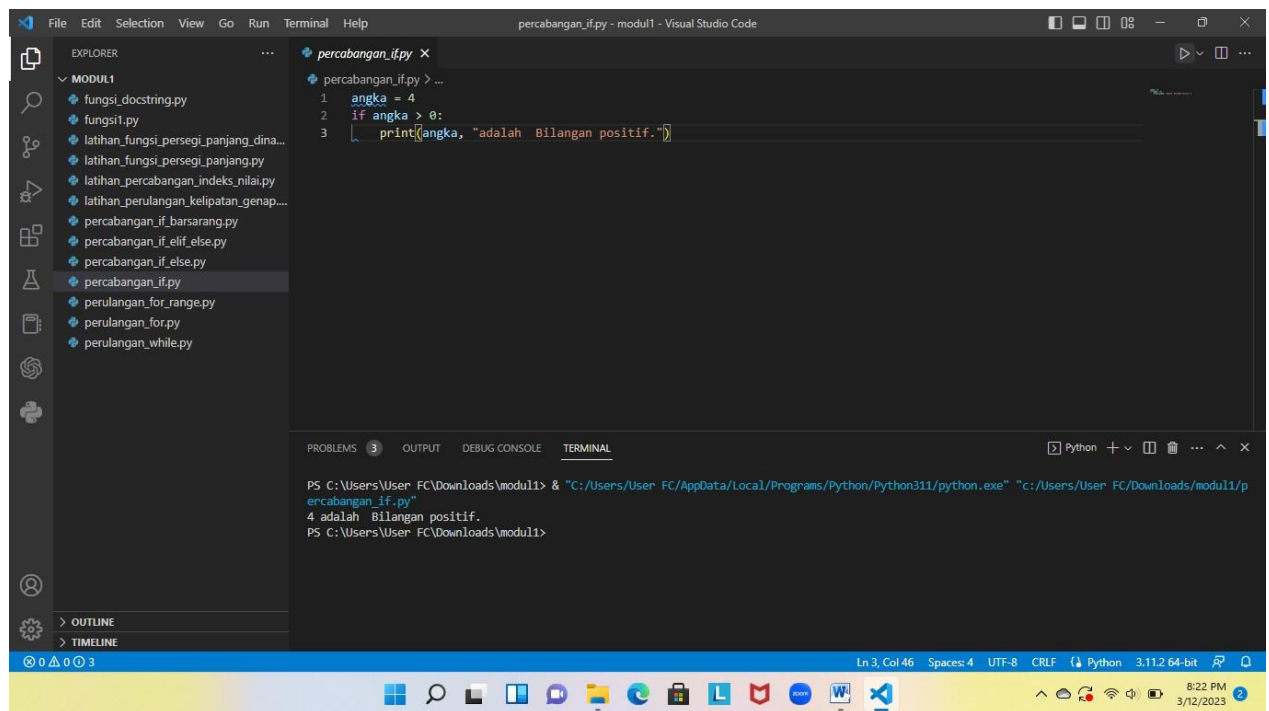


Nama : Armanita Aning
NIM : 20.01.013.002
Kelas : C Pemrograman Python

Python 5 Modul 4

1. Contoh pernyataan if



The screenshot shows the Visual Studio Code interface with a Python file named `percabangan_if.py` open. The file contains the following code:

```
1 angka = 4
2 if angka > 0:
3     print(angka, "adalah Bilangan positif.")
```

