

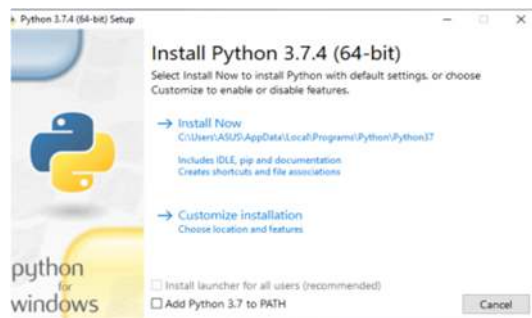
Nama : Nurul Mufliha Puasa

Nim : 20.01.013.014

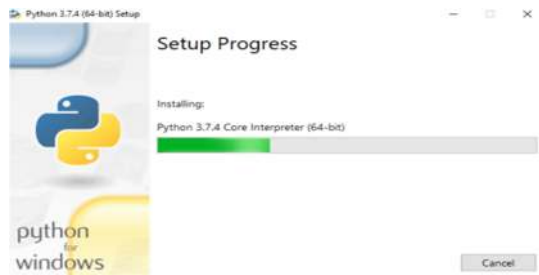
Kelas : C

Modul 1

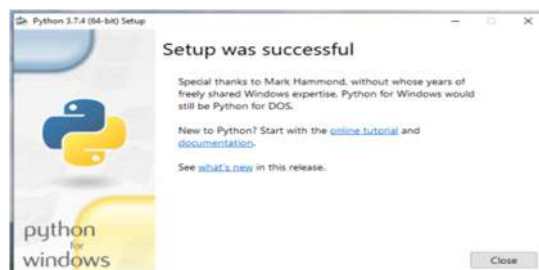
Pilih system python yang sesuai dengan kapasitas Laptop anda. disini saya memakai python(32/64 bit)



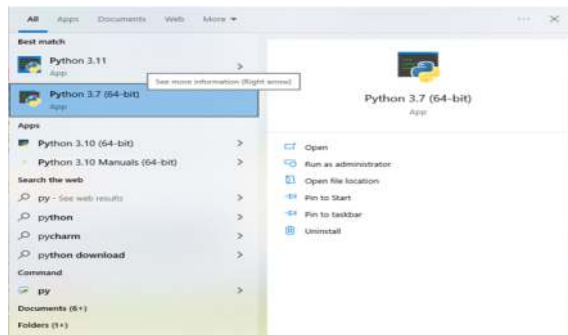
Tunggu beberapa saat hingga proses instalasi selesai.



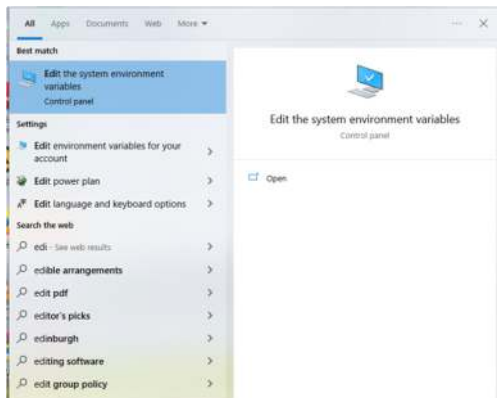
Setelah proses penginstalan sukses klik ok dan klik close agar keluar dari proses penginstalan.



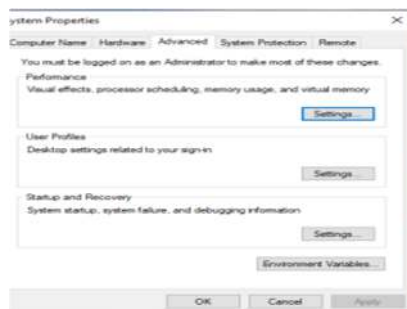
Kalian bisa mengecek apakah python sudah terinstal atau belum di komputer anda

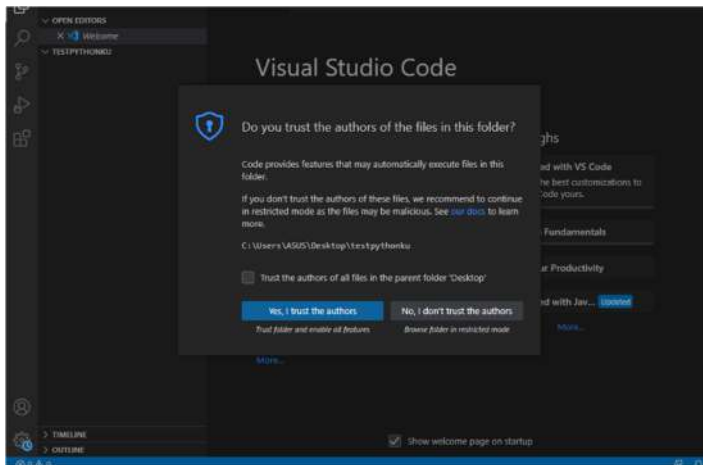


Selanjutnya buka sistem **enviroment variabel** untuk mensetting **path**.bisa di cek di menu searching.

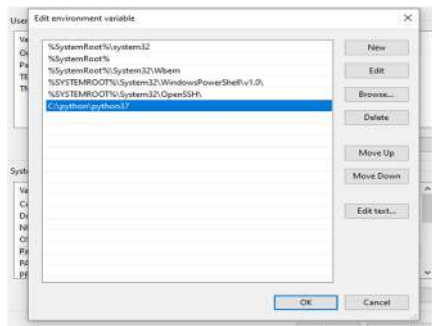


Setelah muncul kotak dialog klik **enviroment variabels** seperti gambar di bawah ini.

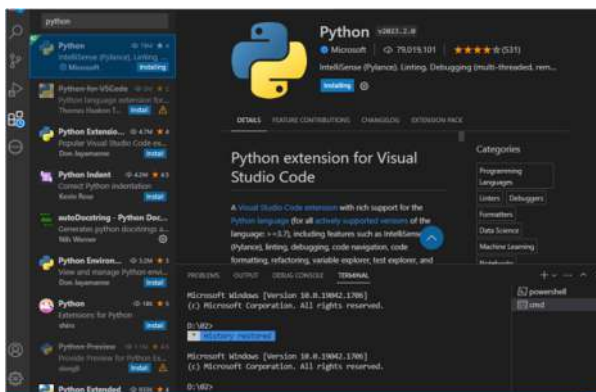




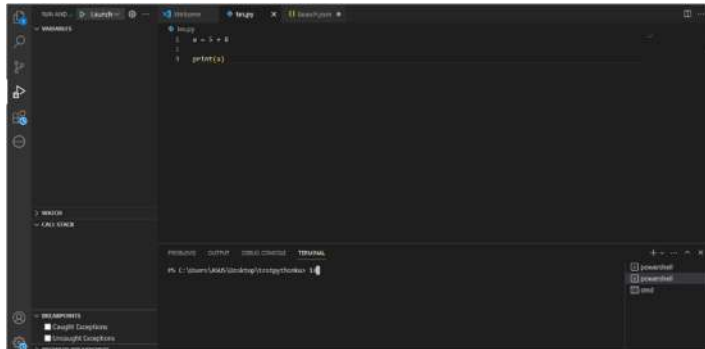
Klik tombol new lalu paste alamat directori yang telah di buat atau di copy



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

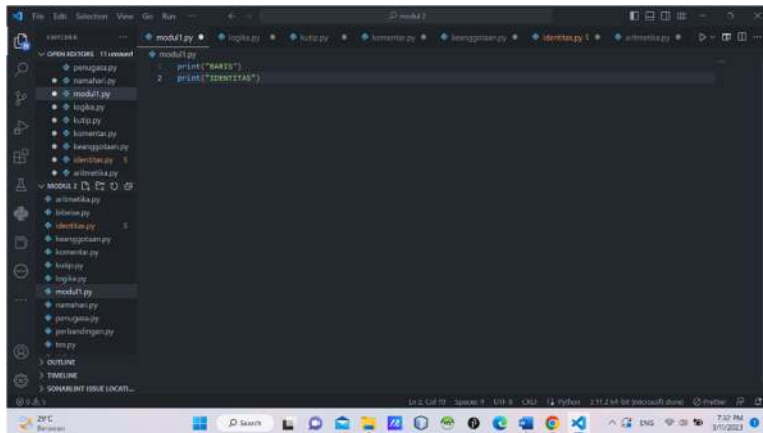


Pada new file lalu buat **folder tes.py**. lalu running project seperti pada gambar di bawah ini.

