

PYTHON MODUL 4

A. Percabangan

Percabangan yaitu cara yang digunakan untuk mengambil keputusan apabila di dalam program terdapat kondisi tertentu. Jumlah kondisinya bisa satu, dua atau lebih terdapat 3 jenis pernyataan yang digunakan untuk percabangan, yaitu if, if else dan if elif else

1. Pernyataan if

Pernyataan if menjalankan satu buah kondisi bila hasilnya benar maka pernyataan di dalam blok if tersebut dieksekusi jika salah, maka pernyataan tidak dieksekusi.

Contoh program Pernyataan if :

```
File Edit Selection View Go Run ... ⌘ PythonB
EXPLORER PYTHONB >_pycache_ >_vscode Marketplace Modul1.py Modul2.py Modul3.py Modul4.py
Modul4.py > ...
1 angka = 4
2 if angka > 0:
3     print(angka, "adalah Bilangan positif.")
4
5

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
4 adalah Bilangan positif.
PS C:\PythonB>

In 5, Col 1  Spaces 4  UTF-8  CRLF  (4 Python 3.11.2 64-bit (microsoft store)) ⌘ Q
```

2. Pernyataan if else

Pernyataan if else menjalankan dua kondisi. Kondisi pertama kalau benar, dan kondisi kedua kalau salah. Contoh program Pernyataan if else:

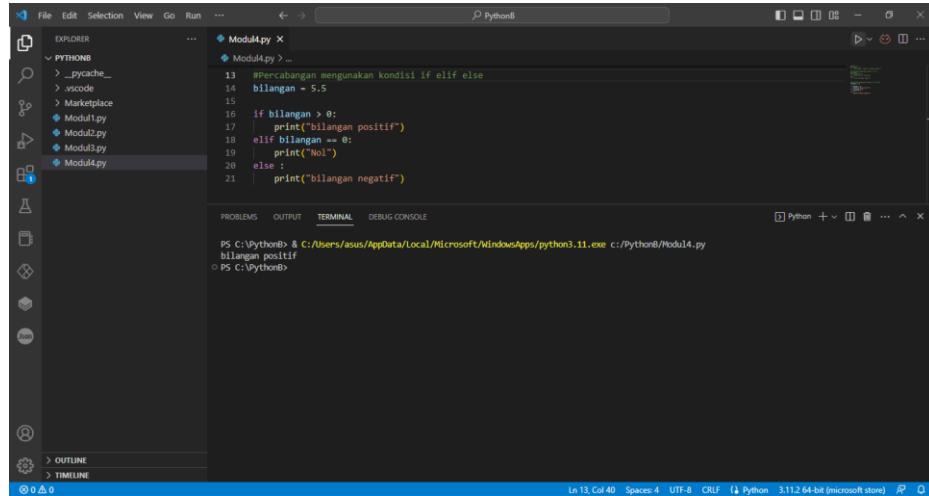
```
File Edit Selection View Go Run ... ⌘ PythonB
EXPLORER PYTHONB >_pycache_ >_vscode Marketplace Modul1.py Modul2.py Modul3.py Modul4.py
Modul4.py > ...
1 #Percabangan menggunakan kondisi if else
2 bilangan = -1
3 if bilangan >= 0:
4     print("Positif atau Nol")
5 else:
6     print("bilangan negatif")
7

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
bilangan negatif.
PS C:\PythonB>

In 5, Col 40  Spaces 4  UTF-8  CRLF  (4 Python 3.11.2 64-bit (microsoft store)) ⌘ Q
```

3. Pernyataan if elif else

Pernyataan if elif else digunakan untuk menjalankan lebih dari dua kondisi. Bila kondisi pada if benar, maka pernyataan di dalamnya yang dieksekusi. Bila salah, maka masuk ke kondisi elif. Dan bila tidak ada if atau elif yang benar, maka yang dijalankan adalah yang else. Contoh Pernyataan if elif else :



A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a folder named 'PYTHONB' containing files: '_pycache_'.vscode', 'Marketplace', 'Modul1.py', 'Modul2.py', 'Modul3.py', and 'Modul4.py'. The 'Modul4.py' file is open in the editor. The code is as follows:

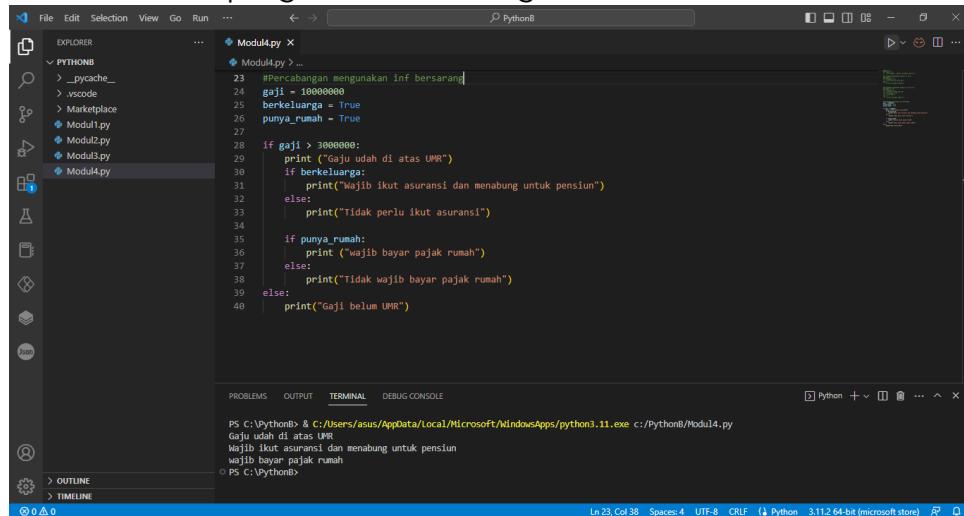
```
13 #Percabangan menggunakan kondisi if elif else
14 bilangan = 5.5
15
16 if bilangan > 0:
17     print("bilangan positif")
18 elif bilangan == 0:
19     print("nol")
20 else:
21     print("bilangan negatif")
```

The terminal below shows the output of running the script:

```
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
bilangan positif
PS C:\PythonB>
```

4. Tambahan : if Bersarang

Sebuah kondisional dapat disimpan di dalam if lain. Berikut ini adalah contoh program if bersarang :



A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a folder named 'PYTHONB' containing files: '_pycache_'.vscode', 'Marketplace', 'Modul1.py', 'Modul2.py', 'Modul3.py', and 'Modul4.py'. The 'Modul4.py' file is open in the editor. The code is as follows:

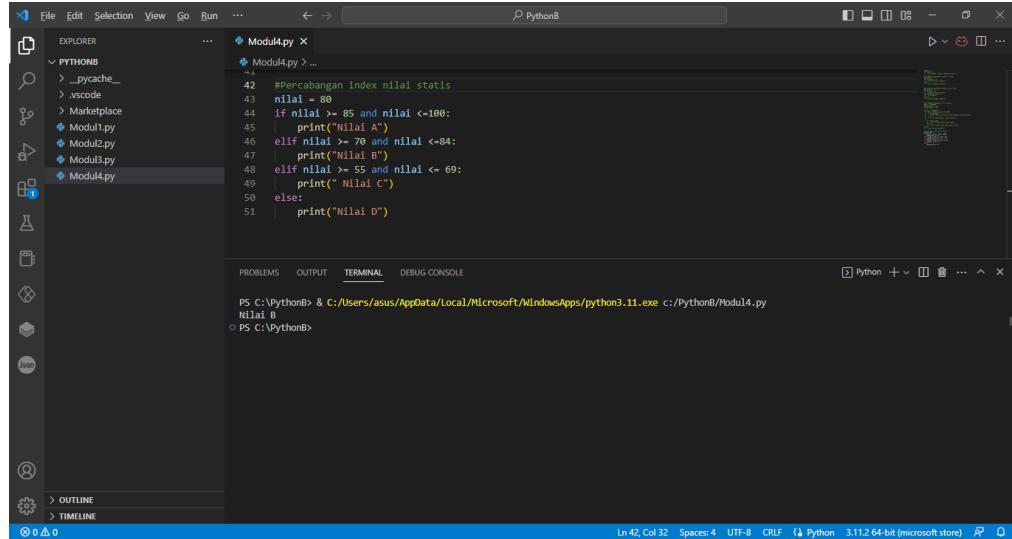
```
23 #percabangan menggunakan if bersarang
24 gaji = 10000000
25 berkeluarga = True
26 punya_rumah = True
27
28 if gaji > 3000000:
29     print ("Gaji udah di atas UMR")
30     if berkeluarga:
31         print("Wajib ikut asuransi dan menabung untuk pensiun")
32     else:
33         print("Tidak perlu ikut asuransi")
34
35 if punya_rumah:
36     print ("wajib bayar pajak rumah")
37 else:
38     print("Tidak wajib bayar pajak rumah")
39 else:
40     print("Gaji belum UMR")
```

The terminal below shows the output of running the script:

```
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Gaji udah di atas UMR
Wajib ikut asuransi dan menabung untuk pensiun
wajib bayar pajak rumah
PS C:\PythonB>
```

5. Program Percabangan Indeks Nilai Statis

Ketentuan : Nilai 85 s/d 100 indeks A, nilai 70 s/d 84 indeks B, nilai 55 s/d 69 indeks C, nilai dibawah 55 indeks D. Contoh Program Percabangan Indeks Nilai Statis :



```
#Percabangan index nilai statis
nilai = 88
if nilai >= 85 and nilai <=100:
    print("Nilai A")
elif nilai >= 70 and nilai <=84:
    print("Nilai B")
elif nilai >= 55 and nilai <= 69:
    print(" Nilai C")
else:
    print("Nilai D")
```

B. Perulangan

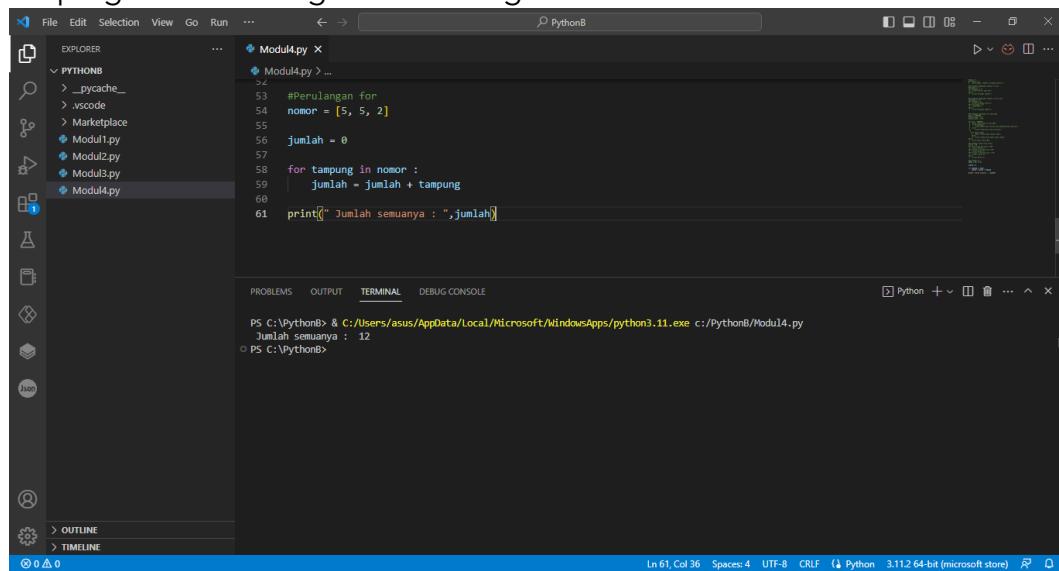
Secara umum, python mengeksekusi program baris perbaris. Mulai dari baris satu, dua, dan seterusnya. perulangan bisa dilakukan dengan dua cara atau metode, yaitu menggunakan for dan menggunakan while.

1. Perulangan dengan Menggunakan for

Perulangan dengan menggunakan for memiliki sintaks seperti berikut :

```
for var in sequence: body of
```

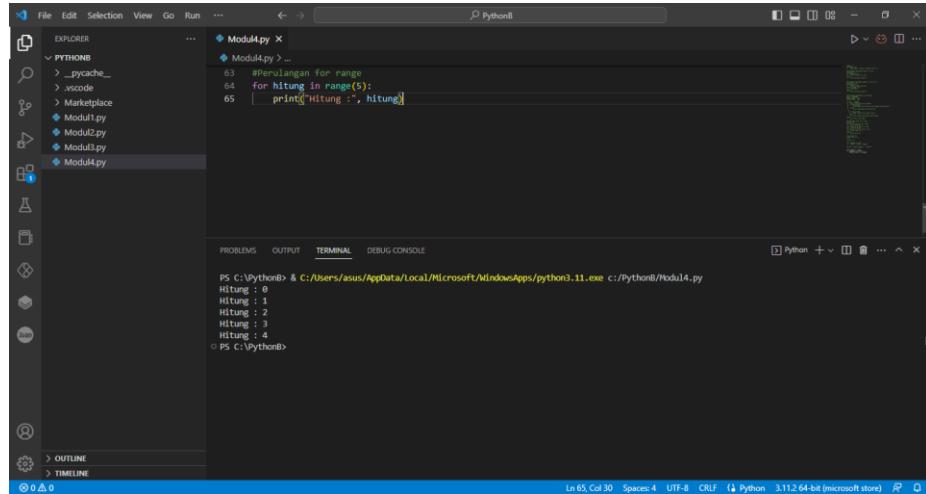
for var adalah variabel digunakan untuk menampung sementara nilai dari sequence pada saat terjadi perulangan. Sequence adalah tipe data berurut seperti string, list, dan tuple. Perulangan terjadi sampai looping mencapai elemen atau anggota terakhir dari sequence. Bila loop sudah sampai ke elemen terakhir dari sequence, maka program akan keluar dari looping. Contoh Program Perulangan for :



```
#Perulangan for
nomor = [5, 5, 2]
jumlah = 0
for tumpang in nomor :
    jumlah = jumlah + tumpang
print(" Jumlah semuanya : ",jumlah)
```

2. Perulangan for dengan range

Fungsi range() digunakan untuk menghasilkan deret bilangan. range(10) akan menghasilkan bilangan dari 0 sampai dengan 9 (10 bilangan). Contoh Program Perulangan for dengan Range :



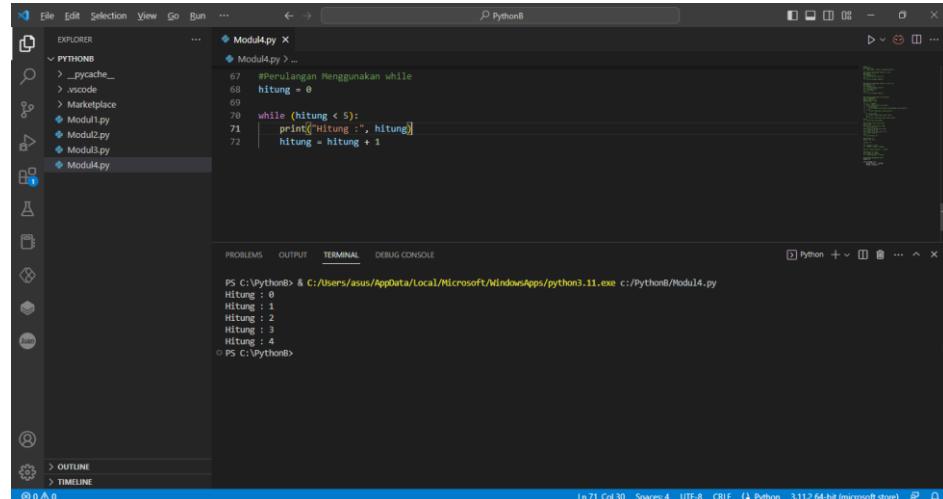
```
File Edit Selection View Go Run ... PythonB
EXPLORER PYTHONB
Modul4.py ...
Modul4.py > ...
63 #Perulangan for range
64 for hitung in range(5):
65     print("Hitung : ", hitung)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\PythonB>

Ln 65, Col 30  Spaces: 4  UTF-8  CR/LF  Python  3.11.2 64-bit (microsoft store)  ⌂ Q
```

3. Perulangan Menggunakan while

Perulangan menggunakan while akan menjalankan blok pernyataan terus menerus selama kondisi bernilai benar. Bila kondisi salah, maka program akan keluar dari while dan lanjut ke baris pernyataan di luar while. Contoh Program Perulangan while :