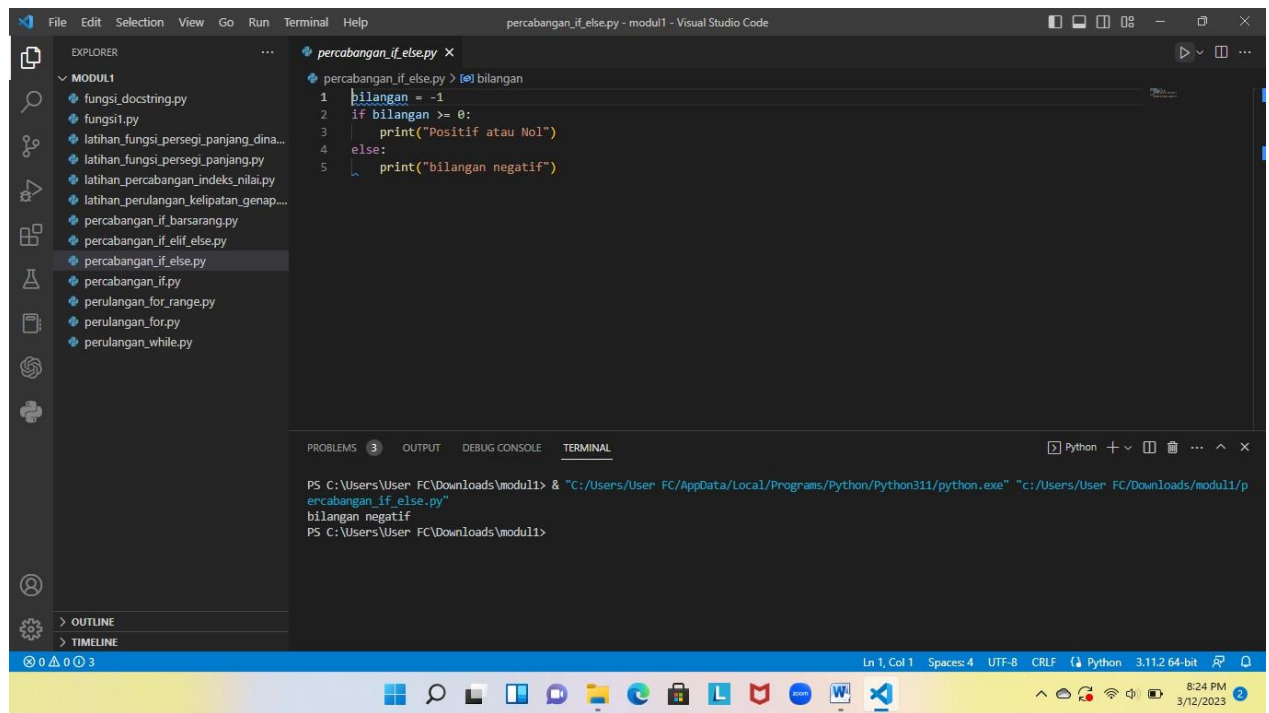
The Explorer sidebar on the left shows a list of files under the `MODUL1` folder, including `fungsi_docstring.py`, `fungsi1.py`, `latihan_fungsi_persegi_panjang_dina...`, `latihan_fungsi_persegi_panjang.py`, `latihan_percabangan_indeks_nilai.py`, `latihan_perulangan_kelipatan_genap...`, `percabangan_if_barsarang.py`, `percabangan_if_elif_else.py`, `percabangan_if_else.py`, `percabangan_if.py` (selected), `perulangan_for_range.py`, `perulangan_for.py`, and `perulangan_while.py`.

The Terminal at the bottom shows the execution of the script:

```
PS C:\Users\User FC\Downloads\modul1> & "C:\Users\User FC\AppData\Local\Programs\Python\Python311\python.exe" "c:\Users\User FC\Downloads\modul1\percabangan_if.py"
4 adalah Bilangan positif.
PS C:\Users\User FC\Downloads\modul1>
```

The status bar at the bottom indicates the current line is 3, column 46, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.11.2 64-bit, and a file icon.

