

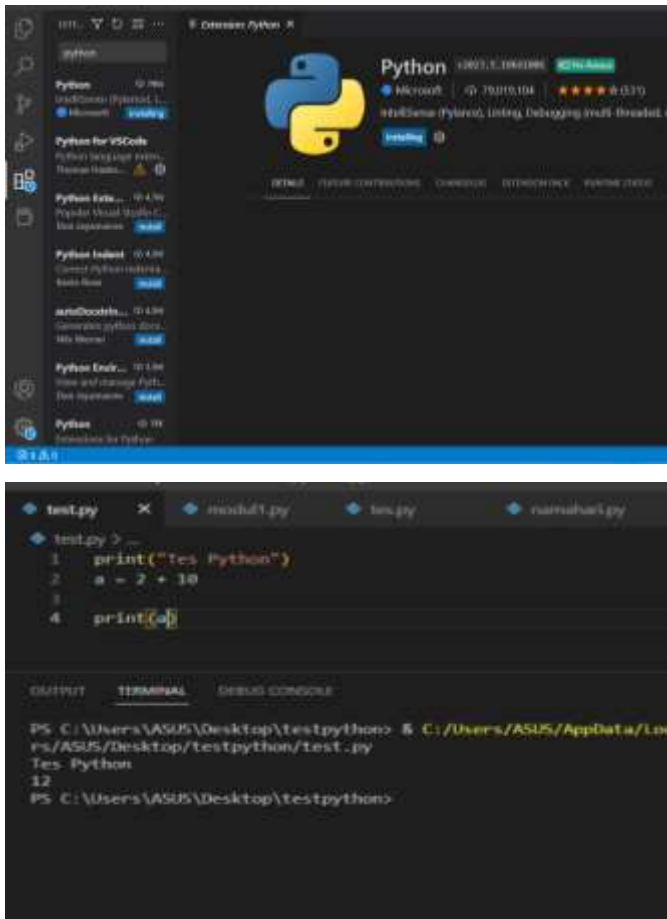
Nama : Novitasari

Nim : 20.01.013.012

MK : Pemrograman Python\_D

## 1. Python Instalasi

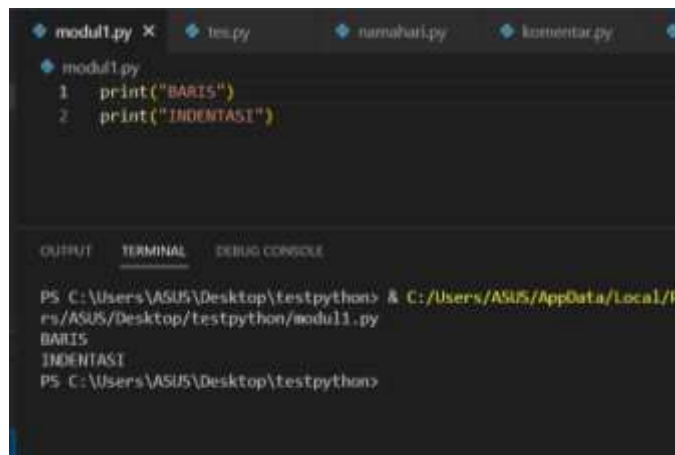
Sebelum memulai membuat codingan di python langkah pertama yang dilakukan yaitu menginstall python, dan juga menginstall fitur-fitur yang ada di python.



## 2. Python Modul 1

### a. Baris dan Indentasi

Adalah kode pada python yang menggunakan spasi, dan spasi yang digunakan pada baris harus sama. Contohnya:



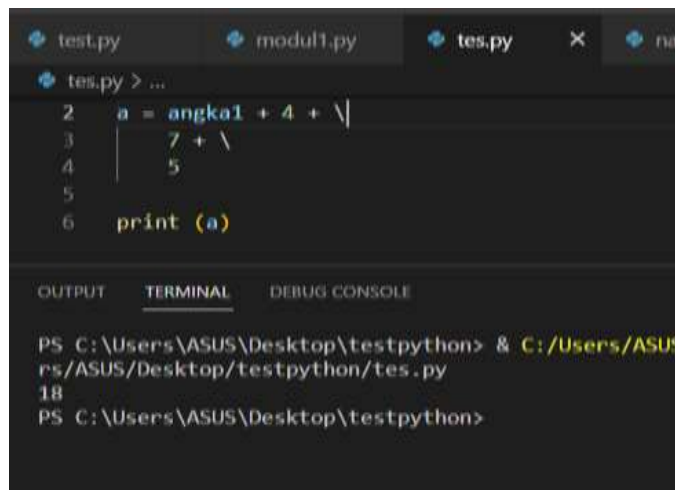
```
modul1.py
1 print("BARIS")
2 print("INDENTASI")

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\modul1.py
BARIS
INDENTASI
PS C:\Users\ASUS\Desktop\testpython>
```

## b. Pernyataan Multibaris

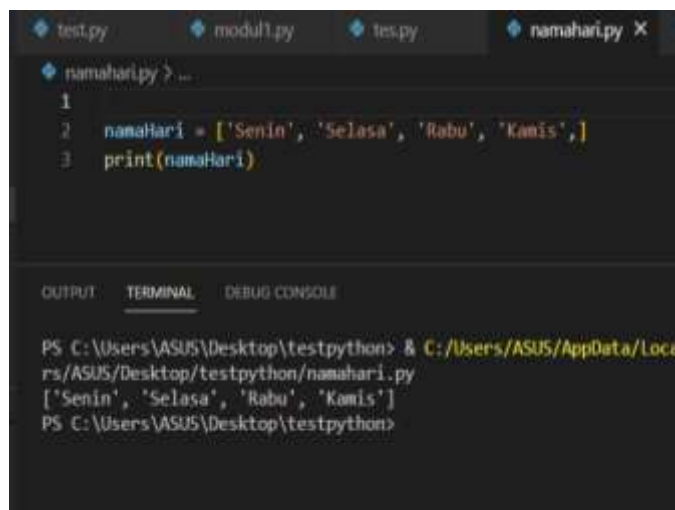
Di python, akhir dari sebuah statement adalah karakter baris baru, ditandai dengan penggunaan tanda backlash [\]. Contohnya:



```
tes.py
2 a = angka1 + 4 + \
3     7 + \
4     5
5
6 print (a)

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\tes.py
18
PS C:\Users\ASUS\Desktop\testpython>
```