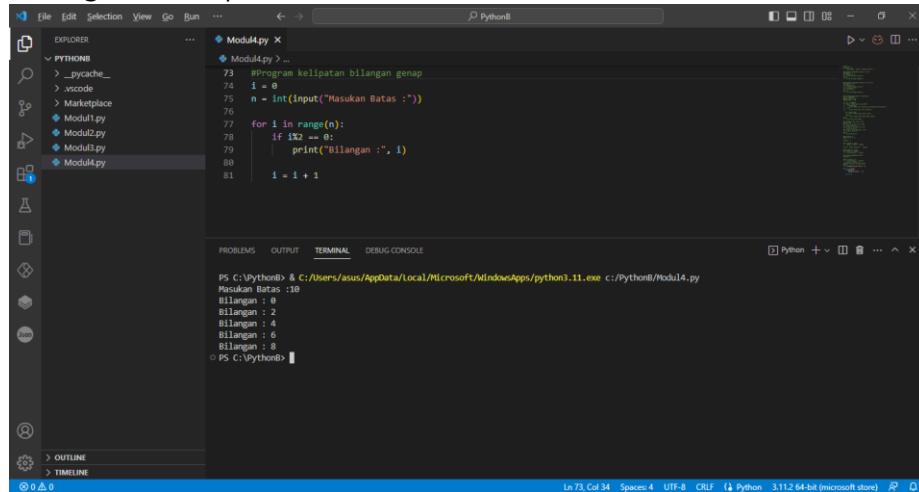
```
File Edit Selection View Go Run ... PythonB
EXPLORER PYTHONB
Modul4.py ...
Modul4.py > ...
57 #Perulangan Menggunakan while
58 hitung = 0
59
60 while (hitung < 5):
61     print("Hitung : ", hitung)
62     hitung = hitung + 1

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\PythonB>

Ln 71, Col 30  Spaces: 4  UTF-8  CR/LF  Python  3.11.2 64-bit (microsoft store)  ⌂ Q
```

4. Program Kelipatan Bilangan Genap

Program pengulangan dengan for. Contoh Program Kelipatan Bilangan Genap :



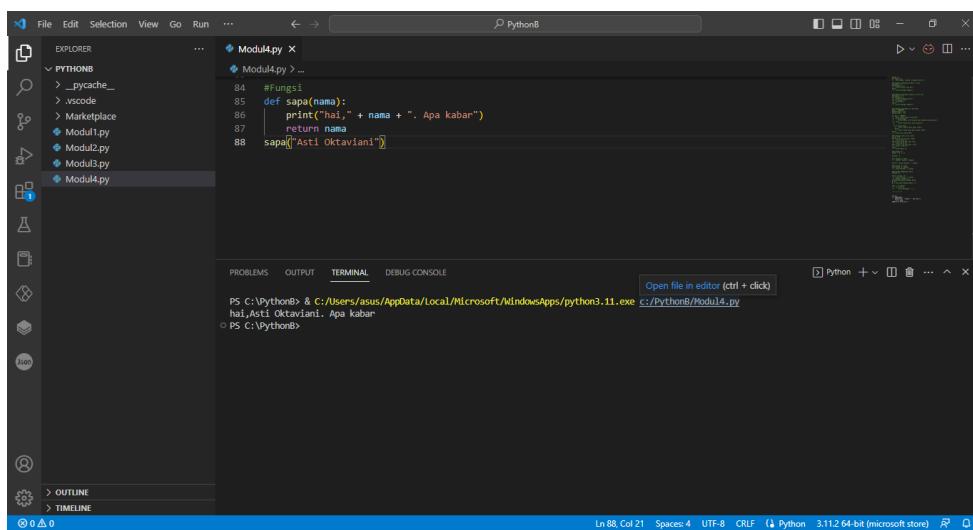
A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: Modul1.py, Modul2.py, Modul3.py, and Modul4.py. The current file is Modul4.py, which contains the following code:

```
73 #Program kelipatan bilangan genap
74 i = 0
75 n = int(input("Masukan Batas :"))
76
77 for i in range(n):
78     if i%2 == 0:
79         print("Bilangan :", i)
80
81     i = i + 1
```

The right side of the screen shows the terminal output. It starts with "Masukan Batas : 10", followed by a series of even numbers from 0 to 8, each preceded by "Bilangan :". The terminal window title is "PythonB".

C. Fungsi

Fungsi adalah grup/blok program untuk melakukan tugas tertentu yang berulang. Fungsi membuat kode program menjadi reusable, artinya hanya di definisikan sekali saja, dan kemudian bisa digunakan berulang kali dalam program. Contoh Program Fungsi :



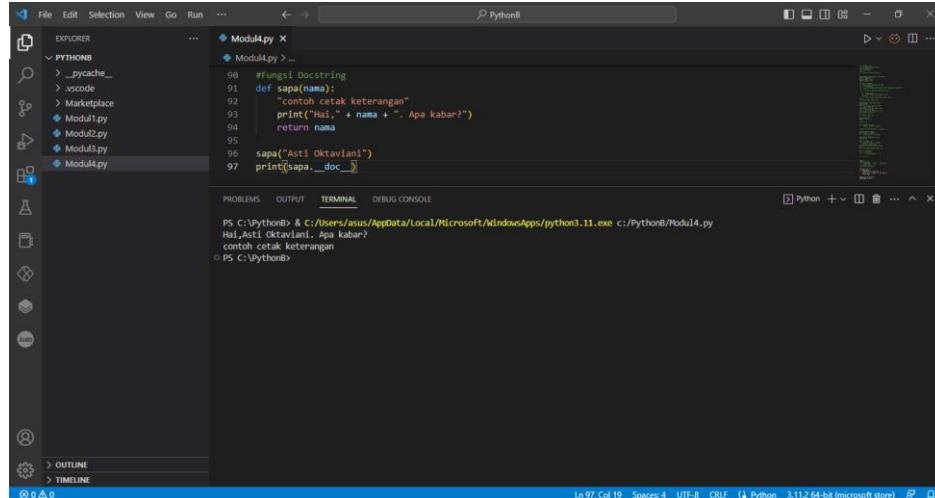
A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: Modul1.py, Modul2.py, Modul3.py, and Modul4.py. The current file is Modul4.py, which contains the following code:

```
84 #FungsI
85 def sapa(nama):
86     print("hai," + nama + ". Apa kabar")
87
88 sapa("Asti Oktaviani")
```

The right side of the screen shows the terminal output. It shows the command "c:\PythonB> & c:/Users/asus/appData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py" followed by the output "hai,Asti Oktaviani. Apa kabar". The terminal window title is "PythonB".

1. Docstring

Docstring adalah singkatan dari documentation string berfungsi sebagai dokumentasi atau keterangan singkat tentang fungsi yang kita buat. Contoh Program Docstring



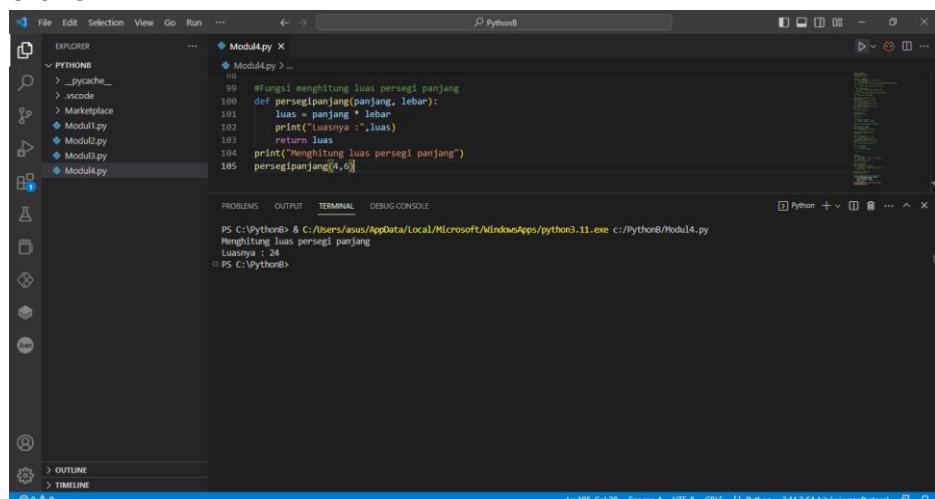
A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: Modul1.py, Modul2.py, Modul3.py, and Modul4.py. The main editor window displays a Python script named Modul4.py. The code contains a function definition:

```
#Fungsi Docstring
def sapa(nama):
    """contoh cetak keterangan"""
    print("Hai," + nama + ". Apa kabar?")
    return nama
sapa("Asti Oktaviani")
print(sapa.__doc__)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\PythonB & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Hai,Asti Oktaviani. Apa kabar?
contoh cetak keterangan
○ PS C:\PythonB>
```

2. Contoh Program Luas Persegi Panjang dengan Fungsi. Contoh Program Statis



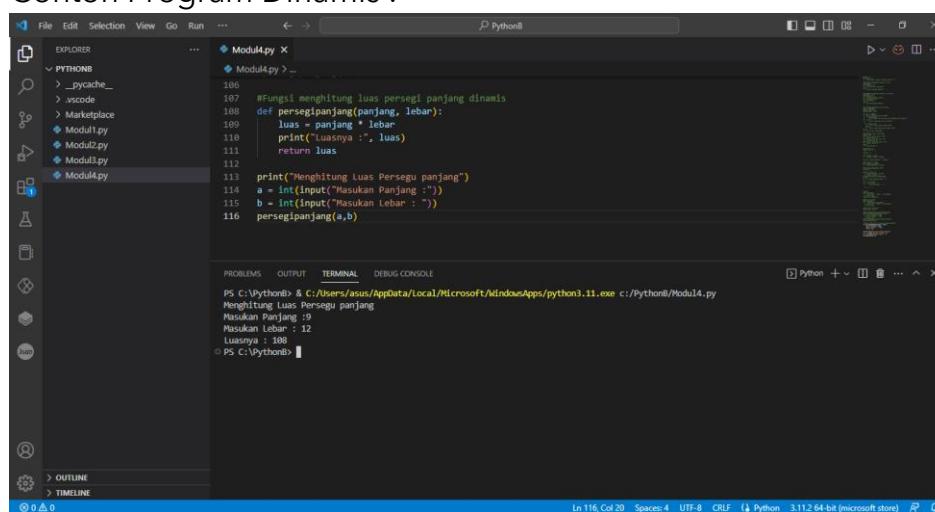
A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: Modul1.py, Modul2.py, Modul3.py, and Modul4.py. The main editor window displays a Python script named Modul4.py. The code contains a function definition:

```
#Fungsi menghitung luas persegi panjang
def persegipanjang(panjang, lebar):
    luas = panjang * lebar
    print("Luasnya :",luas)
    return luas
print("Menghitung luas persegi panjang")
persegipanjang(4,6)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\PythonB & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Menghitung luas persegi panjang
Luasnya : 24
○ PS C:\PythonB>
```

Contoh Program Dinamis :



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: Modul1.py, Modul2.py, Modul3.py, and Modul4.py. The main editor window displays a Python script named Modul4.py. The code contains a function definition:

```
#Fungsi menghitung luas persegi panjang dinamis
def persegipanjang(panjang, lebar):
    luas = panjang * lebar
    print("Luasnya :",luas)
    return luas
print("Menghitung Luas Persegi panjang")
a = int(input("Masukan Panjang :"))
b = int(input("Masukan Lebar :"))
persegipanjang(a,b)
```

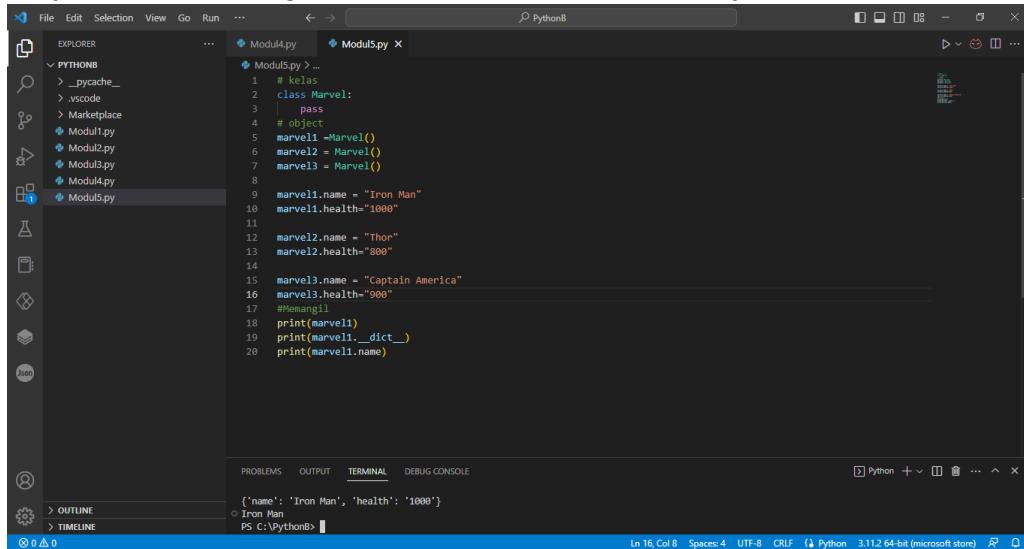
The terminal at the bottom shows the output of running the script, demonstrating user input:

```
PS C:\PythonB & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Menghitung Luas Persegi panjang
Masukan Panjang :9
Masukan Lebar :12
Luasnya : 108
○ PS C:\PythonB>
```

PYTHON MODUL 5

A. Object Oriented Programming

Object Oriented Programming (OOP) merupakan suatu konsep pemrograman yang menekankan pada paradigma atau cara pandang terhadap suatu masalah berdasarkan object. Istilah dalam OOP yaitu sebagai berikut : Kelas, Variabel, Data member. Overloading, Overloading Operator, Variabel, Pewarisan/Inheritansi (Inheritance), Instance, Instansiasi , Metode, Objek. Contoh Program Perkenalan Kelas dan Object



The screenshot shows the Microsoft Visual Studio Code interface with the Python extension installed. The Explorer sidebar shows files like Modul4.py, Modul5.py, and several __init__.py files. The code editor displays the following Python script:

```
# Kelas
class Marvel:
    pass
# object
marvel1 = Marvel()
marvel2 = Marvel()
marvel3 = Marvel()

marvel1.name = "Iron Man"
marvel1.health = "1000"

marvel2.name = "Thor"
marvel2.health = "800"

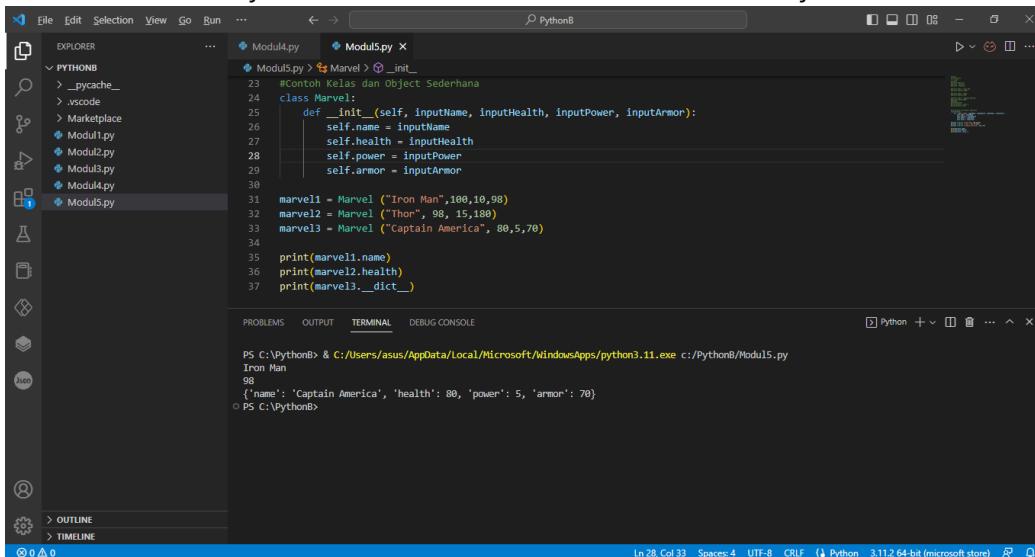
marvel3.name = "Captain America"
marvel3.health = "900"

# Memanggil
print(marvel1)
print(marvel1.__dict__)
print(marvel1.name)
```

The terminal below shows the output of running the script:

```
{'name': 'Iron Man', 'health': '1000'}
Iron Man
PS C:\PythonB>
```

1. Kelas dan Object Sederhana. Contoh Kelas dan Object Sederhana



The screenshot shows the Microsoft Visual Studio Code interface with the Python extension installed. The Explorer sidebar shows files like Modul4.py, Modul5.py, and several __init__.py files. The code editor displays the following Python script:

```
# Contoh Kelas dan Object Sederhana
class Marvel:
    def __init__(self, inputName, inputHealth, inputPower, inputArmor):
        self.name = inputName
        self.health = inputHealth
        self.power = inputPower
        self.armor = inputArmor

    marvel1 = Marvel("Iron Man", 100, 10, 98)
    marvel2 = Marvel("Thor", 98, 15, 180)
    marvel3 = Marvel("Captain America", 80, 5, 70)

print(marvel1.name)
print(marvel1.health)
print(marvel1.__dict__)
```

The terminal below shows the output of running the script:

```
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul5.py
Iron Man
98
{'name': 'Captain America', 'health': 80, 'power': 5, 'armor': 70}
PS C:\PythonB>
```