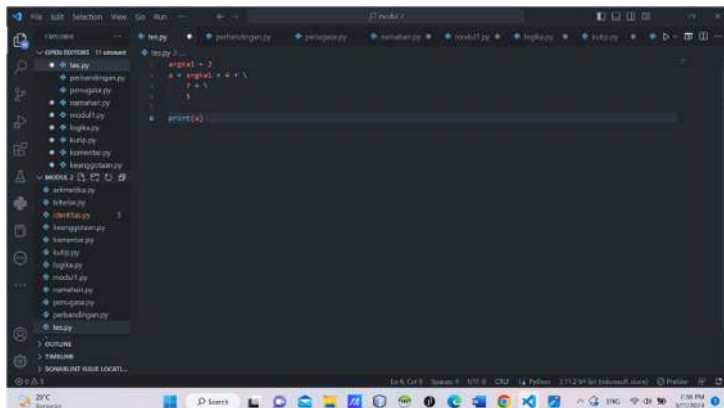


Modul 2

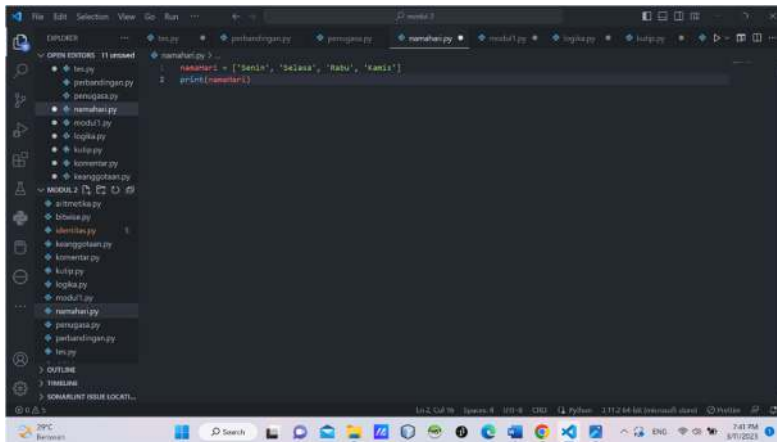
pada modul ini pertama akan membahas baris dan identitas seperti gambar yang ada di bawah ini



selanjutnya masuk ketahap multi baris dengan menggunakan tanda (\)

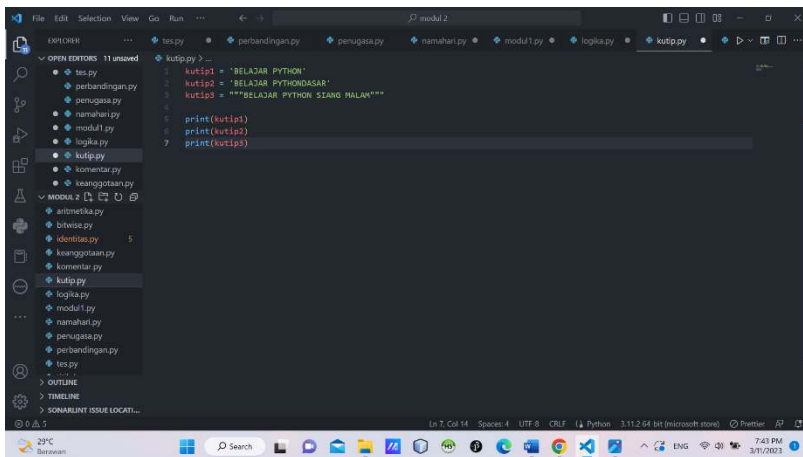


disini ada contoh Latihan yaitu menginput nama-nama hari



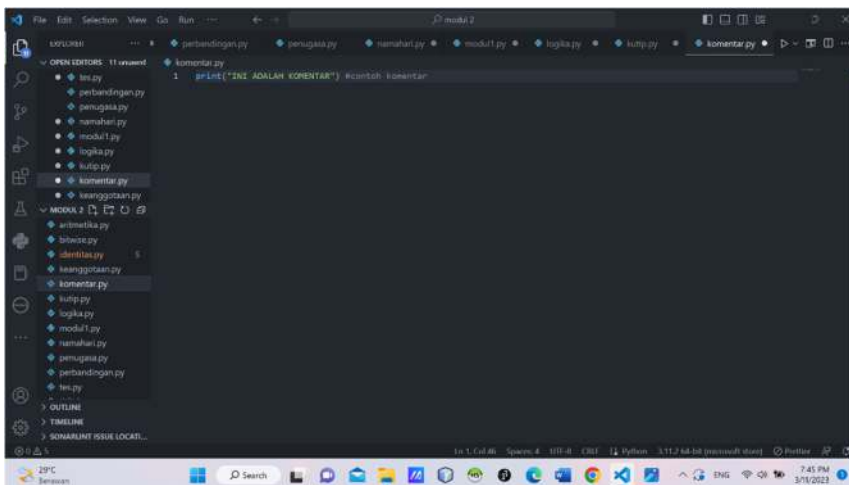
```
nama_hari = ["Senin", "Selasa", "Rabu", "Kamis"]  
print(nama_hari)
```

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



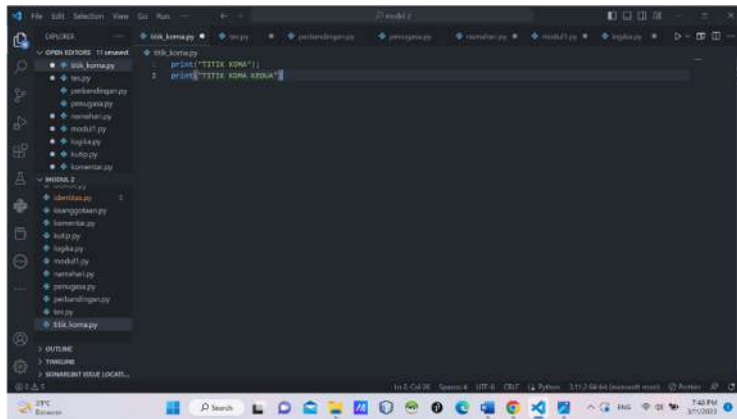
```
kutip1 = 'BELAJAR PYTHON'  
kutip2 = "BELAJAR PYTHON DASAR"  
kutip3 = """BELAJAR PYTHON SIANG MALAM"""  
print(kutip1)  
print(kutip2)  
print(kutip3)
```

disini juga kita bisa menulis komentar seperti di bawah ini



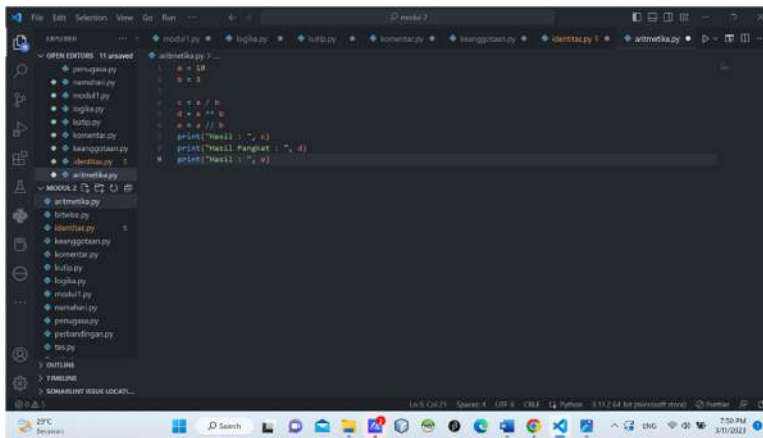
```
print("INI ADALAH KOMENTAR") #ini komentar
```

digunakan bila terdapat dua pertanyaan dalam 1 baris kode



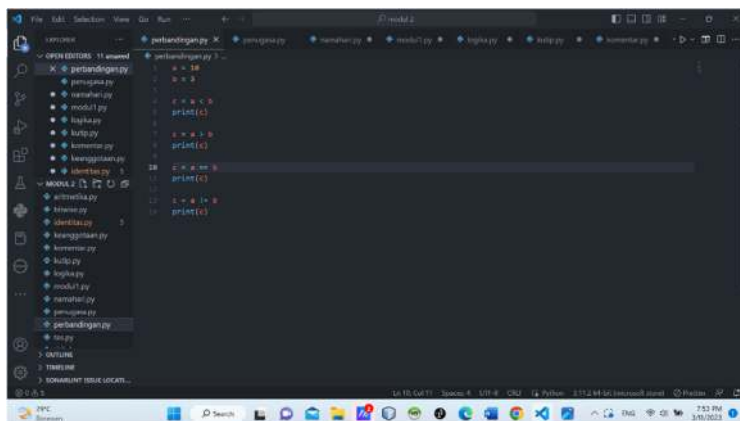
```
1 print('TITIK KONA')  
2 print('TITIK KONA KIRI')
```

