

TASK 3 MODUL 4

Nama: Sedri Sella Jumeni

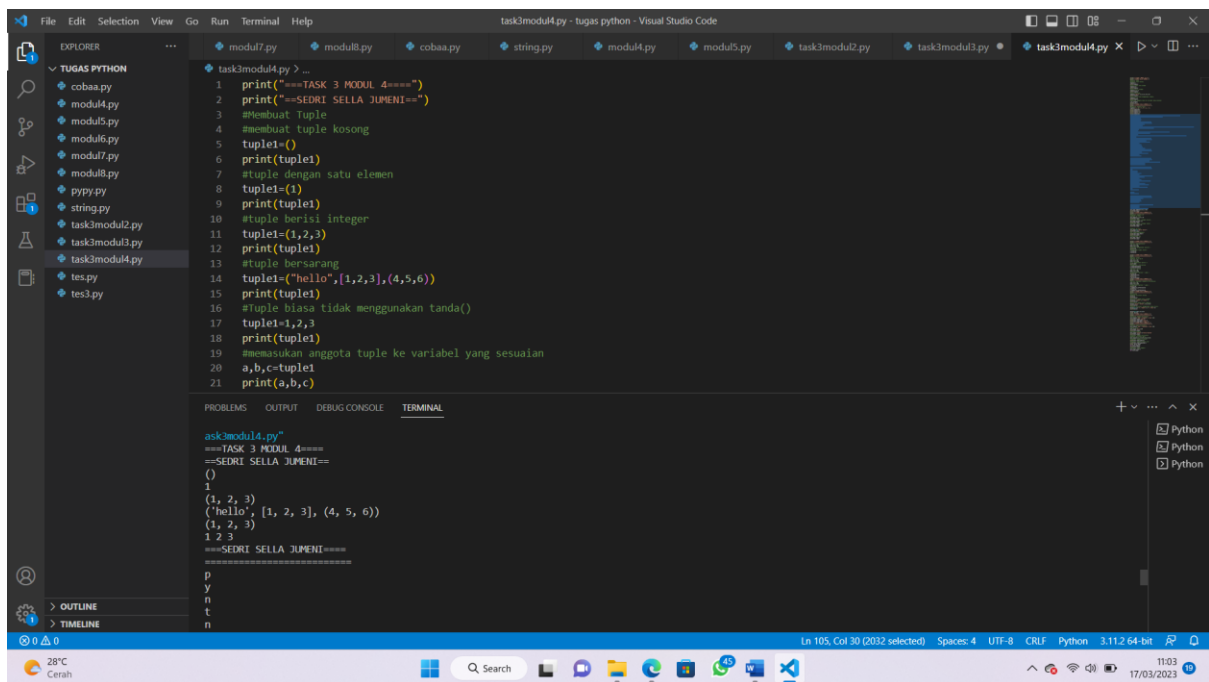
Nim: 211001073

Kelas: Pemrograman Python D

Tuple

Tuple mirip dengan list. Bedanya, tuple bersifat immutable, sehingga anggotanya tidak bisa diubah. **Membuat Tuple** : Tuple dibuat dengan meletakkan semua anggota di dalam tanda kurung (), masingmasing dipisahkan oleh tanda koma.

Contoh:



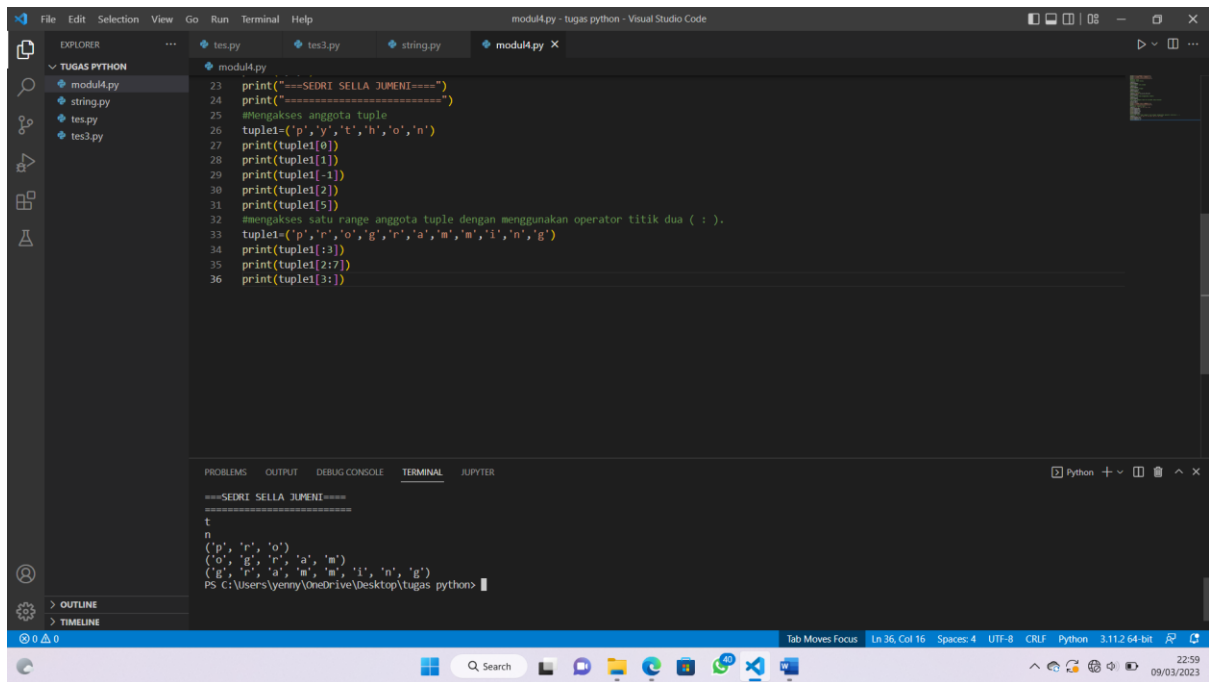
```
task3modul4.py > ...
1 print("==TASK 3 MODUL 4==")
2 print("==SEDRI SELLA JUMENI==")
3 #membuat tuple
4 #membuat tuple kosong
5 tuple1=()
6 print(tuple1)
7 #tuple dengan satu elemen
8 tuple1=(1)
9 print(tuple1)
10 #tuple berisi integer
11 tuple1=(1,2,3)
12 print(tuple1)
13 #tuple bersarang
14 tuple1=("hello",[1,2,3],[4,5,6])
15 print(tuple1)
16 #tuple biasa tidak menggunakan tanda()
17 tuple1=1,2,3
18 print(tuple1)
19 #memasukan anggota tuple ke variabel yang sesuaikan
20 a,b,c=tuple1
21 print(a,b,c)
```

```
task3modul4.py"
==TASK 3 MODUL 4==
==SEDRI SELLA JUMENI==
()
1
(1, 2, 3)
('hello', [1, 2, 3], (4, 5, 6))
(1, 2, 3)
1 2 3
==SEDRI SELLA JUMENI==
=====
p
y
n
t
n
```