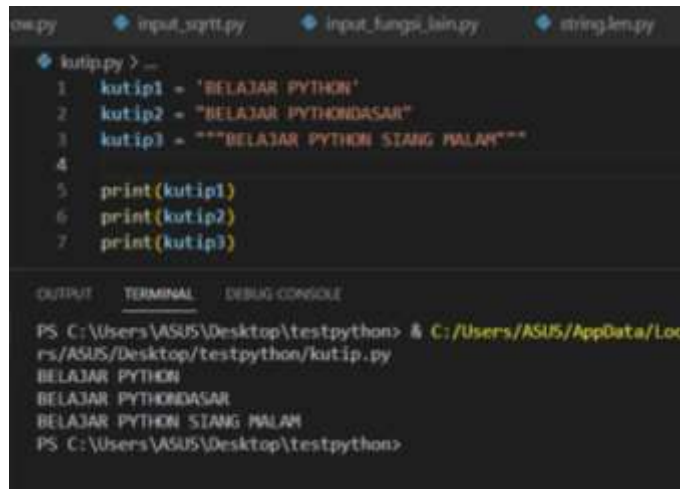


```
namahari.py
1
2 namaHari = ['Senin', 'Selasa', 'Rabu', 'Kamis',]
3 print(namaHari)

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\namahari.py
['Senin', 'Selasa', 'Rabu', 'Kamis']
PS C:\Users\ASUS\Desktop\testpython>
```

### c. Tanda Kutip



The screenshot shows a Python IDE with a file named `kutip.py`. The code defines three string variables: `kutip1` as a single-quoted string, `kutip2` as a double-quoted string, and `kutip3` as a triple-quoted string. Each variable is then printed. The terminal output shows the execution of the script, displaying the three strings on separate lines.

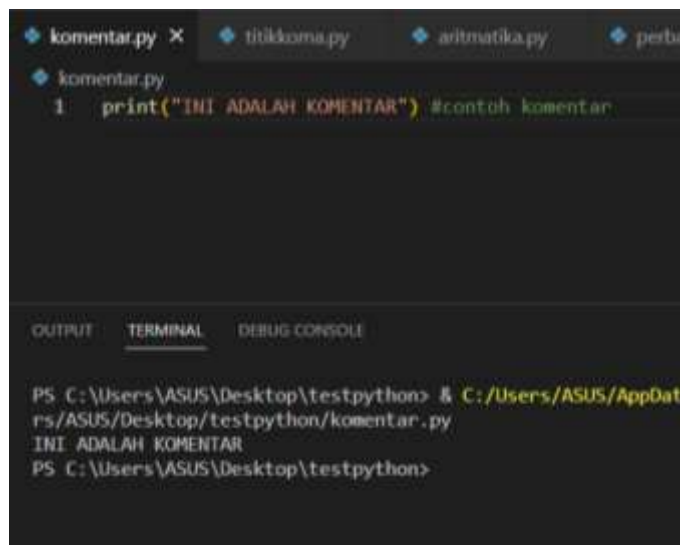
```
1 kutip1 = 'BELAJAR PYTHON'
2 kutip2 = "BELAJAR PYTHONDASAR"
3 kutip3 = """BELAJAR PYTHON SIANG MALAM"""
4
5 print(kutip1)
6 print(kutip2)
7 print(kutip3)
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\kutip.py
BELAJAR PYTHON
BELAJAR PYTHONDASAR
BELAJAR PYTHON SIANG MALAM
PS C:\Users\ASUS\Desktop\testpython>
```

### d. Komentar

Komentar pada python ditandai dengan tanda (#). Contohnya:



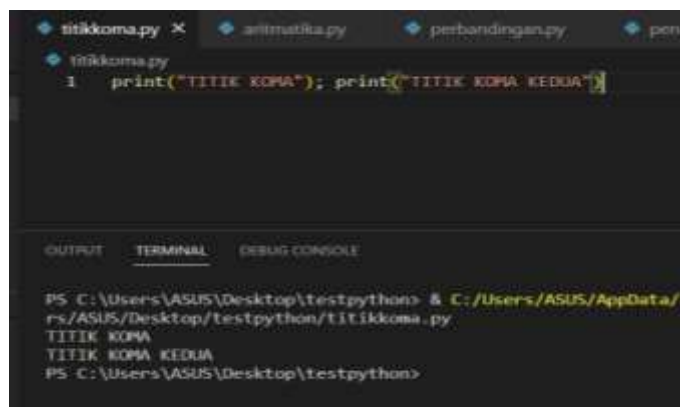
The screenshot shows a Python IDE with a file named `komentar.py`. The code consists of a single line: `print("INI ADALAH KOMENTAR")`, followed by a comment `#contoh komentar`. The terminal output shows the execution of the script, displaying the string "INI ADALAH KOMENTAR".

```
1 print("INI ADALAH KOMENTAR") #contoh komentar
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\komentar.py
INI ADALAH KOMENTAR
PS C:\Users\ASUS\Desktop\testpython>
```

### e. Dua pernyataan dalam satu baris



The screenshot shows a Python IDE with a file named `titikkoma.py`. The code consists of a single line: `print("TITIK KOMA"); print("TITIK KOMA KEDUA")`. The terminal output shows the execution of the script, displaying "TITIK KOMA" on the first line and "TITIK KOMA KEDUA" on the second line.

```
1 print("TITIK KOMA"); print("TITIK KOMA KEDUA")
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\ASUS\Desktop\testpython\titikkoma.py
TITIK KOMA
TITIK KOMA KEDUA
PS C:\Users\ASUS\Desktop\testpython>
```

## f. Operator Aritmatika

```
titikoma.py  aritmatika.py x  perbandingan.py  |
+-----+
+ aritmatika.py > ...
1  a = 10
2  b = 3
3
4  c = a / b
5  d = a ** b
6  e = a // b
7  print("Hasil : ", c)
8  print("Pangkat : ", d)
9  print("Hasil : ", e)

OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData
rs/ASUS/Desktop/testpython/aritmatika.py
Hasil : 3.3333333333333335
Pangkat : 1000
Hasil : 3
PS C:\Users\ASUS\Desktop\testpython> |
```

## g. Operator Perbandingan

```
titikoma.py  aritmatika.py  perbandingan.py x  |
+-----+
+ perbandingan.py > ...
1  a = 10
2  b = 3
3
4  c = a < b
5  print(c)
6
7  c = a > b
8  print(c)
9
10 c = a == b
11 print(c)
12
13 c = a != b
14 print(c)

OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData
rs/ASUS/Desktop/testpython/perbandingan.py
False
True
False
True
PS C:\Users\ASUS\Desktop\testpython>
```

## h. Operator Penugasan

```
titikoma.py  aritmatika.py  perbandingan.py  |
+-----+
+ penugasan.py > ...
1  a = 10
2  b = 3
3
4  a += b
5  print(a)
6
7  a = 10
8  b = 3
9
10 a -= b
11 print(a)

OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/AS
rs/ASUS/Desktop/testpython/penugasan.py
13
7
PS C:\Users\ASUS\Desktop\testpython>
```

### i. Operator Logika

```
titikoma.py  aritmatika.py  perbandingan.py
logika.py > ...
1  a = True
2  b = False
3  c = True
4
5  d = a and c
6  print(d)
7
8  d = a and b
9  print(d)
10
11 d = a or b
12 print(d)
13
14 d = a or c
15 print(d)
```