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



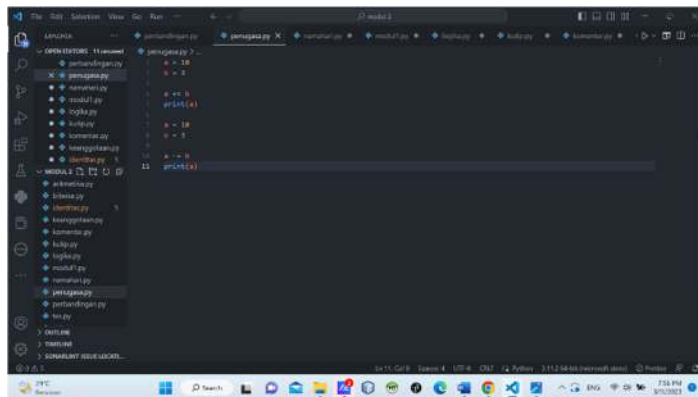
```
1 a = 10  
2 b = 5  
3  
4 c = a + b  
5 d = a - b  
6 e = a // b  
7 print('Hasil : ', c)  
8 print('Hasil Pengkat : ', d)  
9 print('Hasil : ', e)
```

digunakan untuk membandingkan apakah hasilnya true atau false



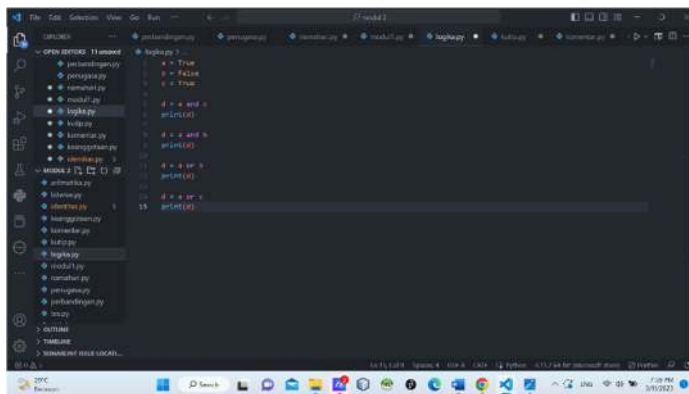
```
1 a = 10  
2 b = 5  
3  
4 c = a < b  
5 print(c)  
6  
7 d = a > b  
8 print(d)  
9  
10 e = a == b  
11 print(e)  
12  
13 f = a != b  
14 print(f)
```

ini di gunakan untu memberi nilai pada variabel



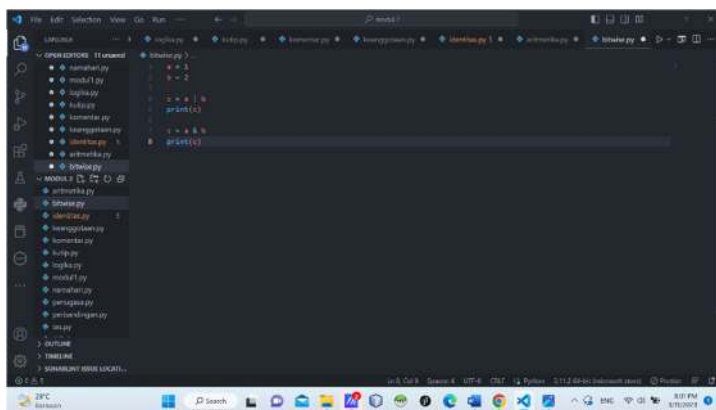
```
peraga.py
a = 10
print(a)
```

Digunakan untuk mengoprasiakan logika



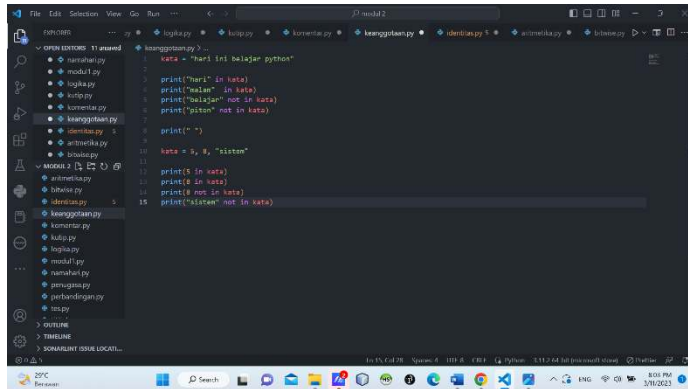
```
logika.py
a = True
print(a)
```

Digunakan untuk mengoprasikan bit per bit sesuai dengan Namanya



```
bitwise.py
a = 5
print(a)
```

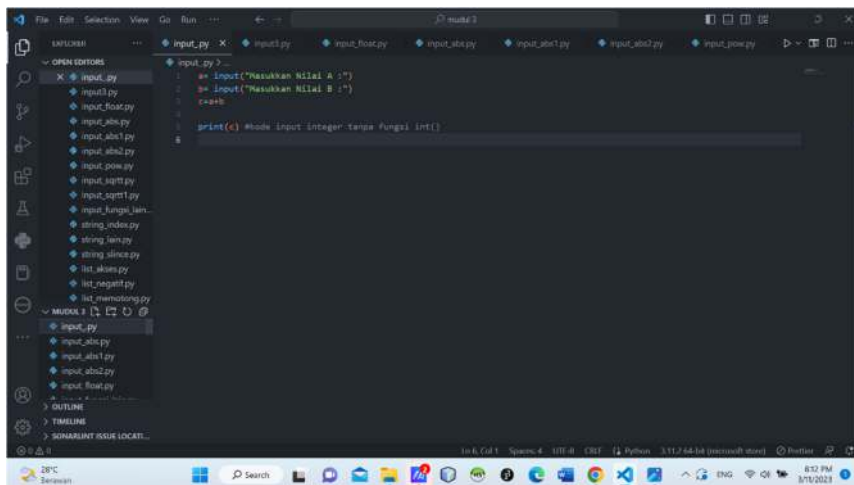
Digunakan untuk memeriksa apakah nilai atau variabel merupakan anggota



```
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)
```

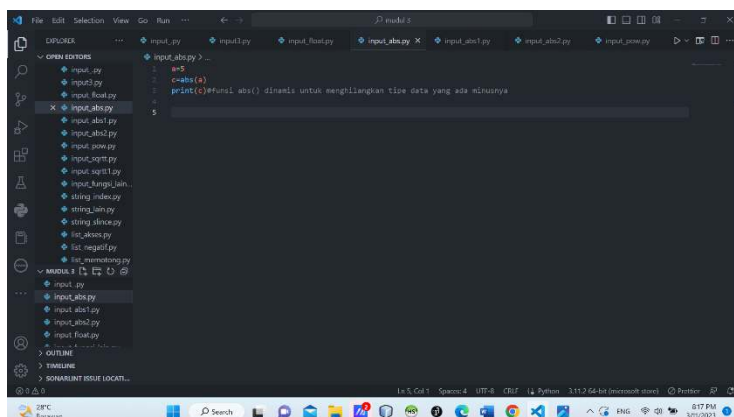
Modul 3

Pertama membahas tentang pengimputan seperti yang ada di bawah ini



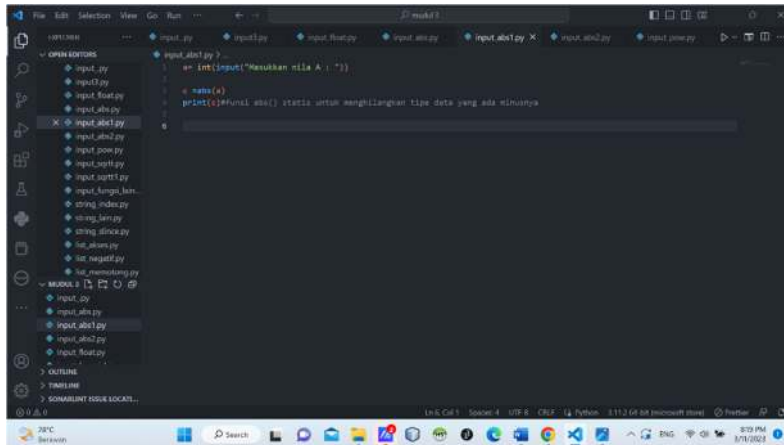
```
1 a = input("Masukkan Nilai A :")  
2 b = input("Masukkan Nilai B :")  
3 c = a + b  
4  
5 print(a) #Maka input integer tanpa fungsi int()
```