Mengakses Anggota Tuple: kita bisa mengakses anggota tuple lewat indeksnya menggunakan format `namatuple[indeks]`. Indeks dimulai dari 0 untuk anggota pertama. Selain itu, indeks negatif juga bisa dipakai mulai dari -1 untuk anggota terakhir tuple.

Contoh:

TASK 3 MODUL 4



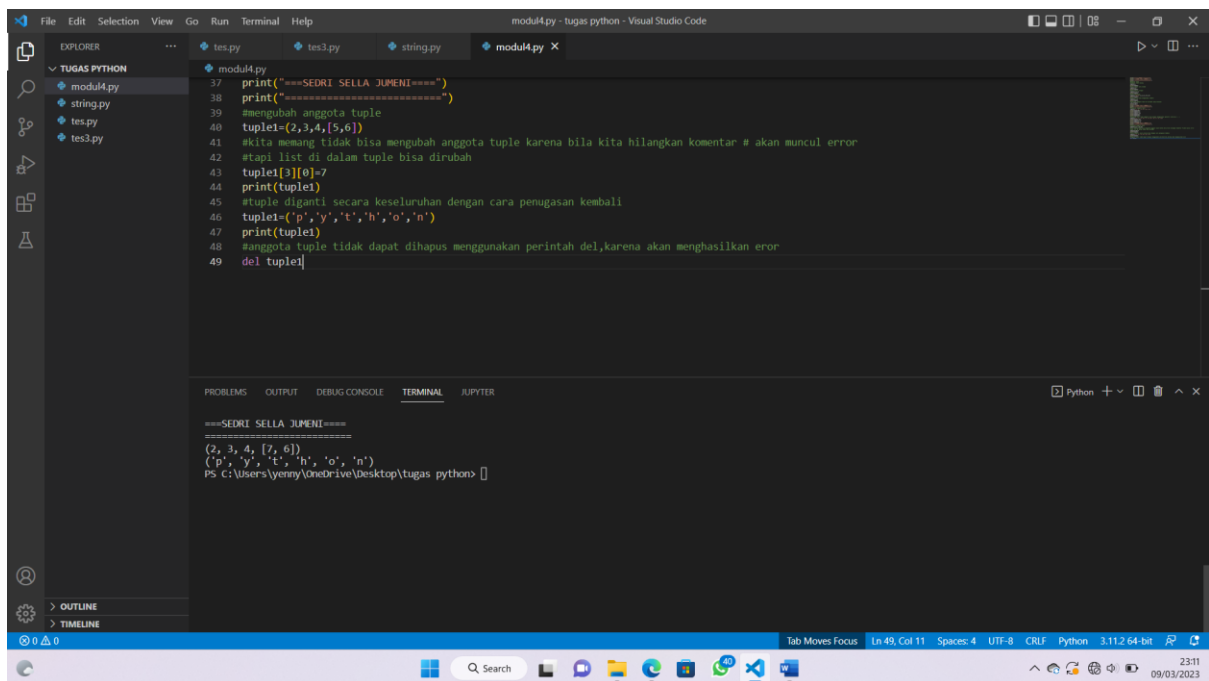
The screenshot shows a Visual Studio Code editor with a Python file named `modul4.py`. The code defines a tuple `t` and demonstrates various indexing and slicing operations. The terminal output shows the execution of the script, displaying the tuple and its elements.

```
23 print("==SEDRi SELLA JUMENI==")
24 print("=====")
25 #Menggakses anggota tuple
26 tuple1=('p','y','t','h','o','n')
27 print(tuple1[0])
28 print(tuple1[1])
29 print(tuple1[-1])
30 print(tuple1[2])
31 print(tuple1[5])
32 #mengakses satu range anggota tuple dengan menggunakan operator titik dua ( : ).
33 tuple1=('p','r','o','g','r','a','m','m','i','n','g')
34 print(tuple1[3])
35 print(tuple1[2:7])
36 print(tuple1[3:])
```

```
==SEDRi SELLA JUMENI==
=====
t
n
('p', 'r', 'o')
('o', 'g', 'r', 'a', 'm')
('g', 'r', 'a', 'm', 'm', 'i', 'n', 'g')
PS C:\Users\yenny\OneDrive\Desktop\tugas python>
```

Mengubah Anggota Tuple: Setelah tuple dibuat, maka anggota tuple tidak bisa lagi diubah atau dihapus. Akan tetapi, bila anggota tuple-nya adalah tuple bersarang dengan anggota seperti list, maka item pada list tersebut dapat diubah.

Contoh:



The screenshot shows a Visual Studio Code editor with a Python file named `modul4.py`. The code demonstrates how to modify a tuple by replacing it with a new tuple, where one of the elements is a list. The terminal output shows the execution of the script, displaying the tuple and its elements.