OUTPUT    TERMINAL    DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/A
rs/ASUS/Desktop/testpython/logika.py
True
False
True
True
PS C:\Users\ASUS\Desktop\testpython>
```

### j. Operator Bitwise

```
titikoma.py  aritmatika.py  perbandingan.py
bitwise.py > ...
1  a = 1
2  b = 2
3
4  c = a | b
5  print(c)
6
7  c = a & b
8  print(c)
9
```

OUTPUT    TERMINAL    DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/User
rs/ASUS/Desktop/testpython/bitwise.py
3
0
PS C:\Users\ASUS\Desktop\testpython>
```

### k. Operator Identitas

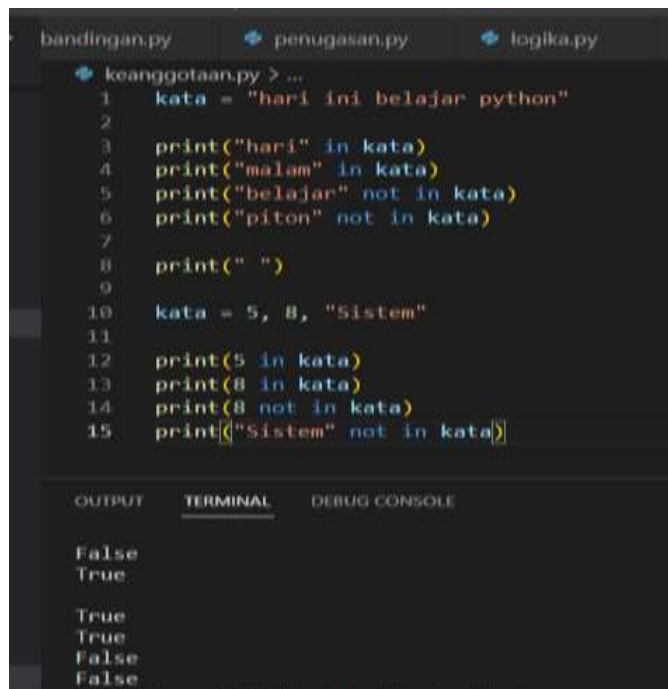
```
identitas.py > ...
1  a = 1
2  b = 2
3
4  print(1 is a)
5  print(2 is b)
6  print(3 is a)
7  print(1 is not a)
8  print(2 is not b)
9
10 print(" ")
11
12 print(type(a) is int)
13 print(type(b) is float)
```

OUTPUT    TERMINAL    DEBUG CONSOLE

```
True
False
False
False
False

True
False
PS C:\Users\ASUS\Desktop\testpython>
```

## 1. Operator Keanggotaan



```
bandingan.py  penugasan.py  logika.py
keanggotaan.py > ...
1  kata = "hari ini belajar python"
2
3  print("hari" in kata)
4  print("malam" in kata)
5  print("belajar" not in kata)
6  print("piton" not in kata)
7
8  print(" ")
9
10 kata = 5, 8, "Sistem"
11
12 print(5 in kata)
13 print(8 in kata)
14 print(8 not in kata)
15 print("Sistem" not in kata)
```

OUTPUT    TERMINAL    DEBUG CONSOLE

```
False
True

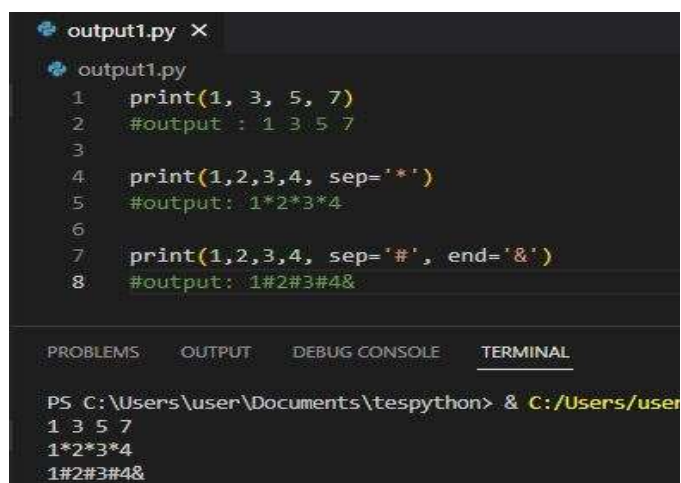
True
True
False
False
```

## 3. Python Module 2

### a. Output

Adalah fungsi bawaan dari pemrograman dan operasi yang digunakan adalah print.

Contohnya:

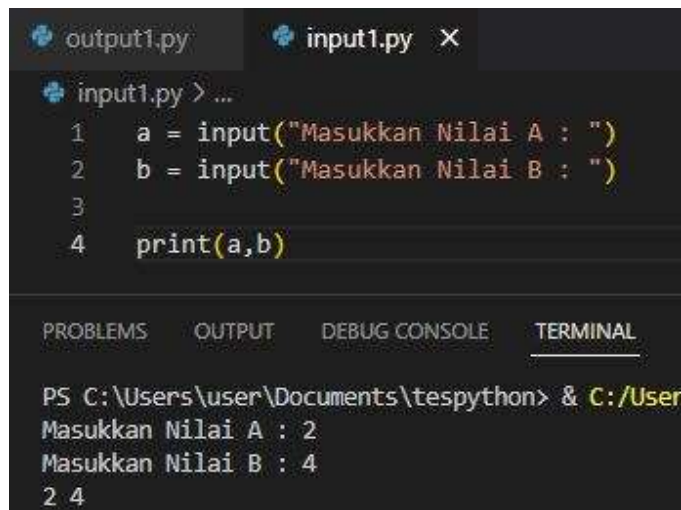


```
output1.py X
output1.py
1  print(1, 3, 5, 7)
2  #output : 1 3 5 7
3
4  print(1,2,3,4, sep='*')
5  #output: 1*2*3*4
6
7  print(1,2,3,4, sep='#', end='&')
8  #output: 1#2#3#4&
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
PS C:\Users\user\Documents\tespython> & C:/Users/user
1 3 5 7
1*2*3*4
1#2#3#4&
```

**b. Input**

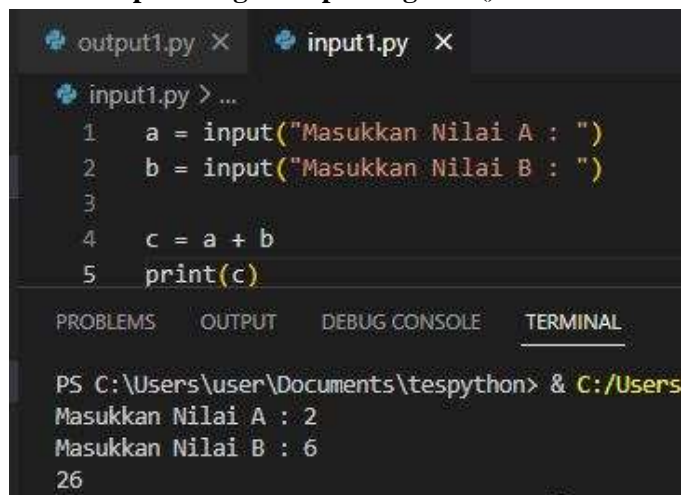


```
output1.py  input1.py X
input1.py > ...
1  a = input("Masukkan Nilai A : ")
2  b = input("Masukkan Nilai B : ")
3
4  print(a,b)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/...  
Masukkan Nilai A : 2  
Masukkan Nilai B : 4  
2 4

- **Input Integer tanpa fungsi int()**