Di bawah ini proses input untuk data dinamis yang minus



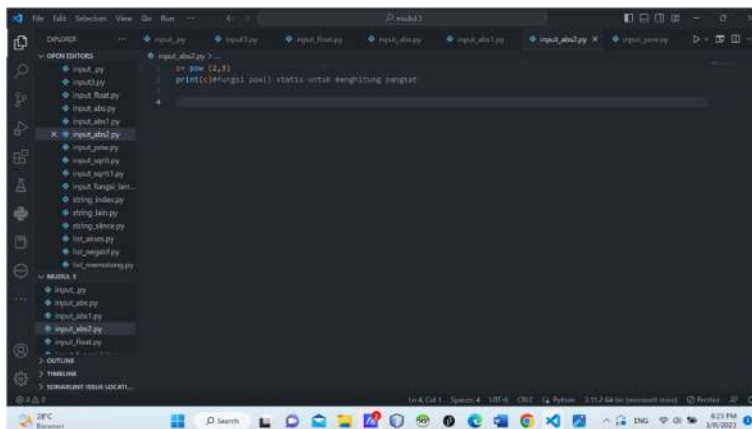
```
1 a = 5  
2  
3 print(a)  
4  
5 print(abs(a)) #Maka abs() dinamis untuk menghilangkan tipe data yang ada minusnya
```


Untuk statis menghilangkan type data yang ada minusnya



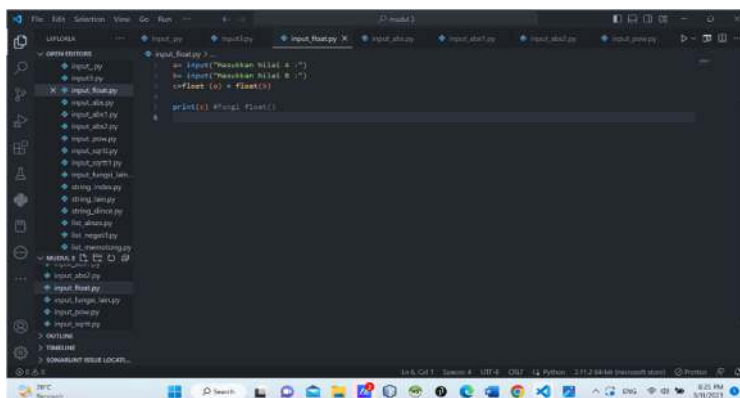
```
input_abs.py >
1 a= int(input("Masukkan nilai A : "))
2
3 a = abs(a)
4 print(a) fungsi abs() statis untuk menghilangkan tipe data yang ada minusnya
```

fungsi pow yang ada statisnya menghitung pangkat



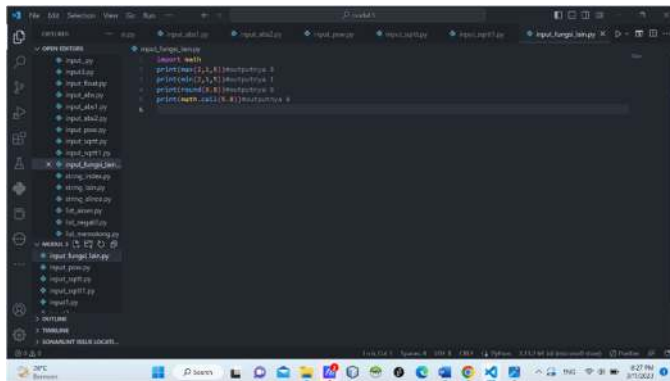
```
input_pow.py >
1 a= pow(2,3)
2
3 print(a) fungsi pow() statis untuk menghitung pangkat
```

Digunakan untuk fungsi float

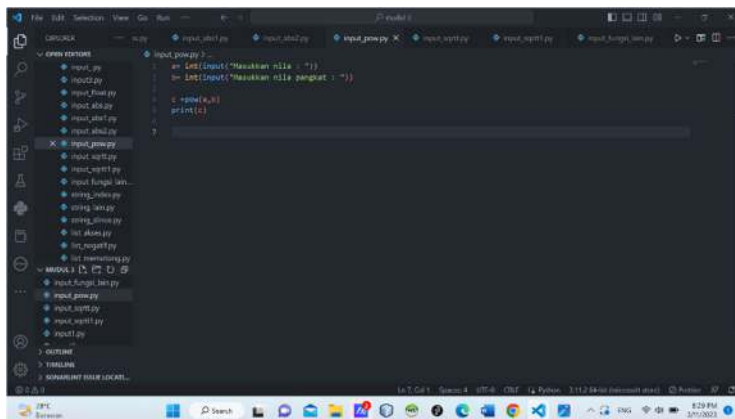


```
input_float.py >
1 a= input("Masukkan bilal : ")
2 b= input("Masukkan bilal : ")
3 c=float(a) + float(b)
4
5 print(c) fungsi float()
```