2. Variabel Kelas dan Object. Contoh program Variabel Kelas dan Object

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, and Modul5.py. The current file is Modul5.py. The code defines a class Marvel with class variables and instance variables. It creates multiple instances of the class and prints their details.

```
40 #Variabel Kelas dan Object
41 class Marvel:
42     # Class Variable
43     jumlah = 0
44     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
45         # instance variable
46         self.name = inputName
47         self.health = inputHealth
48         self.power = inputPower
49         self.armor = inputArmor
50         Marvel.jumlah += 1
51         print("Hero Marvel dengan nama : " + inputName)
52
53 marvel1 = Marvel ("Iron Man", 1000, 900, 800)
54 print(Marvel.jumlah)
55 marvel2 = Marvel ("Thor", 900, 1800, 900)
56 print(Marvel.jumlah)
57 marvel3 = Marvel ("Captain America", 800, 700, 600)
58 print(Marvel.jumlah)
```

The terminal output shows:

```
PS C:\PythonB> & c:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul5.py
Hero Marvel dengan nama : Iron Man
Hero Marvel dengan nama : Thor
Hero Marvel dengan nama : Captain America
3
PS C:\PythonB>
```

3. Method. Contoh Program Method

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, and Modul5.py. The current file is Modul5.py. The code defines a class Marvel with methods like __init__, siapa, healthTambah, getHealth, and a void function. It demonstrates calling these methods and printing their results.

```
60 #METHOD
61 class Marvel:
62     # instance variable
63     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
64         self.health = inputHealth
65         self.name = inputName
66         self.power = inputPower
67         self.armor = inputArmor
68     # void function, method tanpa return
69     def siapa (self):
70         print("Namaku adalah: " + self.name)
71     #method dengan argumen
72     def healthTambah (self, tambah):
73         self.health+tambah
74     #method dengan return.
75     def getHealth (self):
76         return self.health
77
78 marvel1 = Marvel ("Iron Man", 1000, 900, 800)
79 marvel2 = Marvel ("Thor", 900, 1800, 900)
80 marvel3 = Marvel ("Iron Man", 800, 700, 600)
81 #pemanggilan method
82 marvel1.siapa()
83 #pemakaian method dengan argumen marvel1.healthTambah(10) print(marvel1.health)
#mengembalikan nilai dengan method print(marvel1.getHealth())
```

The terminal output shows:

```
PS C:\PythonB> & c:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul5.py
Namaku adalah: Iron Man
PS C:\PythonB>
```

4. Game dengan OOP

Contoh program game dengan OOP :

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, and Modul5.py. The current file is Modul5.py. The code defines a class Marvel with methods for attacking and being attacked. It creates instances of Iron Man and Thor, and simulates a fight between them.

```
85 #Game dengan OOP
86 class Marvel:
87     def __init__(self, name, health, attackPower, armorNumber):
88         self.name = name
89         self.health = health
90         self.attackPower = attackPower
91         self.armorNumber = armorNumber
92     def serang (self, lawan):
93         print(self.name + " menyerang " + lawan.name)
94         lawan.serang (self, self.attackPower)
95     def diserang (self, lawan, attackPower_lawan):
96         print(lawan.name + " diserang " + self.name)
97         self.health -= attackPower_lawan
98         print("Darah " + self.name + " tersisa " + str(self.health))
99         self.health -= attackPower_lawan
100        print("Serangan terasa : " + str(attackPower_lawan))
101 ironman = Marvel ("Iron Man",100,10,5)
102 thor = Marvel ("Thor", 95,15,10)
103 #ironman.serang()
104 ironman.serang (thor)
105 #print("\n")
106 #ironman.serang (thor)
107 #print("\n") #thor.serang (ironman)
```

The terminal output shows:

```
PS C:\PythonB> & c:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul5.py
Iron Man menyerang Thor
Thodiserang Iron Man
Darah Thor tersisa 95
Serangan terasa : 10
PS C:\PythonB>
```

MODUL 6

PYTHON (FOR DATA SCIENCE)

A. Data Science

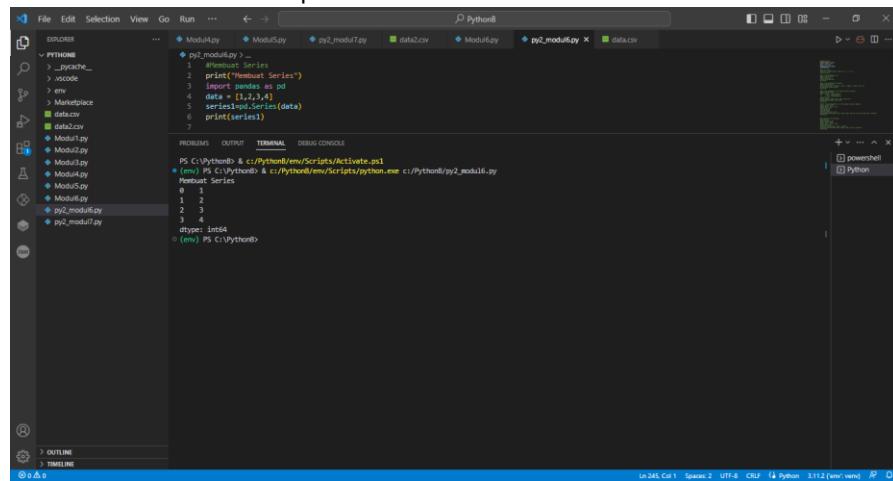
Adalah ilmu yang secara khusus mempelajari soal data terutama data kuantitatif atau data numerik. Secara umum data science adalah penggalian atau bisa juga disebut mengekstrak data agar dapat difilter serta didapatkan data yang benar untuk menghasilkan produk data yang sebenar-benarnya.

B. Pandas DataFrame

1. Pandas

Berasal dari kata python data analysis library mendukung data multi-dimensi yang artinya menggunakan 2 buah index. Sedangkan data satu dimensi adalah elemen pada data dapat diakses hanya dengan 1 buah index.

- 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



```
Python8

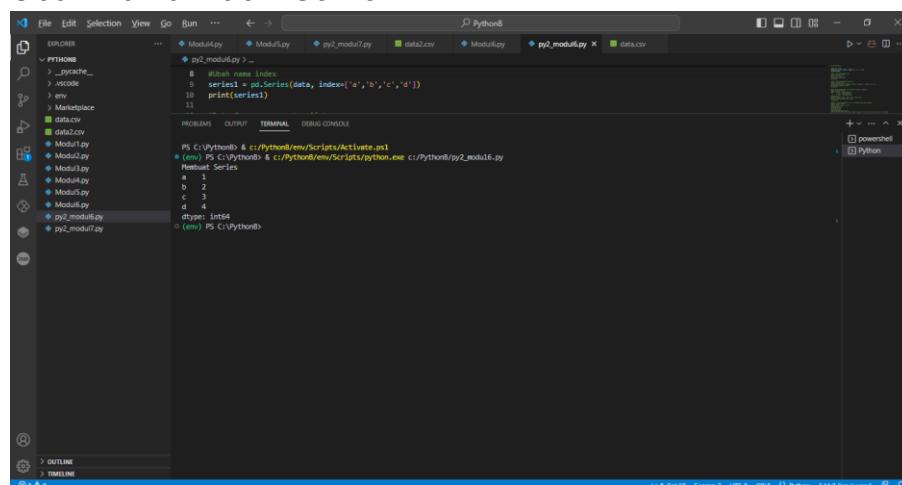
1 # membuat Series
2 print("membuat Series")
3 import pandas as pd
4 data = [1,2,3,4]
5 series1=pd.Series(data)
6 print(series1)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
```

PS C:\Python8\ & c:/Python8/Scripts/Activate.ps1
[enter] PS C:\Python8\ & c:/Python8/Scripts/python.exe c:/Python8/py2_modul6.py

```
1
2
3
4
dtype: int64
(enter) PS C:\Python8\
```

- Ubah Nama Index. Contoh



```
Python8

1 # membuat Series
2 print("membuat Series")
3 series1 = pd.Series(data, index=['a','b','c','d'])
4 print(series1)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
```

PS C:\Python8\ & c:/Python8/Scripts/Activate.ps1
[enter] PS C:\Python8\ & c:/Python8/Scripts/python.exe c:/Python8/py2_modul6.py

```
a    1
b    2
c    3
d    4
dtype: int64
(enter) PS C:\Python8\
```

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

d) Contoh DataFrame Menggunakan Dictionary

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

- File Explorer:** Shows a folder named "PYTHON" containing files: "Modul1.py", "Modul2.py", "py2_modul17.py", "data1.csv", "Modul3.py", "py2_modul8.py", and "data2.csv".
- Editor:** Displays the content of "py2_modul8.py". The code uses the pandas library to create a dictionary from a CSV file and prints it.
- Terminal:** Shows the command PS C:\Python38\ & c:/Python38/ems/Scripts/Activate.ps1 followed by the output of the script execution. The output shows a dictionary with items: Rambutan: 11, Nanas: 4, Anggur: 7.
- Output:** Shows the results of the script execution.
- Problems:** Shows no errors or warnings.
- Terminal:** Shows the command PS C:\Python38\ & c:/Python38/ems/Scripts/Activate.ps1 followed by the output of the script execution. The output shows a dictionary with items: Rambutan: 11, Nanas: 4, Anggur: 7.
- Debug Console:** Shows the results of the script execution.

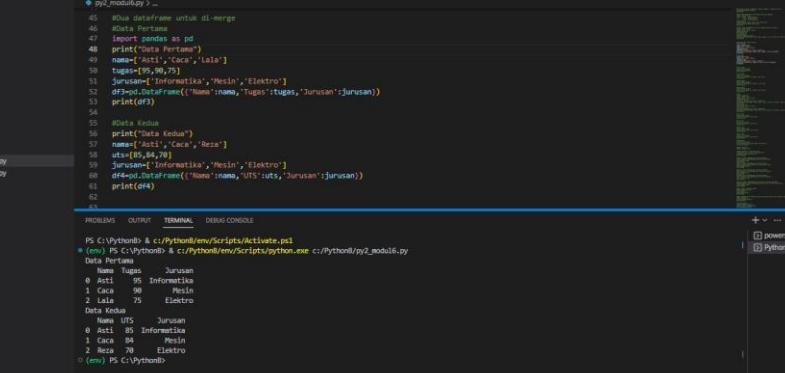
e) Contoh DataFrame Menggunakan List dengan Tipe Data Campuran

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

2. Merge, Join, & Concatenate DataFrame

Merupakan operasi penggabungan.

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



The screenshot shows a Visual Studio Code (VS Code) interface with the Python extension installed. The left sidebar displays a file tree with several Python files and a CSV file named 'data2.csv'. The main editor area contains two snippets of Python code. The first snippet reads 'data1.csv' and prints its contents as a DataFrame:

```
43 # Baca Dataframe untuk di merge
44 dfData Pertama
45 import pandas as pd
46 print("Data Pertama")
47 dfData = pd.read_csv('data1.csv')
48 dfData
49 dfData = dfData[['Nama','Tugas']]
50 tugas=[85,90,75]
51 Jurusan=['Informatika','Mesin','Elektro']
52 df=pd.DataFrame({'Nama':nama,'Tugas':tugas,'Jurusan':jurusan})
53 print(df)
54
55 dfData Kedua
56 print("Data Kedua")
57 dfData = pd.read_csv('data2.csv')
58 uts=[85,84,76]
59 Jurusan=['Informatika','Mesin','Elektro']
60 df2=pd.DataFrame({'Nama':nama,'UTS':uts,'Jurusan':jurusan})
61 print(df2)
62
```

The second snippet reads 'data2.csv' and prints its contents as a DataFrame:

```
PS C:\Python38 & c:\Python\env\Scripts\activate.ps1
(echo 1 > data1.csv) & python.exe c:/Python38/py2_module.py
Data Pertama Jurusan
0 Asti 95 Informatika
1 Caca 84 Mesin
2 Lala 75 Elektro
Data Kedua
Nama Tugas Jurusan
0 Asti 85 Informatika
1 Caca 84 Mesin
2 Reza 76 Elektro

```

b) Contoh Inner 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_modul6.py, and py2_modul7.py. The terminal window displays the following Python code and its output:

```
PS C:\Python38 & c:/Python38/Scripts/Activate.ps1
* (env) PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Inner Merge
Name Tugs Jurusan_ITS Jurusan_y
0 Asti 95 Informatika 85 Informatika
1 Caca 98 Mesin 84.0 Mesin
2 Ida 75 Elektro NaN NaN
* (env) PS C:\Python38
```

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_modul6.py, and py2_modul7.py. The terminal window displays the following Python code and its output:

```
PS C:\Python38 & c:/Python38/Scripts/Activate.ps1
* (env) PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Left Merge
Name Tugs Jurusan_x Jurusan_y
0 Asti 95 Informatika 85 Informatika
1 Caca 98 Mesin 84.0 Mesin
2 Ida 75 Elektro NaN NaN
* (env) PS C:\Python38
```

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_modul6.py, and py2_modul7.py. The terminal window displays the following Python code and its output:

```
PS C:\Python38 & c:/Python38/Scripts/Activate.ps1
* (env) PS C:\Python38 & c:/Python38/Scripts/python.exe c:/Python38/py2_modul6.py
Right Merge
Name Tugs Jurusan_x Jurusan_y
0 Asti 95.0 Informatika 85 Informatika
1 Caca 98.0 Mesin 84.0 Mesin
2 Ida NaN Elektro NaN NaN
* (env) PS C:\Python38
```

e) Contoh Outer Merge

The screenshot shows the Visual Studio Code interface with the Python extension installed. The Explorer sidebar on the left lists files: py2_modul7.py, Modul8.py, Modul9.py, py2_modul7.py, data2.csv, Modul8.py, py2_modul8.py, and data.csv. The terminal at the bottom shows the following session:

```
PS C:\Python36 & c:/Python36/Scripts/activate.ps1
(Env) PS C:\Python36 & c:/Python36/env/Scripts/python -c "Python8/py2_modul8.py
Outer Merge
Name Tugas Jurusan_x LTS Jurusan_y
0 Astuti 80.0 Mesin 84.0 Mesin
1 Caca 90.0 Mesin 94.0 Mesin
2 Laia 75.0 Elektro NaN NaN
3 Reza NaN Elektro 76.0 Elektro
(Env) PS C:\Python36
```

3. Join

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

a) Data join sebagai berikut :

b) Contoh Inner Join

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

- File Explorer:** Shows files in the current workspace, including `Modul1.py`, `Modul1Say`, `py2_modul7.py`, `data2.csv`, `Modul6.py`, `py2_modul6.py`, and `data.csv`.
- Terminal:** Displays the output of running Python scripts. The terminal window shows:
 - Execution of `Activate.ps1` in the `C:\Python2\env\Scripts` directory.
 - Execution of `py2_modul16.py` in the `C:\Python2` directory, displaying student data:

| Nama | Tugas | Jurusan | Nama B | UTS | Jurusan B |
|------|-------|---------|---------|------|-----------|
| L2 | Caca | 98 | Mesin | Asti | 85 |
| L3 | Pj | 92 | Elektro | Caca | BA |
| | | | | | Mesin |

- Output:** Shows the standard output of the executed Python script.
- Problems:** Shows no errors or warnings.
- Terminal Tab:** Active tab in the bottom navigation bar.
- Bottom Status Bar:** Shows the current file is `py2_modul6.py`, the line is 100, column is 1, and the status is `[In 100, Col 1]`.

c) Contoh Left Join

The screenshot shows a VS Code interface with a Python file named `py2_modul6.py` open. The code performs a left join between two CSV files, `data1.csv` and `data2.csv`, using the `left=df1.join(df2, how='left')` command. The output in the terminal shows the resulting data frame.

```
#Left Join
left=df1.join(df2, how='left')
print(left)
```

```
Name Tugas Jurusan Nama B UTS Jurusan B
L1 Asti 95 Informatika NaN NaN
L2 Caca 98.0 Mesin Asti 85.0 Informatika
L3 Reza 75 Elektro Caca 94.0 Mesin
L4 Nahilah Nahil Nahil Bima 78 Mesin
```

d) Contoh Right Join

The screenshot shows a VS Code interface with a Python file named `py2_modul6.py` open. The code performs a right join between two CSV files, `data1.csv` and `data2.csv`, using the `right=df2.join(df1, how='right')` command. The output in the terminal shows the resulting data frame.

```
#Right Join
right=df2.join(df1, how='right')
print(right)
```

```
Name Tugas Jurusan Nama B UTS Jurusan B
L1 Asti 95 Informatika NaN NaN
L2 Caca 98.0 Mesin Asti 85 Informatika
L3 Reza 75 Elektro Caca 94 Mesin
L4 Nahilah Nahil Nahil Bima 78 Mesin
```

e) Contoh Outer Join

The screenshot shows a VS Code interface with a Python file named `py2_modul6.py` open. The code performs an outer join between two CSV files, `data1.csv` and `data2.csv`, using the `outer=df2.join(df1, how='outer')` command. The output in the terminal shows the resulting data frame.

```
#Outer Join
outer=df2.join(df1, how='outer')
print(outer)
```

```
Name Tugas Jurusan Nama B UTS Jurusan B
L1 Asti 95.0 Informatika NaN NaN
L2 Caca 98.0 Mesin Asti 85.0 Informatika
L3 Reza 75.0 Elektro Caca 94.0 Mesin
L4 Nahilah Nahil Nahil Bima 78.0 Mesin
```

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

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

- File Explorer:** Shows a file tree with a folder named "PYTHON" containing "ModulA.py", "ModulB.py", "py2_modul7.py", "data2.csv", and "ModulC.py".
- Terminal:** The terminal window displays Python code for concatenating two CSV files:

```
py2_modul7.py ...
122 #CONCATINATE
123 df3 = pd.read_csv("data2.csv")
124 df4 = pd.read_csv("data2.csv")
125 C=df3.concat([df3,df4], sort=False)
126 print(C)
127
```

The terminal also shows the command `=(env) PS C:\Python\`.