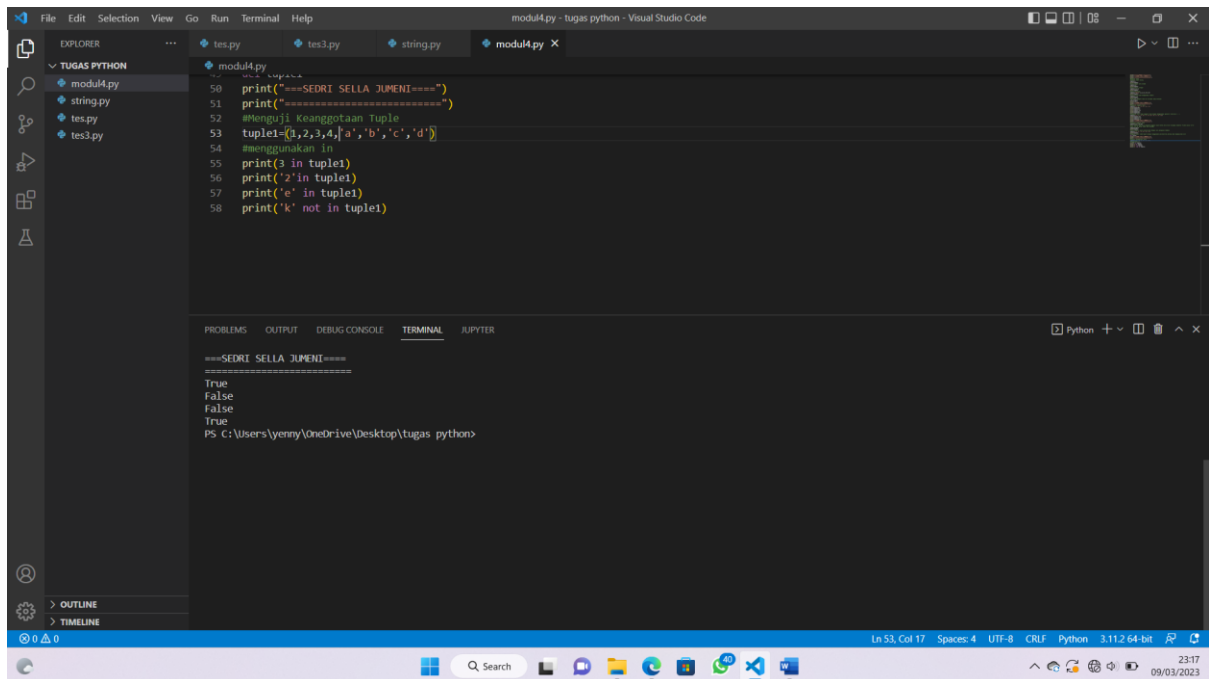
```
37 print("==SEDRi SELLA JUMENI==")
38 print("=====")
39 #mengubah anggota tuple
40 tuple1=(2,3,4,[5,6])
41 #kita memang tidak bisa mengubah anggota tuple karena bila kita hilangkan komentar # akan muncul error
42 #tapi list di dalam tuple bisa dirubah
43 tuple1[3][0]=7
44 print(tuple1)
45 #tuple diganti secara keseluruhan dengan cara penugasan kembali
46 tuple1=('p','y','t','h','o','n')
47 print(tuple1)
48 #anggota tuple tidak dapat dihapus menggunakan perintah del,karena akan menghasilkan eror
49 del tuple1
```

```
==SEDRi SELLA JUMENI==
=====
(2, 3, 4, [7, 6])
('p', 'y', 't', 'h', 'o', 'n')
PS C:\Users\yenny\OneDrive\Desktop\tugas python>
```

TASK 3 MODUL 4

Menguji Keanggotaan Tuple : kita bisa menguji apakah sebuah objek adalah anggota dari tuple atau tidak, yaitu dengan menggunakan operator `in` atau `not in` untuk kebalikannya.

Contoh:

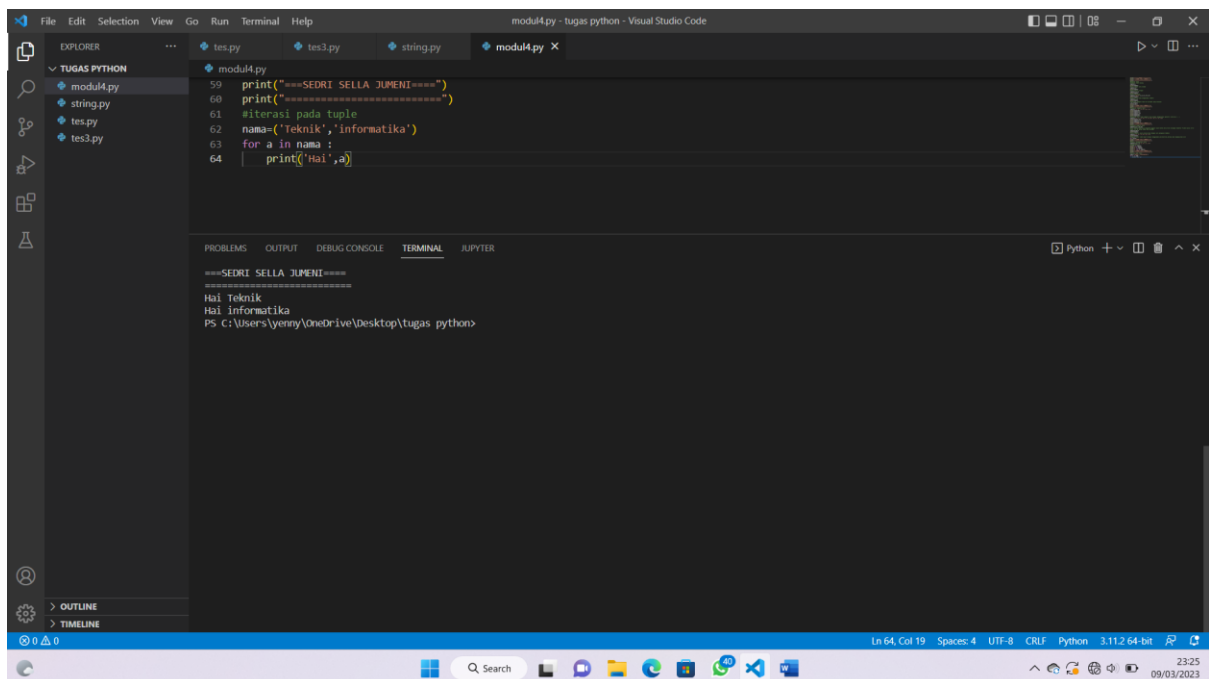


```
modul4.py
50 print("==SEDRi SELLA JUMENI==")
51 print("=====")
52 #menguji Keanggotaan Tuple
53 tuple1=(1,2,3,4,'a','b','c','d')
54 #menggunakan in
55 print(3 in tuple1)
56 print('2' in tuple1)
57 print('e' in tuple1)
58 print('k' not in tuple1)
```

```
==SEDRi SELLA JUMENI==
=====
True
False
True
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

Iterasi pada Tuple : Kita bisa menggunakan `for` untuk melakukan iterasi pada tiap anggota dalam tuple.

Contoh:



```
modul4.py
59 print("==SEDRi SELLA JUMENI==")
60 print("=====")
61 #iterasi pada tuple
62 nama=('Teknik','informatika')
63 for a in nama :
64     print('Hai',a)
```

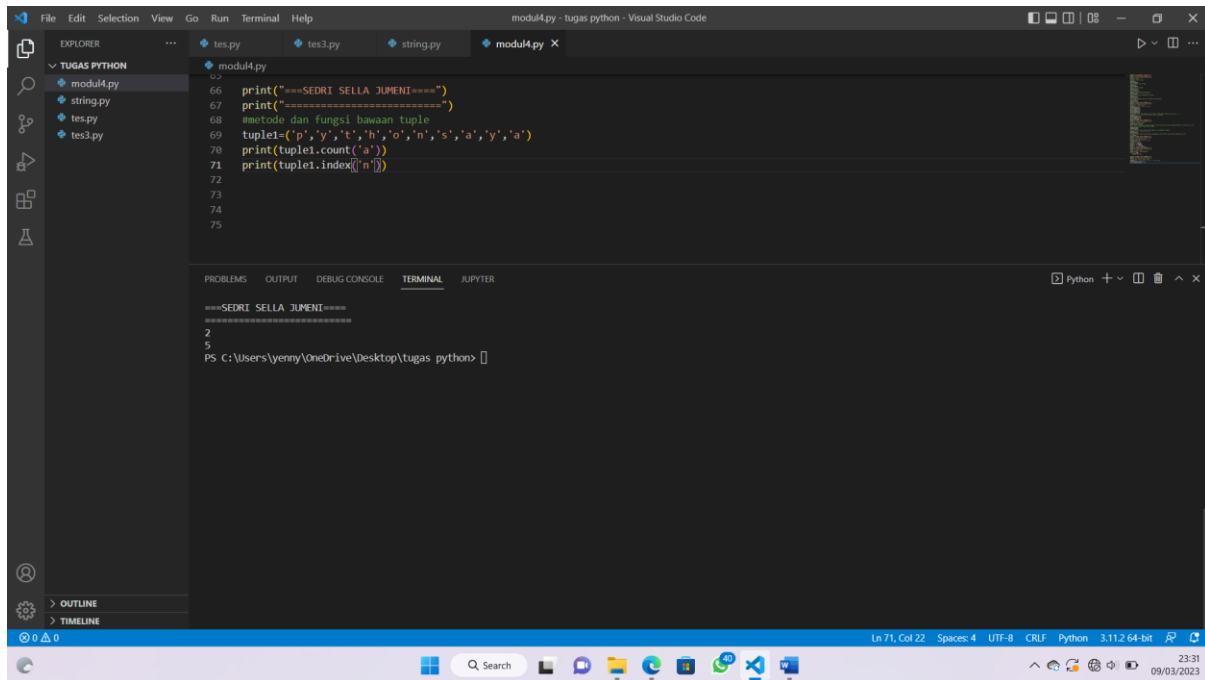
```
==SEDRi SELLA JUMENI==
=====
Hai Teknik
Hai informatika
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

TASK 3 MODUL 4

Metode dan Fungsi Bawaan Tuple : Tuple hanya memiliki dua buah metode yaitu count() dan index().

- Metode count(x) berfungsi mengembalikan jumlah item yang sesuai dengan x pada tuple.
- Metode index(x) berfungsi mengembalikan indeks dari item pertama yang sama dengan x.

Contoh:



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'TUGAS PYTHON' with files 'modul4.py', 'string.py', 'tes.py', and 'tes3.py'. The main editor displays 'modul4.py' with the following code:

```
66 print("==SEORI SELLA JUMENI==")
67 print("=====")
68 #metode dan fungsi bawaan tuple
69 tuple1=('p','y','t','h','o','n','s','a','y','a')
70 print(tuple1.count('a'))
71 print(tuple1.index('n'))
72
73
74
75
```

