

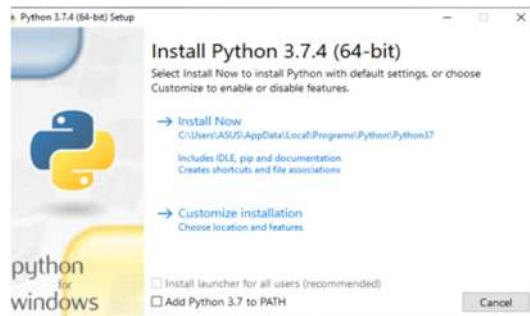
Nama : Nurul Mufliha Puasa

Nim : 20.01.013.014

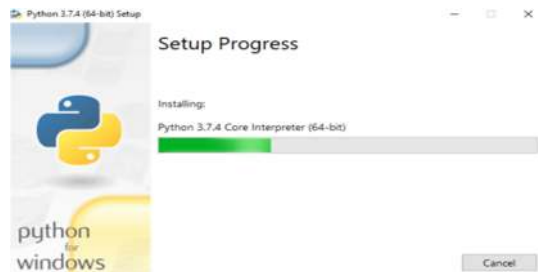
Kelas : C

Modul 1

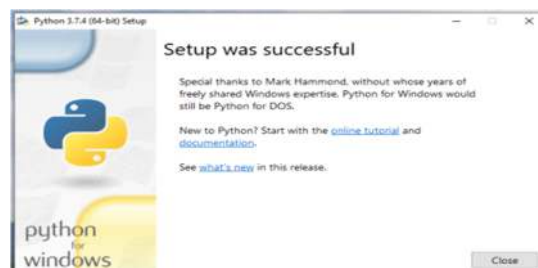
Pilih system python yang sesuai dengan kapasitas Laptop anda. disini saya memakai python(32/64 bit)



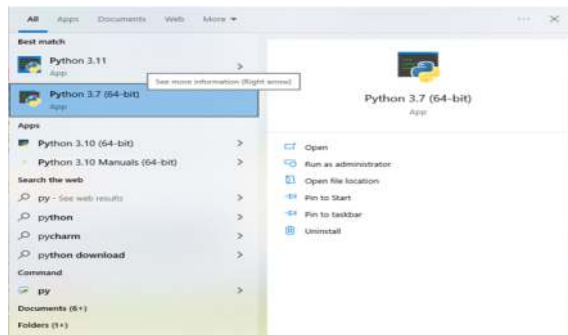
Tunggu beberapa saat hingga proses instalasi selesai.



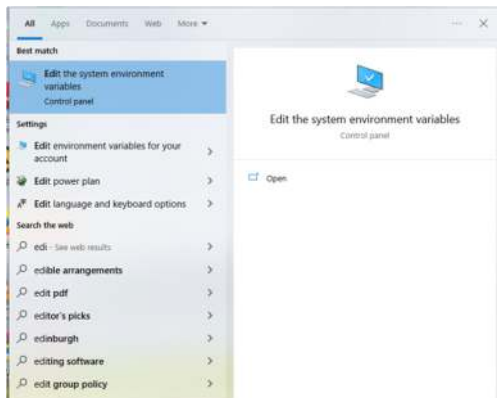
Setelah proses penginstalan sukses klik ok dan klik close agar keluar dari proses penginstalan.



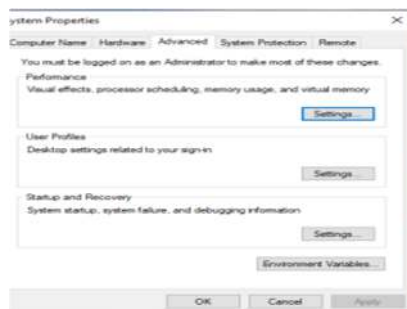
Kalian bisa mengecek apakah python sudah terinstal atau belum di komputer anda

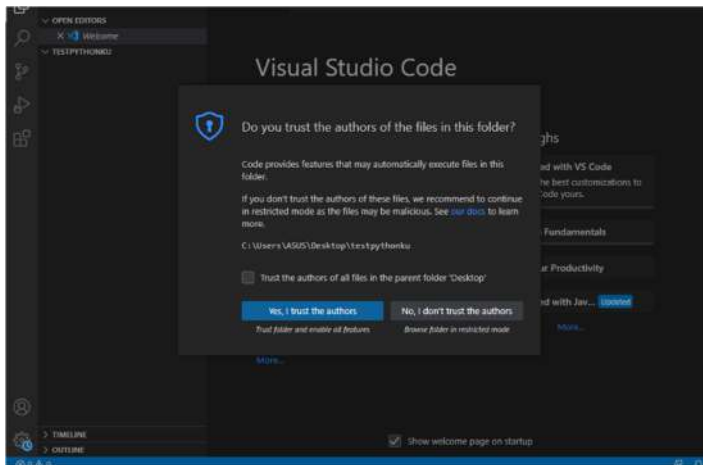


Selanjutnya buka sistem **enviroment variabel** untuk mensetting **path**.bisa di cek di menu searching.

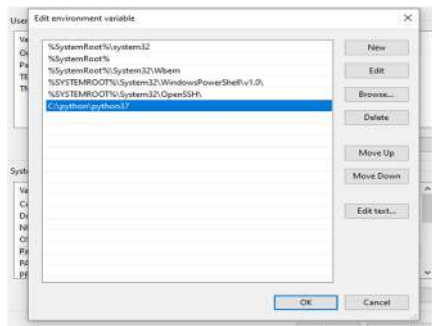


Setelah muncul kotak dialog klik **enviroment variabels** seperti gambar di bawah ini.

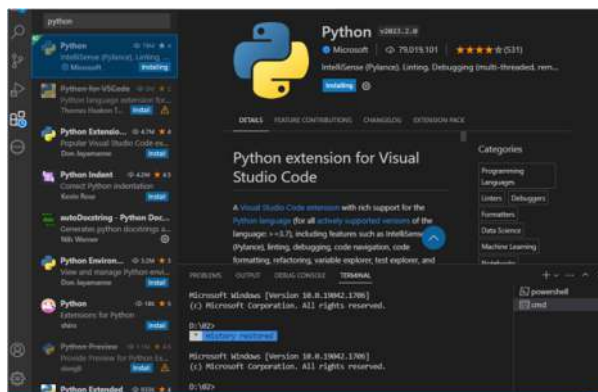




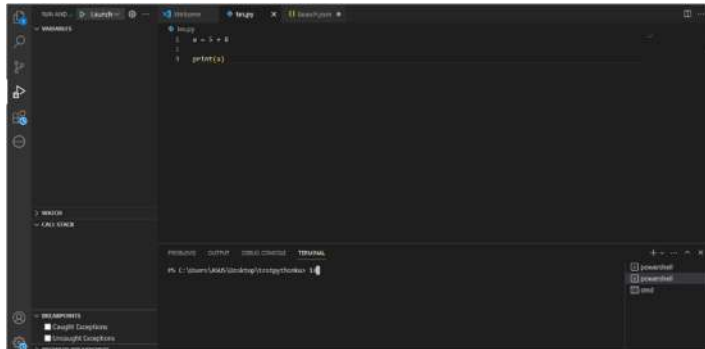
Klik tombol new lalu paste alamat directori yang telah di buat atau di copy



Lalu buka vs code pilih menu **extension** lalu searching python kemudian di install.

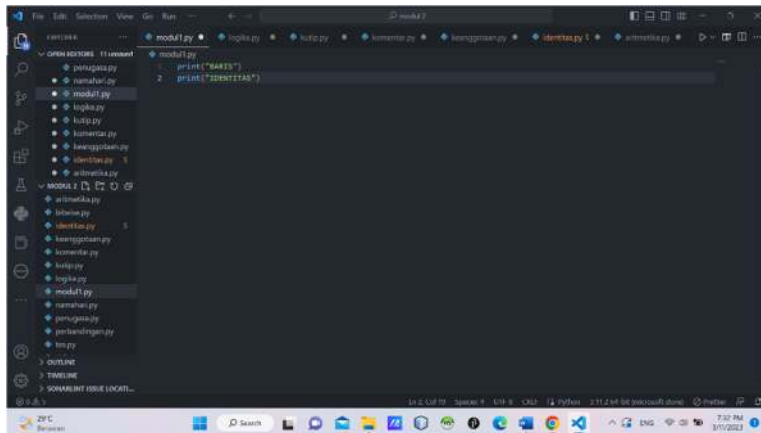


Pada new file lalu buat **folder tes.py**. lalu running project seperti pada gambar di bawah ini.