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

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

- File Explorer:** Shows files like `__pycache__`, `data.csv`, and `py2_mod16.py`.
- Code Editor:** Displays Python code for reading a CSV file into a DataFrame.
- Terminal:** Shows the command `PS C:\Python38 & c:\Python\env\Scripts\Activates.ps1` followed by the output of a script named `py2_mod16.py`. The output lists YouTube channel statistics:

| Rank | Nama Youtuber | Jenis | Kategori | Subscribe |
|------|----------------|-------|-------------|-----------|
| 0 | Redit Dika | L | Komedি | 7000000 |
| 1 | Syuraini Prima | P | Daili | 1200000 |
| 2 | Arief Muhamad | L | Daily Vlog | 3000000 |
| 3 | Anisa Aziza | P | Food Travel | 600000 |
| 4 | Sarah Vilash | P | game | 2000000 |
| 5 | Amelia ARI | L | Komedি | 900000 |
| 6 | Chendre Liow | L | Sketsa | 300000 |

- ### b) Macam - Macam Operasi

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

- 1) Fungsi head(). Contoh penerapannya

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

- File Structure (EXPLORER):** Shows a project named "Modul16" containing files like "Modul16.py", "Modul16_Solusi.py", "py2_modul16.py", "data.csv", and "Modul16_Solusi.py".
- Code Editor:** Displays Python code for reading CSV files using pandas. The code includes:
 - Importing pandas as pd.
 - Reading "data.csv" into a DataFrame named "head".
 - Printing the first 5 rows of "head" using ".head(5)".
 - Printing the first 2 rows of "head" using ".head(2)".
- Terminal:** Shows the command "python3 py2_modul16.py" being run, followed by the output:

| No | Name_YouTuber | Jenis_Kelamin | Umur | Kategori | Subscribe |
|----|---------------|---------------|------|-------------|-----------|
| 1 | Radiq Dika | L | 24 | Dewasa | 7000000 |
| 2 | Statuan Prod | L | 29 | Daily Vlog | 120000 |
| 3 | Aristi Andri | L | 26 | Daily Vlog | 3000000 |
| 4 | Aldila Aldila | P | 25 | Food Travel | 6000000 |
| 5 | Sarah Vilid | P | 23 | Gamer | 2000000 |
- Status Bar:** Shows "PS C:\Python3\ & C:\Python3\venv\Scripts\python.exe -u teste.py".

2) Fungsi tail(), Contoh penerapannya

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

- File Structure (EXPLORER):** Shows a project named "PYTHON" with files like "Modul1.py", "Modul2.py", "Modul3.py", "Modul4.py", "Modul5.py", "Modul6.py", "Modul7.py", "Modul8.py", "py2_modul8.py", and "data.csv".
- Code Editor:** Displays Python code for reading CSV files and printing their contents. The code uses the `csv` module to handle the CSV data.
- TERMINAL:** Shows the command line output of running the script, which prints the contents of the "data.csv" file.
- PROBLEMS:** Shows no errors or warnings.
- STATUS BAR:** Shows the current file is "py2_modul8.py", and the bottom right corner shows "Line 151 Col 12 Spaces: 2 Left: 4880 Right: 3112 (new term)".

3) Fungsi shape, contoh penerapannya

The screenshot shows the Visual Studio Code interface with the Python extension installed. The Explorer sidebar on the left lists files like `Module1.py`, `Module1.ipynb`, `Module1.ipynb`, `data.csv`, and `py2_modul1.py`. The terminal at the bottom displays the output of running `py2_modul1.py`, which includes importing `pandas`, defining a sample DataFrame, and printing its shape. The status bar at the bottom right indicates the file is 16151 bytes large, has 3 changes, and is 100% typed.

```
PS C:\Python> & c:/Python/env/Scripts/Activate.ps1
(mv) PS C:\Python> & c:/Python/env/Scripts/python.exe c:/Python/py2_mod16.py
Operasi Shape() Menampilkan Jumlah Baris dan Kolom DataFrame
(7, 3)
(mv) PS C:\Python>
```

4) Contoh Mean, Median, Standar Deviasi

5) Contoh Max, Min, Count

The screenshot shows the VS Code interface with the Python extension installed. The code in the editor demonstrates how to use the `max()`, `min()`, and `count()` methods on lists and data frames.

```
... + py2_modul6.py ... + Modul8py + Modul5py + py2_modul7.py + data2.csv + Modul8py + py2_modul8.py + data.csv
File Edit Selection View Go Run ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
...
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())
178 print("MIN untuk mencari nilai terendah : ")
179 print(min)
180 print(min())
181 print("COUNT untuk penjumlahan non null record pada setiap kolom : ")
182 print(count)
183 print("Count untuk penjumlahan non null record pada setiap kolom : ")
184 print(count())
185 print(count())
186 print(COUNT)

MAX untuk mencari nilai tertinggi :
Name_YouTuber      Statemen_Profil
Jenis_Kelamin          P
Umur                  29
Kategori            Sketsa
Subscribe        7000000
dtype: object

MIN untuk mencari nilai terendah :
Name_YouTuber      Anisa_Aisha
Jenis_Kelamin          F
Umur                  23
Kategori       Daily Vlog
Subscribe        1200000
dtype: object

COUNT untuk penjumlahan non null record pada setiap kolom :
Name_YouTuber      Statemen_Profil
Jenis_Kelamin          P
Umur                  7
Kategori            Sketsa
Subscribe        7
dtype: int64
= (env) PS C:\Python8
```

6) Contoh describe()

The screenshot shows the VS Code interface with the Python extension installed. The code in the editor demonstrates how to use the `describe()` method to generate a descriptive statistics report for a DataFrame.

```
... + py2_modul8.py ... + Modul8py + Modul5py + py2_modul7.py + data2.csv + Modul8py + py2_modul8.py + data.csv
File Edit Selection View Go Run ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
...
PS C:\Python8& c:/python/env/Scripts/activate.ps1
* (env) PS C:\Python8 & c:/Python/env/Scripts/python.exe c:/Python8/py2_modul8.py
DESCRIBE
DESCRIBE
User      Subscrive
count    7.000000e+00
mean   27.857143 2.300000e+06
std     2.826546 2.346314e+06
min    21.000000 1.000000e+05
25K    25.500000 7.000000e+05
50K    28.400000 2.000000e+06
75K    32.000000 2.000000e+06
max   34.000000 7.000000e+06
= (env) PS C:\Python8
```

7) Contoh Rename dan Drop Kolom

The screenshot shows the VS Code interface with the Python extension installed. The code in the editor demonstrates how to use the `rename()` and `drop()` methods to rename and drop specific columns from a DataFrame.

```
... + py2_modul8.py ... + Modul8py + Modul5py + py2_modul7.py + data2.csv + Modul8py + py2_modul8.py + data.csv
File Edit Selection View Go Run ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
...
193 #RENAME KOLOM
194 print("RENAME")
195 df2_modul8.rename(columns={'Name_YouTuber':'Youtuber'}) 
196 print(RENAME)
197 # DROP KOLOM
198 df2_modul8.drop(columns='Jenis_Kelamin')
199 print(DROP)
200 print(DROP)
201
202 #iloc Mengambil record 1 kolom (umur yang berada di index ke-2)
203 #print(ILOC)
204 #iloc[0]=sample.iloc[1,2]
205 #print(ILOC)
206
207 #iloc Mengambil 3 record pertama dari kolom (umur)
208 #print(ILOC)
209 #iloc[0:3]=sample.iloc[0:3,2]
210 #print(ILOC)
211
212 #iloc Mengambil semua baris dalam kolom
213 #print(ILOC)
214 #iloc[0]=sample.iloc[:,1]
215 #print(ILOC)
216
217 #iloc Mengambil data dari Record ke-3 dalam kolom ke-2
218 #print(ILOC)
219 #iloc[0]=sample.iloc[2,1]
220
221 #RENAME
222 Youtuber  Jenis_Kelamin  Umur  Kategori  Subscrive
0 Raditia_Gita  L  34  Komedi  7000000
1 Stateman_Profil  P  29  Daily Vlog  1200000
2 Joko_Widodo  P  35  Food Review  4000000
4 Sarah_Wijaya  P  23  Komedi  2000000
5 Gendra_Like  L  30  Komedi  8000000
6 Dwi_Rahayu  L  26  Sketsa  3000000
0009
= (env) PS C:\Python8& c:/python/env/Scripts/activate.ps1
* (env) PS C:\Python8 & c:/Python/env/Scripts/python.exe c:/Python8/py2_modul8.py
RENAME
Youtuber  Jenis_Kelamin  Umur  Kategori  Subscrive
0  Raditia_Gita  L  34  Komedi  7000000
1  Stateman_Profil  P  29  Daily Vlog  1200000
2  Joko_Widodo  P  35  Food Review  4000000
4  Sarah_Wijaya  P  23  Komedi  2000000
5  Gendra_Like  L  30  Komedi  8000000
6  Dwi_Rahayu  L  26  Sketsa  3000000
0009
```

8) Contoh iloc

The screenshot shows the PyCharm IDE interface with several files listed in the left sidebar, including `py2_modul5.py`, `Modul5.py`, `py2_modul7.py`, `data.csv`, and `Modul6.py`. The main code editor contains the following Python script:

```
203     #iloc[0]: Mengambil record 1 kolom (urut yang berada di index ke-0) iloc[0]: Mengambil record 1 kolom (urut yang berada di index ke-0)
204     print("iloc[0]: Mengambil record 1 kolom (urut yang berada di index ke-0)\n")
205     print(data[0])
206
207     print("iloc[3]: Mengambil 3 record pertama dari kolom (urut)\") iloc[3]: Mengambil 3 record pertama dari kolom (urut)\n")
208     print(data[0:3])
209
210     print("iloc[1:3]: Mengambil 2 record pertama dari kolom (urut)\") iloc[1:3]: Mengambil 2 record pertama dari kolom (urut)\n")
211     print(data[1:3])
212
213     print("iloc[1:3,1]: Mengambil semua baris pada kolom 1\n")
214     print(data[1:3,1])
215
216     print("iloc[3,1]: Mengambil data dari record ke-3 dalam kolom ke-2\") iloc[3,1]: Mengambil data dari record ke-3 dalam kolom ke-2\n")
217     print(data[3,1])
218
219
220
221     print("User, dtype: int64\n")
222     print("Name: User, dtype: int64\n")
223
224     print("iloc[0]: Mengambil semua baris dalam kolom\n")
225     print("Name_Youtuber Jenis_Kelamin Umur Kategori Subscribe\n")
226     print("0 Della Dika L 29 Daily Vlog 700000 1\n")
227     print("1 Sitiawati Pread L 29 Daily Vlog 1200000 1\n")
228     print("2 Arif Muhammad L 28 Daily Vlog 3000000 1\n")
229     print("3 Anisa Aziza P 25 Food Travel 6000000 1\n")
230     print("4 Sarah Vilaid P 23 Gamer 2000000 1\n")
231     print("5 Aulia Iri L 30 Komedi 8000000 1\n")
232     print("6 Chandra Liow L 26 Senstra 5000000 1\n")
233
234     print("iloc[0:3]: Mengambil data dari record ke-0 sampai ke-3 dalam kolom ke-2\n")
235     print("User Kategori Subscribe\n")
236     print("0 25 Food Travel 6000000\n")
237     print("1 25 Food Travel 6000000\n")
238     print("2 28 Daily Vlog 3000000\n")
239
240     print("iloc[0:3,1]: Mengambil semua baris dari kolom 1 dalam kolom ke-2\n")
241     print("Name_Youtuber\n")
242     print("0 Della Dika\n")
243     print("1 Sitiawati\n")
244     print("2 Arif Muhammad\n")
245     print("3 Sarah Vilaid\n")
246     print("4 Anisa Aziza\n")
247     print("5 Aulia Iri\n")
248     print("6 Chandra Liow\n")
249
250     print("Name: Name_Youtuber, dtype: object\n")
251
252     loc = data.loc[0:3,1]
253     print(loc)
254
255     print("loc[0]: Mengambil record sampai indeks ke-0 sampai ke-1 dari kolom name_youtuber\n")
256     print("Name_Youtuber\n")
257     print("0 Della Dika\n")
258     print("1 Sitiawati\n")
259     print("2 Arif Muhammad\n")
260     print("3 Sarah Vilaid\n")
261     print("4 Anisa Aziza\n")
262     print("5 Aulia Iri\n")
263     print("6 Chandra Liow\n")
264
265     print("Name: Name_Youtuber, dtype: object\n")
266
267     loc = data.loc[0:3,1:2]
268     print(loc)
269
270     print("loc[0:3,1:2]: Mengambil record sampai indeks ke-0 sampai ke-3 dari kolom name_youtuber\n")
271     print("Name_Youtuber Jenis_Kelamin Umur\n")
272     print("0 Della Dika L 29\n")
273     print("1 Sitiawati Pread L 29\n")
274     print("2 Arif Muhammad L 28\n")
275     print("3 Anisa Aziza P 25\n")
276     print("4 Sarah Vilaid P 23\n")
277     print("5 Aulia Iri L 30\n")
278     print("6 Chandra Liow L 26\n")
279
280
281
282     print("sample[\"subscribe\"] = 1\n")
283     sample["subscribe"] = 1
284
285     print("print(sample)\n")
286
287     print(sample)
```

The terminal output shows the results of the `iloc` indexing operations, such as printing specific rows or columns from the `data.csv` file.

9) Contoh loc

The screenshot shows the PyCharm IDE interface with several files listed in the left sidebar, including `py2_modul5.py`, `Modul5.py`, `py2_modul7.py`, `data.csv`, and `Modul6.py`. The main code editor contains the following Python script:

```
219     # loc
220     print("loc[0]: Mengambil record 1 value, dengan results nama kolomnya \"name_youtuber\"\n")
221     print(data.loc[0,"Name_Youtuber"])
222     print("loc[0:3]: Mengambil record 3 kolom, dengan results nama kolomnya name_youtuber\n")
223     print(data.loc[0:3])
224     print("loc[0:3,1]: Mengambil record dari indeks ke-0 sampai ke-3 dari kolom \"name_youtuber\"\n")
225     print(data.loc[0:3,1])
226     print("loc[0:3,1:2]: Mengambil record dari indeks ke-0 sampai ke-3 dari kolom \"name_youtuber\"\n")
227     print(data.loc[0:3,1:2])
228     print("loc[0:3,1:2]: Mengambil record sampai indeks ke-0 sampai ke-3 dari kolom name_youtuber\n")
229     print("Name_Youtuber\n")
230     print("0 Della Dika\n")
231     print("1 Sitiawati\n")
232     print("2 Arif Muhammad\n")
233     print("3 Sarah Vilaid\n")
234     print("4 Anisa Aziza\n")
235     print("5 Aulia Iri\n")
236     print("6 Chandra Liow\n")
237
238     print("Name: Name_Youtuber, dtype: object\n")
239
240     loc = data.loc[0:3,1]
241     print(loc)
242
243     print("loc[0]: Mengambil record sampai indeks ke-0 sampai ke-1 dari kolom name_youtuber\n")
244     print("Name_Youtuber\n")
245     print("0 Della Dika\n")
246     print("1 Sitiawati\n")
247     print("2 Arif Muhammad\n")
248     print("3 Sarah Vilaid\n")
249     print("4 Anisa Aziza\n")
250     print("5 Aulia Iri\n")
251     print("6 Chandra Liow\n")
252
253     print("Name: Name_Youtuber, dtype: object\n")
254
255     loc = data.loc[0:3,1:2]
256     print(loc)
257
258     print("loc[0:3,1:2]: Mengambil record sampai indeks ke-0 sampai ke-3 dari kolom name_youtuber\n")
259     print("Name_Youtuber Jenis_Kelamin Umur\n")
260     print("0 Della Dika L 29\n")
261     print("1 Sitiawati Pread L 29\n")
262     print("2 Arif Muhammad L 28\n")
263     print("3 Anisa Aziza P 25\n")
264     print("4 Sarah Vilaid P 23\n")
265     print("5 Aulia Iri L 30\n")
266     print("6 Chandra Liow L 26\n")
267
268
269     print("sample[\"subscribe\"] = 1\n")
270     sample["subscribe"] = 1
271
272     print("print(sample)\n")
273     print(sample)
```

The terminal output shows the results of the `loc` indexing operations, such as printing specific rows or columns from the `data.csv` file.

10) Contoh Mengisi Nilai Sama Untuk 1 Kolom

The screenshot shows the PyCharm IDE interface with several files listed in the left sidebar, including `py2_modul5.py`, `Modul5.py`, `py2_modul7.py`, `data2.csv`, and `Modul6.py`. The main code editor contains the following Python script:

```
226     # Mengisi nilai sama untuk 1 kolom
227     print("Mengisi nilai sama untuk 1 kolom\n")
228     print("sample[\"subscribe\"] = 1\n")
229     sample["subscribe"] = 1
230
231     print("print(sample)\n")
232     print(sample)
```

The terminal output shows the results of the code execution, which prints the modified `sample` DataFrame with the `subscribe` column filled with the value 1.

11) Contoh Sorting

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modu14.py, Modu15.py, py2_modu17.py, data2.csv, and Modu6.py. The terminal window displays the following Python script and its execution:

```
PS C:\Python38\& c:/Python38/env/Scripts/activate.ps1
● (env) PS C:\Python38\& c:/Python38/env/Scripts/python.exe c:/Python38/py2_modu6.py
Sorting
  Nama_Youtuber Jenis_Kelamin Umur Kategori Subscribe
1  Statwon Fronid   L   29 Daily Vlog 120000
2  Arief Puspitasari L   25 Daily Vlog 300000
3  Anisa Aziza     P   25 Food Travel 600000
4  Sarah Viloid    P   23 Gamer 2000000
0  Radita Dika     L   34 Komedi 7000000
5  Muliati IAT    L   30 Komedi 800000
6  Chandra Low    L   26 Sketsa 300000
Sorting
  Nama_Youtuber Jenis_Kelamin Umur Kategori Subscribe
6  Chandra Low    L   26 Sketsa 3000000
0  Radita Dika    L   30 Komedi 7000000
5  Muliati IAT    L   30 Komedi 8000000
4  Sarah Viloid    P   23 Gamer 2000000
3  Anisa Aziza     P   25 Food Travel 600000
1  Statwon Fronid   L   29 Daily Vlog 120000
2  Arief Puspitasari L   25 Daily Vlog 300000
● (env) PS C:\Python38\
```

12) Contoh Filter

The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like Modu14.py, Modu15.py, py2_modu17.py, data2.csv, and Modu6.py. The terminal window displays the following Python script and its execution:

```
PS C:\Python38\& c:/Python38/env/Scripts/activate.ps1
● (env) PS C:\Python38\& c:/Python38/env/Scripts/python.exe c:/Python38/py2_modu6.py
FILTER record yang umurnya lebih dari 28 dan akan menampilkan status true atau false
249 print("FILTER record yang umurnya lebih dari 28 dan akan menampilkan status true atau false ")
250 a=sample['Umur']>28
251 print(a)
252 print("")

#filter memenuhi record
253 print("Filter memenuhi record")
254 filter1=sample['Umur']>28
255 filterbaru=sample[filter1]
256 print(filterbaru)
257 print("")

#filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )
258 print("Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )")
259 print(sample[filter1])
260 print("")

#filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )
261 print("Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )")
262 print(filter1)
263 print(filter2)
264 print(filterbaru[filter1])
265 print(filterbaru)

Name: Umur, dtype: bool

Filter memenuhi record
  Nama_Youtuber Jenis_Kelamin Umur Kategori Subscribe
0  Radita Dika    L   30 Komedi 800000
1  Statwon Fronid   L   29 Daily Vlog 120000
2  Arief Puspitasari L   28 Daily Vlog 300000
3  Anisa Aziza     P   25 Food Travel 600000
4  Sarah Viloid    P   23 Gamer 2000000
5  Muliati IAT    L   34 Komedi 7000000
6  Chandra Low    L   26 Sketsa 3000000

Filter dengan dua parameter (berdasarkan yang umurnya 27 dan kategori daily Vlog )
  Nama_Youtuber Jenis_Kelamin Umur Kategori Subscribe
0  Radita Dika    L   30 Komedi 800000
1  Statwon Fronid   L   29 Daily Vlog 120000
2  Arief Puspitasari L   28 Daily Vlog 300000
3  Anisa Aziza     P   25 Food Travel 600000
4  Sarah Viloid    P   23 Gamer 2000000
5  Muliati IAT    L   34 Komedi 7000000
6  Chandra Low    L   26 Sketsa 3000000
● (env) PS C:\Python38\
```

MODUL 7

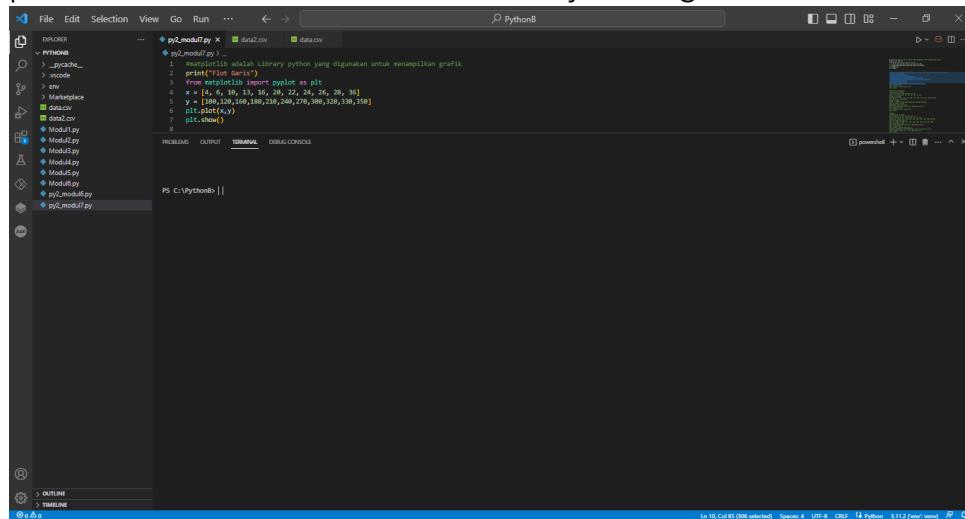
PYTHON (FOR DATA SCIENCE)

A. Matplotlib

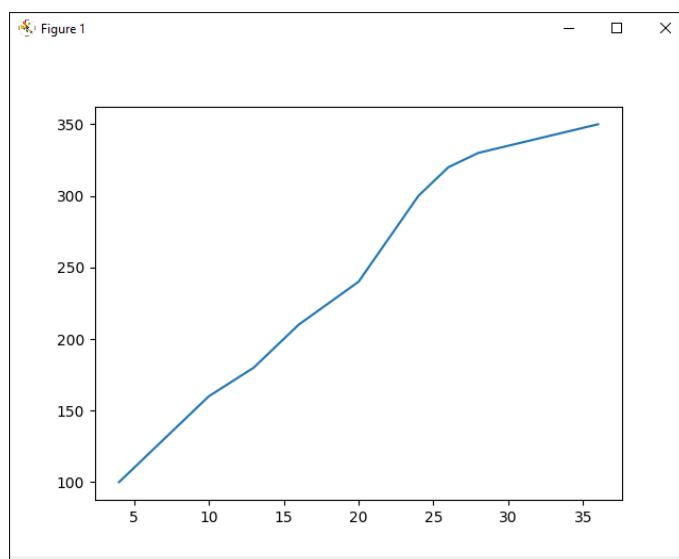
Merupakan library Python 2D yang dapat menghasilkan plot dengan kualitas tinggi dalam berbagai format yang dapat digunakan di banyak platform. Matplotlib dapat digunakan sebagai pembuat grafik dalam berbagai platform, seperti Python dan Jupyter. Grafik yang dapat dibuat beragam, seperti grafik garis,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

1. Jenis Plot Garis

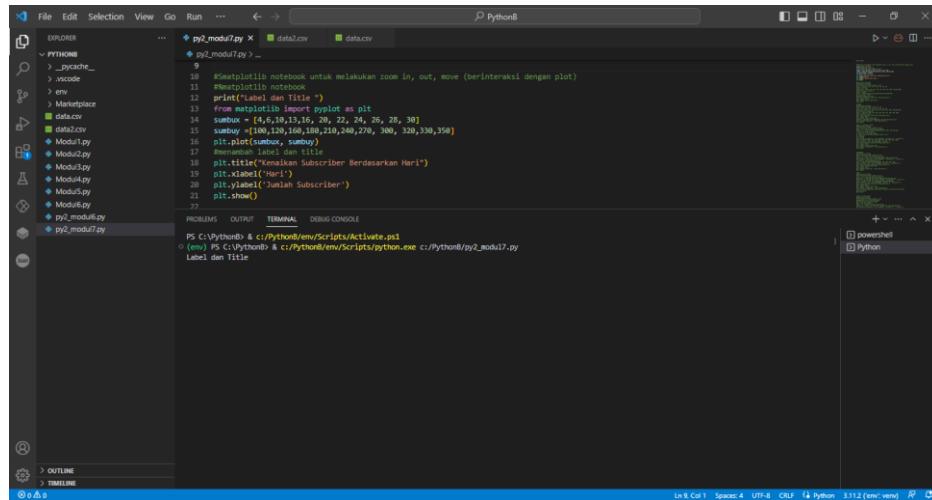
Merupakan representasi berupa garis yang menghubungkan antar posisi koordinat data. Cara membuatnya sebagai berikut.



```
File Edit Selection View Go Run ... ← → ⌘ PythonB
DIPLOER ...
PYTHON ...
py2_modul7.py x data2.csv data.csv
py2_modul7.py ...
matplotlib adalah library python yang digunakan untuk menampilkan grafik
1 print("List Garis")
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 x = [4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
7 y = [100,120,140,160,180,200,220,240,260,280,300,320,340,350]
8
9 plt.plot(x,y)
10
11 plt.show()
12
```



2. Contoh Menambah Label dan Title

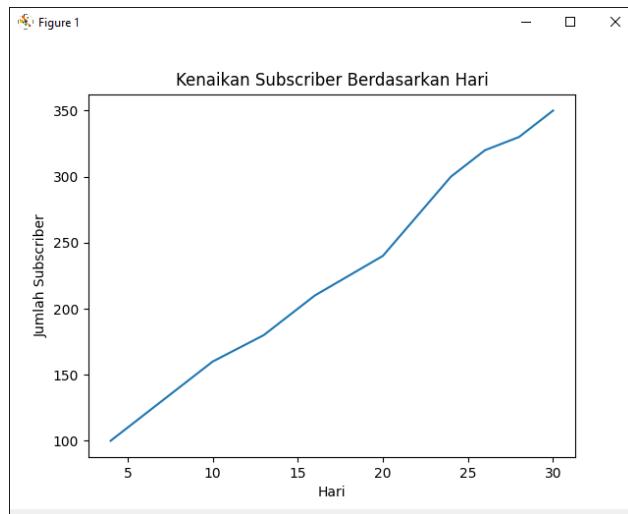


```
File Edit Selection View Go Run ... ⌘P PythonB
EXPLORER ... py2_modul7.py data2.csv data.csv
PYTHON ...
> _pycache_> vscod...
> env> Marketplace
> data.csv> data2.csv
> Modul1.py> Modul2.py
> Modul3.py> Modul4.py
> Modul5.py> Modul6.py
> py2_modul4.py> py2_modul5.py
py2_modul7.py
9 #Import library
10 #Matplotlib notebook untuk melakukan zoom in, out, move (berinteraksi dengan plot)
11 #Neatplotlib notebook
12 print("Label dan Title")
13 #Membuat sumbu x dan y
14 submx = [4,6,10,13,16,20,22,24,26,28,30]
15 sumay = [100,120,140,160,180,210,240,270,300,320,330,350]
16 plt.plot(submx,sumay)
17 #Menambahkan label dan title
18 plt.title("Kenaikan Subscriber Berdasarkan Hari")
19 plt.xlabel("Hari")
20 plt.ylabel("Jumlah Subscriber")
21 plt.show()
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

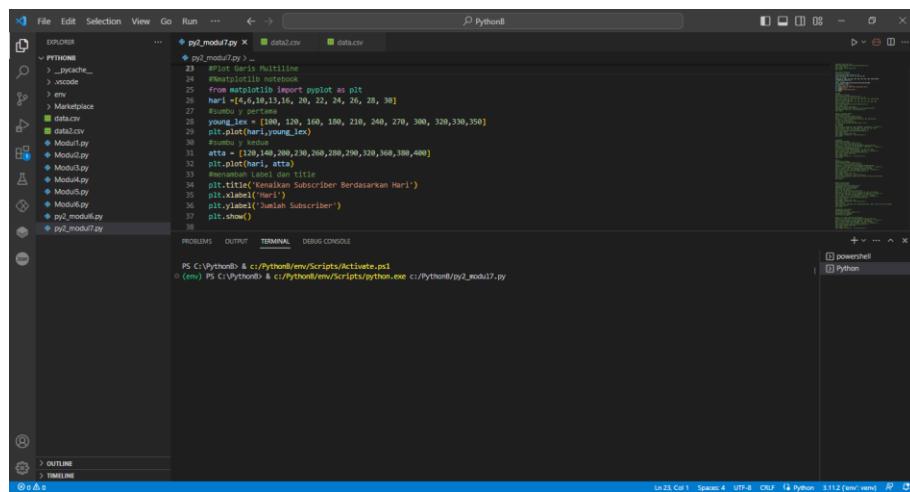
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\PythonB & c:/Python3/env/Scripts/Activate.ps1
(env) PS C:\PythonB & c:/Python3/env/Scripts/python.exe c:/PythonB/py2_modul7.py
Label dan Title

Line 23 Col 1 Space: 4 UFT-8 CRLF & Python 3.11.2 (env:venv) ⚡



3. Contoh Plot Garis Multiline

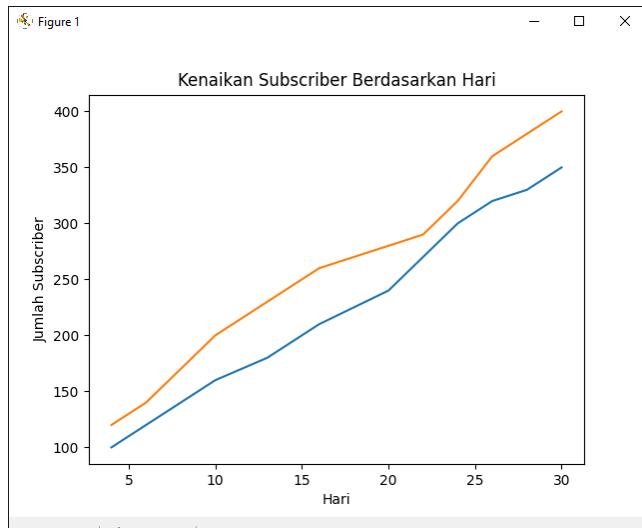


```
File Edit Selection View Go Run ... ⌘P PythonB
EXPLORER ... py2_modul7.py data2.csv data.csv
PYTHON ...
> _pycache_> vscod...
> env> Marketplace
> data.csv> data2.csv
> Modul1.py> Modul2.py
> Modul3.py> Modul4.py
> Modul5.py> Modul6.py
> py2_modul4.py> py2_modul5.py
py2_modul7.py
20 #Import library
21 #Matplotlib Multiline
22 #Neatplotlib notebook
23 from matplotlib import pyplot as plt
24 hari = [4,6,10,13,16,20,22,24,26,28,30]
25 #Membuat sumbu x dan y
26 young_ley = [100, 120, 140, 160, 180, 210, 240, 270, 300, 320, 330, 350]
27 #Membuat sumbu x dan y
28 #Menambahkan label dan title
29 plt.plot(hari,young_ley)
30 #Menambahkan label dan title
31 atta = [120,140,200,240,280,290,330,360,380,400]
32 #Menambahkan label dan title
33 #Menambahkan label dan title
34 plt.title("Kenaikan Subscriber Berdasarkan Hari")
35 plt.xlabel("Hari")
36 plt.ylabel("Jumlah Subscriber")
37 plt.show()
38
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

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

Line 23 Col 1 Space: 4 UFT-8 CRLF & Python 3.11.2 (env:venv) ⚡

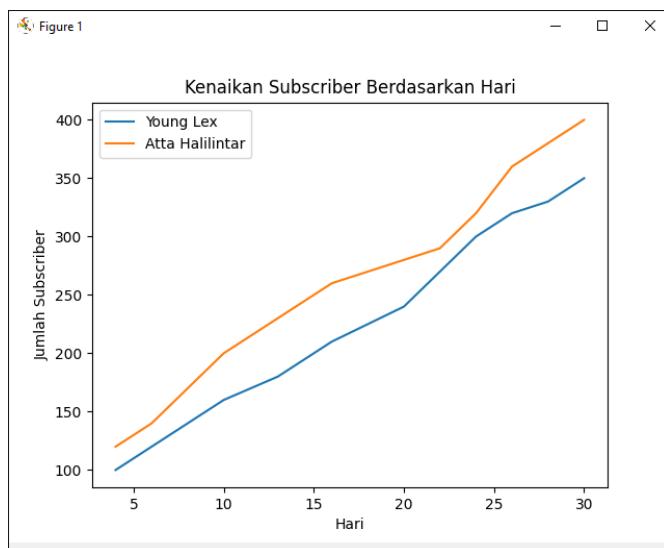


4. Contoh Legend

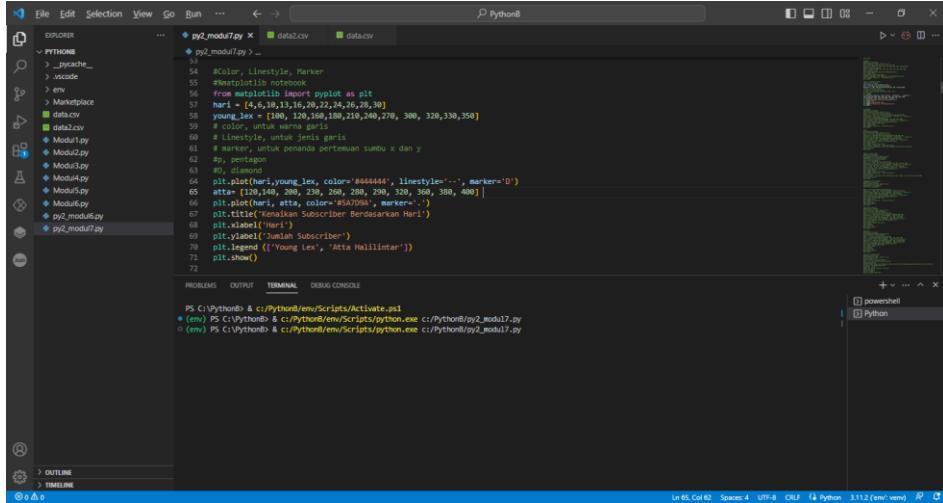
```

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>

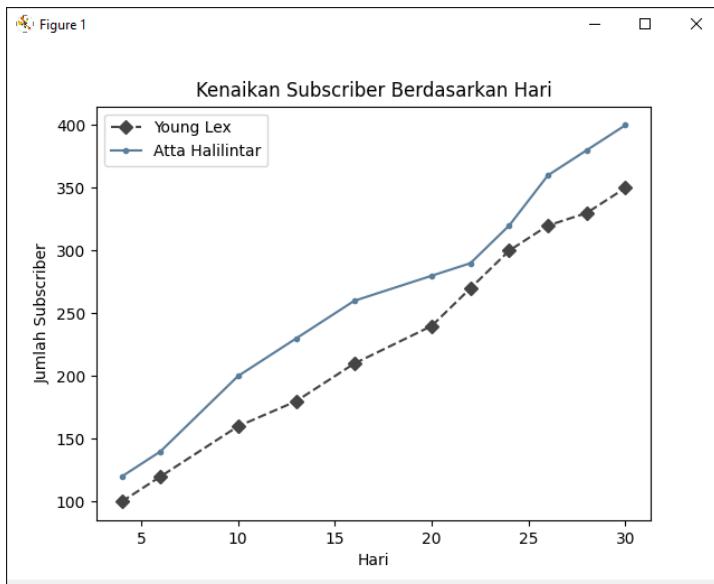
```



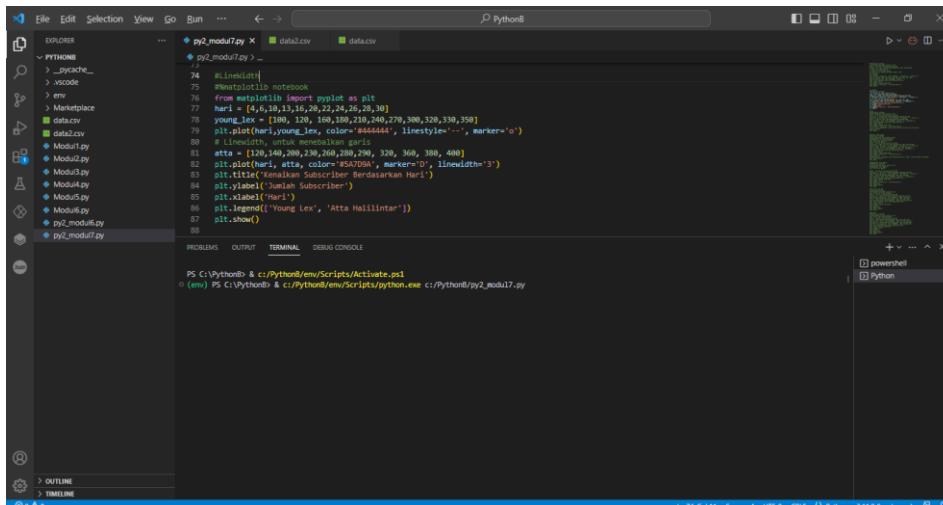
5. Contoh Color, Linestyle, Marker



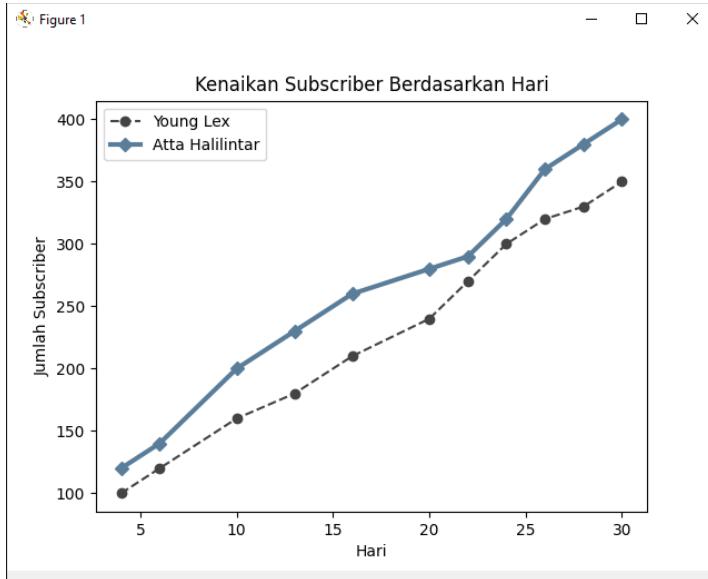
```
# Color, Linestyle, Marker
# Importing required libraries
import matplotlib.pyplot as plt
from matplotlib import colors
# Data
hari = [4,6,10,13,15,16,20,22,24,26,28,30]
young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
attha = [120, 140, 200, 230, 250, 280, 290, 320, 360, 380, 400]
# color, untuk warna garis
# marker, untuk penanda pertemuan sumbu x dan y
# Linestyle, untuk jenis garis
plt.plot(hari,young_lex, color="#444444", linestyle='--', marker='o')
plt.plot(hari,attha, color="#5A7D9A", marker='o')
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.legend(['Young Lex', 'Atta Halilintar'])
plt.show()
```



6. Contoh Linewidth



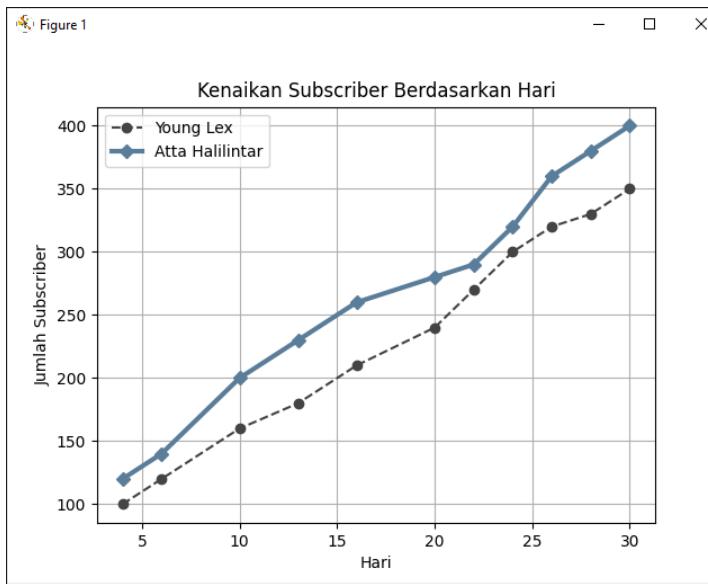
```
# Linewidth
# Importing required libraries
import matplotlib.pyplot as plt
from matplotlib import colors
# Data
hari = [4,6,10,13,15,16,20,22,24,26,28,30]
young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
attha = [120, 140, 200, 230, 260, 280, 320, 360, 380, 400]
# plot(hari,young_lex, color="#444444", linestyle='--', marker='o', linewidth=3)
# plot(hari,attha, color="#5A7D9A", marker='o', linewidth=3)
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.legend(['Young Lex', 'Atta Halilintar'])
plt.show()
```



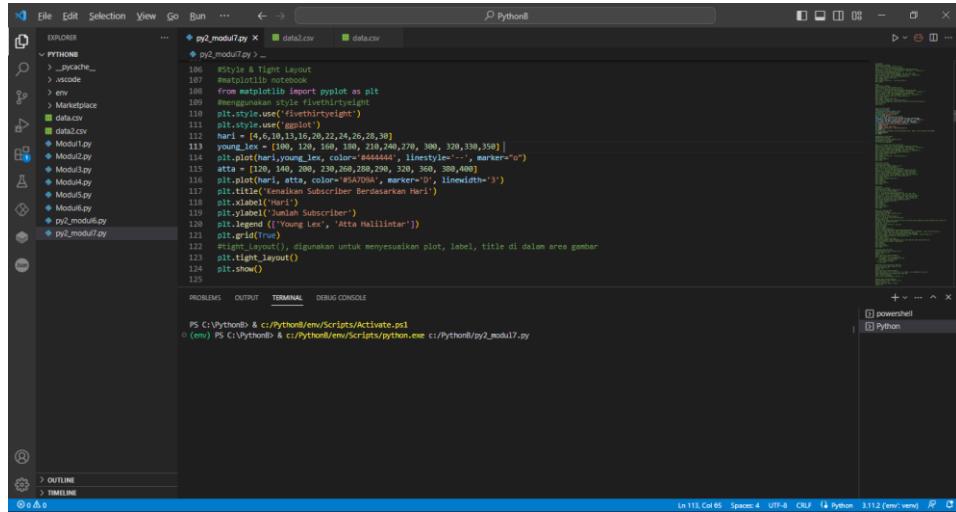
7. Contoh Grid

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

- File Explorer:** Shows files in the current workspace, including `py2_modul7.py`, `data.csv`, and `data2.csv`.
- Code Editor:** Displays the content of `py2_modul7.py`. The code uses Matplotlib to create a scatter plot of 'Young Lex' vs 'Age Hollinger' for different days of the week.
- Terminal:** Shows the command `PS C:\Python38\ & c:/Python38/env/Scripts/activate.ps1` followed by `(env) PS C:\Python38\ & c:/Python38/env/scripts/python.exe c:/Python38/py2_modul7.py`.
- Output:** Shows the execution results of the Python script, including the generated scatter plot image.
- Search:** Shows the search history with terms like `py2_modul7.py`, `data.csv`, and `data2.csv`.
- Problems:** Shows no errors or warnings.
- Timeline:** Shows the history of file changes.



8. Contoh Menggunakan Style (fivethirtyeight) dan Tight Layout

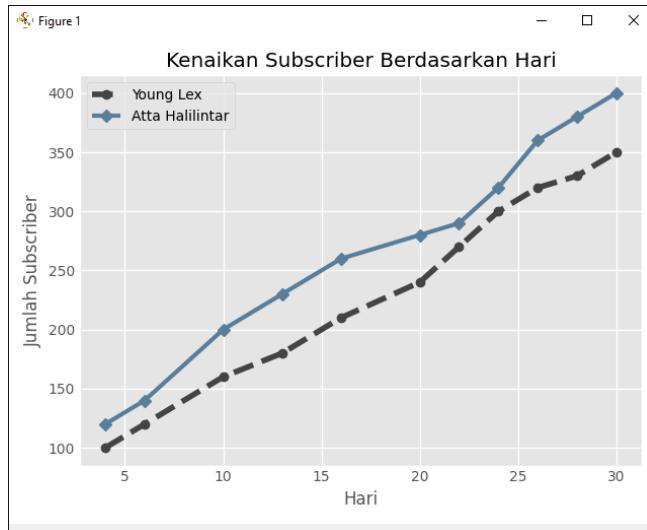


```
File Edit Selection View Go Run ... < > Python8
EXPLORER ... py2_modul7.py x data.csv data.csv
py2_modul7.py ...
106 #style & Tight Layout
107 #matplotlib notebook
108 from matplotlib import pyplot as plt
109 #menggunakan style fivethirtyeight
110 plt.style.use('fivethirtyeight')
111 plt.style.use('ggplot')
112 hari = [4,6,10,13,16,20,22,24,26,28,30]
113 young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
114 atta_halilintar = [120, 140, 200, 230, 260, 290, 320, 360, 380, 400]
115 plt.plot(hari,young_lex,color="#A7D0E4",linestyle='-',marker='o')
116 plt.plot(hari,atta,color="#A7D0E4",marker='o', linewidth=3)
117 plt.title("Kenaikan Subscriber Berdasarkan Hari")
118 plt.xlabel("Hari")
119 plt.ylabel("Jumlah Subscriber")
120 plt.legend(["Young Lex", "Atta Halilintar"])
121 plt.grid(True)
122 #tight_layout(), digunakan untuk menyesuaikan plot, label, title di dalam area gambar
123 plt.tight_layout()
124 plt.show()
125
```

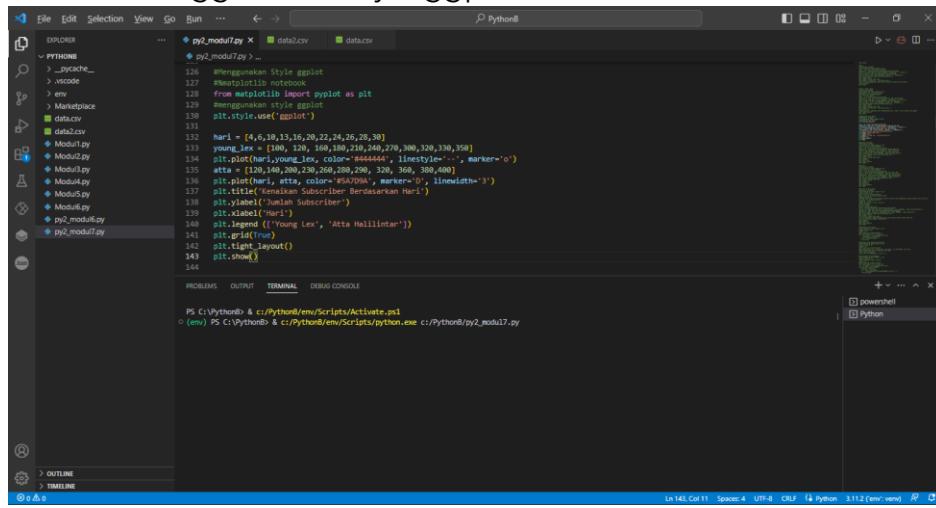
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Python8> & c:/Python8/Scripts/Activate.ps1
o (env) PS C:\Python8> & c:/Python8/env/Scripts/python.exe c:/Python8/py2_modul7.py

Line 113, Col 65 Spaces: 4 UFT-8 CRLF ↵ Python 3.11.2 (env:venv) ⌂ ⌂



9. Contoh Menggunakan Style ggplot

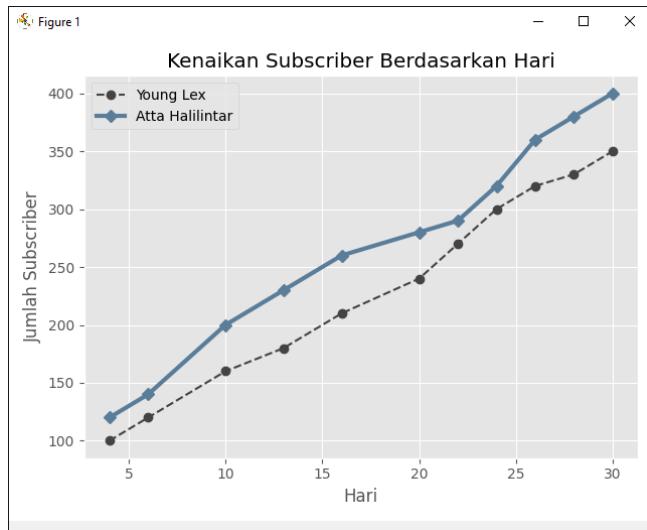


```
File Edit Selection View Go Run ... < > Python8
EXPLORER ... py2_modul7.py x data.csv data.csv
py2_modul7.py ...
126 #menggunakan Style ggplot
127 #matplotlib notebook
128 from matplotlib import pyplot as plt
129 #menggunakan style ggplot
130 plt.style.use('ggplot')
131
132 hari = [4,6,10,13,16,20,22,24,26,28,30]
133 young_lex = [100, 120, 160, 180, 210, 240, 270, 300, 320, 330, 350]
134 plt.plot(hari,young_lex,color="#A7D0E4",linestyle='--',marker='o')
135 atta = [120,140,200,230,260,290,320,360,380,400]
136 plt.plot(hari,atta,color="#A7D0E4",marker='o', linewidth=3)
137 plt.title("Kenaikan Subscriber Berdasarkan Hari")
138 plt.xlabel("Hari")
139 plt.ylabel("Jumlah Subscriber")
140 plt.legend(["Young Lex", "Atta Halilintar"])
141 plt.grid(True)
142 plt.tight_layout()
143 plt.show()
144
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Python8> & c:/Python8/Scripts/Activate.ps1
o (env) PS C:\Python8> & c:/Python8/env/Scripts/python.exe c:/Python8/py2_modul7.py

Line 143, Col 11 Spaces: 4 UFT-8 CRLF ↵ Python 3.11.2 (env:venv) ⌂ ⌂



10. Contoh Plot Garis dan Bar

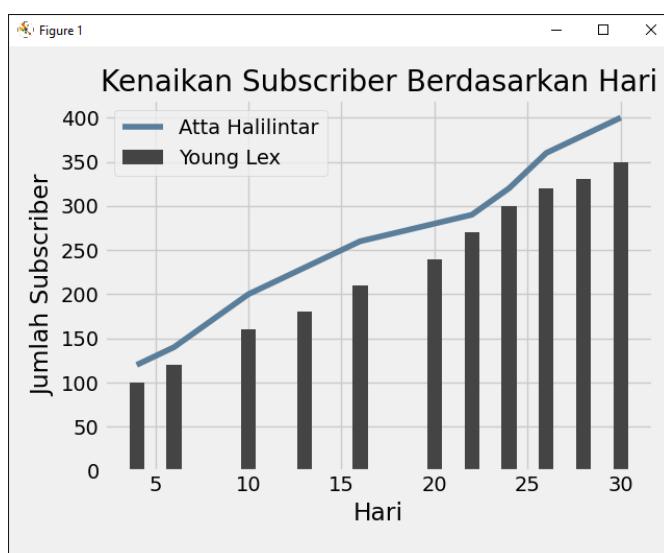
The screenshot shows the PyCharm IDE interface with the following details:

- 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.

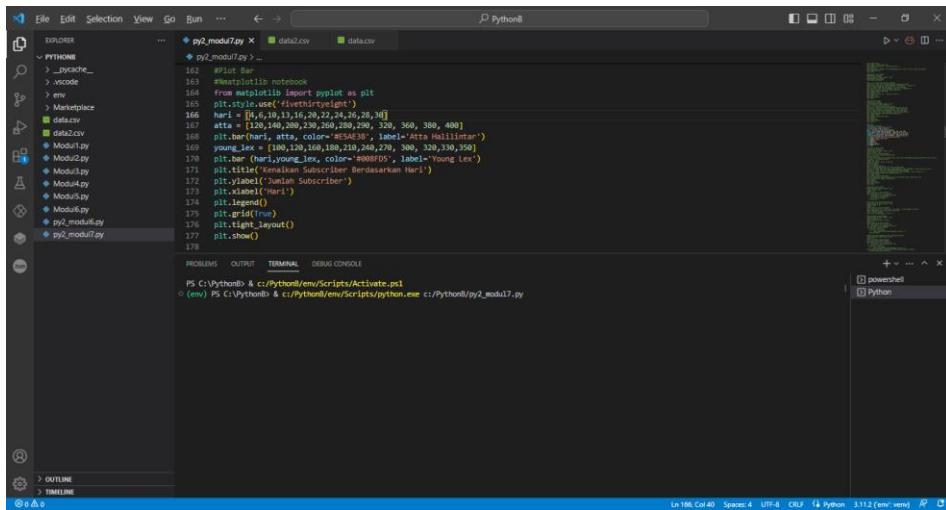
```
143 #!/usr/bin/python
144 # Berikut ini Bar chart dan Bar
145 # scatterplot menggunakan
146 # matplotlib
147 from matplotlib import pyplot as plt
148 plt.style.use('fivethirtyeight')
149 hari = [6,19,13,16,20,24,25,26,28,30]
150 hari_bar = [160, 180, 190, 200, 210, 220, 230, 240, 250, 260]
151 plt.bar(hari, 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
```



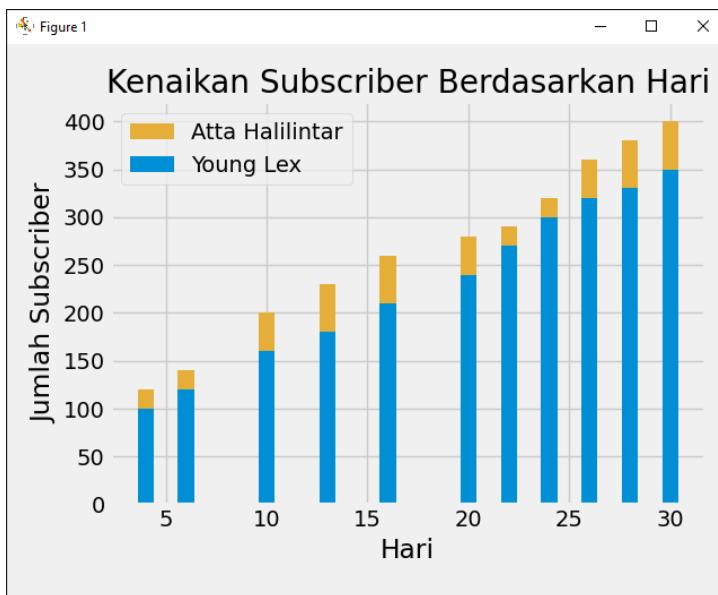
11. Contoh Plot Bar



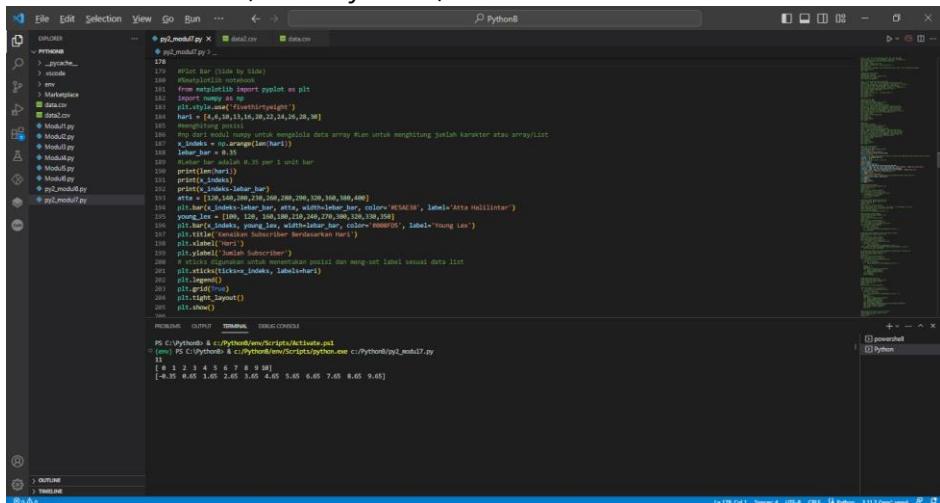
The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like `py2_modul7.py`, `data.csv`, and `data.csv`. The Editor tab contains the following Python code:

```
#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='#E6A38B', 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)

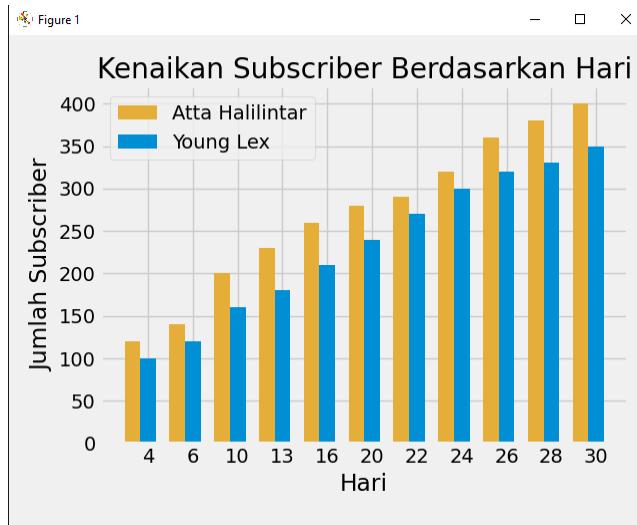


The screenshot shows the VS Code interface with the Python extension installed. The Explorer sidebar shows files like `py2_modul7.py`, `data.csv`, and `data.csv`. The Editor tab contains the following Python code:

```
#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 hari in x_index:
    label_hari.append(str(hari))

def kalkulator(hari):
    print('Hari ke-' + str(hari))
    print('Atta: ' + str(atta[hari]))
    print('Young Lex: ' + str(young_lex[hari]))
    print('Total: ' + str(atta[hari] + young_lex[hari]))
```

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

```

In [1]: #!/usr/bin/python3
# coding: utf-8
# Author: Atta Halilintar
# Date: 2018-07-10
# Description: This script will read a CSV file and print all rows
# Usage: python3 py2_modul17.py <file>
# Example: python3 py2_modul17.py data.csv

import csv
from matplotlib import pyplot as plt
import numpy as np
plt.style.use('fivethirtyeight')
with open('data.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)
    for row in csv_reader:
        print(row)

```

2. Hitung Semua Jumlah Pengguna

```

In [1]: #!/usr/bin/python3
# coding: utf-8
# Author: Atta Halilintar
# Date: 2018-07-10
# Description: This script will read a CSV file and print all rows
# Usage: python3 py2_modul17.py <file>
# Example: python3 py2_modul17.py data.csv

import csv
Counter = 0
for row in csv.reader(open("data.csv")):
    hitung_update(row["jumlahPengguna"])
print(Counter)

def hitung_update(jumlahPengguna):
    global Counter
    Counter += int(jumlahPengguna)

```

3. Tampilkan Jumlah Pengguna Teratas dan Batas Tertentu

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

- File Structure (DIALOGS):** Shows a project named "DIALOGS" with a "PYTHON" folder containing files like "py2_modul7.py", "py2_modul7.csv", and "data.csv".
- Code Editor:** Displays Python code for data analysis, including reading CSV files, calculating the most common value in a column, and plotting data.
- Terminal:** Shows command-line output from running the script, displaying various file paths and counts.
- Status Bar:** Shows the current file (py2_modul7.py), line count (Line 253), character count (Col 5), and other system information.

4. Pisahkan Value Bahasa dan Jumlah Pengguna ke Dalam Masing - Masing List

5. Tampilkan Menjadi Plot Bar

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, Refresh, PythonB
- Explorer:** Shows files in the current directory:
 - py2_modul7.py (active)
 - data.csv
 - data2.csv
- Code Cell:** Python code to read data from 'data.csv', calculate the total number of speakers for each language, and plot a bar chart.

```
215 #Menggunakan Modul Bar
216 from matplotlib import pyplot as plt
217 from collections import Counter
218 import numpy as np
219 import csv
220 plt.style.use('fivethirtyeight')
221 with open('data.csv') as csv_file:
222     csv_reader = csv.reader(csv_file)
223     bahasa = Counter()
224     for row in csv_reader:
225         bahasa.update([row[0].lower()])
226     bahasa = []
227     pengguna = []
228     for item in bahasa.most_common(15):
229         bahasa.append(item[0])
230         pengguna.append(item[1])
231     plt.bar(bahasa, pengguna)
232     plt.xlabel("Bahasa Pemrograman yang Paling Banyak Digunakan")
233     plt.ylabel("Jumlah Pengguna")
234     plt.title("Jumlah Pengguna")
235     plt.show()
```
- Output Cell:** Displays command-line output for activating the Python environment and running the script.

```
PS C:\PythonB\&gt; c:/PythonEnv/Scripts/activate.ps1
❸ PS C:\PythonB & c:/PythonEnv/Scripts/python.exe c:/PythonB/py2_modul7.py
❹ PS C:\PythonB & c:/PythonEnv/Scripts/python.exe c:/PythonB/py2_modul7.py
```
- Bottom Navigation:** OUTLINE, TIMELINE, Help, etc.



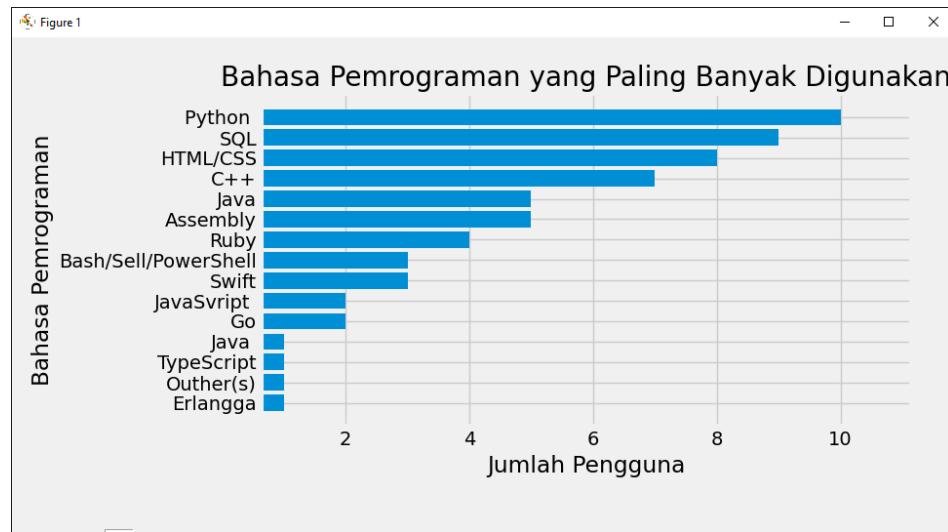
6. Contoh Ubah Data Menjadi Horizontal (Jika Diperlukan)



7. Contoh Reverse (Tampilkan Data Dari terbanyak Ke Sedikit)

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

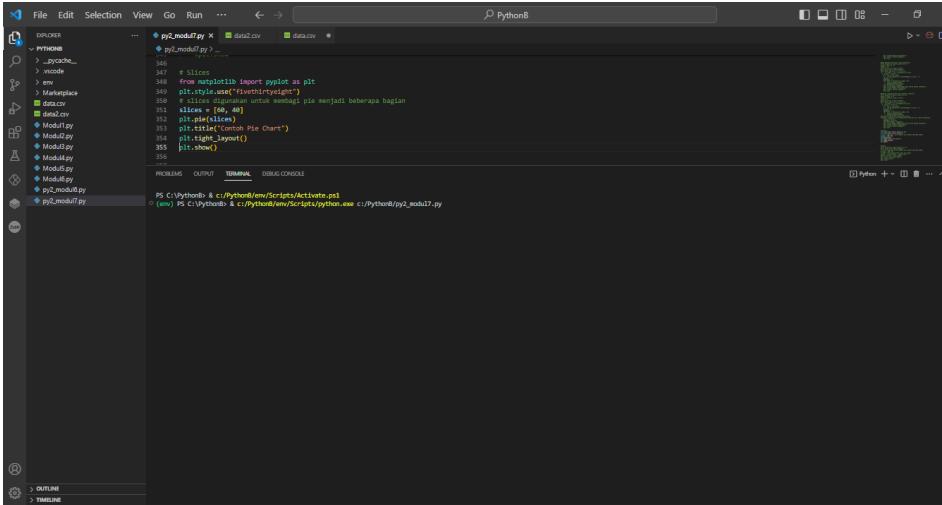
- File Structure:** The left sidebar shows a project named "DIPLODE" with a "PYTHON" section containing files like "py2_modul7.py", "data.csv", and "data2.csv".
- Code Editor:** The main editor window contains Python code for reading CSV files, calculating averages, and plotting bar charts. The code includes imports for os, csv, mathplotlib.pyplot, and collections.
- Output Terminal:** At the bottom, the terminal shows command-line output for running the script: "PS C:\Python3 & cd\PyCharm\myScripts\active\py" and "(env) PS C:\Python3 & cd\PyCharm\myScripts\python\py c:/Python/py2_modul7.py".



C. Plot Pie Chart

Berikut cara menampilkan data menjadi Pie Chart.

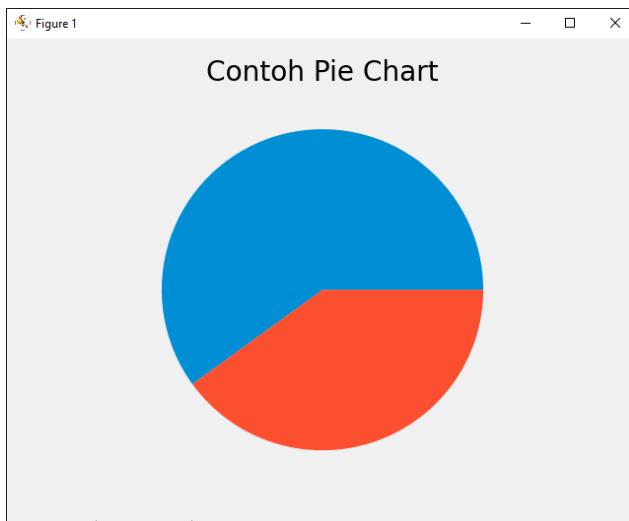
1. Contoh Slices



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

- File Explorer:** On the left, it shows a file tree with several Python files: `py2_modul17.py`, `py2_modul17.csv`, `data2.csv`, and `data.csv`.
- Code Editor:** The main area displays Python code for generating a pie chart from data in `data2.csv`. The code uses `matplotlib` to read the CSV file and create a pie chart titled "contoh Pie Chart".

```
... py2_modul17.py ...  
345  
346  
347 # Slices  
348 from matplotlib import pyplot as plt  
349 plt.pie(data2['jumlah'], labels=data2['kota'], autopct='%.1f')  
350 plt.title('contoh pie bagian')  
351 slices = (60, 40)  
352 labels = ('A', 'B')  
353 plt.title("contoh Pie Chart")  
354 plt.tight_layout()  
355 plt.show()  
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  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2299  
2300  
2301  
2302  
23
```



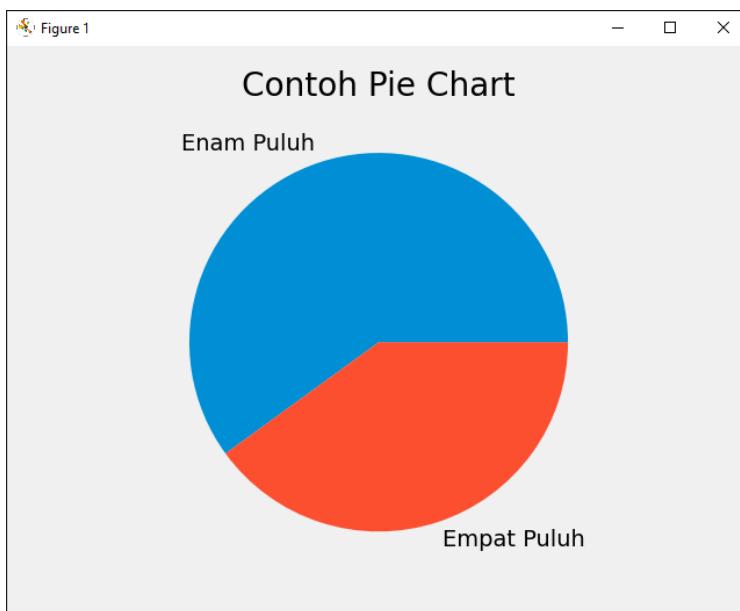
2. Contoh Labels

```
File Edit Selection View Go Run ... < > PythonB
PYTHOND
> _pycache_
> vscod
> env
> __init__.py
> data.csv
> data2.csv
> data3.csv
> Modul1.py
> Modul2.py
> Modul3.py
> Modul4.py
> Modul5.py
> Modul6.py
> Modul7.py
> py2_modul7.py

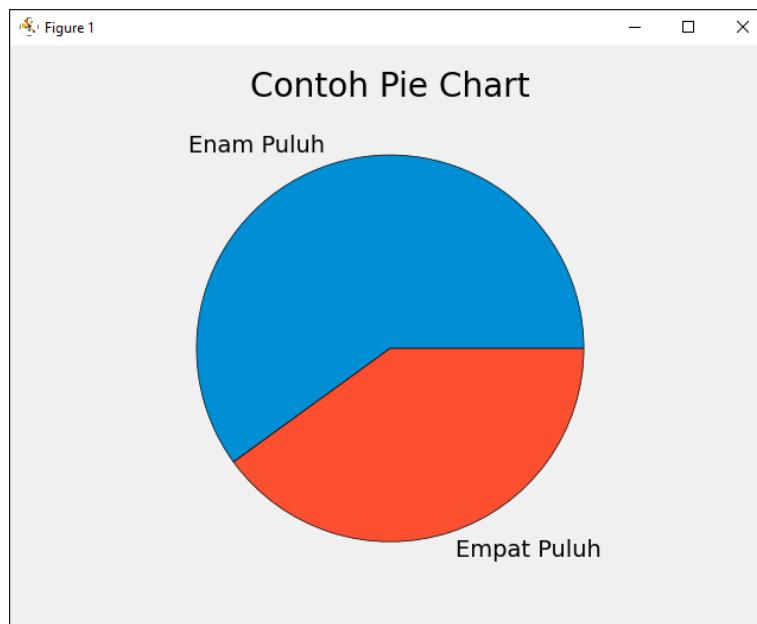
py2_modul7.py
from matplotlib import pyplot as plt
plt.style.use('fivethirtyeight')
slices = [60, 40]
labels = ['Enam Puluh', 'Empat Puluh']
plt.pie(slices, labels=labels)
plt.title('Contoh Pie Chart')
plt.tight_layout()
plt.show()

PROBLEMS OUTPUT DEBUG CONSOLE
PS C:\PythonB & c:/Python/env/Scripts/activate.ps1
(c) PS C:\PythonB & c:/Python/env/Scripts/python.exe c:/PythonB/py2_modul7.py

```



3. Contoh Wedgeprops



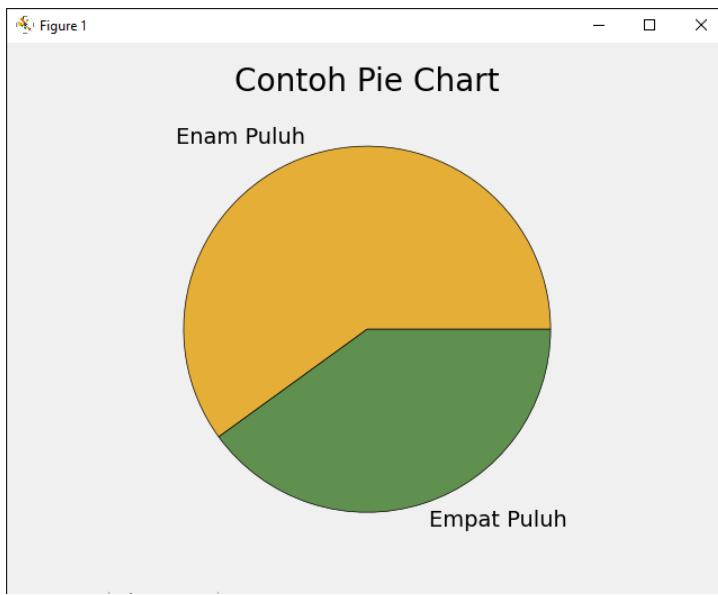
4. Contoh Colors

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

- File Structure:** The left sidebar shows a project named "DONGK" with files like `pyLabeled.py`, `den1.csv`, `den2.csv`, and `den3.csv`.
- Code Editor:** The main window displays the `pyLabeled.py` script:

```
... * pyLabeled.py x den1.csv den2.csv den3.csv
  pyLabeled.py

  381
  382
  383     from matplotlib import pyplot as plt
  384
  385     plt.style.use("fivethirtyeight")
  386
  387     # Mengambil data untuk membuat pie yang jadi beberapa bagian
  388     slices = [60, 40]
  389
  390     # Labels untuk menampilkan label per bagian
  391     label = ["Banyak Pulih", "Anggap Pulih"]
  392
  393     # Mengatur warna
  394     warna = ["#E9967A", "#F0B88C"]
  395
  396     plt.pie(slices, labels=label, colors=warna, wedgeprops={"edgecolor": "black"})
  397     plt.title("Sektor Pie Chart")
  398
  399     plt.tight_layout()
  400
  401     plt.show()
```
- Terminal:** At the bottom, the terminal shows the command `python3 pyLabeled.py` being run.
- Bottom Status Bar:** Shows the Python version (Python 3.11.2) and other system information.



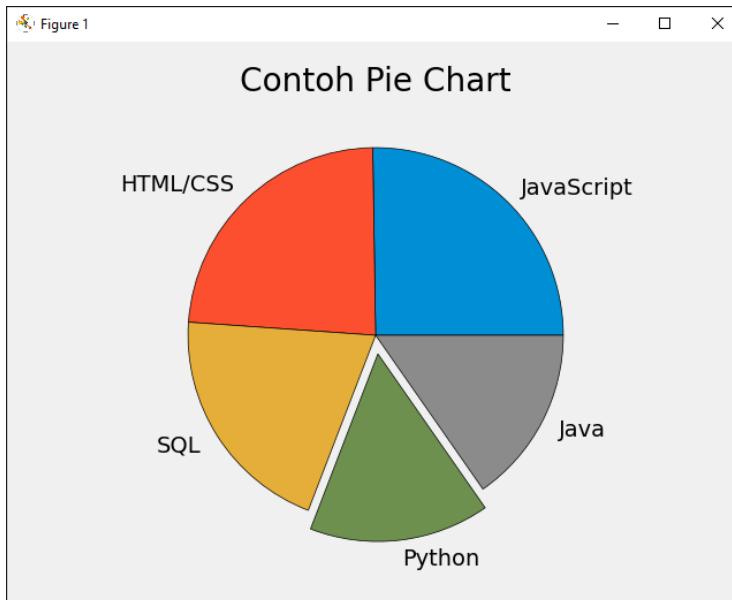
5. Contoh Explode

```

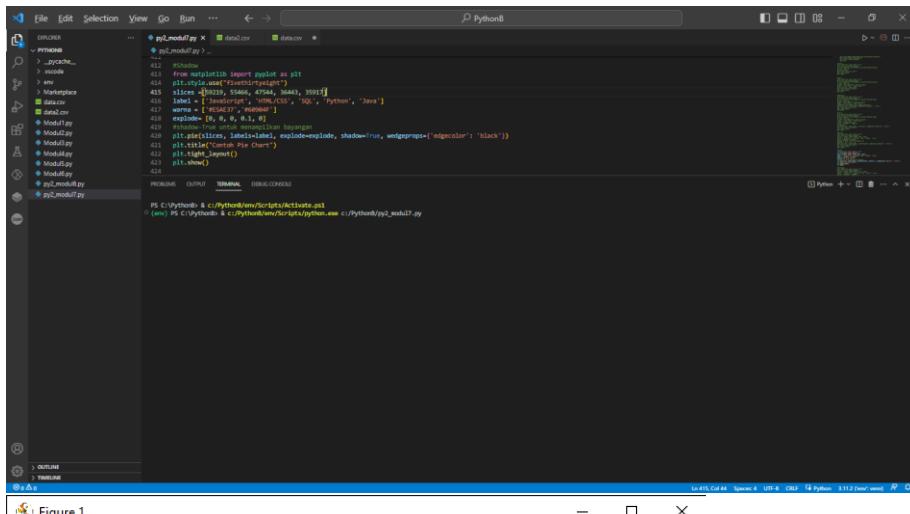
File Edit Selection View Go Run ... < > PythonB
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\PythonB> c:\Python\env\Scripts\activate.ps1
(c:\env) PS C:\PythonB> c:\Python\env\Scripts\python.exe c:/PythonB/py4_modul17.py

```

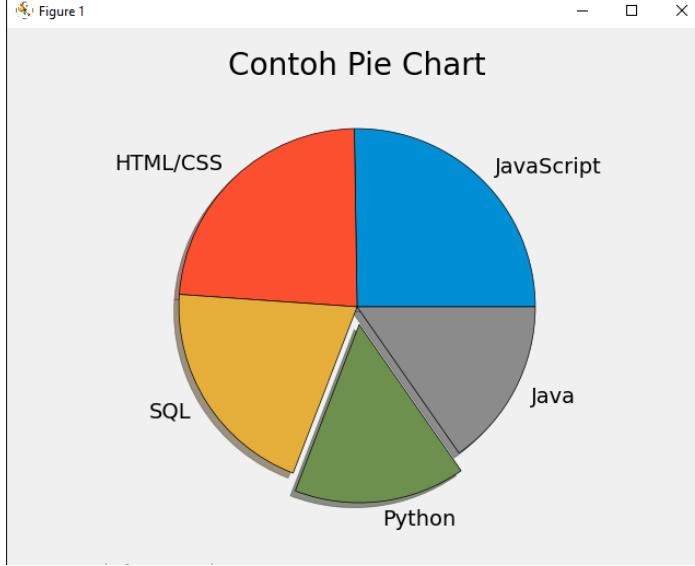
The screenshot shows a Python script named `py4_modul17.py` in a VS Code editor. The script uses the `matplotlib.pyplot` module to create a pie chart. The chart is exploded, with the first slice labeled "JavaScript" being pushed outwards. The labels for the slices are "HTML/CSS", "SQL", "Java", and "Python". The code includes styling like a light gray background and black wedge borders.



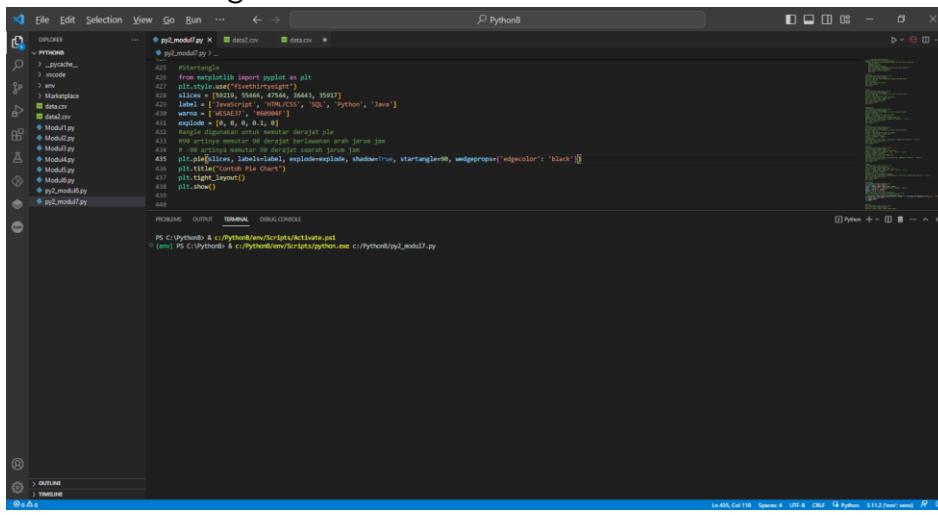
6. Contoh Shadow



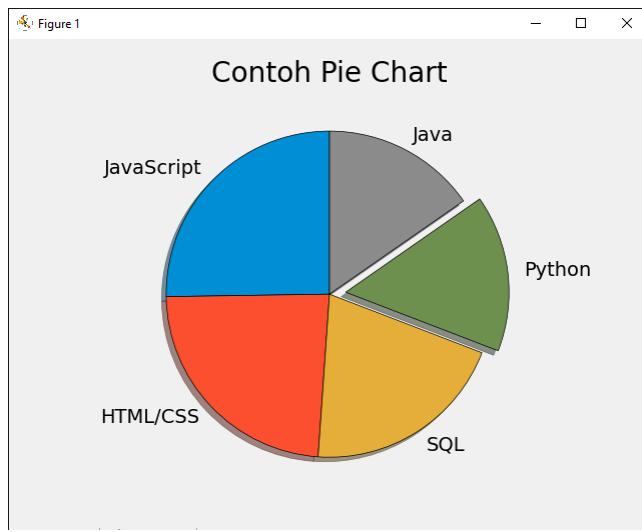
```
py2_modul2.py
411 from matplotlib import pyplot as plt
412 slices = [59219, 55666, 47544, 38441, 38917]
413 label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
414 areas = [59219, 55666, 47544, 38441, 38917]
415 explode = [0, 0, 0, 0.1, 0]
416 plt.pie(slices, labels=label, explode=explode, shadow=True, wedgeprops={'edgecolor': 'black'})
417 plt.title('Contoh Pie Chart')
418 plt.tight_layout()
419 plt.show()
420
```



7. Contoh Startangle



```
py2_modul2.py
425 #startangle
426 from matplotlib import pyplot as plt
427 slices = [59219, 55666, 47544, 38441, 38917]
428 label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
429 areas = [59219, 55666, 47544, 38441, 38917]
430 explode = [0, 0, 0, 0.1, 0]
431 #Buat diagram lingkaran dengan startangle
432 #Start angle ini membuat dia berikanan arah jarum jam
433 #sehingga posisinya berlawanan arah jarum jam
434 #& diatasnya muncul di bawahnya
435 plt.pie(slices, labels=label, explode=explode, startangle=90, wedgeprops={'edgecolor': 'black'})
436 plt.title('Contoh Pie Chart')
437 plt.tight_layout()
438 plt.show()
439
```



8. Contoh Autopct

```

File Edit Selection View Go Run ... ← → ⌘ Python8
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Python8 & c:/Python8/env/Scripts/activate.ps1
(env) PS C:\Python8 & c:/Python8/env/Scripts/python.exe c:/Python8/py2_modul17.py
In 444, Col 45  Specs: 4  UTF-8  CR LF  14 Python  3.11.2 (env\venv) R O

```

```

py2_modul17.py
...
441 #Autopct
442 from matplotlib import pyplot as plt
443 plt.style.use('fivethirtyeight')
444 data = [35, 15, 15, 15, 10]
445 label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
446 warnas=['#E63373', '#D94949']
447 explode=[0, 0, 0, 0, 0]
448 autopct='%.1f%%' # %f artinya menampilkan bilangan float. Harus diawali %
449 #%d artinya menampilkan bilangan integer. Harus diawali %
450 #%0.1f menampilkan bilangan float dengan 1 angka dibelakang koma #%0.1X menampilkan simbol % itu sendiri dan ditutup dengan X
451 plt.pie(data, labels=label, explode=explode, shadow=True, autopct='%.1f%%', wedgeprops={'edgecolor': 'black'})
452 plt.legend(loc='best')
453 plt.tight_layout()
454 plt.show()

```