```
output1.py X  input1.py X
input1.py > ...
1  a = input("Masukkan Nilai A : ")
2  b = input("Masukkan Nilai B : ")
3
4  c = a + b
5  print(c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/...  
Masukkan Nilai A : 2  
Masukkan Nilai B : 6  
26

- **Fungsi int () cara pertama**



```
output1.py  input1.py X
input1.py > ...
1  a = int(input("Masukkan Nilai A : "))
2  b = int(input("Masukkan Nilai B : "))
3
4  c = a + b
5  print(c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/A...  
Masukkan Nilai A : 2  
Masukkan Nilai B : 6  
8

- **Fungsi int () cara kedua**

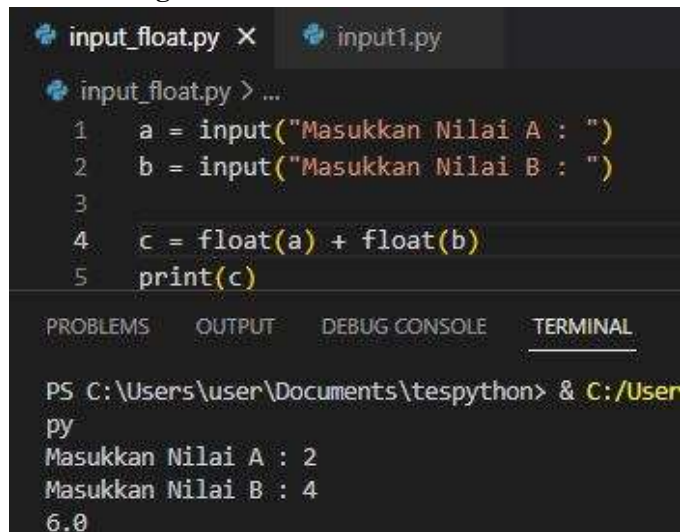


```
output1.py  input1.py X
input1.py > ...
1  a = input("Masukkan Nilai A : ")
2  b = input("Masukkan Nilai B : ")
3
4  c = int(a) + int(b)
5  print(c)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/u
Masukkan Nilai A : 2
Masukkan Nilai B : 4
6
```

- **Fungsi Float**



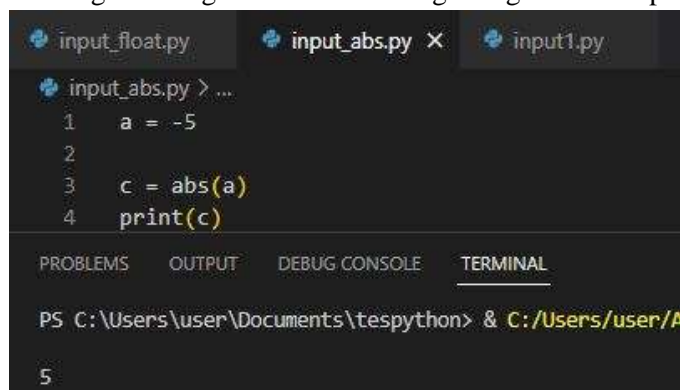
```
input_float.py X  input1.py
input_float.py > ...
1  a = input("Masukkan Nilai A : ")
2  b = input("Masukkan Nilai B : ")
3
4  c = float(a) + float(b)
5  print(c)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\user\Documents\tespython> & C:/User
py
Masukkan Nilai A : 2
Masukkan Nilai B : 4
6.0
```

- **Fungsi abs () statis**

Fungsi abs digunakan untuk menghilangkan minus pada sebuah program



```
input_float.py  input_abs.py X  input1.py
input_abs.py > ...
1  a = -5
2
3  c = abs(a)
4  print(c)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/A
5
```



- **Fungsi abs () dinamis**

```
input_float.py  input_abs.py X  input1.py
input_abs.py > ...
1  a = int(input("Masukkan Nilai A : "))
2
3  c = abs(a)
4  print(c)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData\Local\Microsoft\WindowsApps\python3.9.exe C:\Users\user\Documents\tespython\input_abs.py
Masukkan Nilai A : 2
2
```

- **Fungsi Pow() statis**

```
input_pow.py > ...
1  c = pow(2,3)
2  print(c)

OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & rs/ASUS/Desktop/testpython/testpython/input_pow.py
8
PS C:\Users\ASUS\Desktop\testpython>
```

- **Fungsi pow() dinamis**

```
input_pow.py X
input_pow.py > ...
1  a = int(input("Masukkan Nilai A : "))
2  b = int(input("Masukkan Nilai Pangkat : "))
3
4  c = pow(a,b)
5  print(c)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData\Local\Microsoft\WindowsApps\python3.9.exe C:\Users\user\Documents\tespython\input_pow.py
Masukkan Nilai A : 2
Masukkan Nilai Pangkat : 2
4
```

- Fungsi sqrt() statis

```
input_sqrt.py > ...
1 import math
2
3 c = math.sqrt(36)
4 print(c)
```

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Desktop/testpython/input\_pow.py  
8

- Fungsi sqrt() dinamis

```
input_sqrt.py X
input_sqrt.py > ...
1 import math
2
3 a = input("Masukkan Nilai : ")
4
5 c = math.sqrt(int(a))
6 print(c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/ASUS/Desktop/testpython/input\_sqrt.py  
Masukkan Nilai : 36  
6.0

- Fungsi lain

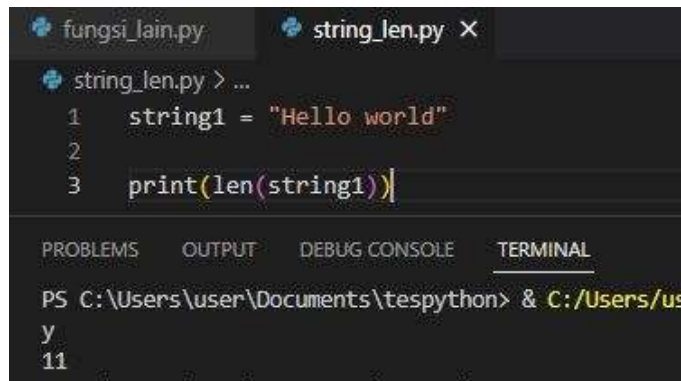
```
input_fungsi_lain.py
1 import math
2
3 print(max(2,1,5)) #outputnya 5
4 print(min(2,1,5)) #outputnya 1
5 print(round(5.8)) #outputnya 6
6 print(math.floor(5.8)) #outputnya 5
7 print(math.ceil(5.8)) #outputnya 6
8
```