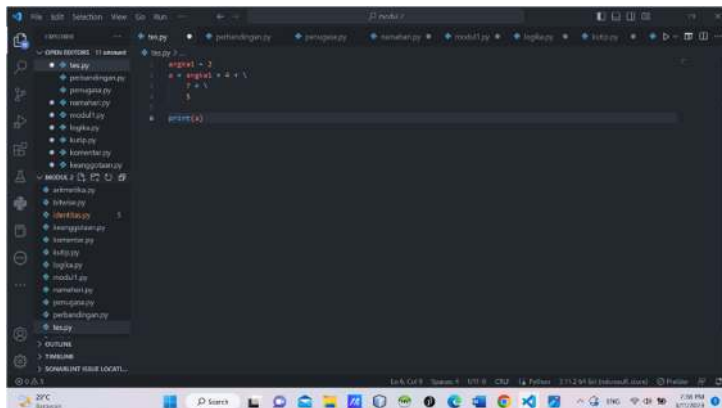


Modul 2

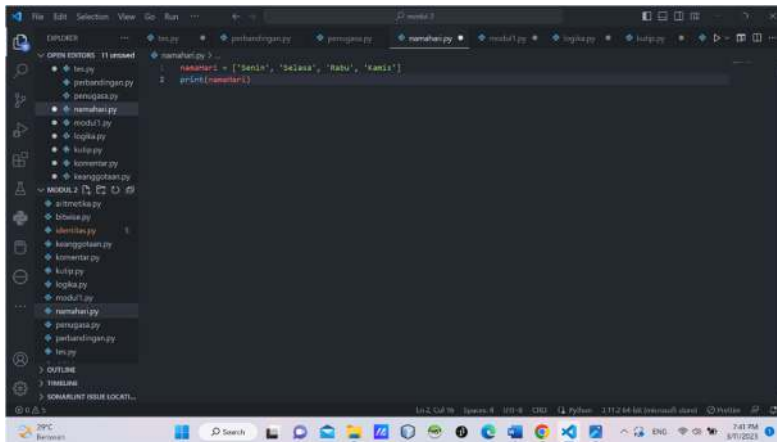
pada modul ini pertama akan membahas baris dan identitas seperti gambar yang ada di bawah ini



selanjutnya masuk ketahap multi baris dengan menggunakan tanda (\)

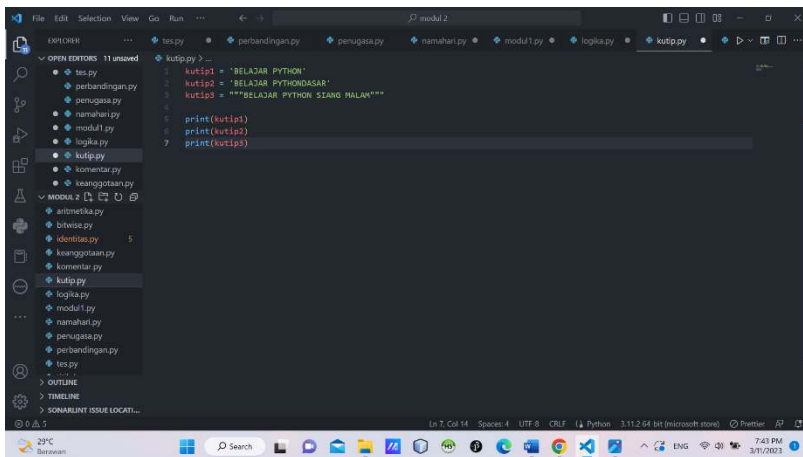


disini ada contoh Latihan yaitu menginput nama-nama hari



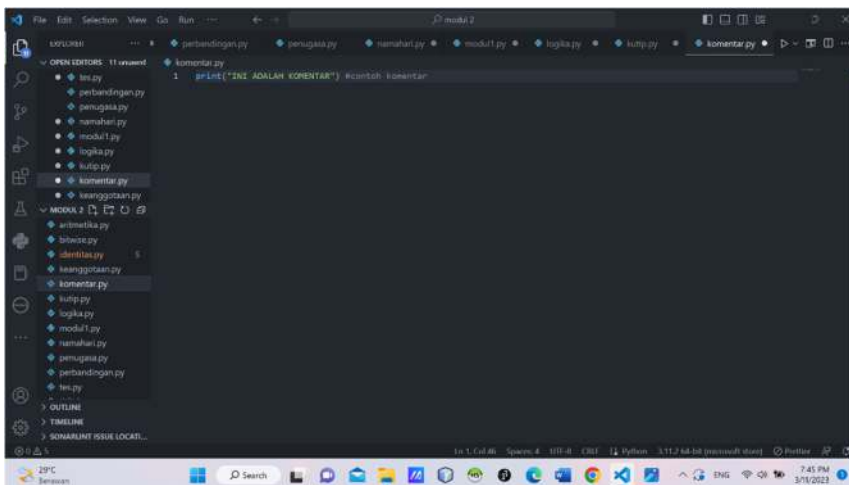
```
nama_hari = ["Senin", "Selasa", "Rabu", "Kamis"]  
print(nama_hari)
```

ada juga meninput kutipan dengan menggunakan ('), (") ataupun (""')



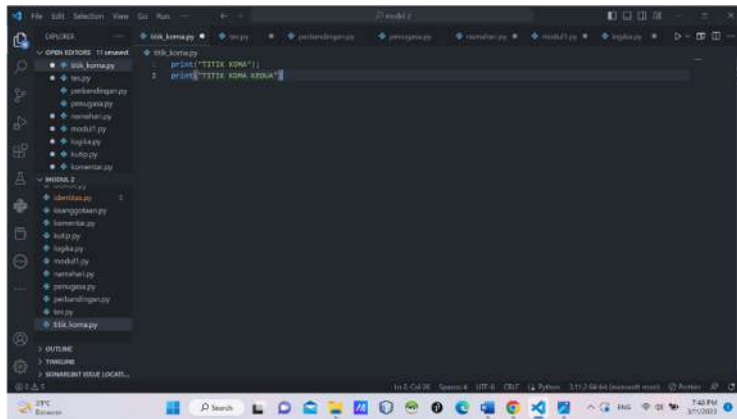
```
kutip1 = 'BELAJAR PYTHON'  
kutip2 = "BELAJAR PYTHONASAR"  
kutip3 = """BELAJAR PYTHON SIANG MALAM"""  
print(kutip1)  
print(kutip2)  
print(kutip3)
```

disini juga kita bisa menulis komentar seperti di bawah ini



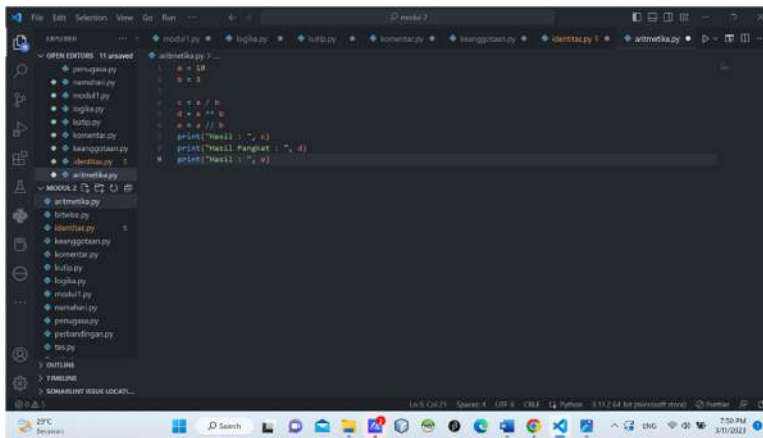
```
print("""INI ADALAH KOMENTAR""") #ini komentar
```

digunakan bila terdapat dua pertanyaan dalam 1 baris kode



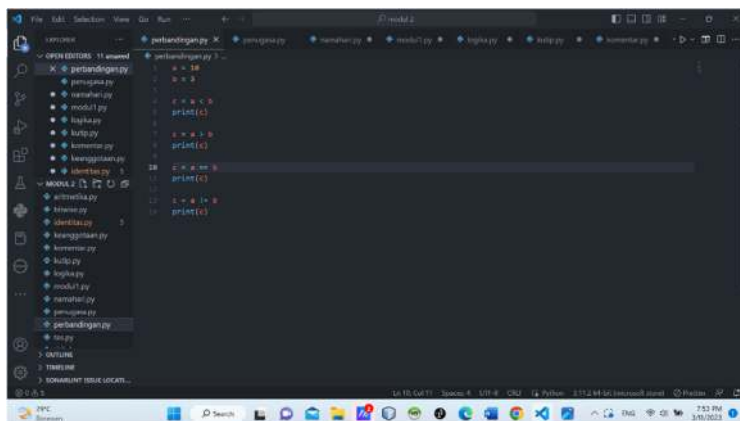
```
1 print("TITIK KONA")  
2 print("TITIK KONA KIRI")
```

Digunakan untuk keperluan matematika seperti penjumlahan, pengurangan dan lain-lain



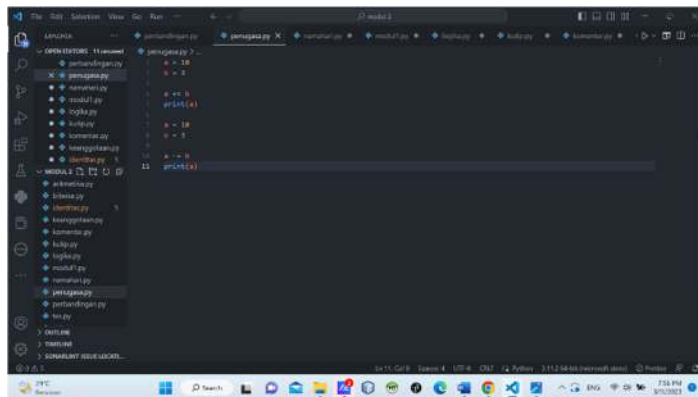
```
1 a = 10  
2 b = 5  
3  
4 c = a + b  
5 d = a - b  
6 e = a // b  
7 print("Hasil : ", c)  
8 print("Hasil Pengkat : ", d)  
9 print("Hasil : ", e)
```