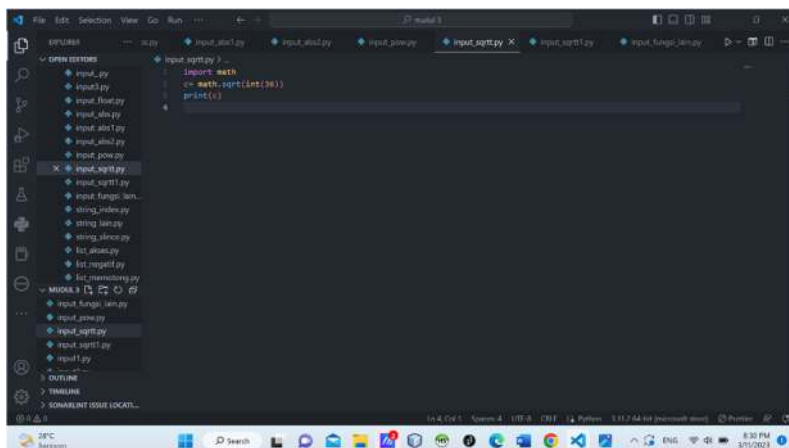
Untuk fungsi lain



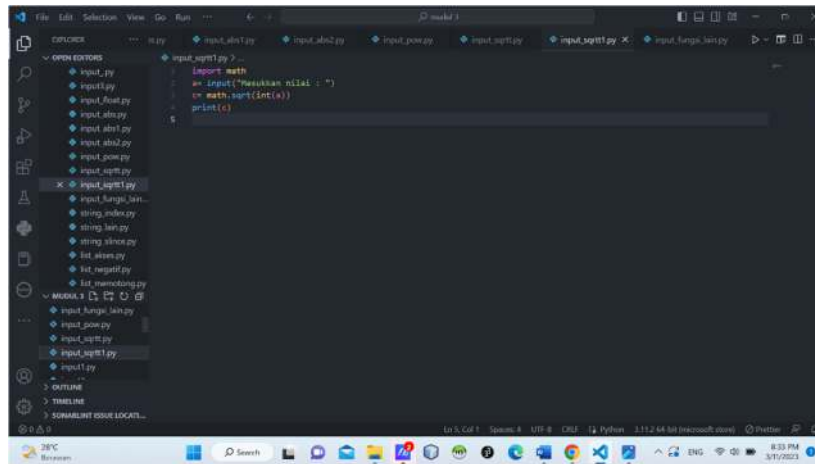
Fungsi fow



Fungsi sqrrt statis

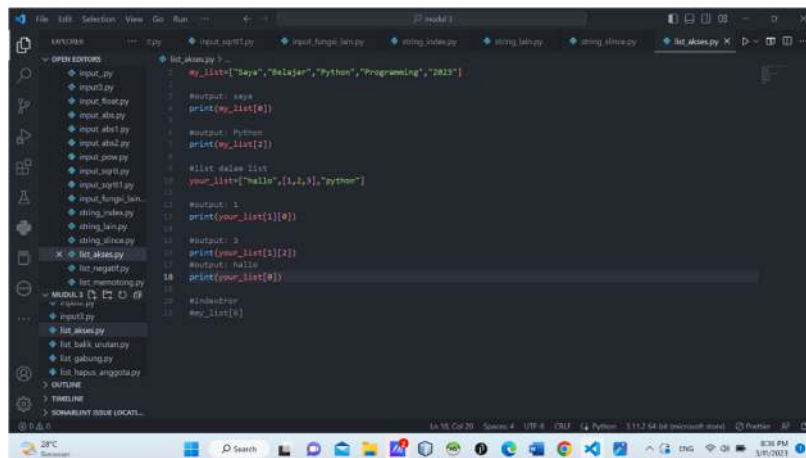


Fungsi sqrtt dinamis



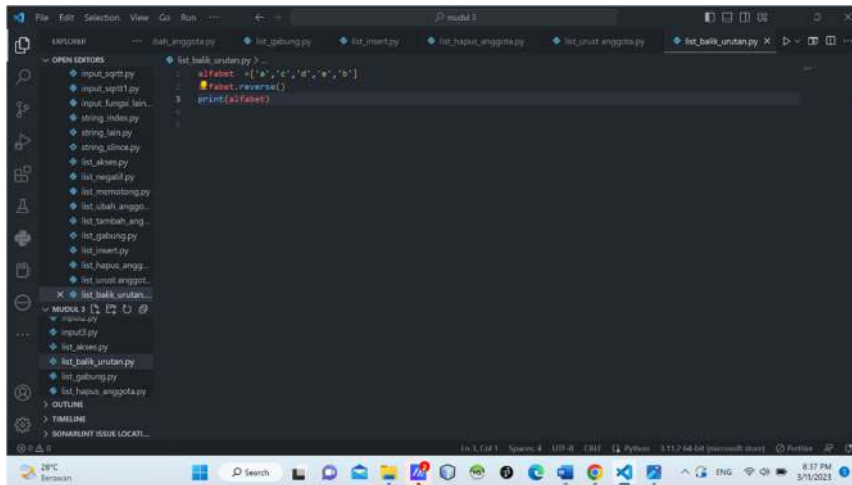
```
File Edit Selection View Go Run ... 27 model 1
input_sqrt1.py
import math
a = input("Masukkan nilai : ")
x = math.sqrt(int(a))
print(x)
```