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Desktop/testpython/input\_fungsi\_lain.py  
5  
1  
6  
5  
6  
PS C:\Users\ASUS\Desktop\testpython>

### c. Operasi String

- Fungsi len()



The screenshot shows a Python IDE with two tabs: 'fungsi\_lain.py' and 'string\_len.py'. The 'string\_len.py' tab is active, displaying the following code:

```
1 string1 = "Hello world"
2
3 print(len(string1))
```

Below the code editor, the 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\user\Documents\tespython> & C:/Users/us
y
11
```

- Fungsi index()



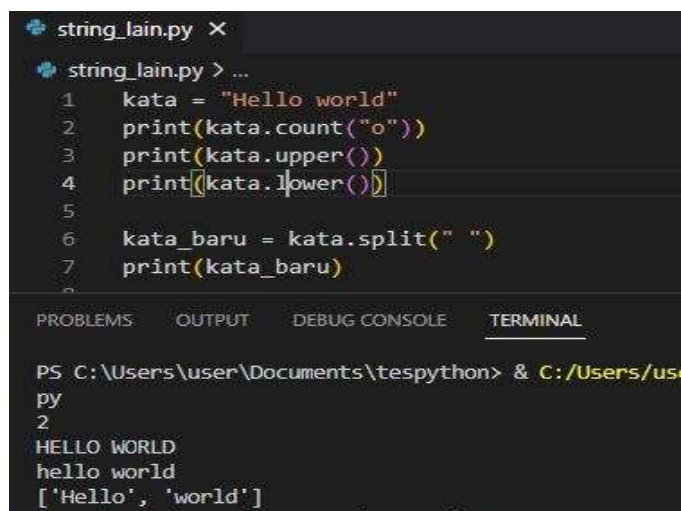
The screenshot shows a Python IDE with two tabs: 'string\_len.py' and 'string\_index.py'. The 'string\_index.py' tab is active, displaying the following code:

```
1 kata = "Hello world"
2 print(kata.index("o"))
```

Below the code editor, the 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\user\Documents\tespython> & C:/U
.py
4
```

- Fungsi string lain



The screenshot shows a Python IDE with a tab 'string\_lain.py'. The code editor displays the following code:

```
1 kata = "Hello world"
2 print(kata.count("o"))
3 print(kata.upper())
4 print(kata.lower())
5
6 kata_baru = kata.split(" ")
7 print(kata_baru)
```

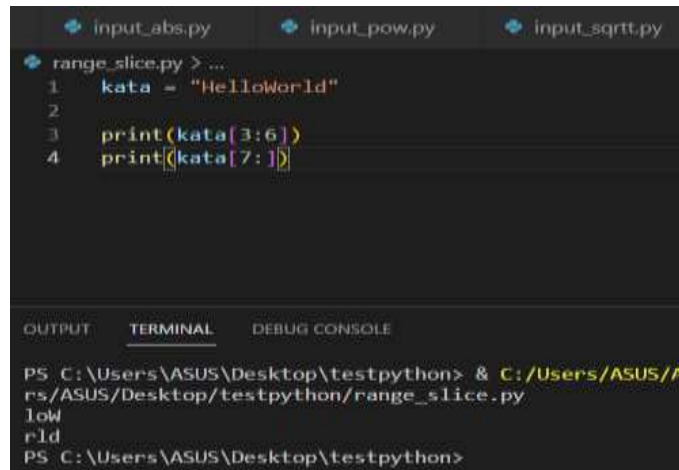
Below the code editor, the 'TERMINAL' tab is selected, showing the command prompt output:

```
PS C:\Users\user\Documents\tespython> & C:/Users/us
py
2
HELLO WORLD
hello world
['Hello', 'world']
```

- **Range slice**

Adalah range karakter dari a mendekati b.

contohnya nama\_variabel [a:b]



The screenshot shows a Python IDE with a file named `range_slice.py` open. The code in the file is as follows:

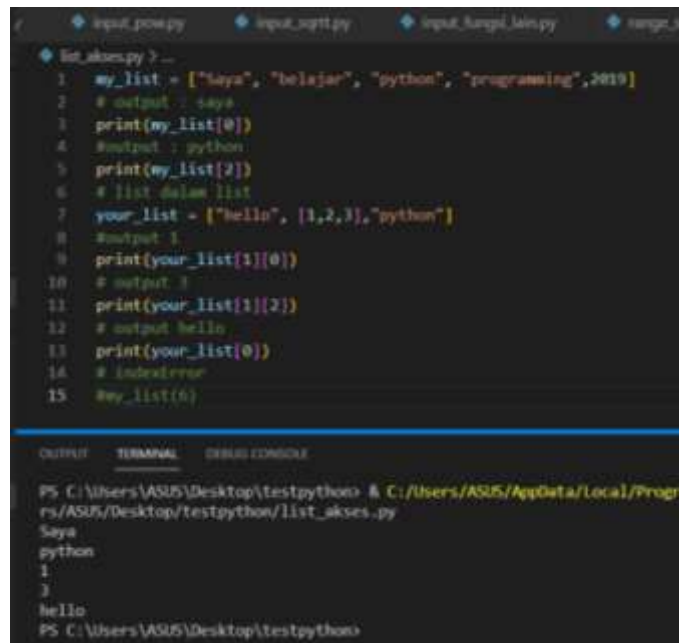
```
1 kata = "HelloWorld"
2
3 print(kata[3:6])
4 print(kata[7:])
```

Below the code editor, the terminal output is displayed:

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python39-64/Python.exe C:/Users/ASUS/Desktop/testpython/range_slice.py
low
rld
PS C:\Users\ASUS\Desktop\testpython>
```

d. List

- **Mengakses anggota list**



The screenshot shows a Python IDE with a file named `list_akses.py` open. The code in the file is as follows:

```
1 my_list = ["saya", "belajar", "python", "programming", 2019]
2 # output : saya
3 print(my_list[0])
4 #output : python
5 print(my_list[2])
6 # list dalam list
7 your_list = ["hello", [1,2,3], "python"]
8 #output 1
9 print(your_list[1][0])
10 # output 1
11 print(your_list[1][2])
12 # output hello
13 print(your_list[0])
14 # IndexError
15 my_list(6)
```

Below the code editor, the terminal output is displayed:

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python39-64/Python.exe C:/Users/ASUS/Desktop/testpython/list_akses.py
saya
python
1
3
hello
PS C:\Users\ASUS\Desktop\testpython>
```

- List dengan indeks negative

```
list_negatif.py > ...
1 my_list = ['p','y','t','h','o','n']
2 #output: n
3 print(my_list[-1])
4 #output: h
5 print(my_list[-3])
```