digunakan untuk membandingkan apakah hasilnya true atau false



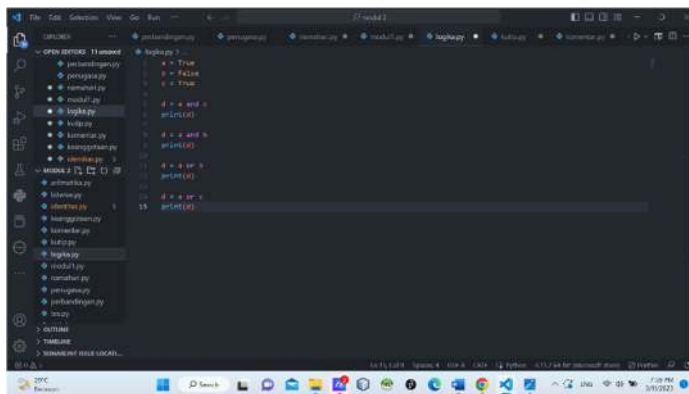
```
1 a = 10  
2 b = 5  
3  
4 c = a < b  
5 print(c)  
6  
7 d = a > b  
8 print(d)  
9  
10 e = a == b  
11 print(e)  
12  
13 f = a != b  
14 print(f)
```

ini di gunakan untu memberi nilai pada variabel



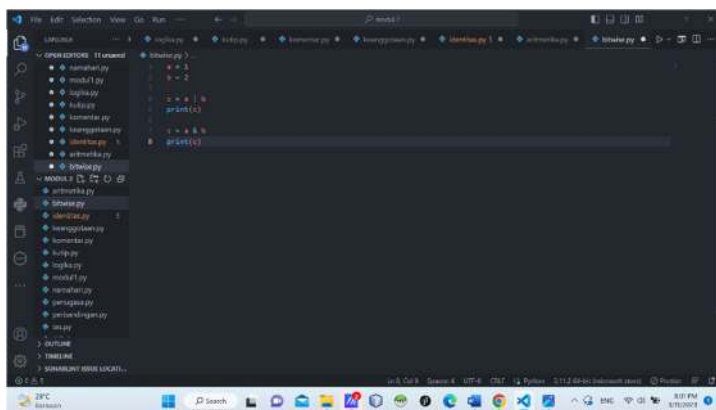
```
peraga.py
a = 10
b = 5
print(a)
print(b)
```

Digunakan untuk mengoprasikan logika



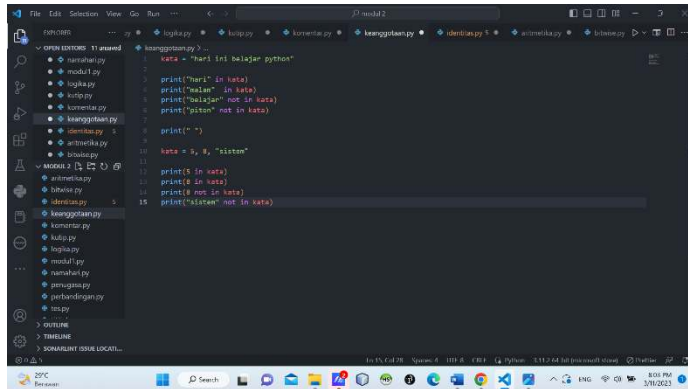
```
logika.py
a = 10
b = 5
print(a > 5 and b > 5)
print(a > 5 or b > 5)
print(not a > 5)
```

Digunakan untuk mengoprasikan bit per bit sesuai dengan Namanya



```
bitwise.py
a = 10
b = 5
print(a & b)
print(a | b)
print(a ^ b)
print(not a)
```

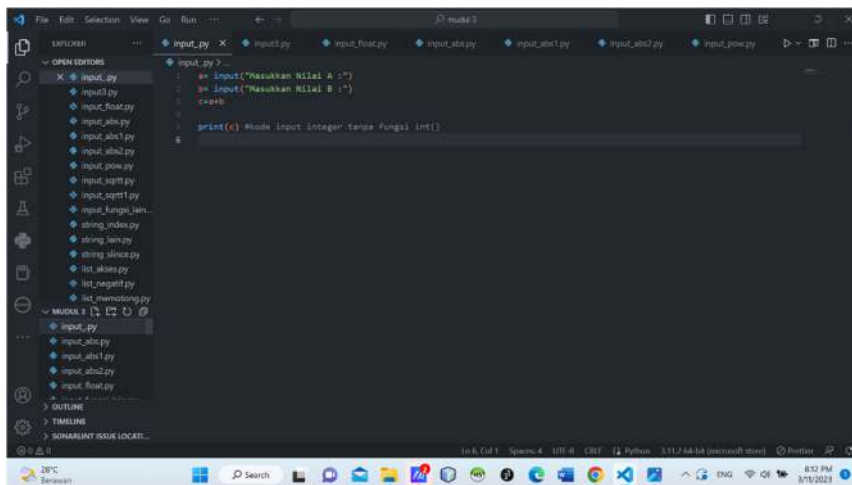
Digunakan untuk memeriksa apakah nilai atau variabel merupakan anggota



```
1 kata = "Hari ini belajar python"  
2  
3 print("Hari" in kata)  
4 print("main" in kata)  
5 print("belajar" not in kata)  
6 print("python" not in kata)  
7  
8 print("-")  
9  
10 kata = 1, 2, "sistem"  
11  
12 print(1 in kata)  
13 print(2 in kata)  
14 print(8 not in kata)  
15 print("sistem" not in kata)
```

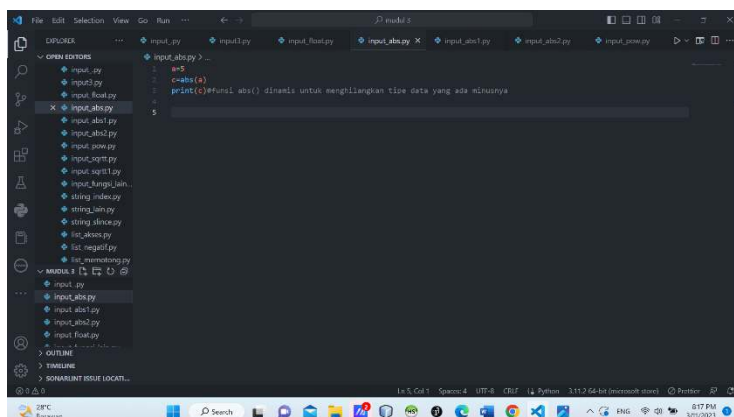
Modul 3

Pertama membahas tentang pengimputan seperti yang ada di bawah ini



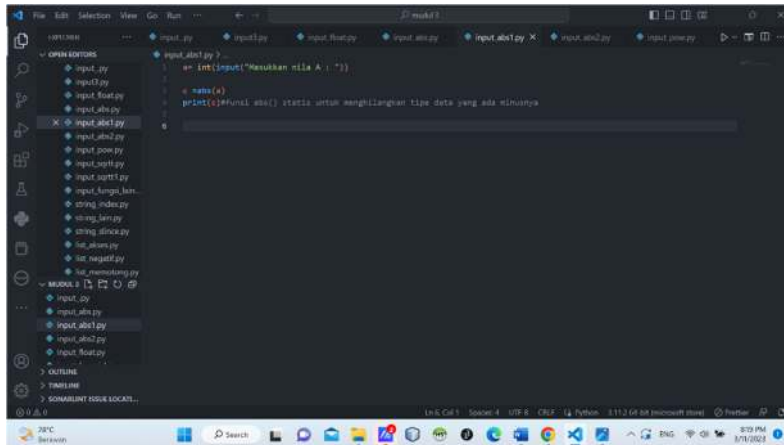
```
1 a = input("Masukkan Nilai A :")  
2 b = input("Masukkan Nilai B :")  
3 c = a + b  
4  
5 print(a) #Maka input integer tanpa fungsi int()
```

Di bawah ini proses input untuk data dinamis yang minus



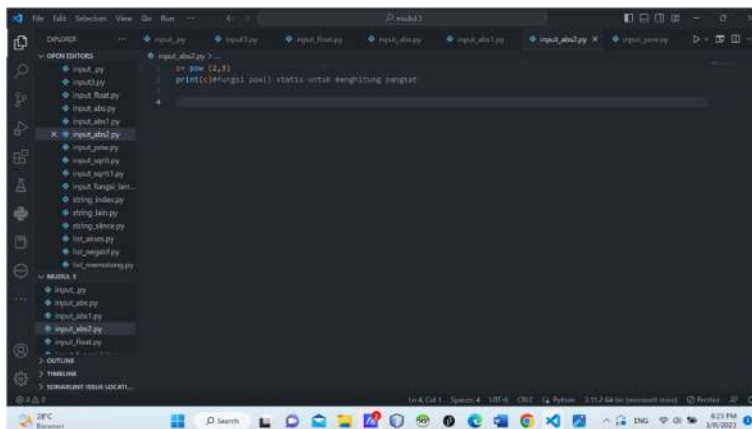
```
1 a = 5  
2  
3 print(a)  
4  
5 print(abs(a)) #Maka abs() dinamis untuk menghilangkan tipe data yang ada minusnya
```