2. Pernyataan if else



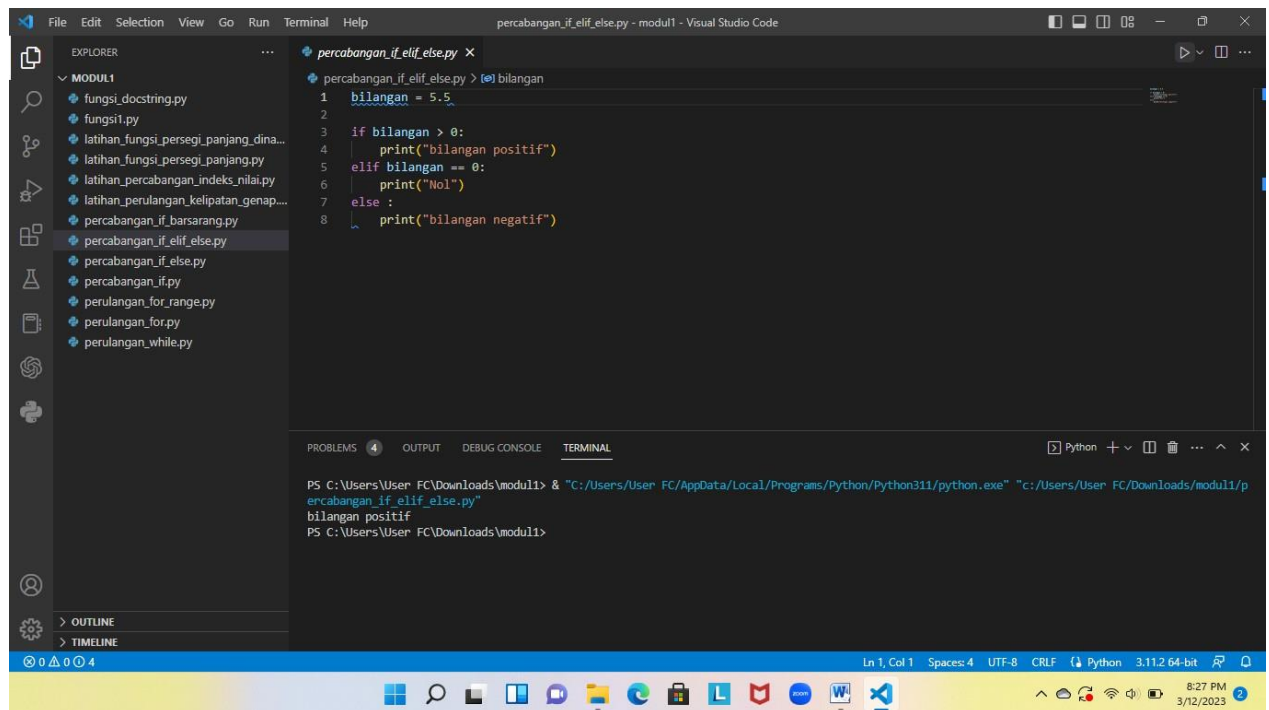
The screenshot shows the Visual Studio Code interface with a file named `percabangan_if_else.py` open. The code in the editor is as follows:

```
1 bilangan = -1
2 if bilangan >= 0:
3     print("Positif atau Nol")
4 else:
5     print("bilangan negatif")
```

The Explorer sidebar on the left shows a project named `MODUL1` with several files, including `percabangan_if_else.py`. The Terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/percabangan_if_else.py"
bilangan negatif
PS C:\Users\User FC\Downloads\modul1>
```

3. Pernyataan if else



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

```
1 bilangan = 5.5
2
3 if bilangan > 0:
4     print("bilangan positif")
5 elif bilangan == 0:
6     print("Nol")
7 else:
8     print("bilangan negatif")
```

The Explorer sidebar on the left shows the same project `MODUL1` with the file `percabangan_if_elif_else.py` selected. The Terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/percabangan_if_elif_else.py"
bilangan positif
PS C:\Users\User FC\Downloads\modul1>
```

4. If bersarang

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1'. The main editor displays the file 'percabangan_if_barsarang.py' with the following Python code:

```
1 gaji = 10000000
2 berkeluarga = True
3 punya_rumah = True
4
5 if gaji > 30000000:
6     print("Gaji udah di atas UMR")
7     if berkeluarga:
8         print("Wajib ikut asuransi dan menabung untuk pensiun")
9     else:
10        print("Tidak perlu ikut asuransi")
11
12     if punya_rumah:
13         print("wajib bayar pajak rumah")
14     else:
15         print("Tidak wajib bayar pajak rumah")
16 else:
17     print("Gaji belum UMR")
```

The terminal at the bottom shows the execution output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/percabangan_if_barsarang.py"
Gaji udah di atas UMR
Wajib ikut asuransi dan menabung untuk pensiun
wajib bayar pajak rumah
PS C:\Users\User FC\Downloads\modul1>
```

5. Percabangan indeks nilai statis

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1'. The main editor displays the file 'latihan_percabangan_indeks_nilai.py' with the following Python code:

```
1 nilai = 80
2 if nilai >= 85 and nilai <=100:
3     print("Nilai A")
4 elif nilai >= 70 and nilai <=84:
5     print("Nilai B")
6 elif nilai >= 55 and nilai <= 69:
7     print("Nilai C")
8 else:
9     print("Nilai D")
```

The terminal at the bottom shows the execution output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/latihan_percabangan_indeks_nilai.py"
Nilai B
PS C:\Users\User FC\Downloads\modul1>
```

6. Perulangan for

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1' with several Python files. The active file is 'perulangan_for.py'. The code in the editor is as follows:

```
1 nomor = [5, 5, 2]
2
3 jumlah = 0
4
5 for tampung in nomor :
6     jumlah = jumlah + tampung
7
8 print("Jumlah semuanya : ",jumlah)
```

The terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/perulangan_for.py"
Jumlah semuanya : 12
PS C:\Users\User FC\Downloads\modul1>
```

7. Perulangan for dengan range

The screenshot shows the Visual Studio Code interface with the same file explorer. The active file is 'perulangan_for_range.py'. The code in the editor is as follows:

```
1 for hitung in range(5):
2     print("Hitung :", hitung)
```

The terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/perulangan_for_range.py"
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\Users\User FC\Downloads\modul1>
```

8. Perulangan while

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1' with several Python files. The active file is 'perulangan_while.py'. The code in the editor is as follows:

```
1 hitung = 0
2
3 while (hitung < 5):
4     print("Hitung :", hitung)
5     hitung = hitung + 1
```

The terminal at the bottom shows the execution of the script, outputting the values of 'hitung' from 0 to 4.

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/perulangan_while.py"
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\Users\User FC\Downloads\modul1>
```

9. Kelipatan bilangan genap

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1' with several Python files. The active file is 'latihan_perulangan_kelipatan_genap.py'. The code in the editor is as follows:

```
1 i = 0
2 n = int(input("Masukan Batas :"))
3
4 for i in range(n):
5     if i%2 == 0:
6         print("Bilangan :", i)
7
8     i = i + 1
```

The terminal at the bottom shows the execution of the script, where the user inputs '5' for the limit, and the program outputs the even numbers 0, 2, and 4.

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/latihan_perulangan_kelipatan_genap.py"
Masukan Batas :5
Bilangan : 0
Bilangan : 2
Bilangan : 4
PS C:\Users\User FC\Downloads\modul1>
```

10. Fungsi

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1'. The main editor displays a file named 'fungsi1.py' with the following Python code:

```
1 def sapa(nama):  
2     print("hai," + nama + ". Apa Kabar")  
3     return nama  
4  
5 # panggilan fungsi  
6 # output : hai, Anna. Apa Kabar?  
7 sapa("Anna")
```

The bottom panel shows the 'TERMINAL' output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/fungsi1.py"  
hai,Anna. Apa Kabar  
PS C:\Users\User FC\Downloads\modul1>
```

11. Docstring

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'MODUL1'. The main editor displays a file named 'fungsi_docstring.py' with the following Python code:

```
1 def sapa(nama):  
2     "contoh cetak keterangan"  
3     print("Hai," + nama + ". Apa Kabar?")  
4     return nama  
5  
6 sapa("Anna")  
7 print(sapa.__doc__)
```

The bottom panel shows the 'TERMINAL' output:

```
PS C:\Users\User FC\Downloads\modul1> & "C:/Users/User FC/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/User FC/Downloads/modul1/fungsi_docstring.py"  
Hai,Anna. Apa Kabar?  
contoh cetak keterangan  
PS C:\Users\User FC\Downloads\modul1>
```

12. Fungsi persegi panjang


```
def persegipanjang(panjang, lebar):  
    luas = panjang * lebar  
    print("Luasnya :", luas)  
    return luas  
  
print("Menghitung luas persegi panjang")  
persegipanjang(4,6)
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