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Desktop/testpython/list\_negatif.py

n

h

PS C:\Users\ASUS\Desktop\testpython>

- Memotong list

```
input_fungsi_lain.py range_slice.py list_akses.py list
memotong_list.py > ...
1 my_list = ['p','y','t','h','o','n','s','a','y','a']
2 # anggota list dari 3 s/d 5 (dari h s/d n)
3 print(my_list[3:6])
4
5 # anggota list dari 4 s/d yang terakhir
6 print(my_list[4:])
7
8 # anggota list dari 6 s/d 4
9 print(my_list[:5])
```

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/ASUS/Desktop/testpython/memotong\_list.py

['h', 'o', 'n']

['o', 'n', 's', 'a', 'y', 'a']

['p', 'y', 't', 'h', 'o']

PS C:\Users\ASUS\Desktop\testpython>

- Mengubah anggota list

```
list_ubah_anggota.py > ...
1 # misal ada nilai yang salah
2 ganjil = [1,3,4,7,9]
3 print("Item Awal : ", ganjil)
4
5 # ubah item ke 3 (indeks ke 2)
6 ganjil[2] = 5
7 print(ganjil)
8
9 # mengubah sekali banyak
10 ganjil[2:5] = [11,13,15]
11 print(ganjil)
```

OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Desktop/testpython/list\_ubah\_anggota.py

Item Awal : [1, 3, 4, 7, 9]

[1, 3, 5, 7, 9]

[1, 3, 11, 13, 15]

PS C:\Users\ASUS\Desktop\testpython>

- Menambah anggota list

```
list_tambah_anggota.py > ...
1 ganjil = [1,3,5,7]
2
3 ganjil.append(9)
4 print(ganjil)
5 [1,3,5,7,9]
6
7 ganjil.extend([11,13,15])
8 print(ganjil)
9 [1,3,5,7,9,11,13,15]
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/ASUS/Desktop/testpython/list_tambah_anggota.py
[1, 3, 5, 7, 9]
[1, 3, 5, 7, 9, 11, 13, 15]
PS C:\Users\ASUS\Desktop\testpython>
```

- Menggabungkan list dengan operator

```
list_gabung.py > ...
1 genap = [2,4,6]
2 print(genap + [8, 10, 12])
3 # output (2, 4, 6, 8, 10, 12)
4
5 print(['p', 'y'] * 2)
6 # output ['p', 'y', 'p', 'y']
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/ASUS/Desktop/testpython/list_gabung.py
[2, 4, 6, 8, 10, 12]
['p', 'y', 'p', 'y']
PS C:\Users\ASUS\Desktop\testpython>
```

- Menyisipkan anggota list

```
list_insert.py > ...
1 ganjil = [5,7,11,13,15]
2
3 #menyisipkan 9 setelah angka 7
4 ganjil.insert(2,9)
5 print(ganjil)
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/ASUS/Desktop/testpython/list_insert.py
[5, 7, 9, 11, 13, 15]
PS C:\Users\ASUS\Desktop\testpython>
```

- Menghapus anggota list

```
list_hapus_anggota.py > ...
1 my_list = ['p', 'y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']
2 my_list.remove('p')
3 # output ['y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']
4 print(my_list)
5
6 my_list = ['p', 'y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']
7 my_list.remove('n')
8 # remove hanya menghapus elemen pertama yang dijumpai
9 # output: ['p', 'y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']
10 print(my_list)
11
12 # output 'y'
13 print(my_list.pop(1))
14
15 del my_list[2]
16 print(my_list)
17
18 my_list.clear()
19 # output []
20 print(my_list)
```

OUTPUT TERMINAL DEBUG CONSOLE

```
['p', 't', 'o', 's', 'a', 'y', 'a']
[]
PS C:\Users\ASUS\Desktop\testpython>
```

- Mengurutkan anggota list

```
list_urut_anggota.py > ...
1 alfabet = ['a', 'b', 'd', 'f', 'e', 'c', 'h', 'g', 'j', 'i']
2 alfabet.sort()
3 print(alfabet)
4 # output ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
5
6 alfabet.sort(reverse=True)
7 print(alfabet)
8 # output ['j', 'i', 'h', 'g', 'f', 'e', 'd', 'c', 'b', 'a']
```

OUTPUT TERMINAL DEBUG CONSOLE

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
['j', 'i', 'h', 'g', 'f', 'e', 'd', 'c', 'b', 'a']
PS C:\Users\ASUS\Desktop\testpython>
```

- Membalik urutan list

```
list_balik_urutan.py > ...
1 alfabet = ['a', 'c', 'd', 'e', 'b']
2 alfabet.reverse()
3 print(alfabet)
4 # output ['b', 'e', 'd', 'c', 'a']
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Desktop/testpython/list_balik_urutan.py
['b', 'e', 'd', 'c', 'a']
PS C:\Users\ASUS\Desktop\testpython>
```



## 4. Python Module 3

### a. Tuple

- Membuat tuple

```
tuple_bersarang.py > ...
1 # membuat tuple kosong
2 # output: ()
3 tuple1 = ()
4 print(tuple1)
5 # tuple dengan 1 elemen
6 # output: (1,)
7 tuple1 = (1,)
8 print(tuple1)
9 # tuple berisi integer
10 # output: (1, 2, 3)
11 tuple1 = (1, 2, 3)
12 print(tuple1)
13 # tuple bersarang
14 # output: ('hello', [1,2,3], (4,5,6))
15 tuple1 = ('hello', [1,2,3], (4,5,6))
16 print(tuple1)
17 # tuple bisa tidak menggunakan tanda ()
18 # output: (1,2,3)
19 tuple1 = 1,2,3
20 print(tuple1)
21 # memasukkan anggota tuple ke variabel yang bersesuaian
22 # a akan berisi 1, b berisi 2, dan c berisi 3
23 # output: 1 2 3
24 a, b, c = tuple1
25 print(a, b, c)
```

```
OUTPUT TERMINAL DEBUG CONSOLE
1 2 3 > & C:/Users/ASUS/AppData/
rs/ASUS/Desktop/testpython/tuple_bersarang.py
()
(1,)
(1, 2, 3)
('hello', [1, 2, 3], (4, 5, 6))
(1, 2, 3)
1 2 3
PS C:\Users\ASUS\Desktop\testpython>
```

- Mengakses tuple

```
tuple_akses.py x
tuple_akses.py > ...
1 tuple1 = ('p','y','t','h','o','n')
2 # Output: 'p'
3 print(tuple1[0])
4
5 # Output: 'y'
6 print(tuple1[1])
7
8 # Output: 'n'
9 print(tuple1[-1])
10
11 # Output: 'o'
12 print(tuple1[-2])
13
14 # IndexError
15 # print(tuple1[6])
16
17 #Gambar 52 Mengakses Tuple
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\user\Documents\testpython> & C:/Users/user/AppData/Local/Prog
py
p
y
n
o
PS C:\Users\user\Documents\testpython>
```



- Mengakses tuple dengan range