Untuk statis menghilangkan type data yang ada minusnya



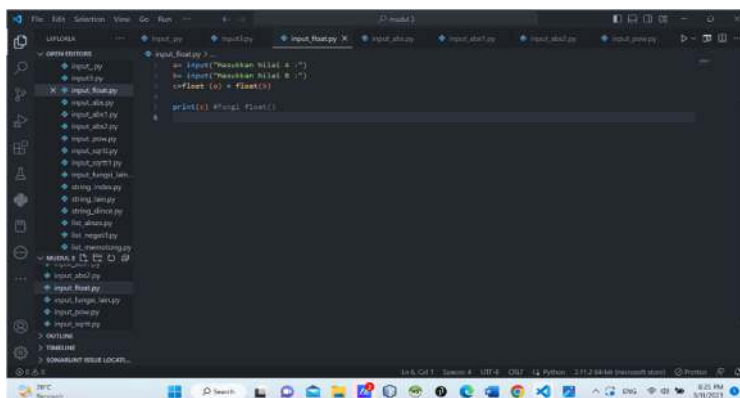
```
input_abs.py >
1 a= int(input("Masukkan nilai A : "))
2
3 a = abs(a)
4 print(a) fungsi abs() statis untuk menghilangkan tipe data yang ada minusnya
```

fungsi pow yang ada statisnya menghitung pangkat



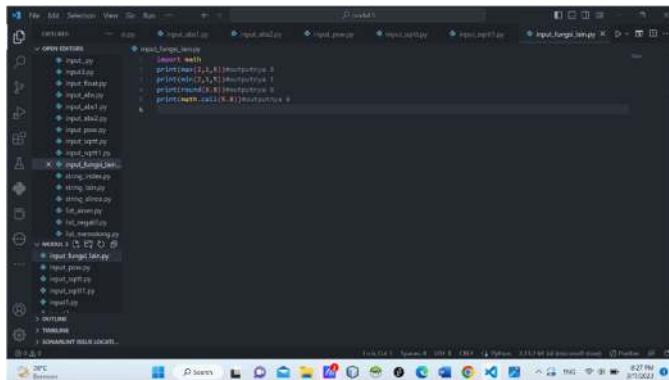
```
input_abs.py >
1 a= pow(2,3)
2
3 print(a) fungsi pow() statis untuk menghitung pangkat
```

Digunakan untuk fungsi float



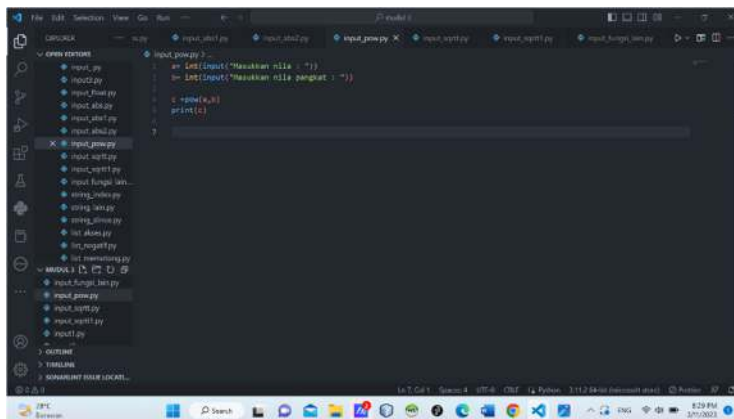
```
input_float.py >
1 a= float(input("Masukkan bilal : "))
2 b= float(input("Masukkan bilal : "))
3 c= float(a) + float(b)
4
5 print(c) fungsi float()
```

Untuk fungsi lain



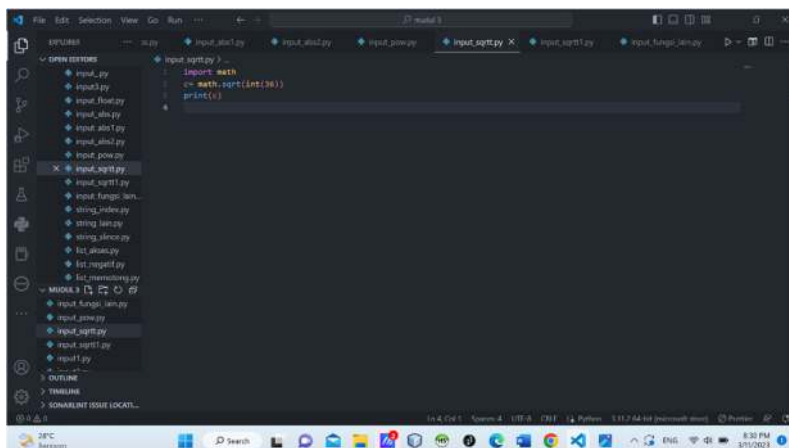
```
import math
def fow(x):
    return math.sqrt(x)
```

Fungsi fow



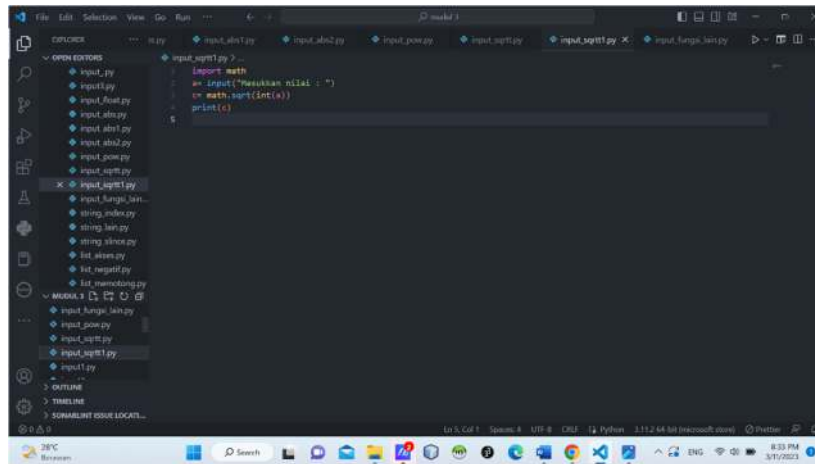
```
def fow(x):
    return math.sqrt(x)
```

Fungsi sqrt statis



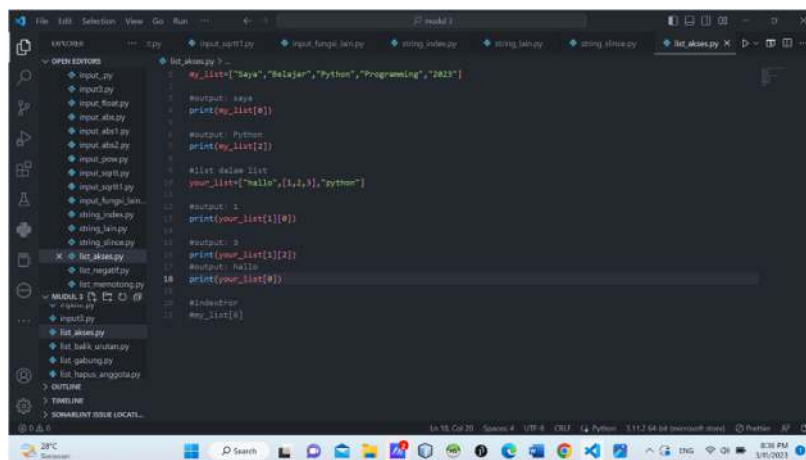
```
import math
def sqrt(x):
    return math.sqrt(x)
```

Fungsi sqrtt dinamis



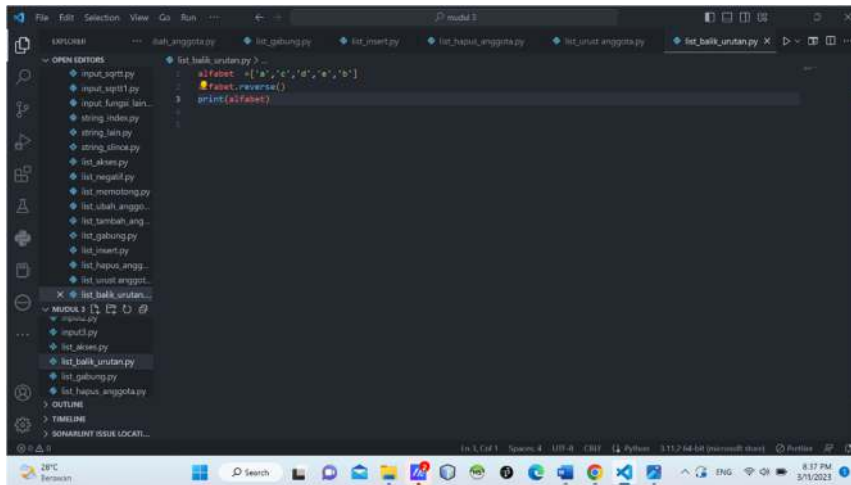
```
File Edit Selection View Go Run ... 27 model 1
input_sqrt1.py
import math
a = input("Masukkan nilai : ")
x = math.sqrt(int(a))
print(x)
```

Fungsi list akses



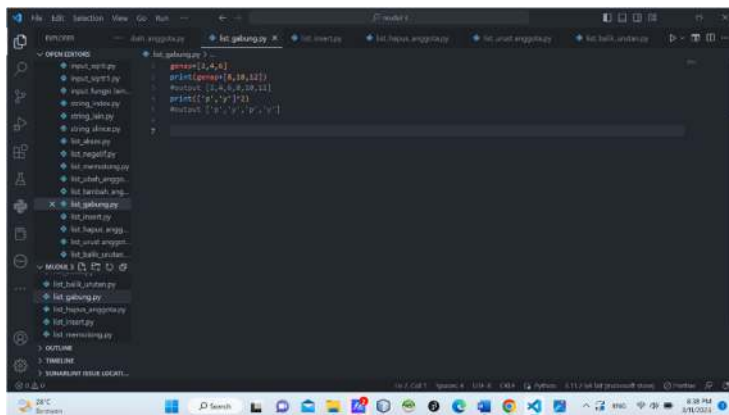
```
File Edit Selection View Go Run ... 27 model 1
list_akses.py
my_list = ["Saya", "Belajar", "Python", "Programming", "2023"]
# Output: saya
print(my_list[0])
# Output: Python
print(my_list[2])
# List index Error
your_list = ["halo", (1,2,3), "python"]
# Output: 1
print(your_list[1][0])
# Output: 3
print(your_list[1][2])
# Output: halo
print(your_list[0])
# IndexError
my_list[6]
```