The TERMINAL panel at the bottom shows the output of the script:

```
==SEORI SELLA JUMENI==
=====
2
5
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

The status bar at the bottom indicates the file is 'Ln 71, Col 22', uses 'Spaces: 4', 'UTF-8' encoding, 'CRLF' line endings, and is a 'Python 3.11.2 64-bit' file.

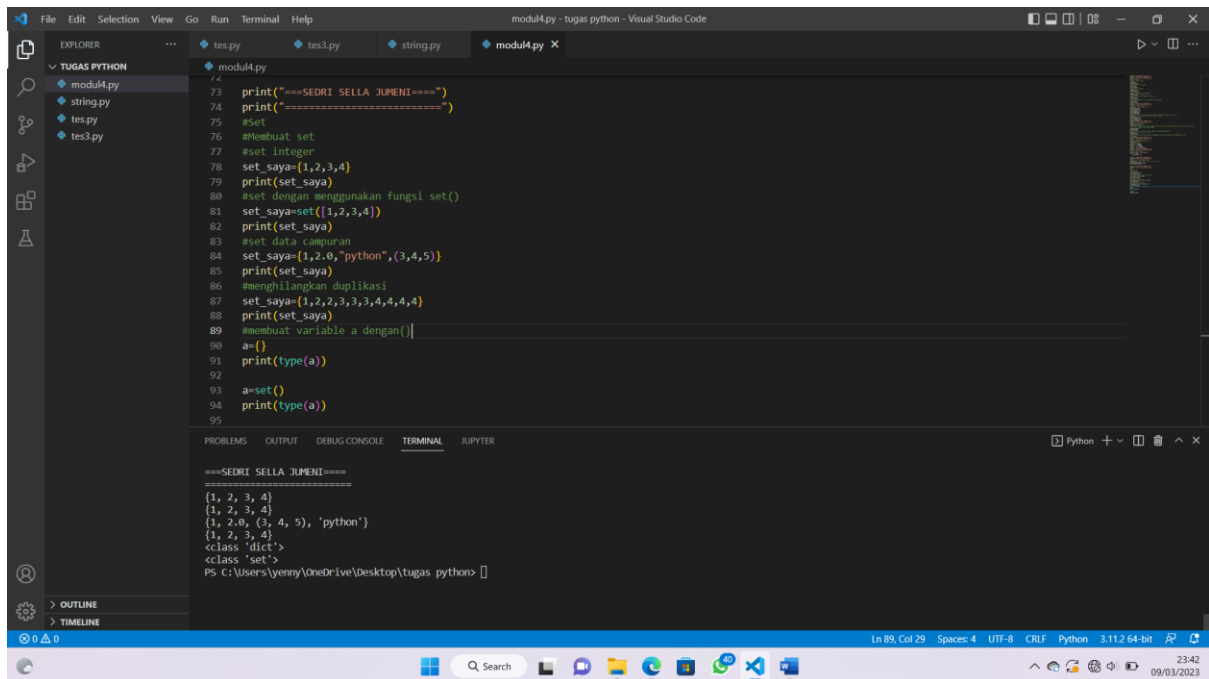
Set

Set adalah salah satu tipe data di Python yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Set bisa digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan lain - lain.

Membuat Set : Set dibuat dengan meletakkan anggota - anggotanya di dalam tanda kurung kurawal {}, dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set().

TASK 3 MODUL 4

Contoh:

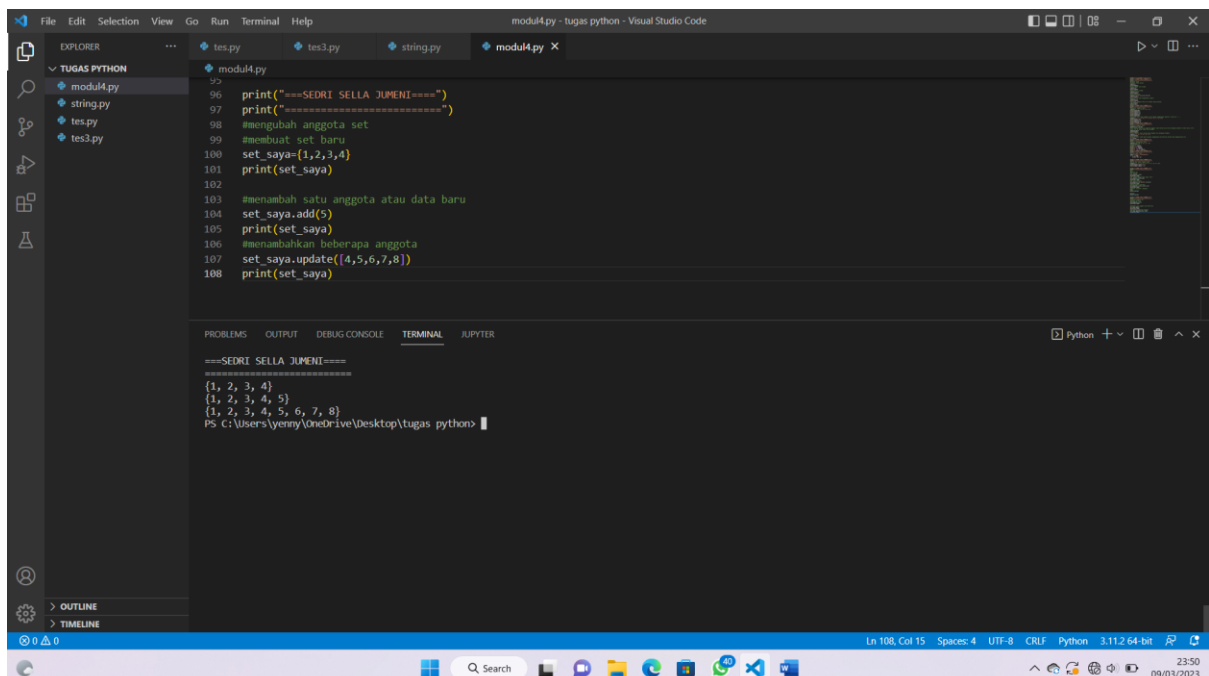


```
modul4.py
74 print("==SEDRI SELLA JUMENI==")
75 print("=====")
76 #set
77 #membuat set
78 #set integer
79 set_saya={1,2,3,4}
80 print(set_saya)
81 #set dengan menggunakan fungsi set()
82 set_saya=set([1,2,3,4])
83 print(set_saya)
84 #set data campuran
85 set_saya={1,2.0,"python",(3,4,5)}
86 print(set_saya)
87 #menghilangkan duplikasi
88 set_saya={1,2,2,3,3,3,4,4,4,4}
89 print(set_saya)
90 #membuat variable a dengan()
91 a={}
92 print(type(a))
93 a=set()
94 print(type(a))
95
```

```
==SEDRI SELLA JUMENI==
=====
{1, 2, 3, 4}
{1, 2, 3, 4}
{1, 2.0, (3, 4, 5), 'python'}
{1, 2, 3, 4}
<class 'dict'>
<class 'set'>
Ps C:\Users\yenny\OneDrive\Desktop\tugas python>
```

Mengubah Anggota Set : Set bersifat mutable. Tapi, karena set adalah tipe data tidak berurut (unordered), maka kita tidak bisa menggunakan indeks. Set tidak mendukung indeks ataupun slicing.

Contoh:



```
modul4.py
96 print("==SEDRI SELLA JUMENI==")
97 print("=====")
98 #mengubah anggota set
99 #membuat set baru
100 set_saya={1,2,3,4}
101 print(set_saya)
102
103 #menambah satu anggota atau data baru
104 set_saya.add(5)
105 print(set_saya)
106 #menambahkan beberapa anggota
107 set_saya.update([4,5,6,7,8])
108 print(set_saya)