```

tuple_dalam_range.py > ...
1 tuple1 = ('p','r','o','g','r','a','m','m','i','n','g')
2 # akses dari indeks 0 s/d 2
3
4 # output: ('p','r','o')
5 print(tuple1[:3])
6
7 # Akses dari indeks 2 s/d 5
8 # output: ('o','g','r','a')
9 print(tuple1[2:6])
10
11 # Akses dari indeks 3 sampai akhir
12 # output: ('g','r','a','m','m','i','n','g')
13 print(tuple1[3:])
14
15 #Gambar 53 Mengakses Tuple dengan Range

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\user\Documents\tespython> & C:\Users\user\AppData\Local\Program\Python\Python38\python.exe tuple_dalam_range.py
('p', 'r', 'o')
('o', 'g', 'r', 'a')
('g', 'r', 'a', 'm', 'm', 'i', 'n', 'g')

```

- Mengubah anggota tuple

```

tuple_ubah_anggota.py > ...
1 tuple1 = (2, 3, 4, [5, 6])
2 # kita tidak bisa mengubah anggota tuple
3 # jika kita hilangkan tanda komentar # pada baris ke 6
4 # akan muncul error: TypeError: 'tuple' object does not support item assignment
5 #tuple1[1] = 9
6 # tapi list di dalam tuple bisa diubah
7 # output: (2, 3, 4, [7, 6])
8 tuple1[3][0] = 7
9 print(tuple1)
10 # tuple bisa diganti secara keseluruhan dengan pengisian kembali
11 tuple1 = ('p','y','t','h','o','n')
12 print(tuple1)
13
14 # anggota tuple juga tidak bisa dihapus menggunakan del
15 # perintah berikut akan menghasilkan error: TypeError
16 # kalau anda menghilangkan tanda komentar #
17
18 del tuple1[0]
19 # kita bisa menghapus tuple keseluruhan
20 del tuple1

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\user\Documents\tespython> & C:\Users\user\AppData\Local\Program\Python\Python38\python.exe tuple_ubah_anggota.py
(2, 3, 4, [7, 6])
('p', 'y', 't', 'h', 'o', 'n')

```

- Menguji anggota tuple

```

tuple_uji_anggota.py > ...
1 tuple1 = (1, 2, 3, 'a','b','c')
2 # menggunakan in
3 # output: True
4 print(3 in tuple1)
5
6 # output: False
7 print('2' in tuple1)
8
9 # output: False
10 print('e' in tuple1)
11
12 # menggunakan not in
13 # output: True
14 print('k' not in tuple1)
15

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\user\Documents\tespython> & C:\Users\user\AppData\Local\Program\Python\Python38\python.exe tuple_uji_anggota.py
True
False
False
True

```

- Literasi pada tuple

```
tuple_literasi.py > ...
1 # Hai Sistem
2 # Hai Informasi
3 # Hai Informasi
4 nama = ('Sistem', 'Informasi')
5 for a in nama:
6     print('Hai', a)
```

OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/Python118/Python.exe C:/Users/ASUS/Desktop/testpython/tuple_literasi.py
Hai Sistem
Hai Informasi
PS C:\Users\ASUS\Desktop\testpython>
```

- Fungsi bawaan tuple

```
tuple_fungsi.py > ...
1 tuple1 = ('p','y','t','o','n','s','a','y','a')
2 # count
3 # output: 2
4 print(tuple1.count('a'))
5
6 # index
7 # Output 4
8 print (tuple1.index('n'))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/Programs/Python/Python118/Python.exe C:/Users/user/Documents/tespython/tuple_fungsi.py
2
4
```

## b. Set

- Membuat set

```
set1.py > ...
1 # set integer
2 set_saya = {1,2,3}
3 print(set_saya)
4 # set dengan menggunakan fungsi set()
5 set_saya = set([1,2,3])
6 print(set_saya)
7
8 # set data campuran
9 set_saya = {1, 2.0, "Python", (3,4,5)}
10 print(set_saya)
11
12 # bila kita mengisi duplikasi, set akan menghilangkan salah satu
13 # output: {1,2,3}
14 set_saya = {1,2,2,2,3,3}
15 print(set_saya)
16
17 # set tidak bisa berisi anggota list
18 # contoh berikut akan muncul error TypeError
19 |
20 set_saya = {1,2,[3,4,5]}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/Programs/Python/Python118/Python.exe C:/Users/user/Documents/tespython/set1.py
{1, 2, 3}
{1, 2, 3}
{1, 2.0, (3, 4, 5), 'Python'}
{1, 2, 3}
Traceback (most recent call last):
  File "C:\Users\user\Documents\tespython\set1.py", line 20, in <module>
    set_saya = {1,2,[3,4,5]}
TypeError: unhashable type: 'list'
```

- Set Kosong

```
set_kosong.py > ...
1 # membuat variabel a dengan 1)
2 a = {}
3 print(type(a))
4 # output <class 'dict'>
5
6 # harus menggunakan fungsi set()
7 a=set()
8 print(type(a))
9 #output <class 'set'>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/y  
<class 'dict'>  
<class 'set'>

- Mengubah Anggota Set

```
set_ubah_anggota.py > ...
1 # membuat set baru
2 set_saya = {1,3,3}
3 print(set_saya)
4
5 # jika kita hilangkan tanda # dari baris 8 akan muncul error TypeError
6 #set_saya[8]
7
8 # menambah satu anggota
9 # output: {1,2,3,4}
10 set_saya.add(4)
11 print(set_saya)
12
13 # menambah beberapa anggota
14 # set akan menghilangkan duplikasi
15 # output: {1,2,3,4,5,6}
16 set_saya.update([3,4,5,6])
17 print(set_saya)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/Programs/Python/Python311/...  
gnta.py  
{1, 2, 3}  
{1, 2, 3, 4}  
{1, 2, 3, 4, 5, 6}

- Menghapus Anggota Set

```
set_hapus_anggota.py > ...
1 # membuat set baru
2 set_saya = {1, 2, 3, 4, 5}
3 print(set_saya)
4
5 # menghapus 4 dengan discard
6 # output: {1, 2, 3, 5}
7 set_saya.discard(4)
8 print(set_saya)
9
10 # menghapus 5 dengan remove
11 # output : {1, 2, 3}
12
13 set_saya.remove(5)
14 print(set_saya)
15
16 # anggota yang mau dihapus tidak ada dalam set
17 # discard tidak akan memunculkan error
18 # output: {1, 2, 3}
19 set_saya.discard(6)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/1...  
ggota.py  
{1, 2, 3, 4, 5}  
{1, 2, 3, 5}  
{1, 2, 3}

- Menghapus Anggota Set Secara Random dengan Pop()