Fungsi list balik urutan



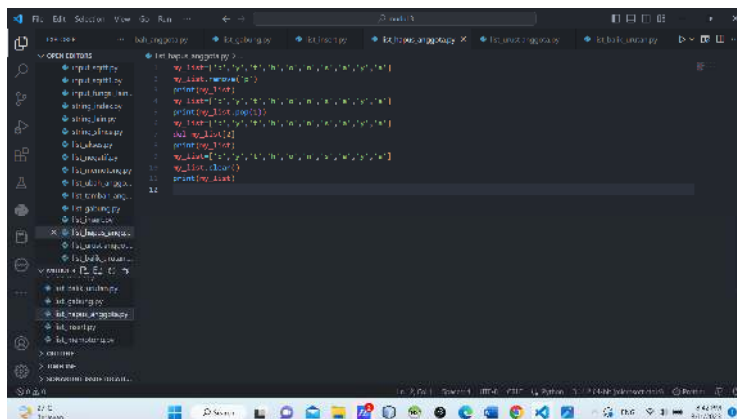
```
def balik_unutan(a):  
    b = a[::-1]  
    return b  
  
a = input("Masukkan list: ")  
b = balik_unutan(a)  
print(b)
```

Fungsi list gabung



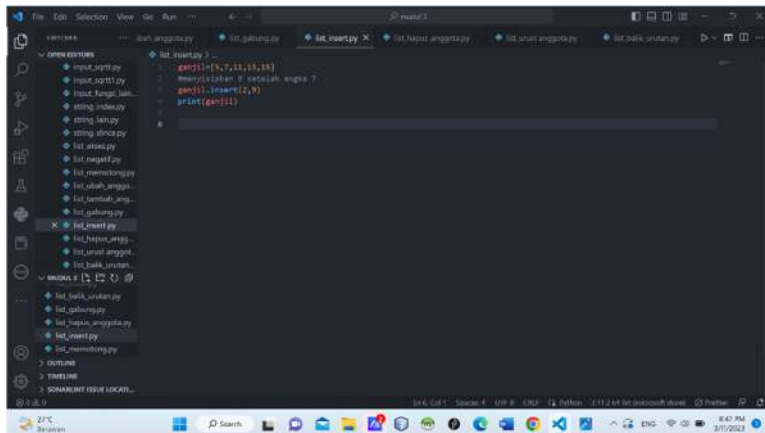
```
def gabung(a, b):  
    c = a + b  
    return c  
  
a = input("Masukkan list 1: ")  
b = input("Masukkan list 2: ")  
c = gabung(a, b)  
print(c)
```

Fungsi untuk menghapus anggota

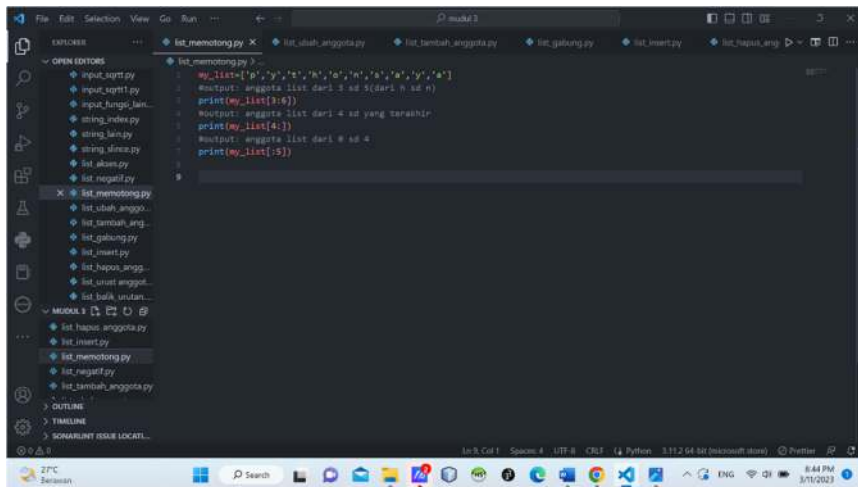


```
def hapus_anggota(a, x):  
    a.remove(x)  
    return a  
  
a = input("Masukkan list: ")  
x = input("Masukkan anggota yang akan dihapus: ")  
b = hapus_anggota(a, x)  
print(b)
```

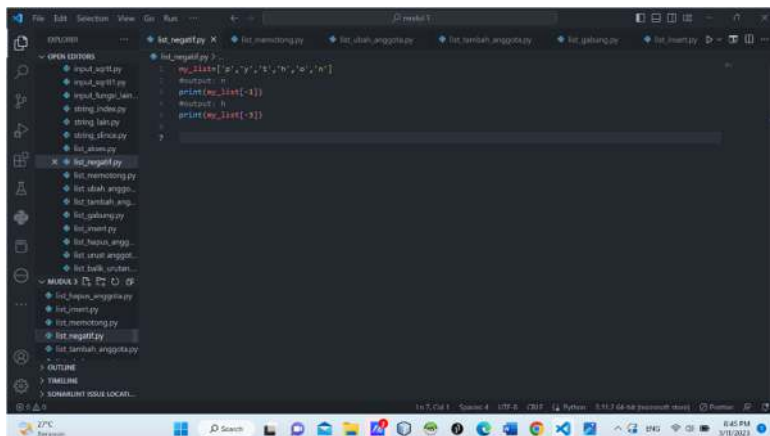
Fungsi list insert



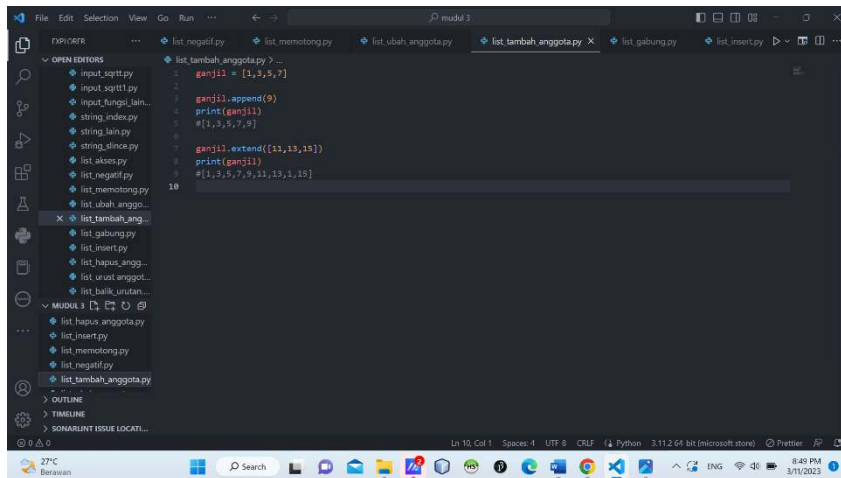
Fungsi list memotong



Fungsi list negatif



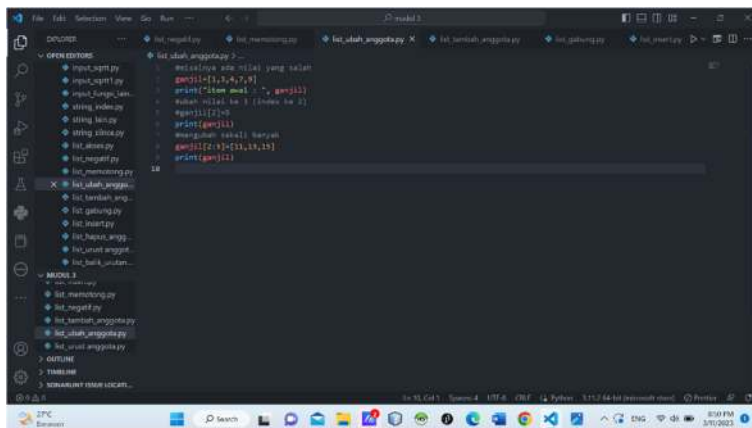
Fungsi list menambah anggota



The screenshot shows a VS Code editor with a Python file named `list_tambah_anggota.py`. The code defines a list `ganjil1` with the values `[1, 3, 5, 7]`. It then uses the `append` method to add the value `9` to the list, followed by a `print` statement. Next, it uses the `extend` method to add the list `[11, 13, 15]` to the end of `ganjil1`, followed by another `print` statement. The file explorer on the left shows a project named `MODUL 3` containing several other Python files related to list operations.

```
1 ganjil1 = [1, 3, 5, 7]
2
3 ganjil1.append(9)
4 print(ganjil1)
5 # [1, 3, 5, 7, 9]
6
7 ganjil1.extend([11, 13, 15])
8 print(ganjil1)
9 # [1, 3, 5, 7, 9, 11, 13, 15]
10
```