Fungsi list akses

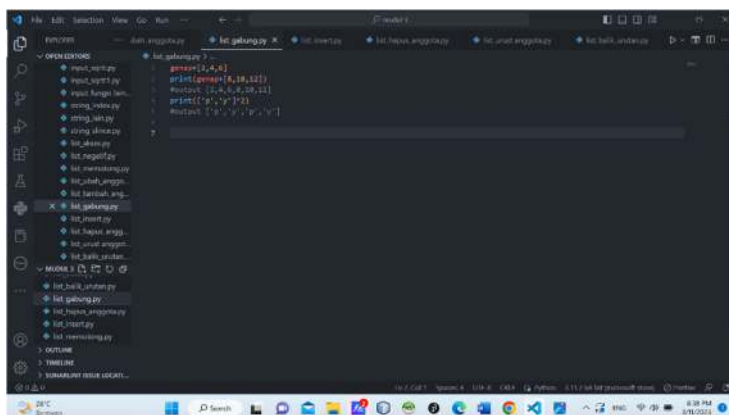


```
File Edit Selection View Go Run ... 27 model 1
list_akses.py
my_list = ["Saya", "Belajar", "Python", "Programming", "2023"]
# Output: saya
print(my_list[0])
# Output: Python
print(my_list[2])
# List index Error
your_list = ["halo", (1,2,3), "python"]
# Output: 1
print(your_list[1][0])
# Output: 3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

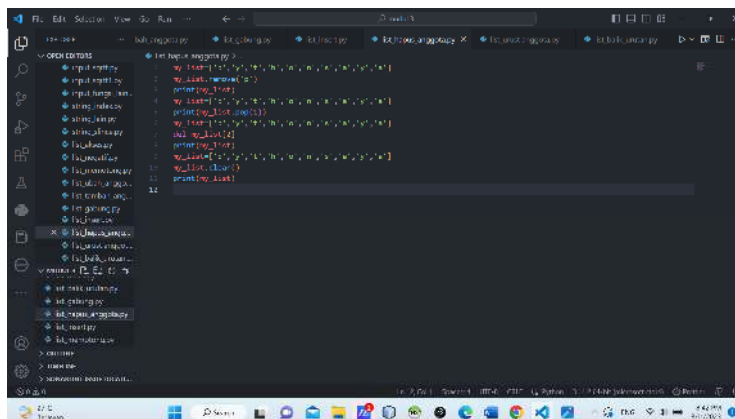
Fungsi list balik urutan



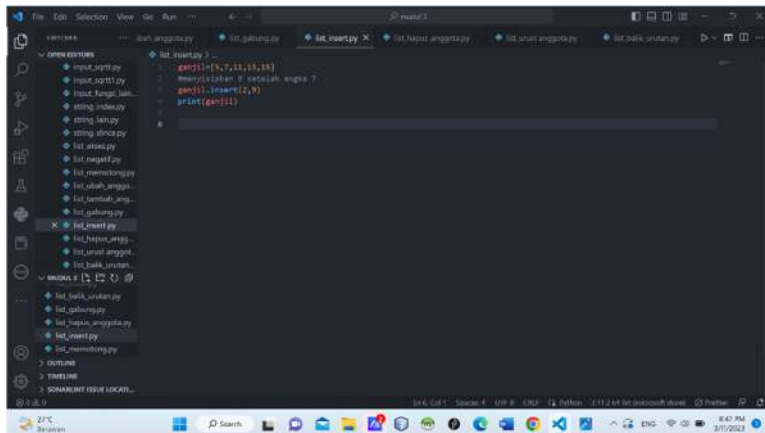
Fungsi list gabung



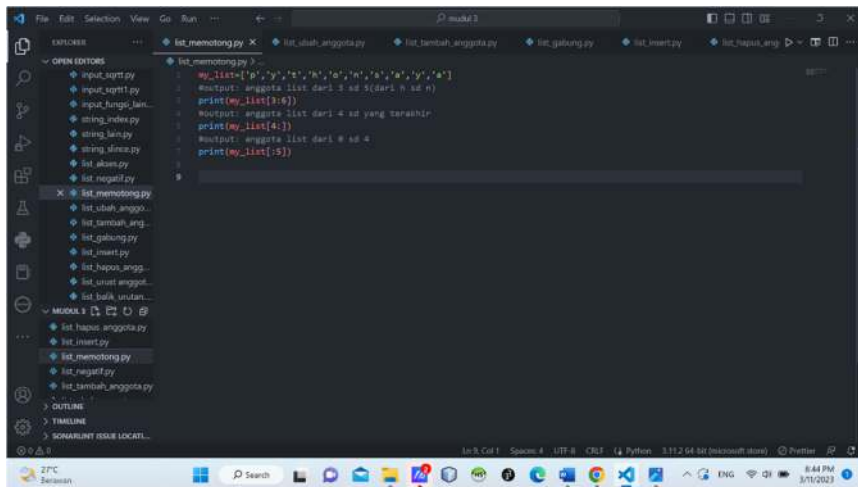
Fungsi untuk menghapus anggota



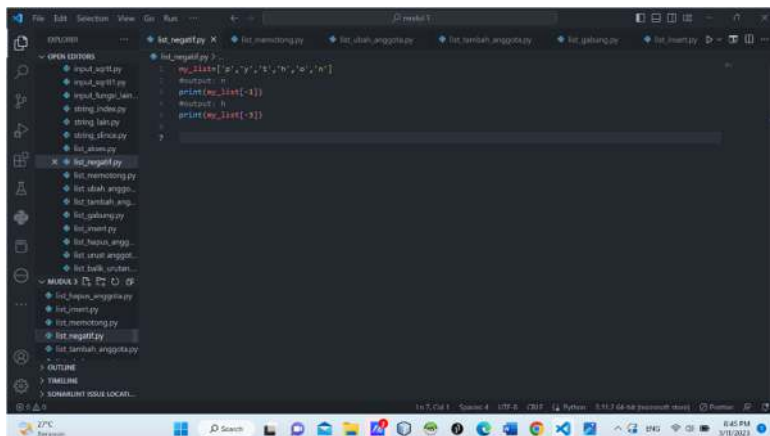
Fungsi list insert



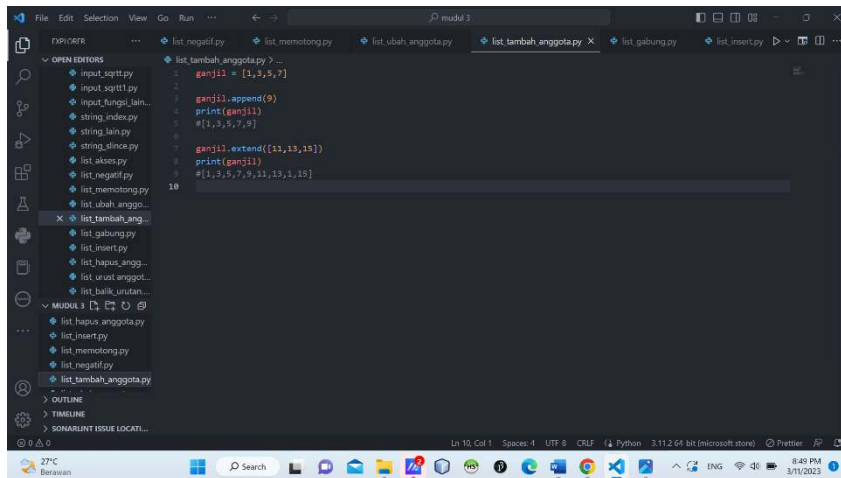
Fungsi list memotong



Fungsi list negatif



Fungsi list menambah anggota

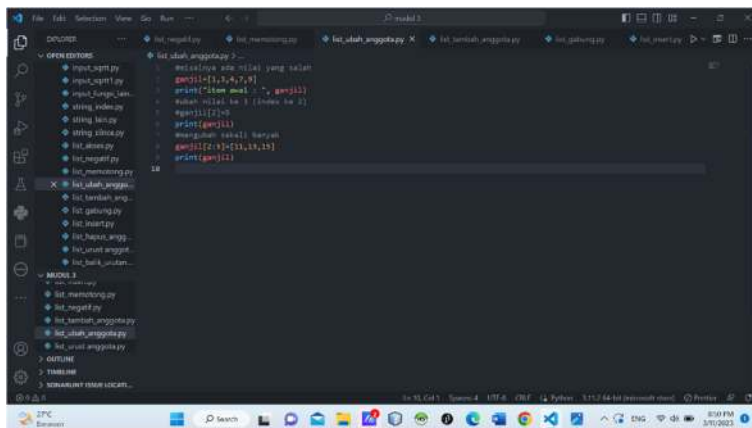


The screenshot shows a VS Code editor with a Python file named `list_tambah_anggota.py`. The code defines a list `ganjil` with the values `[1, 3, 5, 7]`. It then performs the following operations:

- Line 2: `ganjil.append(9)` - Adds the number 9 to the end of the list.
- Line 3: `print(ganjil)` - Prints the list after the first addition.
- Line 4: `#[1, 3, 5, 7, 9]` - A comment showing the expected state of the list.
- Line 7: `ganjil.extend([11, 13, 15])` - Adds the elements 11, 13, and 15 to the end of the list.
- Line 8: `print(ganjil)` - Prints the list after the second addition.
- Line 9: `#[1, 3, 5, 7, 9, 11, 13, 15]` - A comment showing the final state of the list.

The status bar at the bottom indicates the file is at line 10, column 1, with a UTF-8 encoding and CRLF line endings.

Fungsi list mengubah anggota

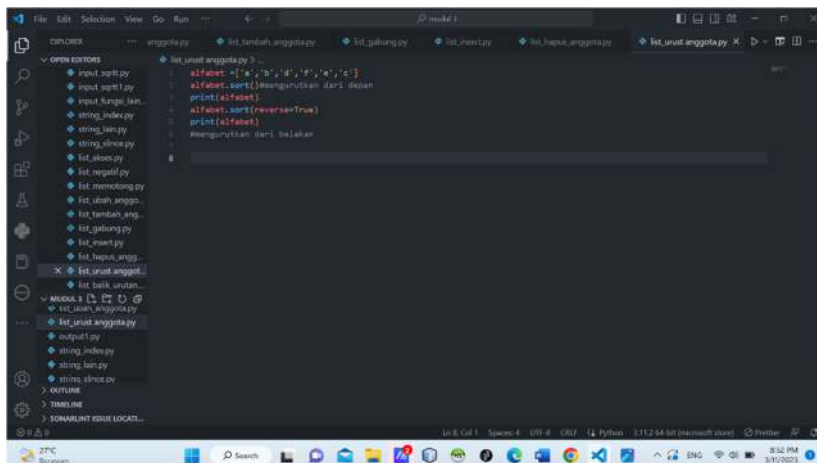


The screenshot shows a VS Code editor with a Python file named `list_ubah_anggota.py`. The code defines a list `ganjil` with the values `[1, 3, 5, 7, 9]`. It then performs the following operations:

- Line 2: `ganjil[2] = 4` - Replaces the element at index 2 (which is 5) with the value 4.
- Line 3: `print("Item awal : ", ganjil)` - Prints the list before the modification.
- Line 4: `#ubah nilai ke 3 (index ke 2)` - A comment explaining the change.
- Line 5: `ganjil[2] = 4` - Repeats the assignment for clarity.
- Line 6: `print(ganjil)` - Prints the list after the modification.
- Line 7: `#mengubah indeks banyak` - A comment indicating the next step.
- Line 8: `ganjil[2:4] = [11, 13, 15]` - Replaces the slice of the list from index 2 to 4 (elements 5, 7, 9) with the new values 11, 13, and 15.
- Line 9: `print(ganjil)` - Prints the final modified list.

The status bar at the bottom indicates the file is at line 10, column 1, with a UTF-8 encoding and CRLF line endings.

Fungsi list mengurutkan anggota



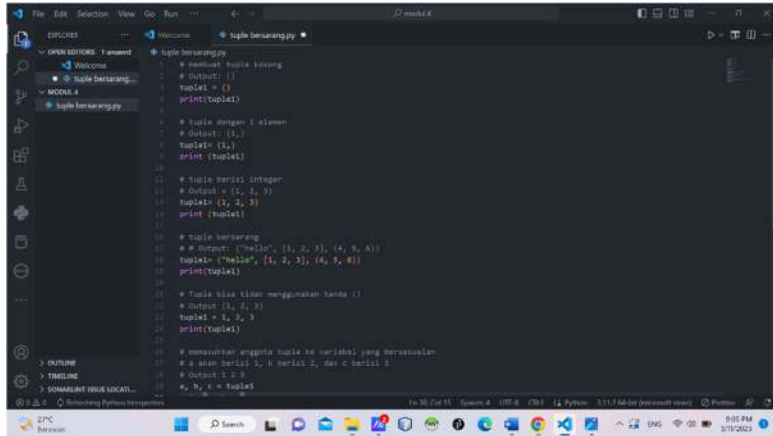
The screenshot shows a VS Code editor with a Python file named `list_urut_anggota.py`. The code defines a list `alfabet` with the values `['a', 'b', 'd', 'f', 'g', 'h', 'c']`. It then performs the following operations:

- Line 2: `alfabet.sort()` - Sorts the list in ascending order.
- Line 3: `print(alfabet)` - Prints the list after sorting.
- Line 4: `alfabet.sort(reverse=True)` - Sorts the list in descending order.
- Line 5: `print(alfabet)` - Prints the list after reverse sorting.
- Line 6: `#mengurutkan dari belakang` - A comment indicating the next step.

The status bar at the bottom indicates the file is at line 6, column 1, with a UTF-8 encoding and CRLF line endings.

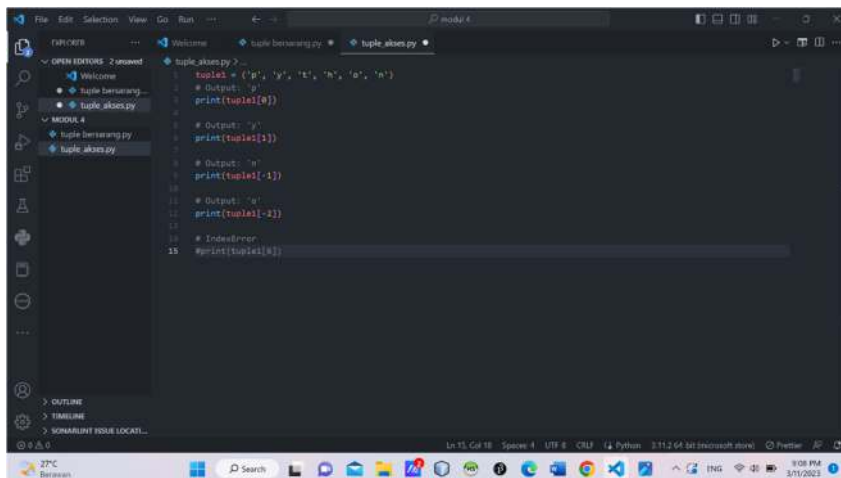
Modul 4

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

The screenshot shows a Windows 10 desktop with a taskbar at the bottom containing icons for the Start menu, File Explorer, Google Chrome, and several instances of VS Code. The VS Code window is open, displaying a Python file named `halo.py`. The code in the file is as follows:

```

1 # slicing
2
3 # slicing dari index 0 s/d 5
4 a = 'halo python'
5
6 # output: 'halo'
7 print(a[0:5])
8
9 # slicing dari index 6 sampai akhir
10
11 # output: 'python'
12 print(a[6:])
13
14

```

The VS Code interface includes a sidebar on the left with a file explorer showing the project structure, a central editor area with the code, and a bottom status bar showing the file's encoding (UTF-8), line/column position (1/1), and the current Python interpreter (Python 3.7.4).

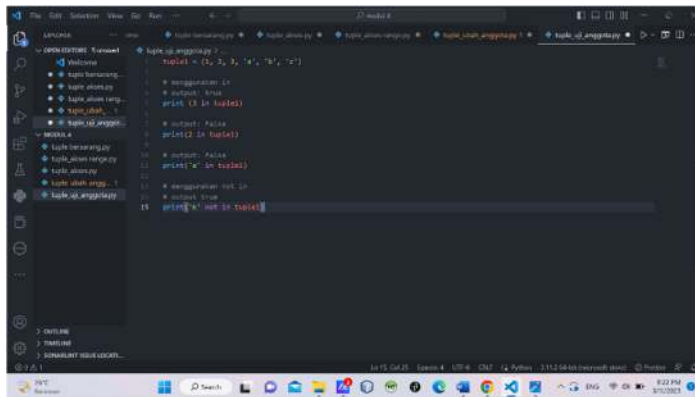
Tuple ubah anggota digunakan untuk tipe bersarang

```

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

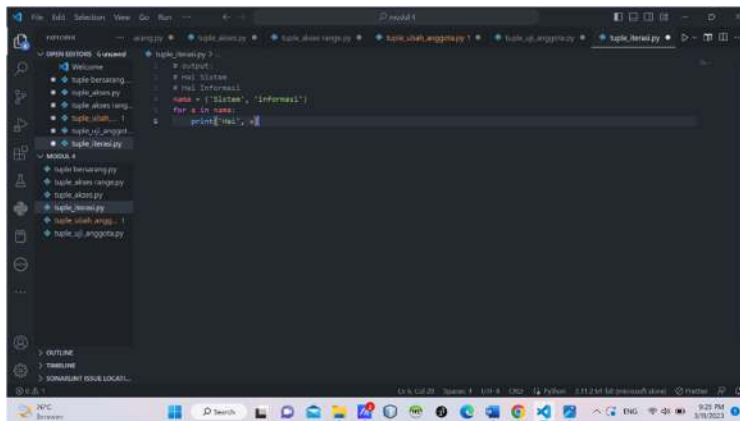
```


Tuple uji anggota



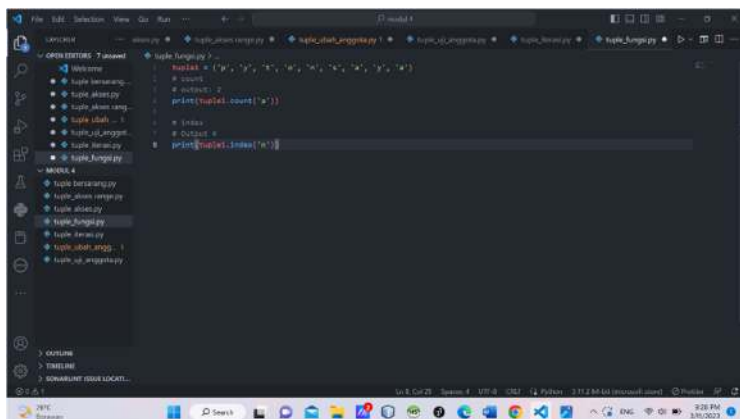
```
1 tuple = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z')
2
3 # Menghasilkan list
4 output = True
5 print('a' in tuple)
6
7 # output: False
8 print('b' in tuple)
9
10 # output: False
11 print('x' in tuple)
12
13 #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

Tuple iterasi



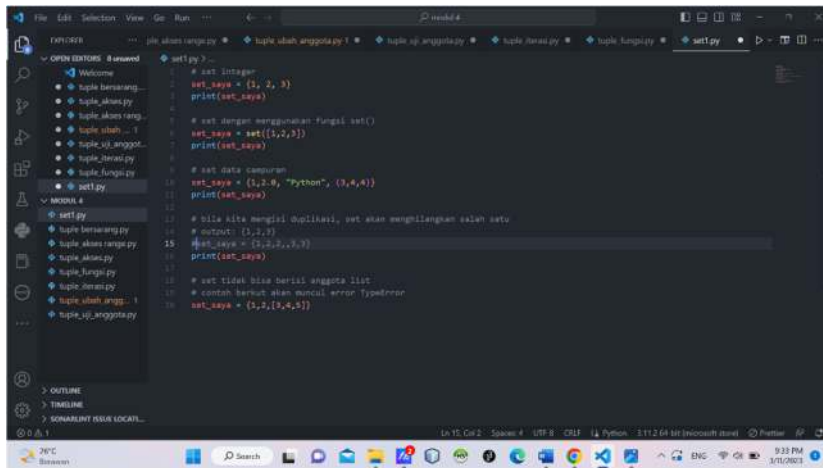
```
1 # output:
2 # list:
3 # list: ['liston', 'informasi']
4 for x in tuple:
5     print('halo', x)
```

Tuple fungsi



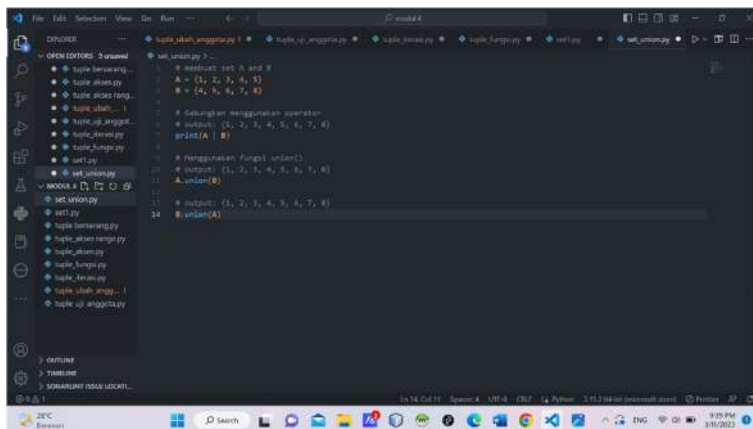
```
1 tuple = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z')
2
3 # count
4 output = 2
5 print(tuple.count('a'))
6
7 # index
8 output = 4
9 print(tuple.index('a'))
```