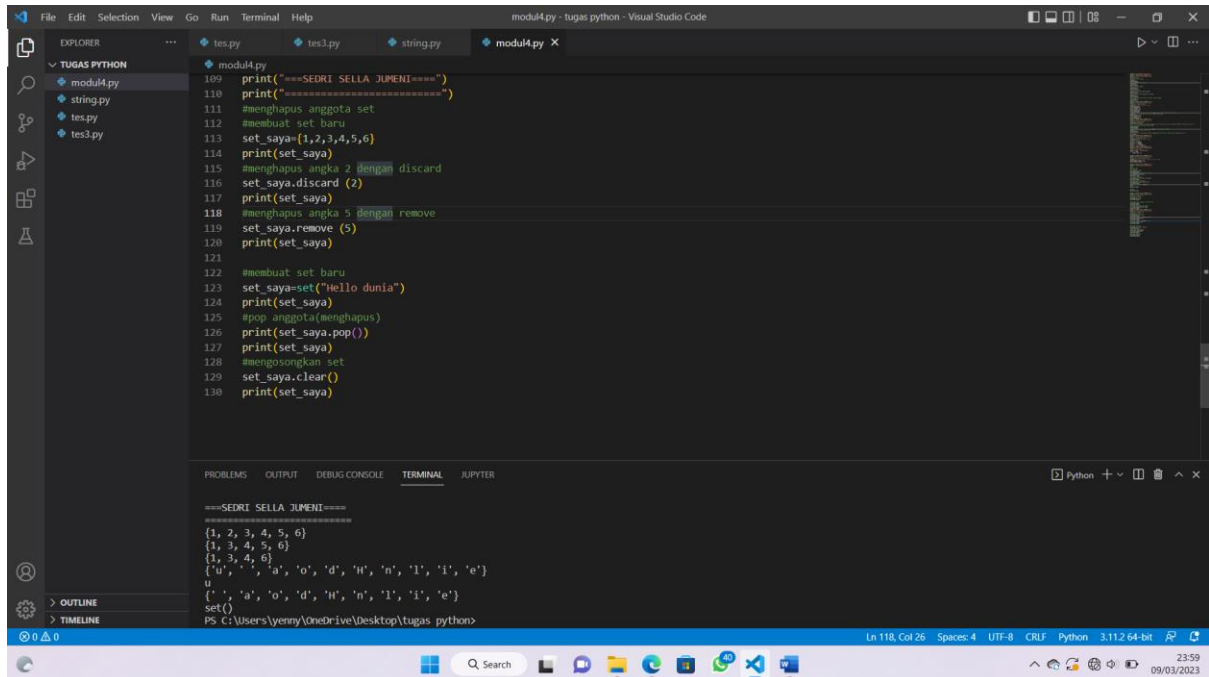
```

```
==SEDRI SELLA JUMENI==
=====
{1, 2, 3, 4}
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5, 6, 7, 8}
Ps C:\Users\yenny\OneDrive\Desktop\tugas python>
```

Menghapus Anggota : Set Kita bisa menghapus anggota set dengan menggunakan fungsi discard() dan remove().

TASK 3 MODUL 4

Contoh:



```
modul4.py
109 print("==SEORI SELLA JUMENI==")
110 print("=====")
111 #menghapus anggota set
112 #membuat set baru
113 set_saya={1,2,3,4,5,6}
114 print(set_saya)
115 #menghapus angka 2 dengan discard
116 set_saya.discard (2)
117 print(set_saya)
118 #menghapus angka 5 dengan remove
119 set_saya.remove (5)
120 print(set_saya)
121
122 #membuat set baru
123 set_saya=set("Hello dunia")
124 print(set_saya)
125 #pop anggota(menghapus)
126 print(set_saya.pop())
127 print(set_saya)
128 #mengosongkan set
129 set_saya.clear()
130 print(set_saya)
```