```

set_hapus_anggota2.py > ...
1  # membuat set baru
2  # output: set berisi anggota yang unik
3  set_saya = set("HelloPython")
4  print(set_saya)
5
6  # pop anggota
7  # Output: anggota acak
8  print(set_saya.pop())
9  print(set_saya)
10
11 # pop anggota lainnya
12 # Output: anggota acak
13 print(set_saya.pop())
14 print(set_saya)
15
16 # mengosongkan set
17 # Output: set()
18 set_saya.clear()
19 print(set_saya)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** [Open file in editor \(ctrl + click\)](#)

```

PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/
ggota2.py
{'n', 't', 'p', 'y', 'h', 'o', 'H', 'l', 'e'}
n
{'t', 'p', 'y', 'h', 'o', 'H', 'l', 'e'}
t
{'p', 'y', 'h', 'o', 'H', 'l', 'e'}
set()

```

- Operasi Gabungan

```

set_union.py > ...
1  # Membuat set A and B
2  A = {1, 2, 3, 4, 5}
3  B = {4, 5, 6, 7, 8}
4
5  # Gabungan menggunakan operator
6  # output: {1, 2, 3, 4, 5, 6, 7, 8}
7  print(A | B)
8
9  # Menggunakan fungsi union()
10 # output: {1, 2, 3, 4, 5, 6, 7, 8}
11 A.union(B)
12
13 #output: {1, 2, 3, 4, 5, 6, 7, 8}
14 B.union(A)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

```

PS C:\Users\user\Documents\tespython> & C:/Users/user/
{1, 2, 3, 4, 5, 6, 7, 8}
PS C:\Users\user\Documents\tespython> & C:/Users/user/
{1, 2, 3, 4, 5, 6, 7, 8}

```

- Operasi Irisan

```

set_intersection.py > ...
1  # Membuat set A and B
2  A = {1, 2, 3, 4, 5}
3  B = {4, 5, 6, 7, 8}
4
5  # Irisan menggunakan operator &
6  # output: {4,5}
7  print(A & B)
8  # Menggunakan fungsi Intersection()
9  # output: {4,5}
10 A.intersection(B)
11
12 # output: {4,5}
13 B.intersection(A)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

PS C:\Users\user\Documents\tespython> & C:/Users/user/tespython/set\_intersection.py  
{4, 5}

- Operasi Selisih (Difference) dengan Set

```

set_difference.py > ...
1  # membuat A and B
2  A = {1, 2, 3, 4, 5}
3  B = {4, 5, 6, 7, 8}
4
5  # Menggunakan operator - pada A
6  # Output: {1, 2, 3}
7  print(A - B)
8
9  # Output: {1, 2, 3}
10 A.difference(B)
11
12 # Menggunakan operator - pada B
13 # Output: {6, 7}
14 print(B - A)
15
16 # Output: {6, 7}
17 B.difference(A)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

PS C:\Users\user\Documents\tespython> & C:/Users/user/tespython/set\_difference.py  
{1, 2, 3}  
{6, 7}

- Operasi Komplemen (Symmetric Difference) dengan Set

```

set_symetric_difference.py > ...
1  # membuat A and B
2  A = {1, 2, 3, 4, 5}
3  B = {4, 5, 6, 7, 8}
4
5  # Menggunakan operator ^ pada A
6  # Output: {1, 2, 3, 6, 7, 8}
7  print(A ^ B)
8
9  # Output: {1, 2, 3, 6, 7, 8}
10 A.symmetric_difference(B)
11
12 # Menggunakan operator ^ pada B
13 # Output: {1, 2, 3, 6, 7, 8}
14 print(B ^ A)
15
16 # Output: {1, 2, 3, 6, 7, 8}
17 B.symmetric_difference(A)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

PS C:\Users\user\Documents\tespython> & C:/Users/user/tespython/set\_symetric\_difference.py  
{1, 2, 3, 6, 7, 8}  
{1, 2, 3, 6, 7, 8}

## c. Dictionary

- Membuat Dictionary

```
dictionary1.py > ...
2 dict1= {}
3 print(dict1)
4
5 # dictionary dengan kunci integer
6 dict1 = {1: 'sepatu', 2: 'tas'}
7 print(dict1)
8
9 # dictionary dengan kunci campuran
10 dict1 = {'warna': 'merah', 1: [2,3,5]}
11 print(dict1)
12
13 # membuat dictionary menggunakan fungsi dict()
14 dict1 = dict([('1', 'sepatu'), ('2','bola')])
15 print(dict1)
16
17 dict1 = dict(m=8, n=9, o=10)
18 print(dict1)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/Programs/Python/Python39-64/python.exe dictionary1.py
{}
{1: 'sepatu', 2: 'tas'}
{'warna': 'merah', 1: [2, 3, 5]}
{'1': 'sepatu', '2': 'bola'}
{'m': 8, 'n': 9, 'o': 10}
```

- Mengakses Anggota Dictionary

```
dictionary_akses_anggota.py > _
1 dict_saya = {'nama': 'Budi', 'usia': 27}
2
3 # output: Budi
4 print(dict_saya['nama'])
5
6 # output 27
7 print(dict_saya.get('usia'))
8
9 # Output None
10 print(dict_saya.get('alamat'))
11
12 # Mengakses kunci yang tidak tersedia menyebabkan KeyError
13 #dict_saya['alamat']
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\user\Documents\tespython> & C:/Users/user/AppData/Local/Programs/Python/Python39-64/python.exe dictionary_akses_anggota.py
Budi
27
None
```



- Mengubah Anggota Dictionary

```
dictionary_ubah_anggota.py > ...
1 dict_saya = {'nama': 'Ikhsan', 'usia': 35}
2
3 # mengupdate nilai
4 dict_saya['usia'] = 36
5 # Output: {'nama': 'Ikhsan', 'usia': 36}
6 print(dict_saya)
7
8 # menambah anggota
9 dict_saya['alamat'] = 'Tanjungpinang'
10 # output: {'alamat': 'Tanjungpinang', 'nama': 'Ikhsan', 'usia': 36}
11 print(dict_saya)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\user\Documents\tespython> & C:\Users\user\AppData\Local\Programs\Python\Python38\python.exe dictionary\_ubah\_anggota.py  
{'nama': 'Ikhsan', 'usia': 36}  
{'nama': 'Ikhsan', 'usia': 36, 'alamat': 'Tanjungpinang'}

- Menghapus Anggota Dictionary

```
dictionary_hapus_anggota.py > ...
1 # membuat dictionary baru
2 dict_saya = {1:1, 2:4, 3:9, 4:16, 5:25}
3 # menghapus anggota tertentu
4 # output: 9
5 print(dict_saya.pop(3))
6 # menghapus anggota secara acak
7 # output: (5, 25)
8 print(dict_saya.popitem())
9 # yang tersisa adalah {1:1, 2:4, 4:16}
10 print(dict_saya)
11 # delete 1
12 del dict_saya[2]
13 # Output: {2:4, 4:16}
14 print(dict_saya)
15
16 # menghapus semua anggota
17 dict_saya.clear()
18 # menghapus dictionary dict_saya
19 del dict_saya
20 # Error karena dict saya sudah dihapus
21 #print(dict_saya)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

hapus\_anggota.py  
9  
(5, 25)  
{1: 1, 2: 4, 4: 16}  
{1: 1, 4: 16}