Set1



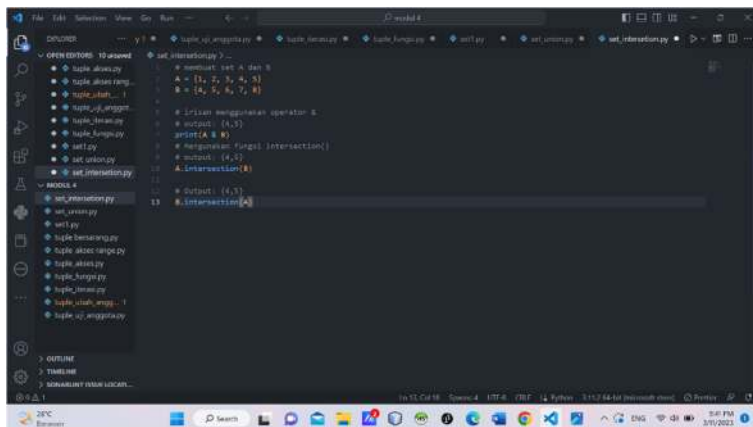
```
1 # set Integer
2 set_saya = {1, 2, 3}
3 print(set_saya)
4
5 # set dengan menggunakan fungsi set()
6 set_saya = set([1, 2, 3])
7 print(set_saya)
8
9 # set data campuran
10 set_saya = {1, 2.0, "Python", (3, 4, 4)}
11 print(set_saya)
12
13 # jika kita mengil duplikasi, set akan menghilangkan salah 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



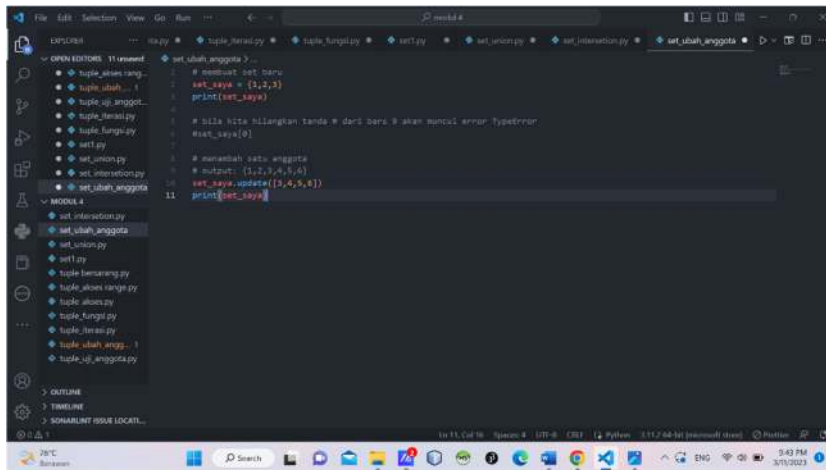
```
1 # membuat set A dan B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Gabungan menggunakan operasi
6 # output: {1, 2, 3, 4, 5, 6, 7, 8}
7 print(A | B)
8
9 # Menggunakan Fungsi union()
10 # output: {1, 2, 3, 4, 5, 6, 7, 8}
11 A.union(B)
12
13 # output: {1, 2, 3, 4, 5, 6, 7, 8}
14 B.union(A)
```

Set intrsetion



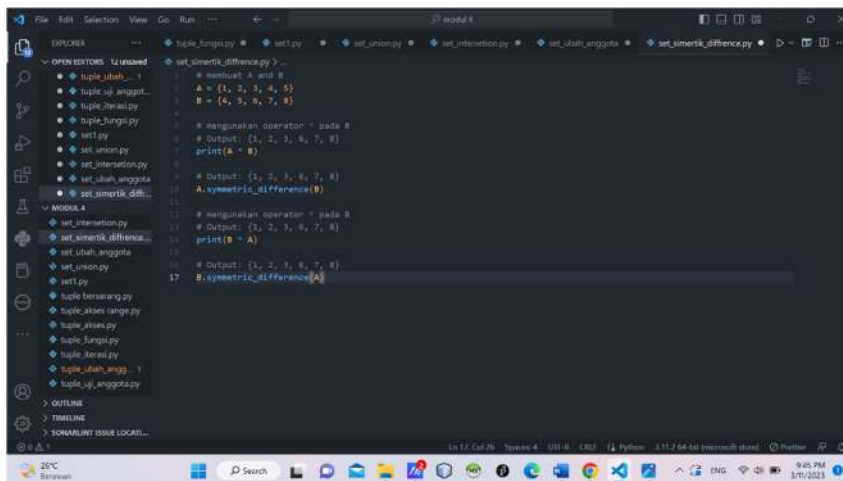
```
1 # membuat set A dan B
2 A = {1, 2, 3, 4, 5}
3 B = {4, 5, 6, 7, 8}
4
5 # Irisan menggunakan operasi &
6 # output: {4, 5}
7 print(A & B)
8
9 # menggunakan fungsi intersection()
10 # output: {4, 5}
11 A.intersection(B)
12
13 # output: {4, 5}
14 B.intersection(A)
```

Set ubah anggota



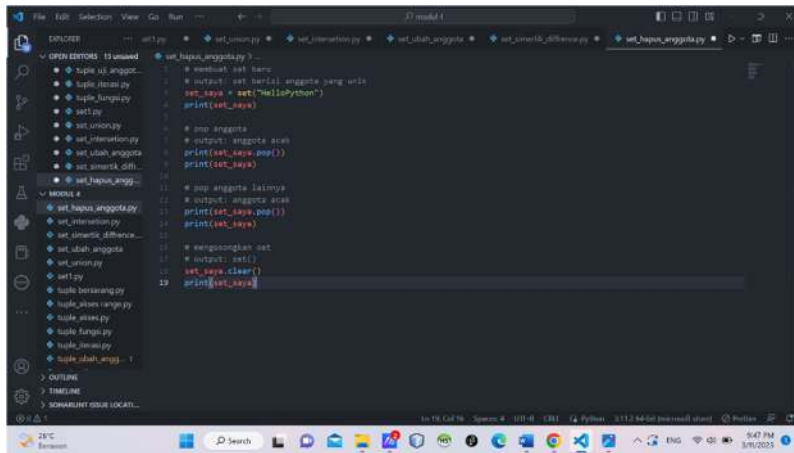
```
1 # membuat set baru
2 set_saya = {1, 2, 3}
3 print(set_saya)
4
5 # jika kita hilangkan tanda # dari baris 9 akan muncul error TypeError
6 set_saya[0]
7
8 # merubah satu anggota
9 # output: {1, 2, 3, 4, 5, 6}
10 set_saya.update([4, 5, 6])
11 print(set_saya)
```

Set dictoanari difference



```
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)
```

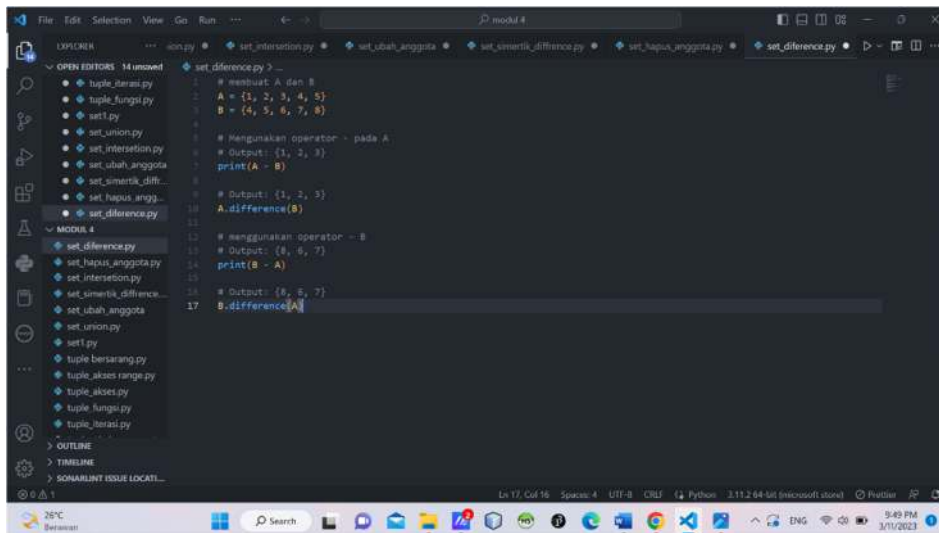
Set hapus anggota



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



```

1 dict_saya = {'nama': 'Ikhwan', 'usia': 35}
2
3 # mengupdate nilai
4 dict_saya['usia'] = 36
5 # Output: {'nama': 'Ikhwan', 'usia': 36}
6 print(dict_saya)
7
8 # merubah anggota
9 dict_saya['alamat'] = 'Tanjungpinang'
10 # Output: {'nama': 'Ikhwan', 'usia': 36}
11 print(dict_saya)

```

Dictionary hapus anggota

The screenshot shows a Python IDE with a file explorer on the left and a code editor on the right. The code editor displays a script named 'modul4.py' that demonstrates dictionary operations. The script defines a dictionary 'dict_saya' with keys 1 through 5 and values 'mawati', 'siti', 'dina', 'dina', and 'dina'. It then performs several operations: printing the dictionary, deleting the value 'dina' (which results in a 'ValueError' error), deleting the key 'dina', and printing the updated dictionary. The output shows the dictionary after removing the value 'dina' and the key 'dina'.

```

1 # membuat dictionary baru
2 dict_saya = {1:1, 2:4, 3:9, 4:16, 5:25}
3
4 # menghapus anggota tertentu
5 # Output: 9
6 print(dict_saya.pop(3))
7
8 # menghapus anggota secara acak
9 # Output: {5: 25}
10 print(dict_saya.popitem())
11
12 # yang tersisa adalah {1:1, 2:4, 4:16}
13 print(dict_saya)
14
15 # delete 5
16 del dict_saya[2]
17
18 # output: {1:1, 4:16}
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)

```

Dictionary akses anggota

The screenshot shows a PyCharm IDE with a Python script open. The script is titled 'week6_4' and contains the following code:

```

1 # dict_saya = {'nama': 'Budi', 'usia': 27}
2
3 # dict_kamu = {'nama': 'Budi', 'usia': 27}
4
5 # output: Budi
6 print(dict_saya['nama'])
7
8 # output: 27
9 print(dict_saya.get('usia'))
10
11 # output: None
12 print(dict_saya.get('alamat'))
13
14 # output: Budi
15 print(dict_kamu['nama'])
16
17 # output: 27
18 print(dict_kamu.get('usia'))
19
20 # output: None
21 print(dict_kamu.get('alamat'))
22
23 # Mengambil kunci yang tidak bersesuaian keyError
24 dict_saya['alamat']

```

The script is executed, and the output is displayed in the console:

```

Budi
27
None
Budi
27
None

```

The IDE interface includes a menu bar at the top with options like File, Edit, Selection, View, Get, Run, and a toolbar with icons for various actions. The left sidebar shows a project explorer with a folder named 'week6_4' containing several Python files. The bottom status bar indicates the current file is 'week6_4.py' and the interpreter is 'Python 3.11.2 64-bit (headless install)'.