```
==SEORI SELLA JUMENI==
=====
{1, 2, 3, 4, 5, 6}
{1, 3, 4, 5, 6}
{'u', ' ', 'a', 'o', 'd', 'H', 'n', 'l', 'e'}
{' ', 'a', 'o', 'd', 'H', 'n', 'l', 'e'}
set()
PS C:\Users\yenny\OneDrive\Desktop\tugas python>
```

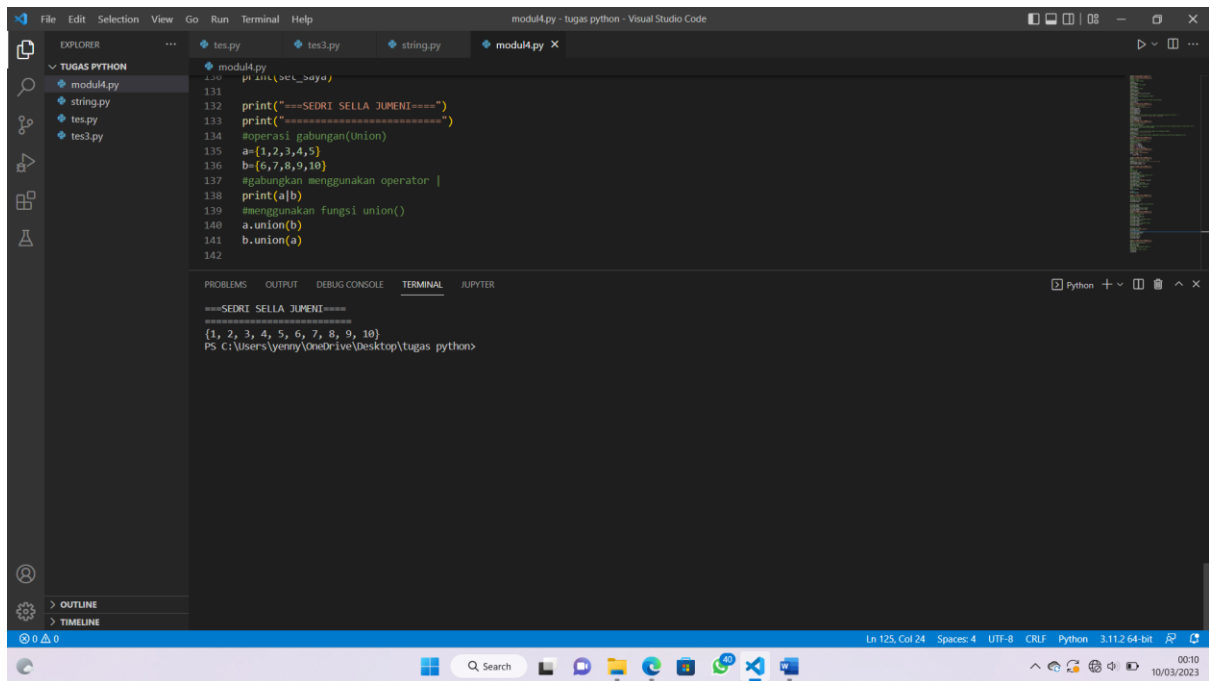
Operasi Set di Python

Set dapat digunakan untuk melakukan operasi himpunan matematika seperti gabungan, irisan, selisih, dan komplemen.

Operasi Gabungan (Union): Gabungan (union) dari A dan B adalah himpunan atau set anggota yang ada di A dan B. Gabungan dapat dibuat dengan menggunakan operator palang (`|`). Selain itu juga bisa dilakukan dengan menggunakan fungsi `union()`.

Contoh:

TASK 3 MODUL 4



The screenshot shows a Visual Studio Code editor with a Python file named `modul4.py`. The code defines two sets, `a` and `b`, and performs union operations using both the `union()` method and the `|` operator. The terminal output shows the execution results.

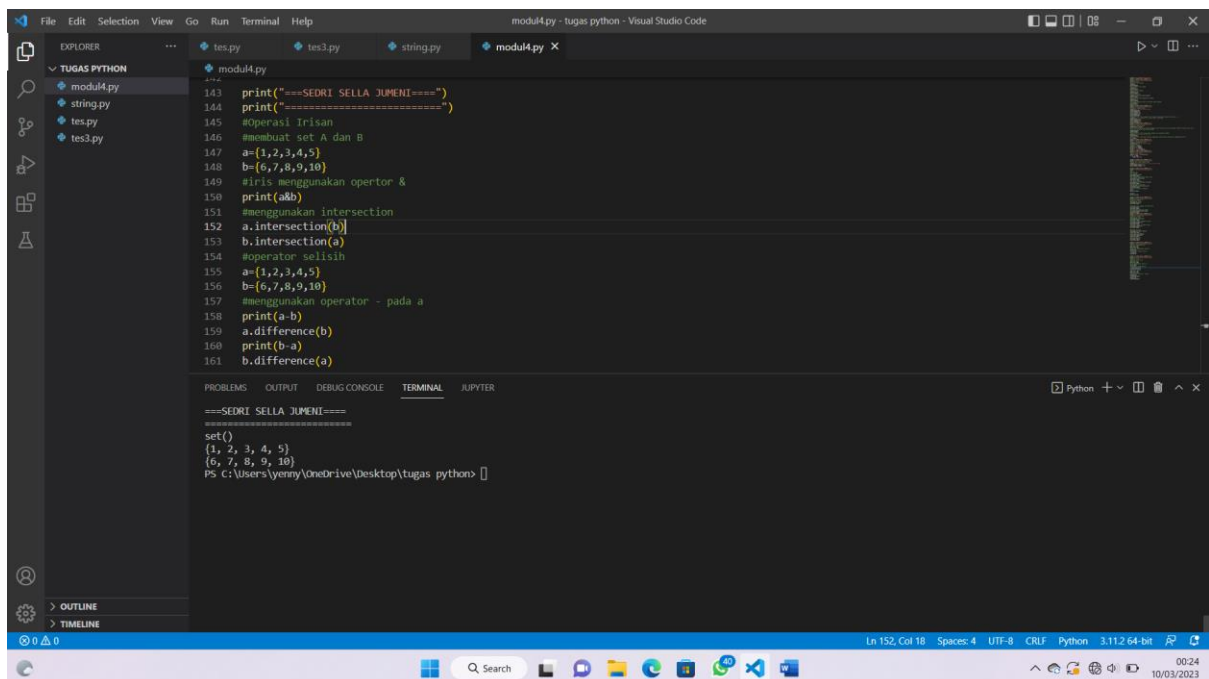
```
modul4.py
130 print(set_a, set_b)
131
132 print("==SEDRIT SELLA JUMENI==")
133 print("=====")
134 operasi_gabungan(Union)
135 a={1,2,3,4,5}
136 b={6,7,8,9,10}
137 #gabungkan menggunakan operator |
138 print(a|b)
139 #menggunakan fungsi union()
140 a.union(b)
141 b.union(a)
142
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
==SEDRIT SELLA JUMENI==
=====
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

Operasi Irisan (Intersection): B. Irisan dilakukan dengan menggunakan operator jangkar (&). Irisan juga bisa dilakukan dengan menggunakan fungsi `intersection()`.

Operasi Selisih (Difference): Selisih dilakukan dengan menggunakan operator kurang (-). Bisa juga dengan menggunakan fungsi `difference()`.

Contoh:



The screenshot shows a Visual Studio Code editor with a Python file named `modul4.py`. The code defines two sets, `a` and `b`, and performs intersection and difference operations using both operators and functions. The terminal output shows the execution results.

```
modul4.py
143 print("==SEDRIT SELLA JUMENI==")
144 print("=====")
145 #operasi Irisan
146 membuat set A dan B
147 a={1,2,3,4,5}
148 b={6,7,8,9,10}
149 #iris menggunakan oporator &
150 print(a&b)
151 #menggunakan intersection
152 a.intersection(b)
153 b.intersection(a)
154 #operator selisih
155 a={1,2,3,4,5}
156 b={6,7,8,9,10}
157 #menggunakan operator - pada a
158 print(a-b)
159 a.difference(b)
160 print(b-a)
161 b.difference(a)

```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
==SEDRIT SELLA JUMENI==
=====
set()
{1, 2, 3, 4, 5}
{6, 7, 8, 9, 10}
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

Operasi Komplemen (Symmetric Difference): himpunan atau set anggota yang ada di A dan di B, tapi tidak di keduanya. Komplemen dilakukan dengan menggunakan operator `^`. Bisa juga dengan menggunakan fungsi `symmetric_difference()`.

Contoh:

TASK 3 MODUL 4

The screenshot shows a Visual Studio Code editor with a Python file named `modul4.py` open. The file contains a script that calculates the symmetric difference of two sets, `a` and `b`.

```

162 print("===SEDRI SELLA JUMENI===")
163 print("=====")
164 superasi_komplemen (Symmetric Difference)
165 a={1,2,3,4,5}
166 b={6,7,8,9,10}
167 #menggunakan operator ^ pada a
168 print(a^b)
169 a.symmetric_difference(b)
170 print(b^a)
171 b.symmetric_difference(a)
172

```

The terminal output shows the execution of the script:

```

===SEDRI SELLA JUMENI===
=====
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
PS C:\Users\yenny\OneDrive\Desktop\tugas python>

```

Dictionary

Dictionary adalah tipe data yang anggotanya terdiri dari pasangan kunci:nilai (key:value). Dictionary bersifat tidak berurut (unordered) sehingga anggotanya tidak memiliki indeks.