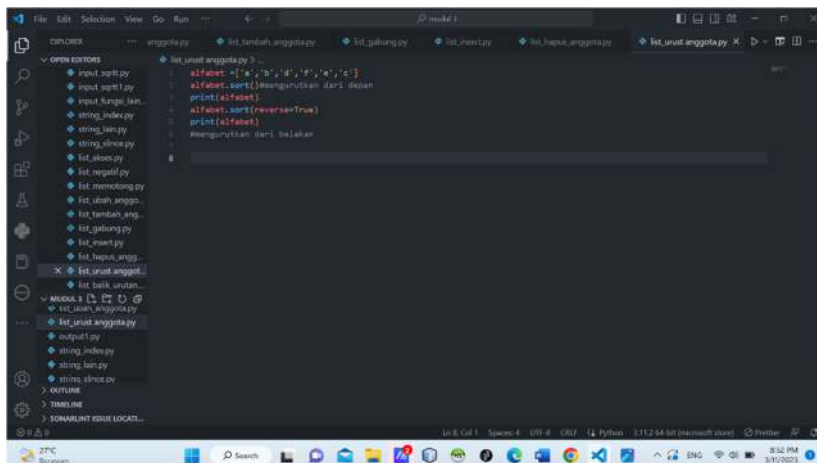
Fungsi list mengubah anggota



The screenshot shows a VS Code editor with a Python file named `list_ubah_anggota.py`. The code defines a list `ganjil1` with the values `[1, 3, 5, 7, 9]`. It then uses the `pop` method to remove the last element (`9`) from the list, followed by a `print` statement. Next, it uses the `insert` method to insert the value `4` at index `2`, followed by a `print` statement. Finally, it uses the `replace` method to replace the value `11` at index `2` with the value `13`, followed by a `print` statement. The file explorer on the left shows a project named `MODUL 3` containing several other Python files related to list operations.

```
1 ganjil1 = [1, 3, 5, 7, 9]
2
3 ganjil1.pop()
4 print(ganjil1)
5 # [1, 3, 5, 7]
6
7 ganjil1.insert(2, 4)
8 print(ganjil1)
9 # [1, 3, 4, 5, 7]
10
11 ganjil1[2] = 13
12 print(ganjil1)
13 # [1, 3, 13, 5, 7]
```

Fungsi list mengurutkan anggota

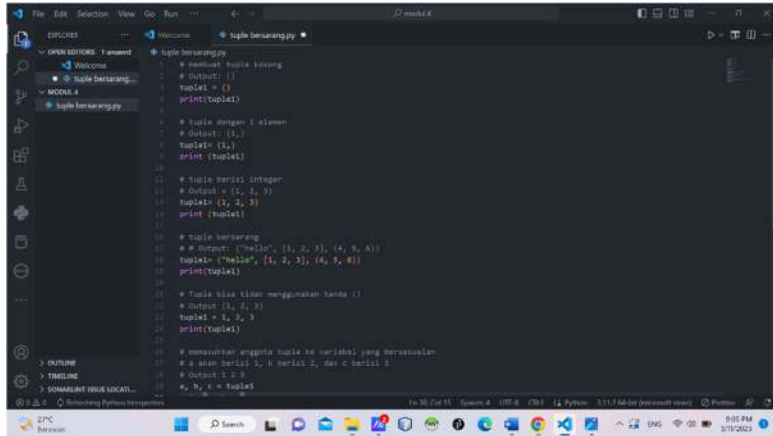


The screenshot shows a VS Code editor with a Python file named `list_urut_anggota.py`. The code defines a list `alfabet` with the values `['a', 'b', 'd', 'f', 'e', 'c']`. It then uses the `sort` method to sort the list in ascending order, followed by a `print` statement. Next, it uses the `sort` method with the `reverse=True` parameter to sort the list in descending order, followed by a `print` statement. The file explorer on the left shows a project named `MODUL 3` containing several other Python files related to list operations.

```
1 alfabet = ['a', 'b', 'd', 'f', 'e', 'c']
2
3 alfabet.sort()
4 print(alfabet)
5 # ['a', 'b', 'c', 'd', 'e', 'f']
6
7 alfabet.sort(reverse=True)
8 print(alfabet)
9 # ['f', 'e', 'd', 'c', 'b', 'a']
```

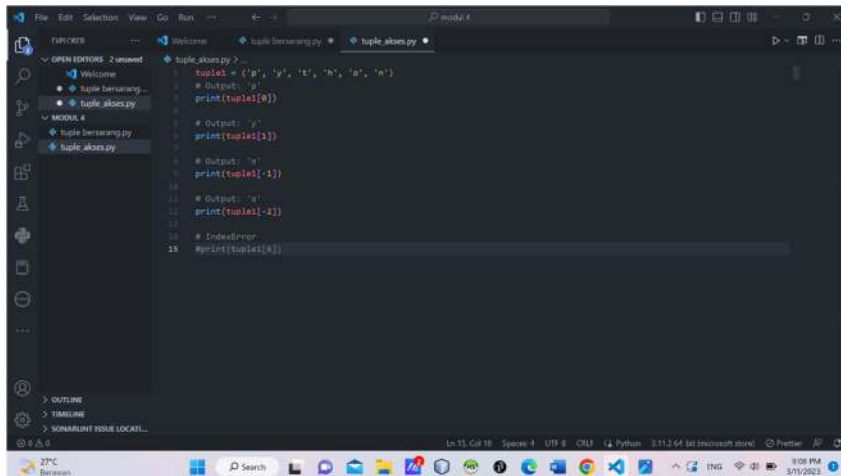
Modul 4

Tuple bersarang digunakan untuk meletakkan tanda () masing-masing dan dipisahkan dengan tanda koma



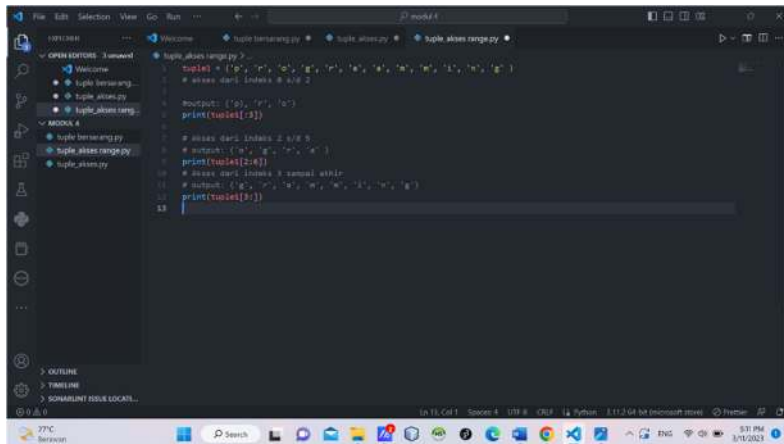
```
1 # membuat tuple kosong
2 # Output: ()
3 tuple1 = ()
4 print(tuple1)
5
6 # tuple dengan 1 elemen
7 # Output: (1,)
8 tuple1 = (1,)
9 print(tuple1)
10
11 # tuple berisi integer
12 # Output: (1, 2, 3)
13 tuple1 = (1, 2, 3)
14 print(tuple1)
15
16 # tuple bersarang
17 # Output: ('hello', (1, 2, 3), (4, 5, 6))
18 tuple1 = ('hello', (1, 2, 3), (4, 5, 6))
19 print(tuple1)
20
21 # Tuple bisa tidak menggunakan tanda ()
22 # Output: (1, 2, 3)
23 tuple1 = 1, 2, 3
24 print(tuple1)
25
26 # menambahkan anggota tuple ke variabel yang bersarang
27 # akan berisi 2, 3 berisi 2, dan 4 berisi 2
28 # Output: 1 > 2
29 # Output: 2 > 2
30 # Output: 4 > 2
```

Tuple akses digunakan untuk untuk format namatuple[[indeks]



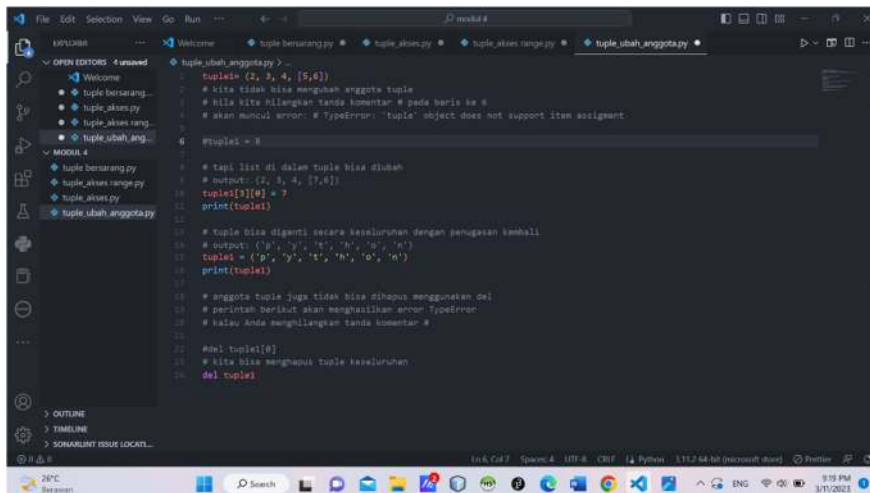
```
1 tuple1 = ('p', 'y', 't', 'h', 'o', 'n')
2 # Output: 'p'
3 print(tuple1[0])
4
5 # Output: 'y'
6 print(tuple1[1])
7
8 # Output: 'n'
9 print(tuple1[-1])
10
11 # Output: 'e'
12 print(tuple1[-2])
13
14 # IndexError
15 print(tuple1[8])
```

Tuple akses range digunakan untuk operator titik dua



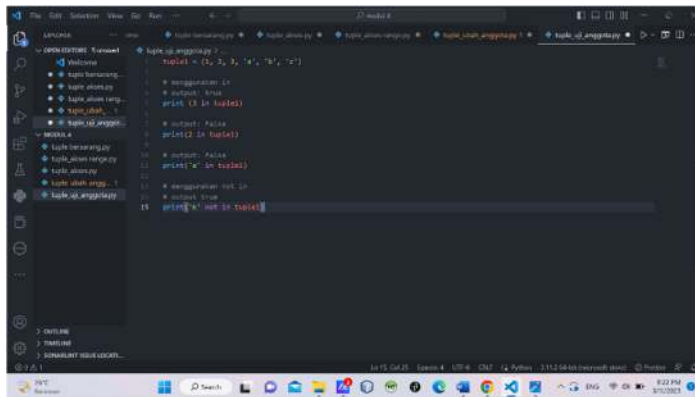
```
1 tuple = ('p', 'r', 'i', 'n', 't', ' ', 'h', 'i', 's', ' ', 'i', 's', ' ', 'a', ' ', 't', 'u', 'p', 'l', 'e')
2 # akses dari index 0 s/d 2
3
4 # output: ('p', 'r', 'i')
5 print(tuple[:3])
6
7 # akses dari index 2 s/d 5
8
9 # output: ('n', 't', ' ', 'h', 'i')
10 print(tuple[2:6])
11
12 # akses dari index 3 sampai akhir
13
14 # output: (' ', 'h', 'i', 's', ' ', 'a', ' ', 't', 'u', 'p', 'l', 'e')
15 print(tuple[3:])
```

Tuple ubah anggota digunakan untuk tipe bersarang



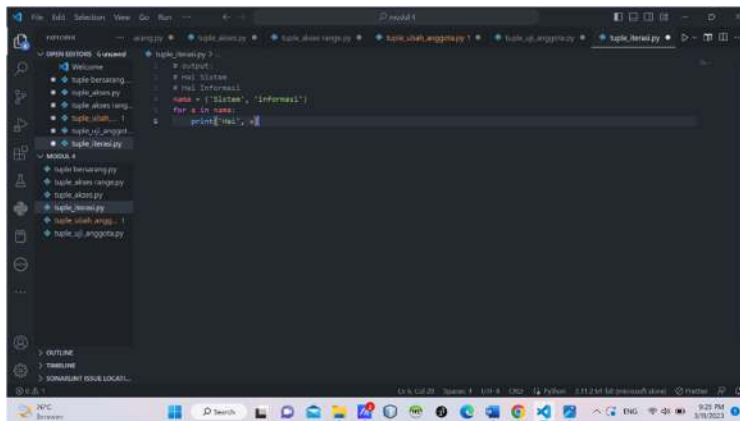
```
1 tuple1 = (2, 3, 4, [5,6])
2 # kita tidak bisa mengubah anggota tuple
3 # kita kita hilangkan tanda komentar # pada baris ke 4
4 # akan muncul error: TypeError: 'tuple' object does not support item assignment
5
6 # tuple1 = 8
7
8 # tapi list di dalam tuple bisa diubah
9 # output: (2, 3, 4, [7,6])
10 tuple1[3][0] = 7
11 print(tuple1)
12
13 # tuple bisa diganti secara keseluruhan dengan pengisian kembali
14 # output: ('p', 'y', 't', 'h', 'i', 'n')
15 tuple1 = ('p', 'y', 't', 'h', 'i', 'n')
16 print(tuple1)
17
18 # anggota tuple juga tidak bisa dihapus menggunakan del
19 # perintah berikut akan menghasilkan error TypeError
20 # kalau Anda menghilangkan tanda komentar #
21
22 #del tuple1[0]
23 # kita bisa menghapus tuple keseluruhan
24 del tuple1
```


Tuple uji anggota



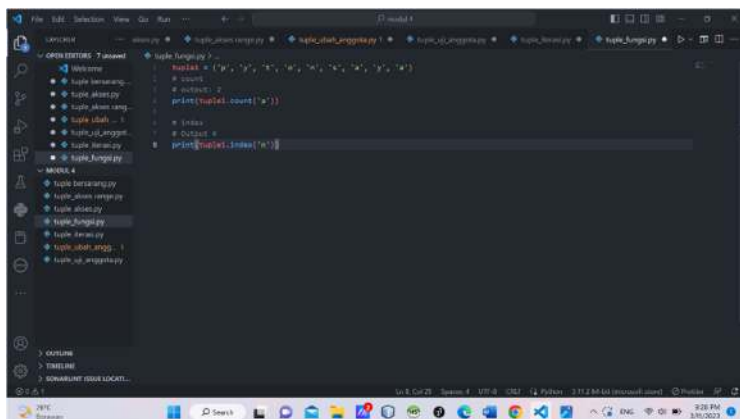
```
1 tuple1 = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z')
2
3 # Menghasilkan list
4 output = True
5 print('a' in tuple1)
6
7 # output: False
8 print('b' in tuple1)
9
10 # output: False
11 print('m' in tuple1)
12
13 # Menghasilkan list
14 output = True
15 print('n' not in tuple1)
```

Tuple iterasi



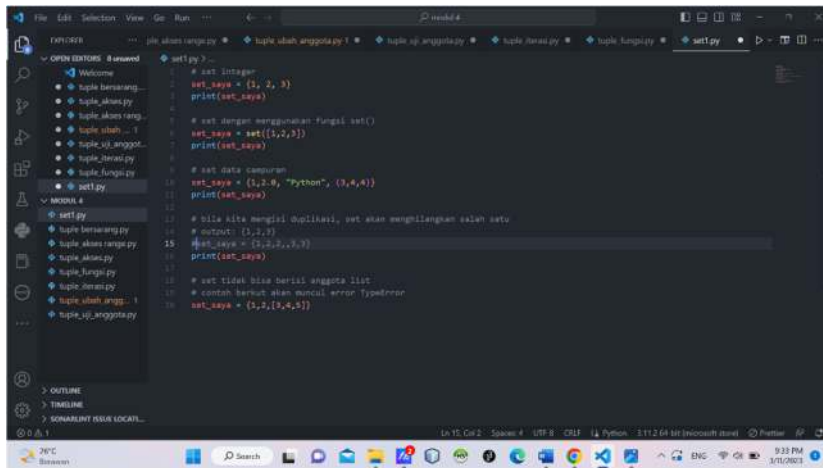
```
1 # output:
2 # msg: listan
3 # msg: Informasi
4 nama = ('listan', 'Informasi')
5 for x in nama:
6     print('Nama: ', x)
```

Tuple fungsi



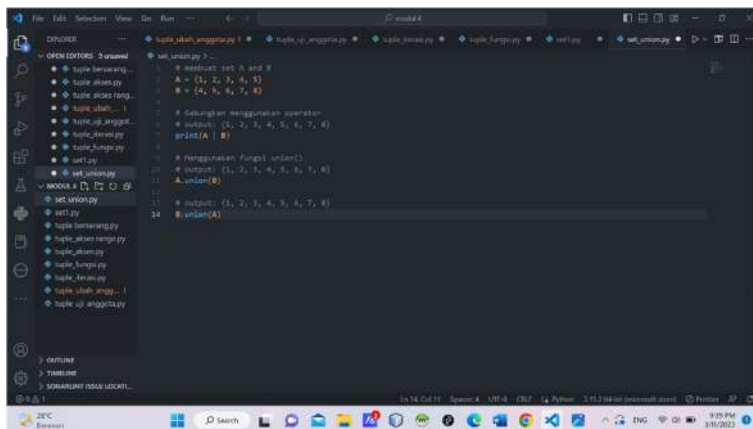
```
1 tuple1 = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z')
2
3 # count
4 output = 2
5 print(tuple1.count('a'))
6
7 # index
8 output = 4
9 print(tuple1.index('a'))
```

Set1



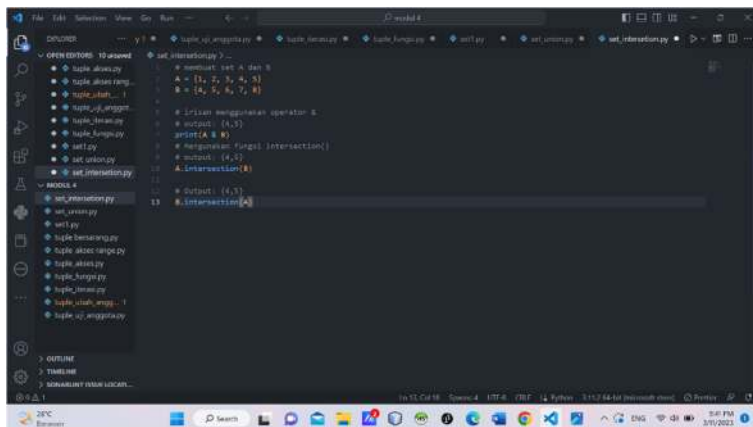
```
1 # set Integer
2 set_saya = {1, 2, 3}
3 print(set_saya)
4
5 # set dengan menggunakan fungsi set()
6 set_saya = set([1, 2, 3])
7 print(set_saya)
8
9 # set data campuran
10 set_saya = {1, 2.0, "Python", (3, 4, 4)}
11 print(set_saya)
12
13 # jika kita mengil duplikasi, set akan menghilangkan salah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

Set union



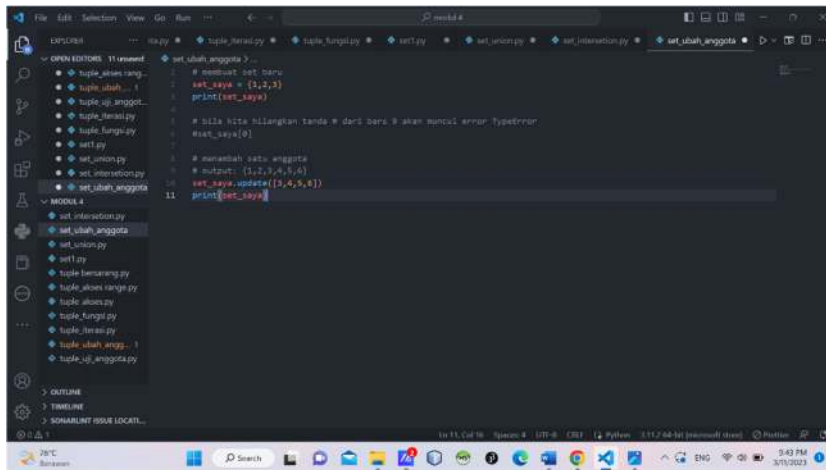
```
1 # membuat set A dan B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Gabungan menggunakan operasi
6 # output: {1, 2, 3, 4, 5, 6, 7, 8}
7 print(A | B)
8
9 # Menggunakan Fungsi union()
10 # output: {1, 2, 3, 4, 5, 6, 7, 8}
11 A.union(B)
12
13 # output: {1, 2, 3, 4, 5, 6, 7, 8}
14 B.union(A)
```

Set intrsetion



```
1 # membuat set A dan B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Irisan menggunakan operasi &
6 # output: {4, 5}
7 print(A & B)
8
9 # menggunakan fungsi intersection()
10 # output: {4, 5}
11 A.intersection(B)
12
13 # output: {4, 5}
14 B.intersection(A)
```

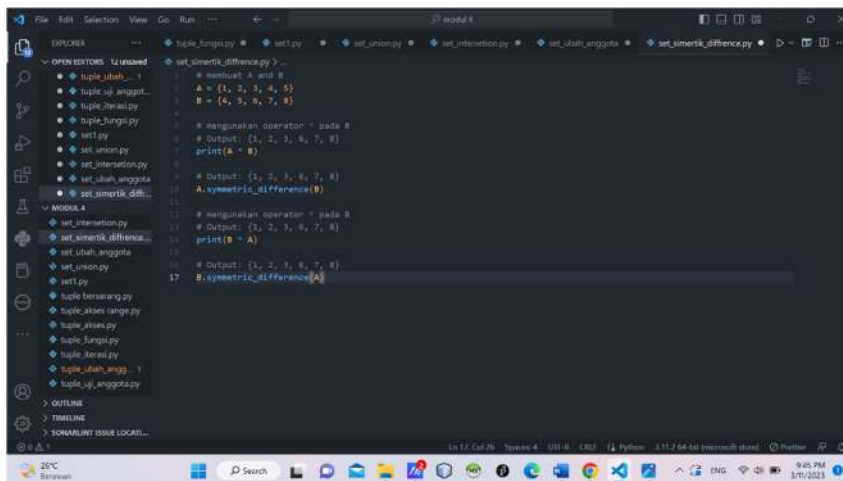
Set ubah anggota



The screenshot shows a VS Code editor with a Python file named `set_ubah_anggota.py`. The code defines a set `set_baru` with elements `{1, 2, 3}` and prints it. It then updates the set with the elements `{4, 5, 6}` and prints the result. Comments explain that the `update` method is used to add new elements and that the original set is modified in place.

```
1 # membuat set baru
2 set_baru = {1, 2, 3}
3 print(set_baru)
4
5 # jika kita hilangkan tanda # dari baris 9 akan muncul error TypeError
6 set_baru[0]
7
8 # merubah satu anggota
9 output = {1, 2, 3, 4, 5, 6}
10 set_baru.update([4, 5, 6])
11 print(set_baru)
```

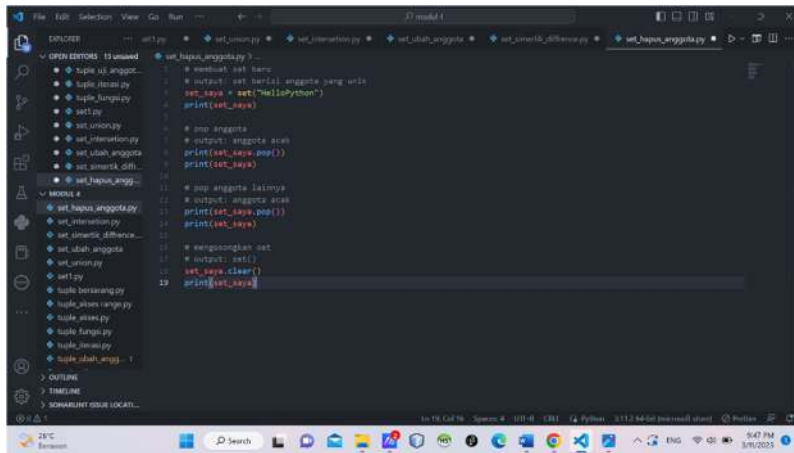
Set dictoanari difference



The screenshot shows a VS Code editor with a Python file named `set_simetrik_difference.py`. The code defines two sets `A` and `B` with elements `{1, 2, 3, 4, 5}` and `{4, 5, 6, 7, 8}` respectively. It then calculates the symmetric difference of the two sets using the `symmetric_difference` method and prints the result. Comments explain that the symmetric difference operation is used to find elements that are in either set but not in both.

```
1 # membuat A and B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # menggunakan operator ^ pada 8
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 8
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)
```

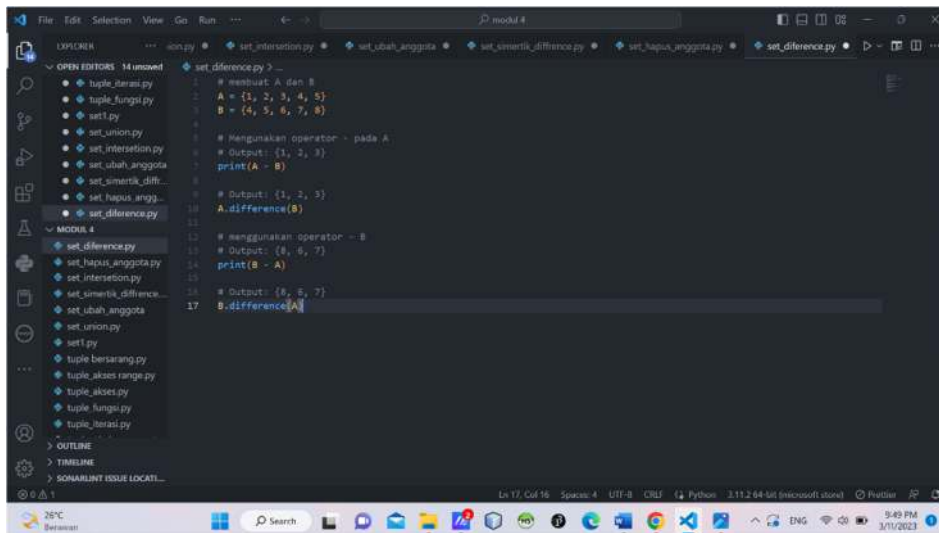
Set hapus anggota



The screenshot shows a VS Code editor with a Python script titled 'set_hapus_anggota.py'. The script demonstrates how to remove elements from a set using the `pop()` and `discard()` methods. It includes comments in Indonesian explaining each step.

```
1 # membuat set baru
2 # membuat 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 # menghapus set
17 # output: set()
18 set_saya.clear()
19 print(set_saya)
```

Set difference



The screenshot shows a VS Code editor with a Python script titled 'set_difference.py'. The script demonstrates how to find the difference between two sets using the `difference()` method. It includes comments in Indonesian explaining each step.

```
1 # membuat A dan 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 - B
13 # Output: {8, 6, 7}
14 print(B - A)
15
16 # Output: {8, 6, 7}
17 B.difference(A)
```

dictionary ubah anggota

The screenshot shows a Windows 10 desktop with a code editor open. The editor contains the following Python code:

```
dict_saya = {'nama': 'Ihsaan', 'usia': 35}
dict_saya2 = {'nama': 'Zishaan', 'usia': 36}
print(dict_saya | dict_saya2)
```

The output of the code is displayed in the console:

```
Output: {'nama': 'Ihsaan', 'usia': 36}
Output: {'nama': 'Zishaan', 'usia': 36}
Output: {'nama': 'Ihsaan', 'usia': 36, 'Tanjungpinang': 35}
```

The code editor also shows a list of files in the left sidebar, including 'dict_saya.py', 'dict_saya2.py', and 'dict_saya3.py'. The status bar at the bottom indicates the system time as 9:52 PM on 3/7/2023.

Dictionary hapus anggota

The screenshot shows a Python IDE with a file named 'modul4.py'. The code is as follows:

```

1 # membuat dictionary baru
2 dict_saya = {'1': 1, '2': 2, '3': 3, '4': 4, '5': 5}
3
4 # menghapus anggota tertentu
5 # Output: 5
6 print(dict_saya.pop(3))
7
8 # menghapus anggota secara acak
9 # Output: {'5': 5}
10 print(dict_saya.popitem())
11
12 # yang tersisa adalah {'1': 1, '2': 2, '4': 4}
13 print(dict_saya)
14
15 # delete 5
16 del dict_saya[2]
17
18 # output: {'1': 1, '4': 4}
19 print(dict_saya)
20
21 # menghapus semua anggota
22 dict_saya.clear()
23
24 # menghapus dictionary dict_saya
25 del dict_saya
26
27 # Error karena dict_saya sudah dihapus
28 print(dict_saya)

```

The output of the script is shown in the console at the bottom:

```

28 print(dict_saya)
Traceback (most recent call last):
  File "modul4.py", line 28, in <module>
    print(dict_saya)
NameError: name 'dict_saya' is not defined

```

Dictionary akses anggota

[illegible]