



MODUL PRAKTIKUM DATA MINING

**Teknik Informatika
Universitas Pelita Bangsa**

DAFTAR ISI

MODUL 1 (MINGGU PERTAMA)	3
PRAKTIKUM 1	4
INSTALASI PYTHON	5
INSTALASI LIBRARY PANDAS, MATPLOTLIB	12
PRAKTIKUM 2	15
Sintaks Dasar Python	15

MODUL 1
(MINGGU PERTAMA)

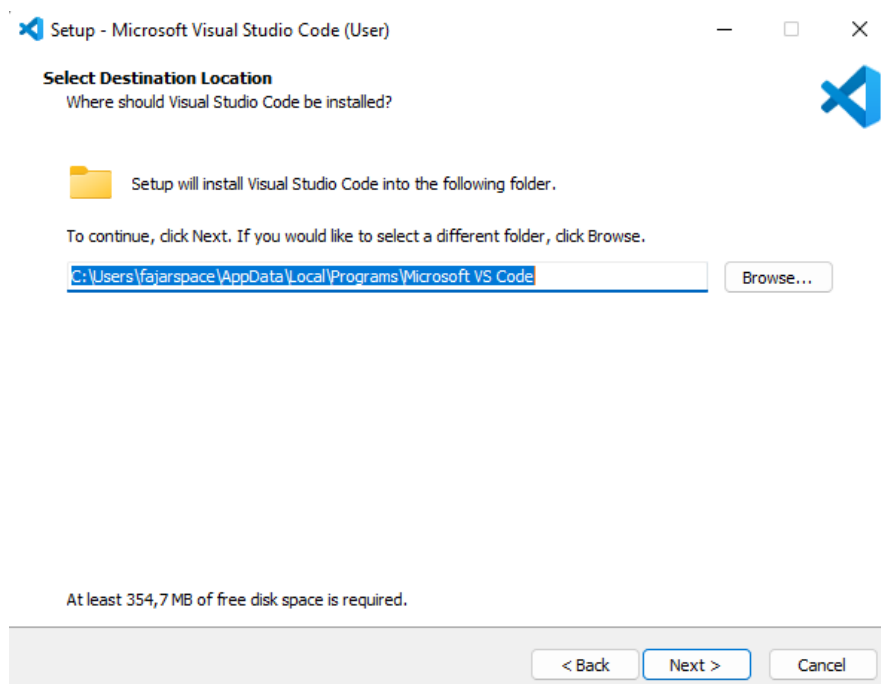
PRAKTIKUM 1

JUDUL : Instalasi VSCode, Python, Library

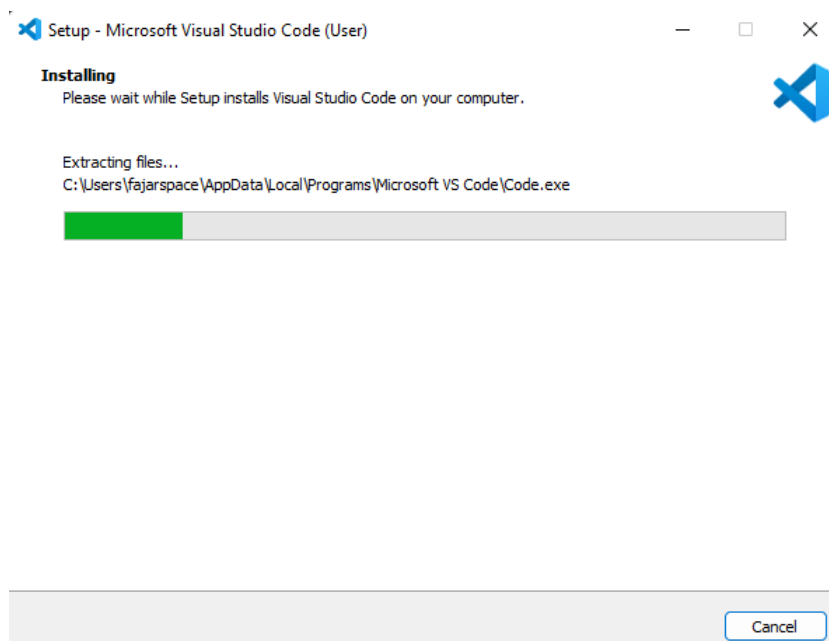
DESKRIPSI : Tujuan dari praktikum ini adalah

INSTALASI

1. Download VSCode di <https://code.visualstudio.com/>
2. lalu diklik, Kemudian klik next saja.
3. Sesuaikan dengan direktori masing-masing



4. Kemudian klik next saja, lalu tunggu hingga selesai



INSTALASI PYTHON

TUJUAN

1. Mahasiswa Mampu Melakukan Instalasi Python
2. Mahasiswa Dapat Mengerti Dasar Pemrograman Python

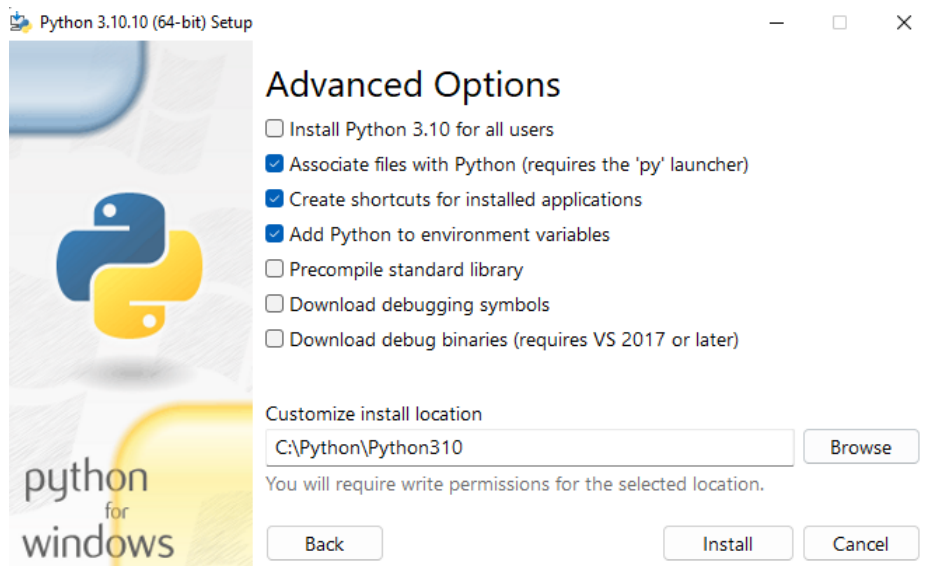
INSTALASI

1. Download Python di <https://www.python.org/downloads/windows/>
Pilih versi Python sesuai dengan tipe sistem operasi Anda (32/64 bit)

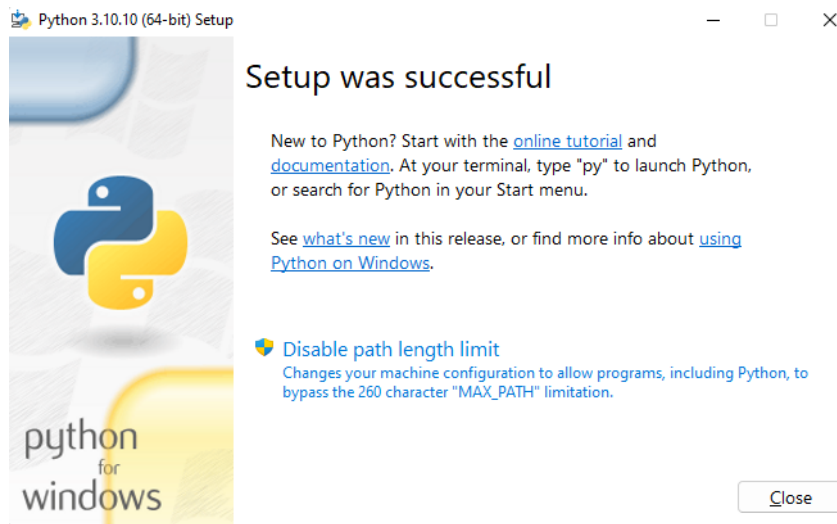


lalu diklik. Setelah muncul kotak dialog seperti dibawah, pilih **Customize installation**.

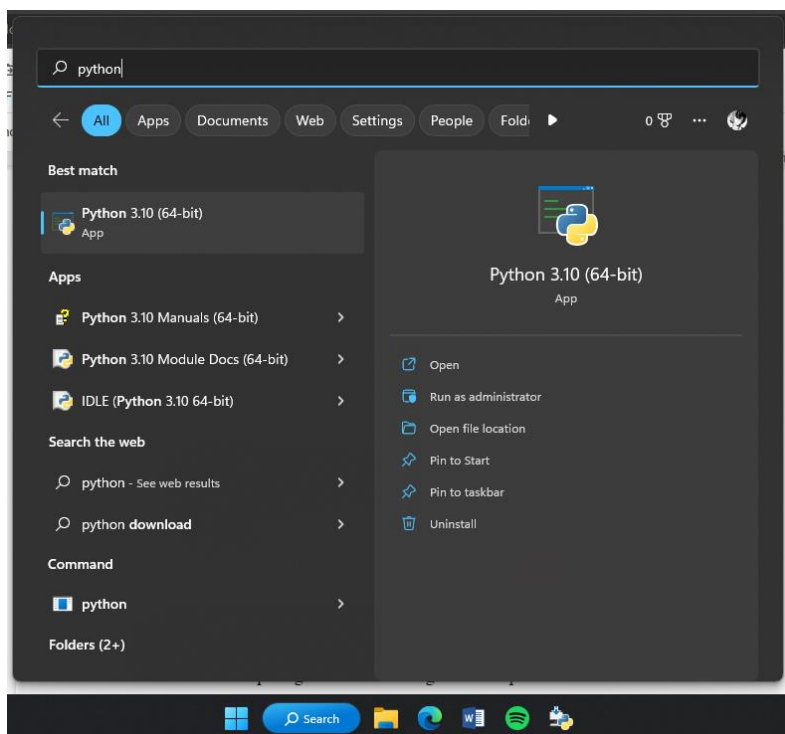
2. Kemudian klik next saja.
3. Pilih direktori dimana Anda ingin menginstal Python. Lokasi yang saya pilih **C:\Python\Python310** agar mudah diakses, Centang '**Add python to environment variables**' Kemudian klik Install.



4. Tunggu hingga selesai
5. Setelah sukses, klik Close

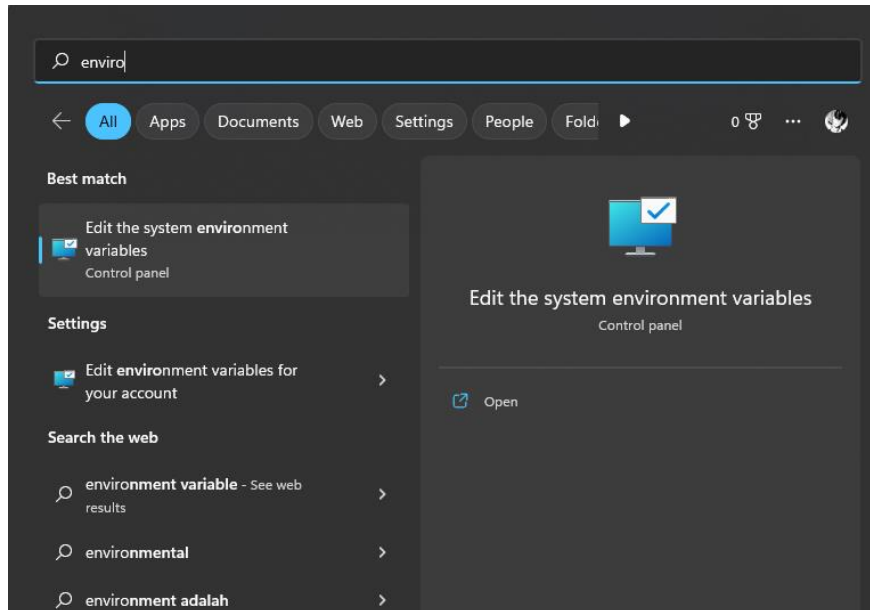


6. Anda bisa mengecek apakah Python sudah diinstal dengan melakukan searching di Windows.

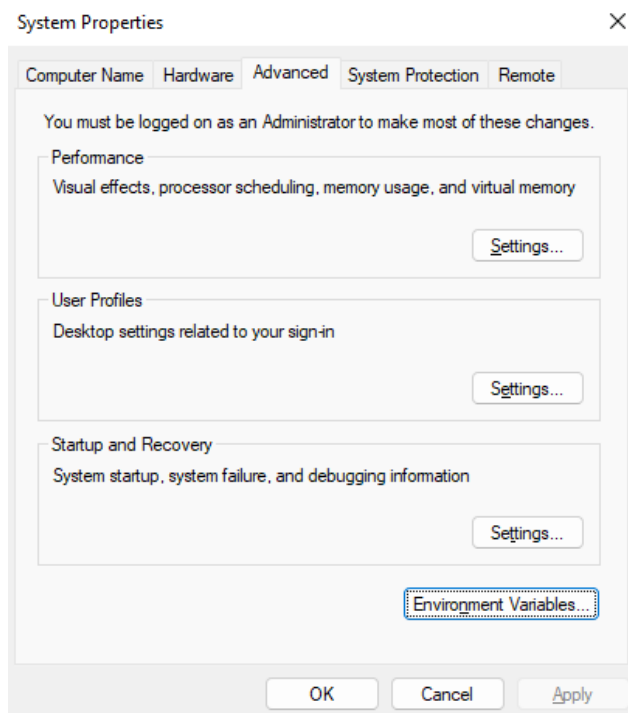


7. Kemudian copy alamat direktori Python **C:\python\python310**

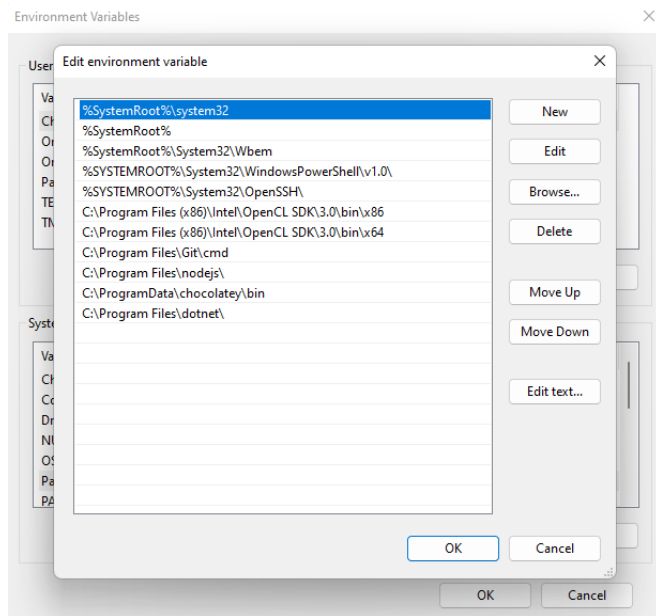
8. Buka System Environment Variables untuk men-setting Path. Buka **Control Panel** lalu searching dengan keyword atau bisa langsung search melalui **Search Box Windows** seperti gambar dibawah agar lebih cepat.



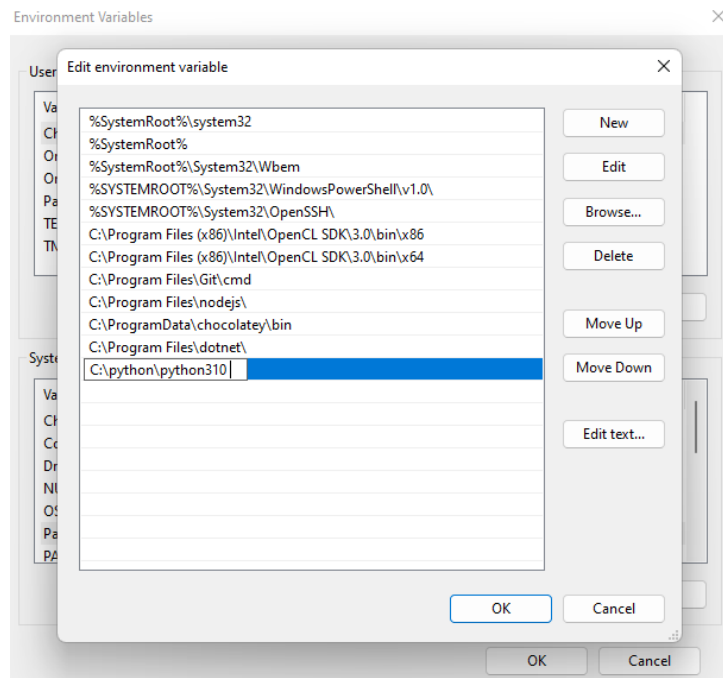
Setelah muncul kotak dialog seperti gambar dibawah klik **Environment Variables**.



9. Pilih bagian **System variables** > **Path** lalu klik Edit.



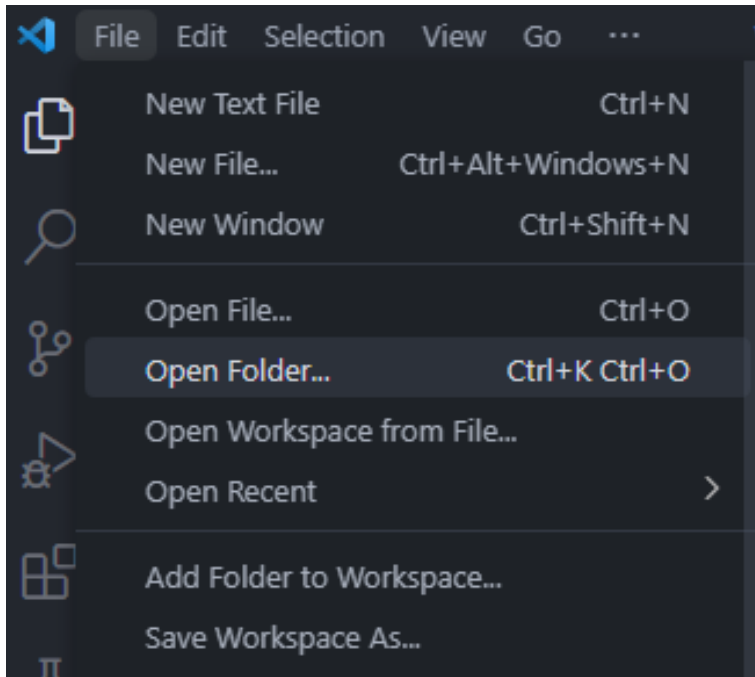
10. Klik tombol **New** kemudian paste alamat **C:\Python\Python310** yang telah di-copy tadi. Setelah selesai klik OK.



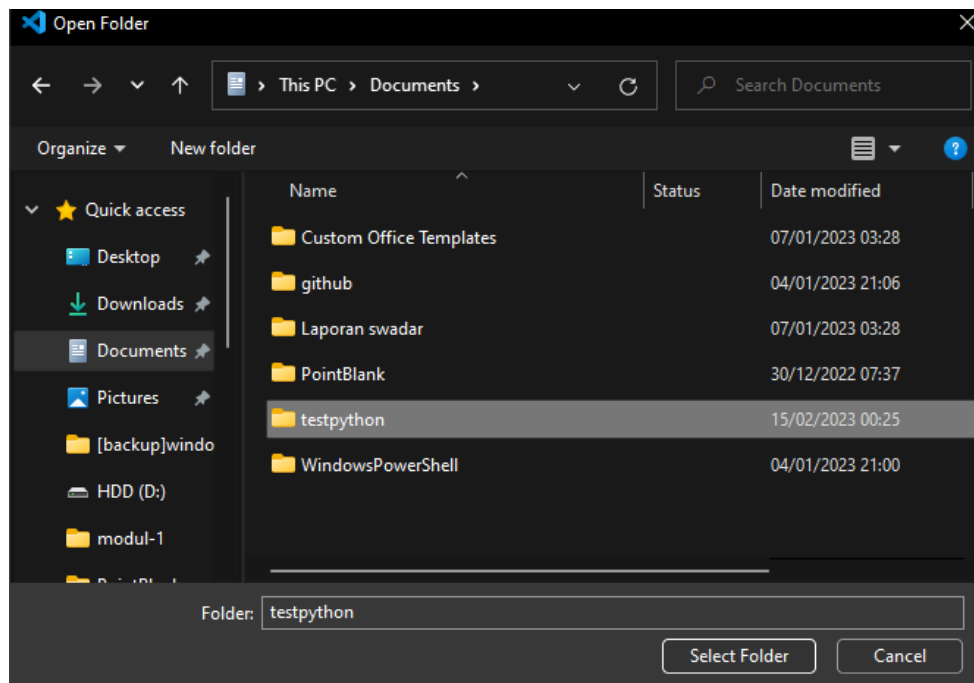
11. Buka VS Code yang sudah terinstal, Lalu klik menu **Extensions** pada **Activity Bar** (sebelah kiri). Ketik **python**, kemudian **Install** dan reload VS Code Anda.



12. Pada bagian **Top Bar**, klik **File > Open Folder**. Mulai dari sini kita akan melakukan tes running project Python.

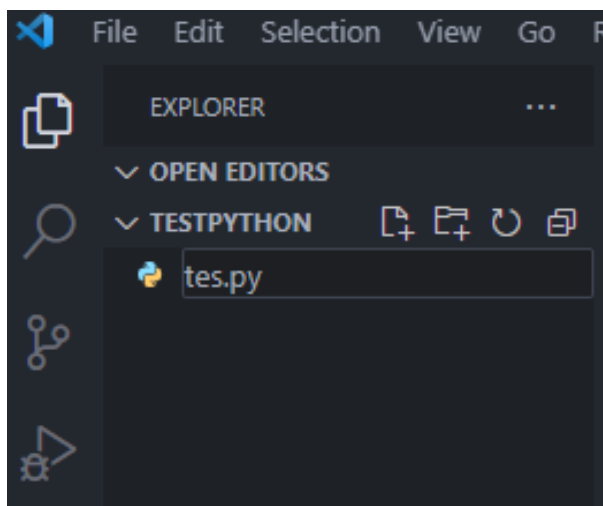


13. Pilih lokasi folder project. Anda bisa membuat folder dimana saja. Pada kasus ini saya membuatnya di **Documents** dengan nama **tespython**. Kemudian klik Select Folder.

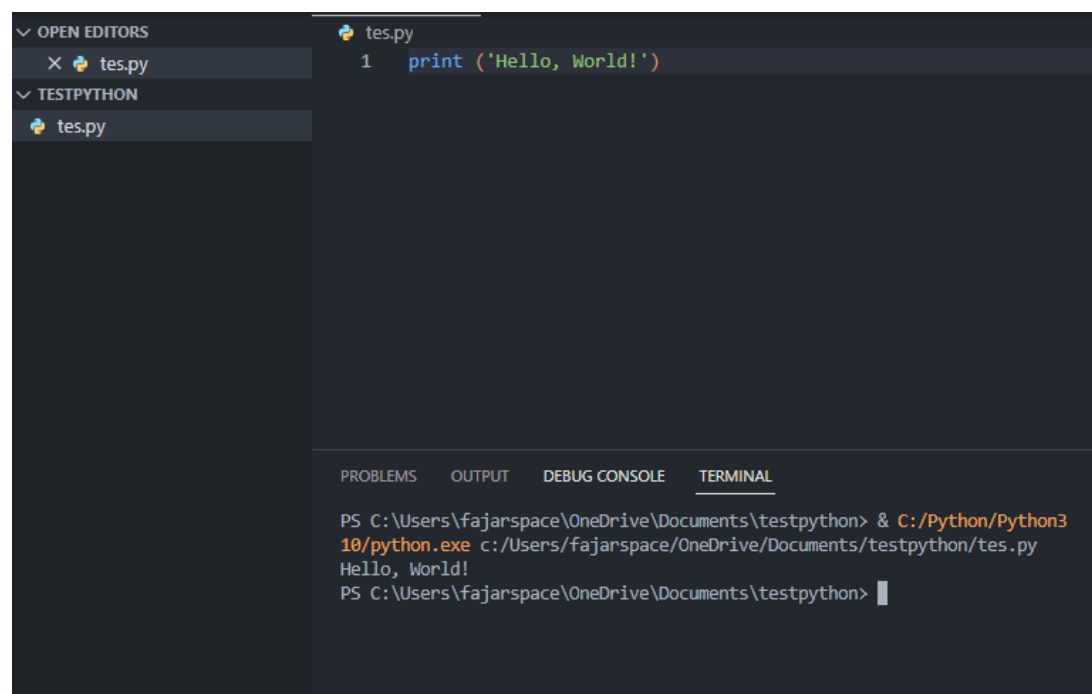
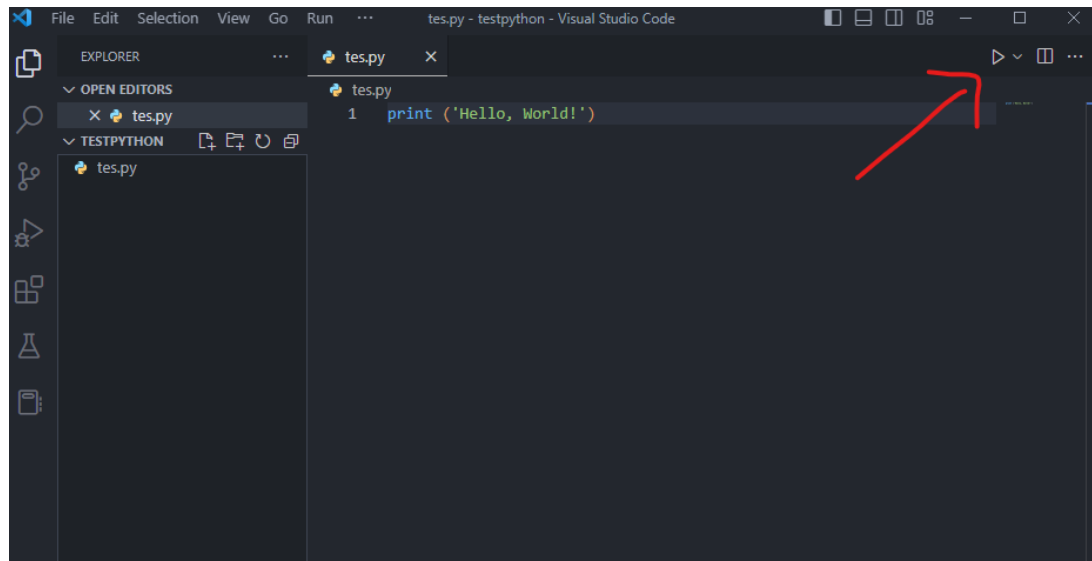


14. Setelah tampil di **Explorer**. Klik **New File** untuk membuat file Python.

15. Ketik nama file yang Anda inginkan. Jangan lupa menambahkan ekstensi **.py** dibelakang nama file. Misalnya **tes.py**



16. Contoh dibawah adalah menampilkan teks Tes Python, kemudian klik tombol **Run** pada pojok kanan atas.



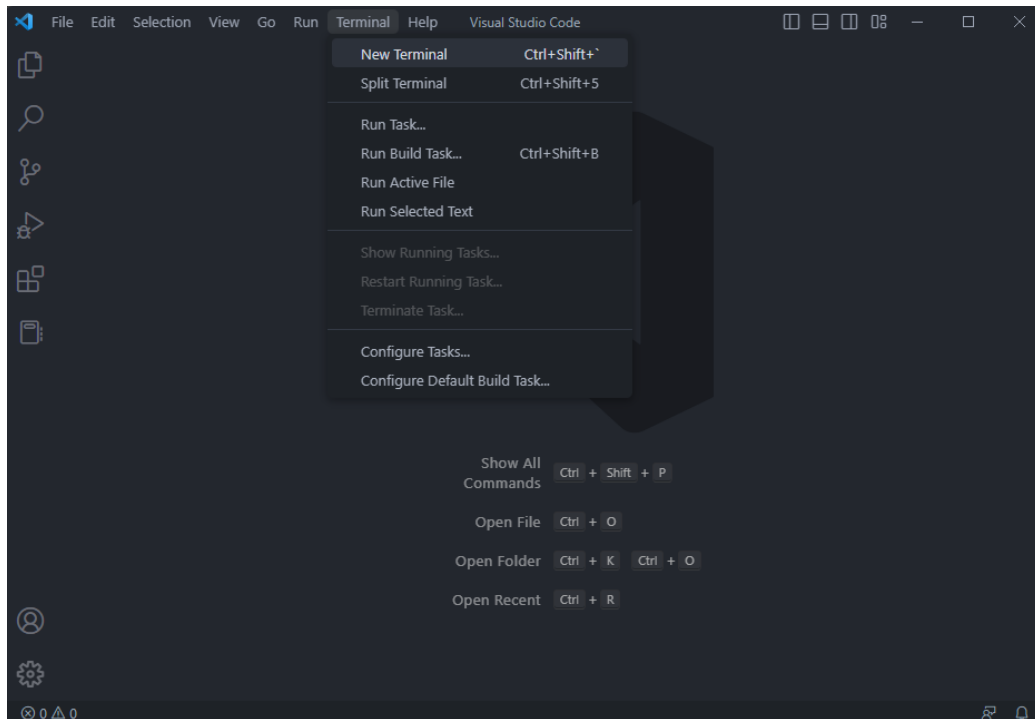
INSTALASI LIBRARY PANDAS, MATPLOTLIB

TUJUAN

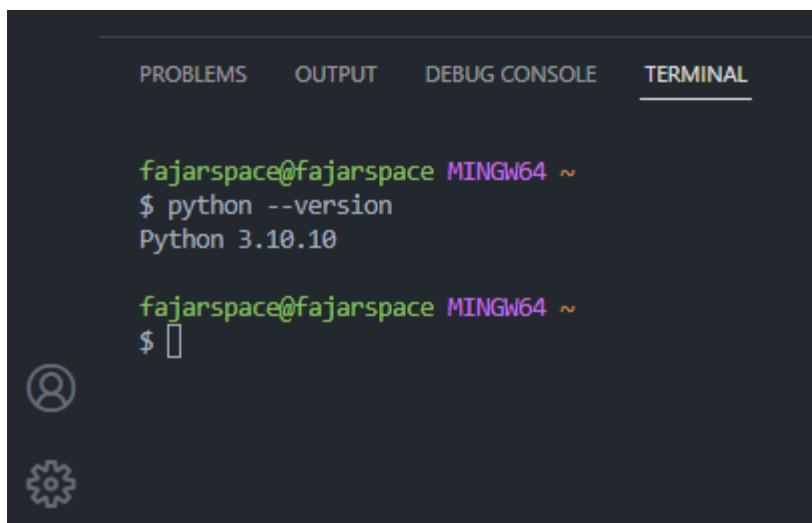
1. Mahasiswa Mampu Melakukan Instalasi Library pandas dan matplotlib

INSTALASI

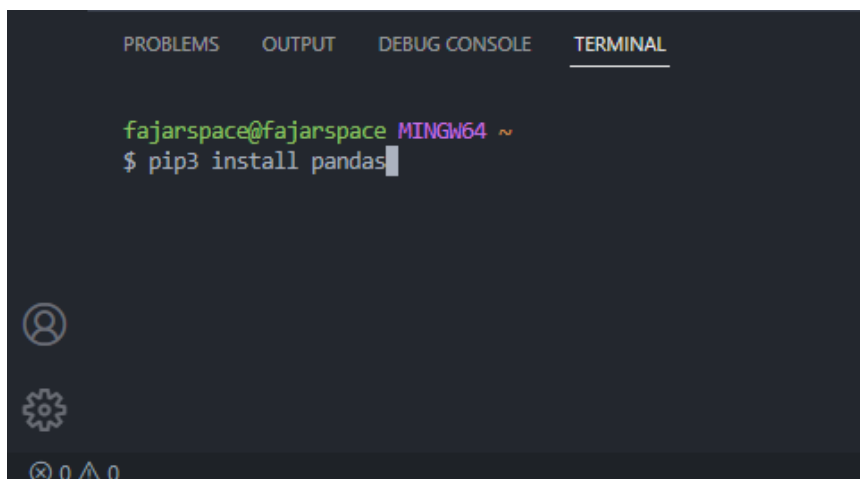
1. Buka VSCode yang sebelumnya sudah di install, lalu klik **terminal > new terminal**



2. Untuk mengecek python yang sudah kita install ketik **python --version** disitu tercetak **python 3.10.10**, yang artinya versi python yang terinstal adalah versi 3.10



3. Untuk dapat menginstall **pandas**, kita bisa menjalankan perintah berikut :

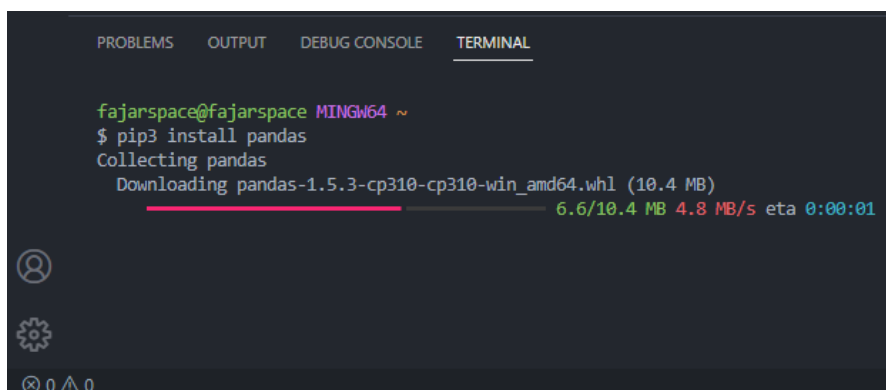


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

fajarspace@fajarspace MINGW64 ~
$ pip3 install pandas
```

The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The prompt 'fajarspace@fajarspace MINGW64 ~' is displayed. Below the prompt, the command '\$ pip3 install pandas' has been entered, and the cursor is at the end of the line. On the left side of the terminal, there are icons for a user profile and a gear (settings). At the bottom left, there are status icons showing 0 errors, 0 warnings, and 0 info messages.

Lalu klik **enter**, tunggu hingga proses selesai.

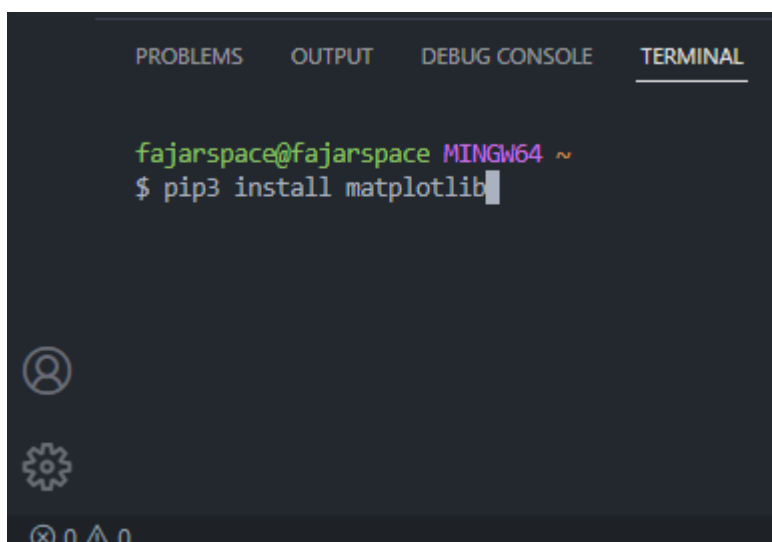


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

fajarspace@fajarspace MINGW64 ~
$ pip3 install pandas
Collecting pandas
  Downloading pandas-1.5.3-cp310-cp310-win_amd64.whl (10.4 MB)
    6.6/10.4 MB 4.8 MB/s eta 0:00:01
```

This screenshot shows the terminal after the 'enter' key is pressed. The command '\$ pip3 install pandas' has been executed. The output shows 'Collecting pandas' followed by 'Downloading pandas-1.5.3-cp310-cp310-win_amd64.whl (10.4 MB)'. A progress bar is visible, showing '6.6/10.4 MB' downloaded at a speed of '4.8 MB/s' with an estimated time remaining of 'eta 0:00:01'. The rest of the terminal interface, including the tabs and status icons, remains the same as in the previous screenshot.

4. Lalu untuk dapat menginstall **matplotlib**, kita bisa menjalankan perintah berikut :

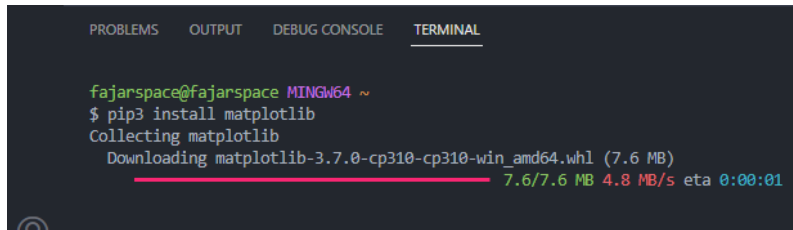


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

fajarspace@fajarspace MINGW64 ~
$ pip3 install matplotlib
```

The screenshot shows a terminal window with the same interface as the previous ones. The prompt 'fajarspace@fajarspace MINGW64 ~' is visible. The command '\$ pip3 install matplotlib' has been entered, and the cursor is at the end of the line. The tabs at the top and the status icons at the bottom are also visible.

Lalu klik **enter**, tunggu hingga proses selesai.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

fajarspace@fajarspace MINGW64 ~
$ pip3 install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.7.0-cp310-cp310-win_amd64.whl (7.6 MB)
    

7.6/7.6 MB 4.8 MB/s eta 0:00:01


```

The image shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal content shows a user prompt 'fajarspace@fajarspace MINGW64 ~' followed by the command '\$ pip3 install matplotlib'. The output shows 'Collecting matplotlib' and then 'Downloading matplotlib-3.7.0-cp310-cp310-win_amd64.whl (7.6 MB)'. Below this, a progress bar is shown with the text '7.6/7.6 MB 4.8 MB/s eta 0:00:01'.

PRAKTIKUM 2

JUDUL : Python Dasar

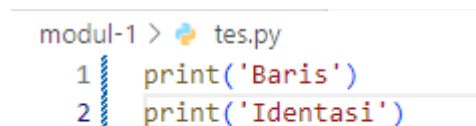
Sintaks Dasar Python

1.1. Identifier

Identifier adalah identitas atau nama yang telah diberikan kepada function, variabel, obyek, class, namespace dan lain-lain.

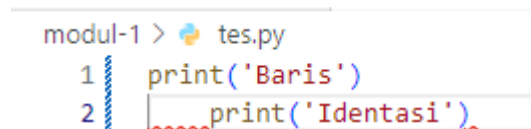
1.2. Baris dan Indentasi

Python tidak menggunakan tanda { } untuk menandai blok/grup kode. Blok kode di python menggunakan tanda indentasi (spasi). Jumlah spasi untuk setiap baris yang ada dalam satu blok kode harus sama.



```
modul-1 > tes.py
1 | print('Baris')
2 | print('Identasi')
```

Gambar 1: Contoh benar



```
modul-1 > tes.py
1 | print('Baris')
2 | print('Identasi')
```

Gambar 2: Contoh Salah

2. Tipe Data

Beberapa tipe data built-in atau bawaan yang cukup lengkap dan tidak sulit untuk digunakan. Beberapa tipe data yang wajib Anda ketahui di Python3 antara lain:

- int, tipe data yang dapat Anda isi dengan bilangan bulat.
- float, tipe data yang dapat Anda isi dengan bilangan koma.
- string, tipe data yang dapat Anda isi dengan sebuah karakter atau kalimat.
- complex, tipe data bilangan kompleks atau bilangan imajiner, seperti 5j, 54j, 1j.
- long, tipe data yang dapat Anda isi dengan bilangan yang sangat besar. Bisa lebih dari 1 milyar.

- boolean, tipe data yang dapat Anda isi untuk mendukung operasi bool. Nilainya hanya True dan False.
- List, adalah tipe data yang berisi item yang berurut. List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung [] dan masing-masing anggotanya dipisahkan oleh tanda koma.
- Tuple, adalah jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak bisa diubah (immutable). Tuple dideklarasikan dengan menggunakan tanda kurung (). dan anggotanya dipisahkan oleh tanda koma.
- Set, adalah salah satu tipe data di Python yang tidak berurut (unordered).
- Dictionary, adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci- nilai (key-value). Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal { }, dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

3. Variabel, Operator, dan Ekspresi

3.1.Variabel

Variabel adalah lokasi di memori yang digunakan untuk menyimpan nilai. Pada saat kita membuat sebuah variabel, kita ‘memesan’ tempat di dalam memori. Tempat tersebut bisa diisi dengan data atau objek, baik itu bilangan bulat (integer), pecahan (float), karakter (string), dan lain - lain.

Di python, variabel tidak perlu dideklarasikan secara eksplisit. Deklarasi atau pembuatan variabel terjadi secara otomatis pada saat kita memberi (menugaskan) suatu nilai ke variabel. Tanda sama dengan (=) digunakan untuk memberikan nilai ke variabel. Misal *a = 'Hello, world!'*

3.2. Operator & Ekspresi

Hampir semua statemen (baris logis) yang Anda tulis akan mengandung ekspresi. Contoh sederhana dari ekspresi adalah 2+3. Sebuah ekspresi dapat diturunkan menjadi operator dan operand. Operator adalah simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Nilai yang padanya dilakukan operasi disebut operand. Misalnya adalah 2 + 3. Di sini tanda + adalah operator penjumlahan. 2 dan 3 adalah operand.

4. Input dan Output

4.1. Input

Input adalah masukan yang kita berikan ke program. Program akan memprosesnya dan menampilkan hasil outputnya. Input, proses, dan output adalah inti dari semua program komputer. Fungsi untuk melakukan operasi input adalah fungsi input()

```
modul-1 > tes.py > ...
1 a = input('Masukkan nilai A :')
2 b = input('Masukkan nilai B :')
3 print(a,b)
```

Gambar 2: Contoh input

4.2. Output

Fungsi bawaan untuk melakukan operasi output adalah print(). Seperti yang sudah sering kita praktekan, kita menggunakan fungsi print() untuk menampilkan data ke perangkat keluaran standar (layar).

```
4
5 print(1, 3, 5, 7)
6 # output: 1 3 5 7
```

Gambar 3: Contoh output

5. Percabangan (if, if else, if elif else)

Percabangan adalah cara yang digunakan untuk mengambil keputusan apabila di dalam program dihadapkan pada kondisi tertentu. Jumlah kondisinya bisa satu, dua atau lebih. Percabangan mengevaluasi kondisi atau ekspresi yang hasilnya benar atau salah. Kondisi atau ekspresi tersebut disebut ekspresi boolean. Hasil dari pengecekan kondisi adalah True atau False. Bila benar (True), maka pernyataan yang ada di dalam blok kondisi tersebut akan dieksekusi. Bila salah (False), maka blok pernyataan lain yang dieksekusi.

5.1.1. Pernyataan if

Pernyataan if menguji satu buah kondisi. Bila hasilnya benar maka pernyataan di dalam blok if tersebut dieksekusi. Bila salah, maka pernyataan tidak dieksekusi.

Sintaksnya adalah seperti berikut:

```
9
10 angka = 4
11 if angka > 0:
12     print(angka, 'adalah bilangan positif')
13
```

Gambar 4: Contoh if

5.1.2. Pernyataan if...else

Pernyataan if...else menguji 2 kondisi. Kondisi pertama kalau benar, dan kondisi kedua kalau salah. Sintaksnya adalah seperti berikut :

```
14 | bilangan = -1
15 | if bilangan >= 0:
16 |     print('Positif atau nol')
17 | else:
18 |     print('Bilangan negatif')
```

Gambar 5: Contoh if else

5.1.3. Pernyataan if...elif...else...

Pernyataan if...elif...else digunakan untuk menguji lebih dari 2 kondisi. Bila kondisi pada if benar, maka pernyataan di dalamnya yang dieksekusi. Bila salah, maka masuk ke pengujian kondisi elif.

```
20 | bilangan = 5.5
21 | if bilangan > 0:
22 |     print('Positif atau nol')
23 | elif bilangan == 0:
24 |     print('Nol')
25 | else:
26 |     print('Bilangan negatif')
```

Gambar 6: Contoh if elif else

5.1.4. Tambahan : if Bersarang

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

```
percabangan_if_bersarang.py ▶ ...
1 | gaji = 10000000
2 | berkeluarga = True
3 | punya_rumah = True
4 |
5 | if gaji > 3000000:
6 |     print ("Gaji sudah diatas UMR")
7 |     if berkeluarga:
8 |         print ("Wajib ikut asuransi dan menabung untuk pensiun")
9 |     else:
10 |         print ("Tidak perlu ikut asuransi")
11 |
12 |     if punya_rumah:
13 |         print ("Wajib bayar pajak rumah")
14 |     else:
15 |         print ("Tidak wajib bayar pajak rumah")
16 | else:
17 |     print ("Gaji belum UMR")
```

6. Perulangan

Secara umum, Python mengeksekusi program baris perbaris. Mulai dari baris satu, dua, dan seterusnya. Ada kalanya, kita perlu mengeksekusi satu baris atau satu blok kode program beberapa kali.

6.1.1. Perulangan dengan Menggunakan for

Perulangan dengan menggunakan for memiliki sintaks seperti berikut :

for var in sequence:
body of for

var adalah variabel yang digunakan untuk penampung sementara nilai dari sequence pada saat terjadi perulangan. **Sequence** adalah tipe data berurut seperti string, list, dan tuple.

```
28 | nomor = [5, 5, 2]
29 | jumlah = 0
30 | for tampung in nomor:
31 |     jumlah = jumlah + tampung
32 | print('jumlah semuanya :', jumlah)
```

Gambar 7: Contoh for

Perulangan for dengan range

Fungsi range() dapat digunakan untuk menghasilkan deret bilangan. range(10) akan menghasilkan bilangan dari 0 sampai dengan 9 (10 bilangan).

```
34 | for hitung in range(5):
35 |     print('Hitung :', hitung)
```

Gambar 8: Contoh for dengan range()

6.1.2. Perulangan Menggunakan while

Perulangan menggunakan while akan menjalankan blok pernyataan terus menerus selama kondisi bernilai benar.

Adapun sintaks dari perulangan menggunakan while adalah :

while expression:
statement (s)

Di sini, **statement (s)** bisa terdiri dari satu baris atau satu blok pernyataan.

Expression merupakan ekspresi atau kondisi apa saja, dan untuk nilai selain nol dianggap True.

```

37 | hitung = 0
38 | while (hitung < 5):
39 |     print('hitung :', hitung)
40 |     hitung = hitung + 1

```

Gambar 9: Contoh while

Contoh Program Kelipatan Bilangan Genap

Ketentuan : Program pengulangan dengan for. Tampilkan bilangan genap dari 0 hingga batas terakhir bilangan input. Misalnya, apabila diinput 10, maka yang tampil adalah : 0 2 4 6 8.

```

42 | i = 0
43 | n = int(input('Masukkan batas :'))
44 | for i in range(n):
45 |     if i%2 == 0:
46 |         print("Bilangan :", i)
47 |         i = i + 1

```

Gambar 10: Contoh program kelipatan genap

6.1.3. Latihan

Buatlah program kelipatan bilangan genap dengan menampilkan banyaknya jumlah. Misalnya, apabila diinput 10, maka yang tampil adalah 0 2 4 6 8 10 12 14 16 18 (**10 bilangan**).

7. 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 dari tempat lain di dalam program.

7.1.1. Mendefinisikan Fungsi

Berikut adalah sintaks yang digunakan untuk membuat fungsi :

def function_name(parameters):

"""function_docstring"""

statement(s)

return [expression]

Penjelasannya dari sintaks fungsi di atas :

- Kata kunci **def** diikuti oleh **function_name** (nama fungsi), tanda kurung dan tanda titik dua (:) menandai header (kepala) fungsi.
- **Parameter/** argumen adalah input dari luar yang akan diproses di dalam tubuh fungsi.

- **"function_docstring"** bersifat opsional, yaitu sebagai string yang digunakan untuk dokumentasi atau penjelasan fungsi. "function_docstring" diletakkan paling atas setelah baris def.
- Setelah itu diletakkan baris-baris pernyataan (**statements**). Jangan lupa indentasi untuk menandai blok fungsi.
- bersifat opsional. Gunanya adalah untuk mengembalikan suatu nilai expression dari fungsi.

```

49 def sapa(nama):
50     print('hai,' + nama + ' . Apa kabar?')
51     return nama
52 sapa('Anna')

```

Gambar 11: Contoh fungsi

7.1.2. Contoh Program Luas Persegi Panjang dengan Fungsi

```

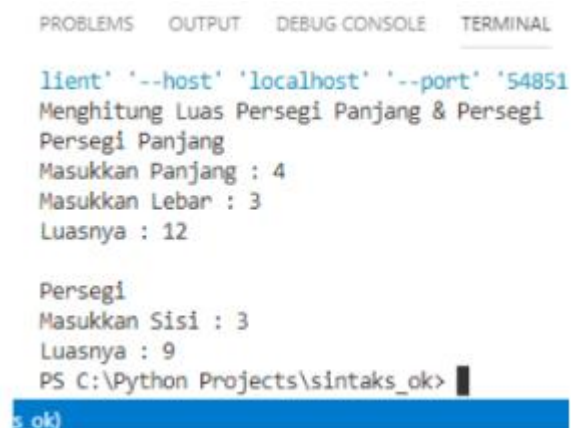
54 def persegipanjang(panjang, lebar):
55     luas = panjang * lebar
56     print('Luasnya :', luas)
57     return luas
58
59 print('menghitung luas persegi panjang')
60 a = int(input('Masukkan panjang :'))
61 b = int(input('Masukkan lebar :'))
62 persegipanjang(a, b)

```

Gambar 12: Contoh Program Luas Persegi Panjang dengan Fungsi

7.1.3. Latihan

Buatlah program dinamis menghitung luas persegi panjang dan persegi dengan menggunakan 1 fungsi. Misalnya, apabila diinput panjang = 4 dan lebar 3, maka tampil luas persegi panjang = 12. Dan apabila diinput sisi persegi = 3, maka tampil luas persegi = 9. Contoh tampilan terminal seperti gambar dibawah.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
lient' '--host' 'localhost' '--port' '54851
Menghitung Luas Persegi Panjang & Persegi
Persegi Panjang
Masukkan Panjang : 4
Masukkan Lebar : 3
Luasnya : 12

Persegi
Masukkan Sisi : 3
Luasnya : 9
PS C:\Python Projects\sintaks_ok>
s ok)

```