Membuat Dictionary: Dictionary dibuat dengan menempatkan anggotanya di dalam tanda kurung kurawal {}, dipisahkan oleh tanda koma.

Contoh:

The image shows a Windows desktop environment. The primary focus is the Visual Studio Code (VS Code) editor, which is open to a file named 'modul4.py' within a workspace titled 'tugas python'. The editor's interface includes a sidebar on the left with the Explorer view showing the project structure, a main editor area displaying the Python code, and a bottom panel with the Output and Debug Console views. The code in 'modul4.py' defines a dictionary 'dict1' with keys 'sepatu' and 'sandal', and another dictionary 'dict2' with keys 'merah' and 'kuning'. It then creates a new dictionary 'dict3' by merging 'dict1' and 'dict2' using the update method. The output of the script is shown in the terminal window at the bottom. The Windows taskbar at the bottom shows the Start button, a search bar, and several pinned applications including File Explorer, Microsoft Edge, and VS Code. The system tray on the right shows the date and time as 10/03/2022 and 00:47.

```

modul4.py
172 print("===SEDRI SELLA JUMENI===")
173 print("=====")
174 #dictionary
175 #membuat dictionary kosong
176 dict1={}
177 print(dict1)
178 #dictionary dengan kunci integer
179 dict1={1:'sepatu',2:'sandal'}
180 print(dict1)
181 #dictionary dengan kunci campuran
182 dict1={'warna':'merah',1:[2,3,4]}
183 print(dict1)
184 #membuat dictionary menggunakan fungsi dict()
185 dict1=dict([('1','sepatu'),('2','kaos')])
186 print(dict1)
187
188 dict1=dict(m=7,n=2,o=2)
189 print (dict1)

```

```

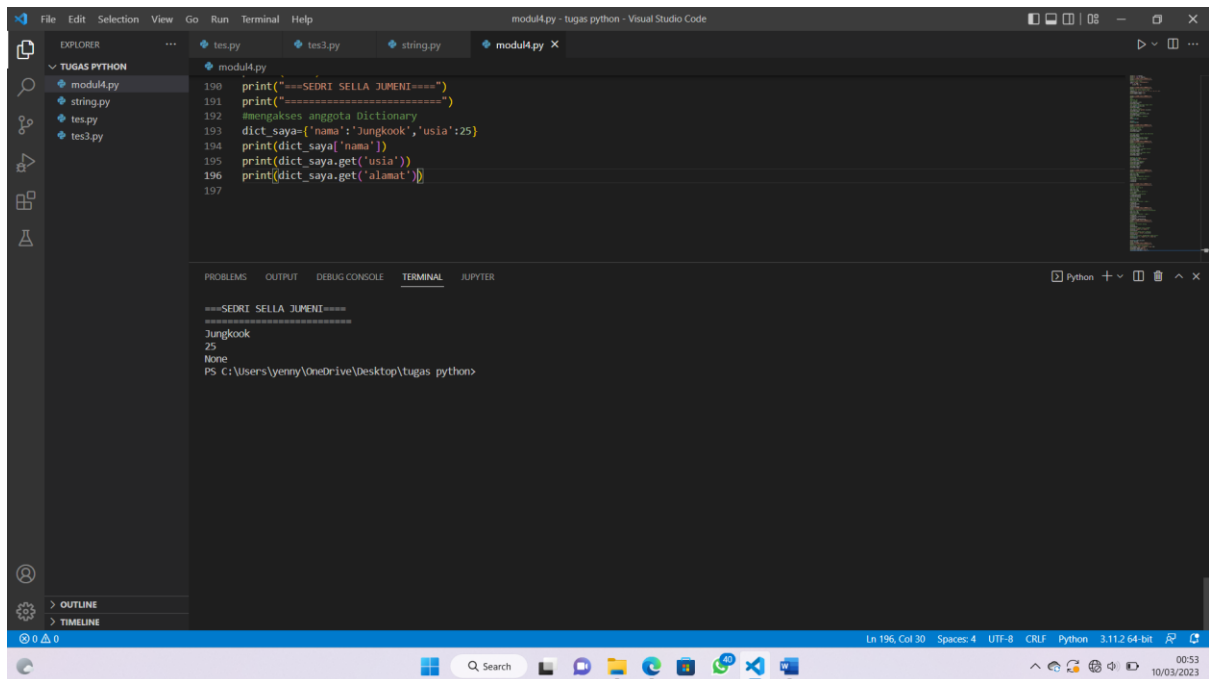
{}
{1: 'sepatu', 2: 'sandal'}
{'warna': 'merah', 1: [2, 3, 4]}
{'1': 'sepatu', '2': 'kaos'}
{'m': 7, 'n': 2, 'o': 2}
PS c:\Users\yenny\OneDrive\Desktop\tugas python>

```

Mengakses Anggota Dictionary: Dictionary tidak menggunakan indeks. Anggota dictionary diakses dengan menggunakan kuncinya. Selain itu, bisa juga diakses dengan menggunakan fungsi `get()`.

TASK 3 MODUL 4

Contoh:



The screenshot shows the Visual Studio Code interface with a Python file named `modul4.py` open. The code in the editor is as follows:

```
190 print("==SEDRI SELLA JUMENI==")
191 print("=====")
192 #mengakses anggota Dictionary
193 dict_saya={'nama':'Jungkook','usia':25}
194 print(dict_saya['nama'])
195 print(dict_saya.get('usia'))
196 print(dict_saya.get('alamat'))
197
```

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

```
==SEDRI SELLA JUMENI==
=====
Jungkook
25
None
PS C:\Users\yenny\OneDrive\Desktop\tugas_python>
```

Mengubah Anggota Dictionary Dictionary: bersifat mutable. Kita bisa menambahkan atau mengubah nilai dari anggotanya menggunakan operator penugasan. Bila kunci sudah ada, maka nilainya yang akan diupdate. Bila kunci belum ada, maka akan ditambahkan sebagai kunci baru.

Menghapus Anggota Dictionary: Kita bisa menghapus anggota tertentu pada dictionary dengan menggunakan fungsi `pop()`.

Contoh: