126  2560 325000000
   127  2580 328000000
   128  2600 330000000
   129  2620 332000000
   130  2640 335000000
   131  2660 338000000
   132  2680 340000000
   133  2700 342000000
   134  2720 345000000
   135  2740 348000000
   136  2760 350000000
   137  2780 352000000
   138  2800 355000000
   139  2820 358000000
   140  2840 360000000
   141  2860 362000000
   142  2880 365000000
   143  2900 368000000
   144  2920 370000000
   145  2940 372000000
   146  2960 375000000
   147  2980 378000000
   148  3000 380000000
   149  3020 382000000
   150  3040 385000000
   151  3060 388000000
   152  3080 390000000
   153  3100 392000000
   154  3120 395000000
   155  3140 398000000
   156  3160 400000000
   157  3180 402000000
   158  3200 405000000
   159  3220 408000000
   160  3240 410000000
   161  3260 412000000
   162  3280 415000000
   163  3300 418000000
   164  3320 420000000
   165  3340 422000000
   166  3360 425000000
   167  3380 428000000
   168  3400 430000000
   169  3420 432000000
   170  3440 435000000
   171  3460 438000000
   172  3480 440000000
   173  3500 442000000
   174  3520 445000000
   175  3540 448000000
   176  3560 450000000
   177  3580 452000000
   178  3600 455000000
   179  3620 458000000
   180  3640 460000000
   181  3660 462000000
   182  3680 465000000
   183  3700 468000000
   184  3720 470000000
   185  3740 472000000
   186  3760 475000000
   187  3780 478000000
   188  3800 480000000
   189  3820 482000000
   190  3840 485000000
   191  3860 488000000
   192  3880 490000000
   193  3900 492000000
   194  3920 495000000
   195  3940 498000000
   196  3960 500000000
   197  3980 502000000
   198  4000 505000000
   199  4020 508000000
   200  4040 510000000
   201  4060 512000000
   202  4080 515000000
   203  4100 518000000
   204  4120 520000000
   205  4140 522000000
   206  4160 525000000
   207  4180 528000000
   208  4200 530000000
   209  4220 532000000
   210  4240 535000000
   211  4260 538000000
   212  4280 540000000
   213  4300 542000000
   214  4320 545000000
   215  4340 548000000
   216  4360 550000000
   217  4380 552000000
   218  4400 555000000
   219  4420 558000000
   220  4440 560000000
   221  4460 562000000
   222  4480 565000000
   223  4500 568000000
   224  4520 570000000
   225  4540 572000000
   226  4560 575000000
   227  4580 578000000
   228  4600 580000000
   229  4620 582000000
   230  4640 585000000
   231  4660 588000000
   232  4680 590000000
   233  4700 592000000
   234  4720 595000000
   235  4740 598000000
   236  4760 600000000
   237  4780 602000000
   238  4800 605000000
   239  4820 608000000
   240  4840 610000000
   241  4860 612000000
   242  4880 615000000
   243  4900 618000000
   244  4920 620000000
   245  4940 622000000
   246  4960 625000000
   247  4980 628000000
   248  5000 630000000
   249  5020 632000000
   250  5040 635000000
   251  5060 638000000
   252  5080 640000000
   253  5100 642000000
   254  5120 645000000
   255  5140 648000000
   256  5160 650000000
   257  5180 652000000
   258  5200 655000000
   259  5220 658000000
   260  5240 660000000
   261  5260 662000000
   262  5280 665000000
   263  5300 668000000
   264  5320 670000000
   265  5340 672000000
   266  5360 675000000
   267  5380 678000000
   268  5400 680000000
   269  5420 682000000
   270  5440 685000000
   271  5460 688000000
   272  5480 690000000
   273  5500 692000000
   274  5520 695000000
   275  5540 698000000
   276  5560 700000000
   277  5580 702000000
   278  5600 705000000
   279  5620 708000000
   280  5640 710000000
   281  5660 712000000
   282  5680 715000000
   283  5700 718000000
   284  5720 720000000
   285  5740 722000000
   286  5760 725000000
   287  5780 728000000
   288  5800 730000000
   289  5820 732000000
   290  5840 735000000
   291  5860 738000000
   292  5880 740000000
   293  5900 742000000
   294  5920 745000000
   295  5940 748000000
   296  5960 750000000
   297  5980 752000000
   298  6000 755000000
   299  6020 758000000
   300  6040 760000000
   301  6060 762000000
   302  6080 765000000
   303  6100 768000000
   304  6120 770000000
   305  6140 772000000
   306  6160 775000000
   307  6180 778000000
   308  6200 780000000
   309  6220 782000000
   310  6240 785000000
   311  6260 788000000
   312  6280 790000000
   313  6300 792000000
   314  6320 795000000
   315  6340 798000000
   316  6360 800000000
   317  6380 802000000
   318  6400 805000000
   319  6420 808000000
   320  6440 810000000
   321  6460 812000000
   322  6480 815000000
   323  6500 818000000
   324  6520 820000000
   325  6540 822000000
   326  6560 825000000
   327  6580 828000000
   328  6600 830000000
   329  6620 832000000
   330  6640 835000000
   331  6660 838000000
   332  6680 840000000
   333  6700 842000000
   334  6720 845000000
   335  6740 848000000
   336  6760 850000000
   337  6780 852000000
   338  6800 855000000
   339  6820 858000000
   340  6840 860000000
   341  6860 862000000
   342  6880 865000000
   343  6900 868000000
   344  6920 870000000
   345  6940 872000000
   346  6960 875000000
   347  6980 878000000
   348  7000 880000000
   349  7020 882000000
   350  7040 885000000
   351  7060 888000000
   352  7080 890000000
   353  7100 892000000
   354  7120 895000000
   355  7140 898000000
   356  7160 900000000
   357  7180 902000000
   358  7200 905000000
   359  7220 908000000
   360  7240 910000000
   361  7260 912000000
   362  7280 915000000
   363  7300 918000000
   364  7320 920000000
   365  7340 922000000
   366  7360 925000000
   367  7380 928000000
   368  7400 930000000
   369  7420 932000000
   370  7440 935000000
   371  7460 938000000
   372  7480 940000000
   373  7500 942000000
   374  7520 945000000
   375  7540 948000000
   376  7560 950000000
   377  7580 952000000
   378  7600 955000000
   379  7620 958000000
   380  7640 960000000
   381  7660 962000000
   382  7680 965000000
   383  7700 968000000
   384  7720 970000000
   385  7740 972000000
   386  7760 975000000
   387  7780 978000000
   388  7800 980000000
   389  7820 982000000
   390  7840 985000000
   391  7860 988000000
   392  7880 990000000
   393  7900 992000000
   394  7920 995000000
   395  7940 998000000
   396  7960 1000000000
   397  7980 1002000000
   398  8000 1005000000
   399  8020 1008000000
   400  8040 1010000000
   401  8060 1012000000
   402  8080 1015000000
   403  8100 1018000000
   404  8120 1020000000
   405  8140 1022000000
   406  8160 1025000000
   407  8180 1028000000
   408  8200 1030000000
   409  8220 1032000000
   410  8240 1035000000
   411  8260 1038000000
   412  8280 1040000000
   413  8300 1042000000
   414  8320 1045000000
   415  8340 1048000000
   416  8360 1050000000
   417  8380 1052000000
   418  8400 1055000000
   419  8420 1058000000
   420  8440 1060000000
   421  8460 1062000000
   422  8480 1065000000
   423  8500 1068000000
   424  8520 1070000000
   425  8540 1072000000
   426  8560 1075000000
   427  8580 1078000000
   428  8600 1080000000
   429  8620 1082000000
   430  8640 1085000000
   431  8660 1088000000
   432  8680 1090000000
   433  8700 1092000000
   434  8720 1095000000
   435  8740 1098000000
   436  8760 1100000000
   437  8780 1102000000
   438  8800 1105000000
   439  8820 1108000000
   440  8840 1110000000
   441  8860 1112000000
   442  8880 1115000000
   443  8900 1118000000
   444  8920 1120000000
   445  8940 1122000000
   446  8960 1125000000
   447  8980 1128000000
   448  9000 1130000000
   449  9020 1132000000
   450  9040 1135000000
   451  9060 11380
```

a) Visualisasikan data menjadi scatter plot sebagai berikut :

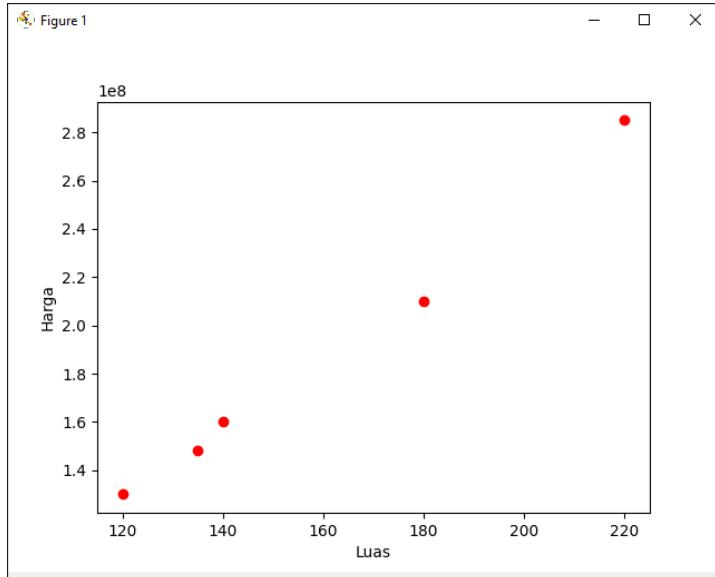


The screenshot shows the PyCharm IDE interface. The left sidebar displays a file tree with various Python files and CSV datasets. The main editor window contains the following Python code:

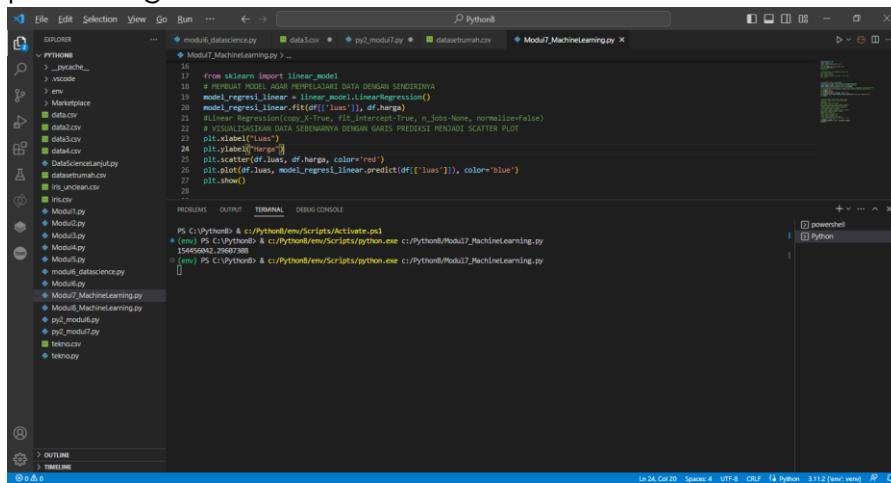
```
8 # VISUALISASIKAN DATA MENJADI SCATTER PLOT
9 plt.xlabel("Luas")
10 plt.ylabel("Harga")
11 df = pd.read_csv('dataset.csv')
12 plt.scatter(df.luas, df.harga, color="red")
13 plt.show()
14
15
```

The terminal tab at the bottom shows the command run in the terminal: PS C:\Python3\ & c:/Python3/Scripts/activate.ps1 & python Modul7_MachineLearning.py. The output in the terminal shows the command and the file path.

Output :



b) Visualisasikan data sebenarnya dengan garis prediksi menjadi scatter plot sebagai berikut :

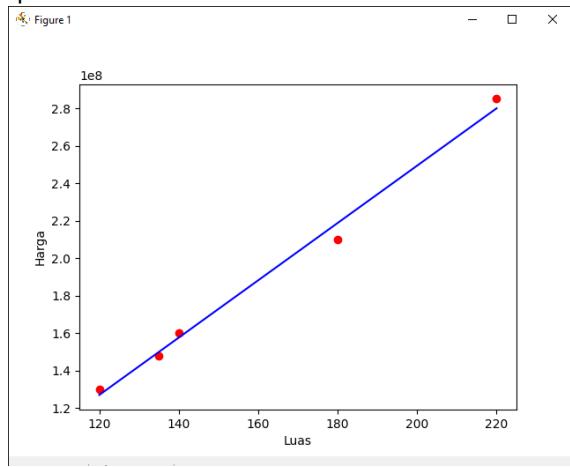


The screenshot shows the PyCharm IDE interface. The left sidebar displays a file tree with various Python files and CSV datasets. The main editor window contains the following Python code:

```
16
17 from sklearn import linear_model
18 # Membuat model agar mempelajari data dengan seridinha
19 model_regress_linear = linear_model.LinearRegression()
20 model_regress_linear.fit(df[['luas']], df.harga)
21 # Linear Regression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
22
23 plt.scatter(df.luas, df.harga, color="red")
24 plt.plot(df.luas, model_regress_linear.predict(df[['luas']]), color="blue")
25 plt.title("Harga")
26 plt.xlabel("Luas")
27 plt.ylabel("Harga")
28 plt.show()
```

The terminal tab at the bottom shows the command run in the terminal: PS C:\Python3\ & c:/Python3/Scripts/activate.ps1 & python Modul7_MachineLearning.py. The output in the terminal shows the command and the file path.

Output :



- c) Prediksi harga untuk rumah seluas 138m²

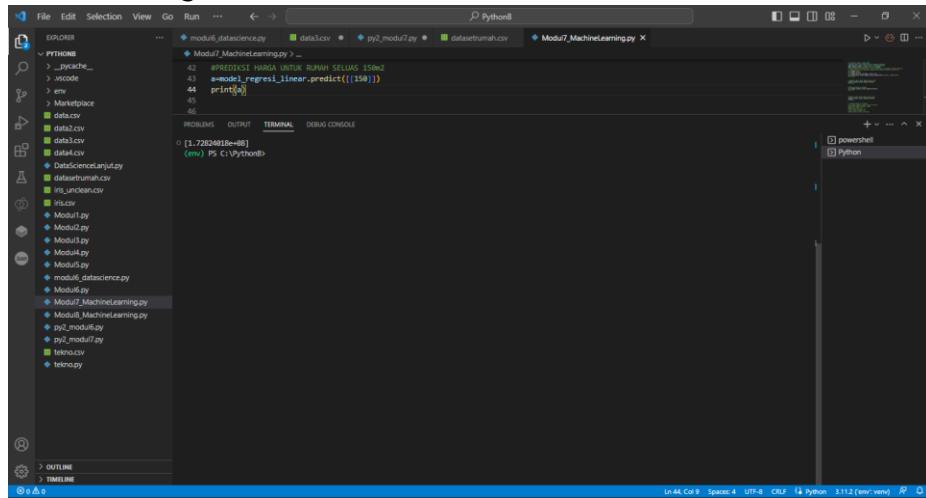
- d) Tampilkan harga ke integer

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `modul6_datascience.py`, `data3.csv`, `py2_modul7.py`, `datasetrumah.csv`, and `Modul7_MachineLearning.py`.
- Terminal:** Displays the command `PS C:\Python3` and the output `154459642`.
- Code Editor:** The file `Modul7_MachineLearning.py` is open, containing the following code:

```
35 #INPUT LAMAR HARGA KE INTEGER
36 a=int(modul_regres_linear.predict([[138]]))
37 print(a)
38
39
40
```
- Bottom Status Bar:** Shows line 95, column 45, spacing 4, UTF-8, CR/LF, Python 3.11.2 (min-vend), and a file icon.

e) Prediksi harga untuk rumah seluas 150m²



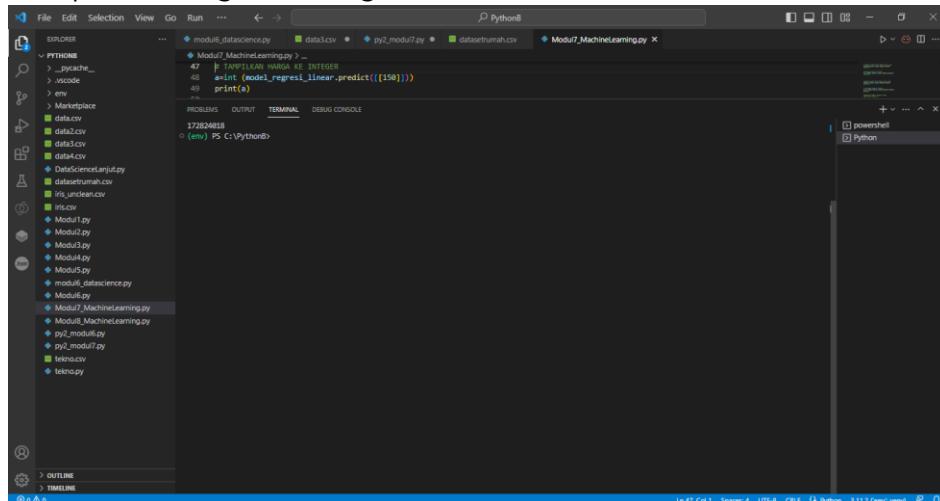
A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a file tree with several Python files and CSV datasets. The main editor window contains the following Python code:

```
#PREDIKSI HARGA UNTUK RUMAH SELUAS 150m2
41 import numpy as np
42 from sklearn.linear import LinearRegression
43 model = LinearRegression()
44 model.fit(X, Y)
45 print(model.predict([[150]]))
```

The terminal at the bottom shows the output of the code execution:

```
[1,7284918e+08]
(eme) PS C:\Python8
```

f) Tampilkan harga ke integer



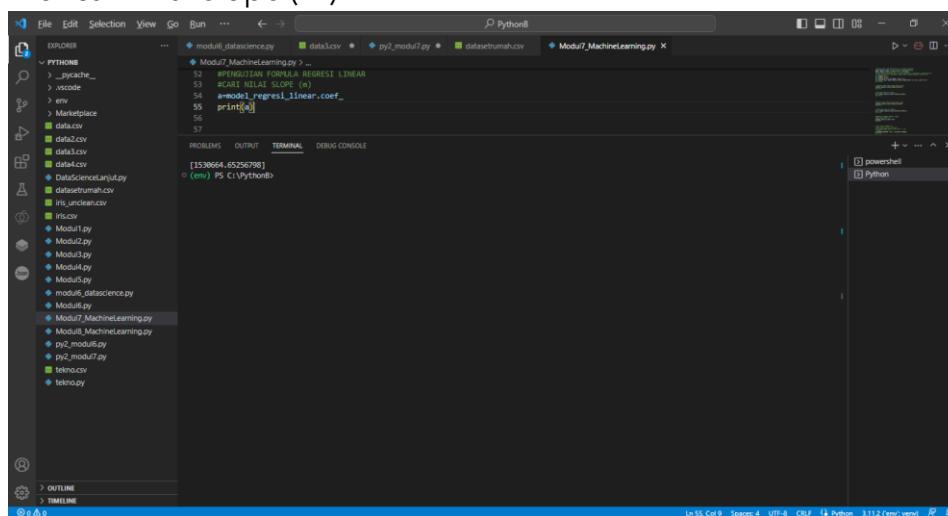
A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a file tree with several Python files and CSV datasets. The main editor window contains the following Python code:

```
#TAMPILKAN HARGA KE INTEGER
47 model = LinearRegression()
48 model.fit(X, Y)
49 print(int(model.predict([[150]])))
```

The terminal at the bottom shows the output of the code execution:

```
17284918
(eme) PS C:\Python8
```

g) Mencari nilai slope (m)



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a file tree with several Python files and CSV datasets. The main editor window contains the following Python code:

```
#FIND SLOPE OF REGRESSION LINE
52 print("REGRESI SLOPE : ", model.coef_[0])
53 print("INTERSECT FORMULA REGRESI LINEAR")
54 print("y = " + str(model.intercept_) + " + " + str(model.coef_[0]) + "x")
55 print(model.coef_[0])
56
57
```

The terminal at the bottom shows the output of the code execution:

```
[153664.65256798]
(eme) PS C:\Python8
```

h) Cari nilai intersep (b)

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files in the current workspace, including `modul7_dataScience.py`, `modul7_modul7.py`, `datasetumrah.csv`, and `modul7_MachineLearning.py`.
- Code Editor:** The main area displays Python code for a linear regression model:

```
58
59 # CARI NILAI INTERSEPT (b)
60 a=modul7_regress_linear.intercept_
61 print(a)
62
```
- Terminal:** Below the code editor, the terminal window shows the output of a command:

```
-56775079.7588823
```
- Bottom Status Bar:** Shows the file `modul7_MachineLearning.py`, line 59, column 25, with 4 spaces, 8 tabs, and 100% zoom.

i) Uji untuk rumah seluas 138m² atau $x = 138$

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows files in the current workspace, including `modul1_datascience.py`, `modul2_MachineLearning.py`, `modul3.csv`, `py2_modul7.py`, `datasetrumah.csv`, and `Modul7_MachineLearning.py`.
- Terminal:** Displays the command `python Modul7_MachineLearning.py` and its output:

```
60
61 # UJI UNTUK RUMAH SELUAS 118m2 ATAU x = 138
62 # FORMULA: y = m + b
63
64 y = 0.002214 * 138 + -56775679.75830812
65
66 print(y)
```
- Output:** Shows the result of the calculation: `154456842.29667308`.
- Problems:** No problems are listed.
- Debug Console:** No entries are present.
- Status Bar:** Shows the current file is `Modul7_MachineLearning.py`, the line number is 60, and the column number is 1.

HASILNYA : Rp. 154.456.042

Maka dapat disimpulkan terdapat beberapa model yang dapat digunakan berdasarkan datarumah.csv dengan hasil akhir dari model menetapkan harga rumah sebesar Rp. 154.456.042

MODUL 8

MACHINE LEARNING PREDIKSI HARGA RUMAH DENGAN REGRESI LINEAR MULTI VARIABEL

A. Implementasi Dengan Python

Pada implementasi ini kita akan memprediksi harga rumah dengan variabel luas rumah, banyaknya kamar tidur dan kamar mandi.



1. Menampilkan data yang ada pada file csv

2. Menerapkan data dan mengisi data yang hilang

3. Mengganti nilai 0 menjadi NaN (Not A Number)

4. Mencari nilai median dari kolom kamar tidur

The screenshot shows the PyCharm IDE interface with the following details:

- File Structure:** On the left, there's a tree view of files and folders. The 'Modul18_MachineLearning' folder is expanded, showing 'Modul18_MachineLearning.py' and 'Modul18_MachineLearning.ipynb'. Other files like 'data1.csv', 'data2.csv', etc., are also listed.
- Code Editor:** The main area displays Python code:

```
# mencari nilai median dari kolom kematidur
z1 = df['kematidur'].median()
print(z1)
```
- Terminal:** At the bottom, the terminal window shows the command and output:

```
PS C:\Python3 & c:\Python\myScripts\activate.ps1
(env) PS C:\Python3 & c:\Python\myScripts\python.exe c:/Python/modul18_MachineLearning.py
4.0
(env) PS C:\Python3
```
- Status Bar:** The bottom right corner shows the status bar with 'Ln 21, Col 58' and other system information.

5. Membulatkan nilai

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it lists several Python files and CSV files under the "Python" folder.
- Code Editor:** The main area displays a Python script named "Modul18_MachineLearning.py". The code includes importing pandas, reading a CSV file, calculating the median of a column, and printing the result.
- Terminal:** At the bottom, the terminal window shows the command to activate the Python environment and run the script, followed by the output "4".

6. Fillna untuk mengganti NaN value

7. Menampilkan value yang baru kedalam DataFrame

The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays a file structure with several Python files (MachineLearning.py, Modul8/MachineLearning.py, etc.) and CSV files (dataframe1.csv, data3.csv, etc.). The right side features a terminal window titled 'Python' where the command `python3 dataframe1.py` is run, resulting in the following output:

```
PS C:\Python3 & cd\Python\env\Scripts\Activate.ps1
* (env) PS C:\Python3 & cd\Python\env\Scripts\python.exe c:/Python3/Modul8_MachineLearning.py
    0   100      3   130000000
    1   135      3   140000000
    2   140      4   160000000
    3   130      3   210000000
    4   220      6   250000000
* (env) PS C:\Python3
```

8. Menampilkan nilai slope jika dibutuhkan

9. Menampilkan nilai intersep jika dibutuhkan

The screenshot shows the PyCharm IDE interface. The left sidebar displays a file tree with various CSV files and Python scripts. The main editor window shows a Python script named `Modul8_MachineLearning.py`. The terminal tab at the bottom shows the following command-line session:

```
PS C:\Python8\ & c:\Python8\env\Scripts\activate.ps1
(env) PS C:\Python8\ & c:\Python8\env\Scripts\python.exe c:/Python8/Modul8_MachineLearning.py
-58148876,404494435
(env) PS C:\Python8\
```

10. Prediksi