Python 3.11.2 64-bit

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

8:39 PM 3/12/2023

13. Persegi panjang dinamis

```
def persegipanjang(panjang, lebar):  
    luas = panjang * lebar  
    print("Luasnya :", luas)  
    return luas  
  
print("Menghitung Luas Persegi panjang")  
a = int(input("Masukan Panjang :"))  
b = int(input("Masukan Lebar :"))  
persegipanjang(a,b)
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

Python 3.11.2 64-bit

Ln 9, Col 20 Spaces: 4 UTF-8 CRLF

8:41 PM 3/12/2023

The screenshot shows a Visual Studio Code editor with a Python file named `modul5.py`. The Explorer sidebar on the left shows a project structure with a `python` folder containing several `modulX.py` files. The main editor displays the following code:

```
1 # kelas
2 class Marvel:
3     pass
4
5 # object
6 marvel1 = Marvel()
7 marvel2 = Marvel()
8 marvel3 = Marvel()
9
10 marvel1.name = "Iron Man"
11 marvel1.health = "1000"
12
13 marvel2.name = "Thor"
14 marvel2.health = "800"
15
16 marvel3.name = "Captain America"
17 marvel3.health = "900"
18
19 # pemanggilan
20 print(marvel1.name)
21 print(marvel1.health)
22 print(marvel1.__dict__)
23
24
25 class Marvel:
26     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
27         self.name = inputName
28         self.health = inputHealth
29         self.power = inputPower
30         self.armor = inputArmor
31
32
33 marvel1 = Marvel("Iron Man", 100, 10, 90)
```

The status bar at the bottom indicates the cursor is at line 108, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.10.4 64-bit.

The screenshot shows the same Visual Studio Code editor with `modul5.py`, but with more code added. The Explorer sidebar is the same. The main editor displays the following code:

```
29     self.power = inputPower
30     self.armor = inputArmor
31
32 marvel1 = Marvel("Iron Man", 100,10,90)
33 marvel2 = Marvel("Thor", 90,15,100)
34 marvel3 = Marvel("Captain America", 80,5,70)
35
36 print(marvel1.name)
37 print(marvel2.health)
38 print(marvel3.__dict__)
39
40 class Marvel:
41     # class variable
42     jumlah = 0
43
44     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
45         # instance variable
46         self.name = inputName
47         self.health = inputHealth
48         self.power = inputPower
49         self.armor = inputArmor
50         Marvel.jumlah += 1
51         print("Hero Marvel dengan nama : " + inputName)
52
53 marvel1 = Marvel("Iron Man", 1000,900,800)
54 print(Marvel.jumlah)
55 marvel2 = Marvel("Thor", 900,1000,900)
56 print(Marvel.jumlah)
57 marvel3 = Marvel("Captain America", 800,700,600)
58 print(Marvel.jumlah)
59
60 class Marvel:
```

The status bar at the bottom indicates the cursor is at line 31, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.10.4 64-bit.


```
.vscode > pratikum > python > modul5.py > ...
57 marvel3 = Marvel("Captain America", 800,700,600)
58 print(Marvel.jumlah)
59
60 class Marvel:
61
62     # instance variable
63     self.name = inputName
64     self.health = inputHealth
65     self.power = inputPower
66     self.armor = inputArmor
67
68     # void function, method tanpa return
69     def siapa(self):
70         print("Namaku adalah : " + self.name)
71
72     # method dengan argumen
73     def healthTambah(self, tambah):
74         self.health += tambah
75
76     # method dengan return
77     def getHealth(self):
78         return self.health
79
80
81 marvel1 = Marvel("Iron Man", 1000,900,800)
82 marvel2 = Marvel("Thor", 900,1000,900)
83 marvel3 = Marvel("Captain America", 800,700,600)
84
85 # pemanggilan method
86 marvel1.siapa()
87
88 #pemakaian method dengan argumen
89 marvel1.healthTambah(10)
```

```
.vscode > pratikum > python > modul5.py > ...
60 class Marvel:
61
62     # instance variable
63     self.name = inputName
64     self.health = inputHealth
65     self.power = inputPower
66     self.armor = inputArmor
67
68     # void function, method tanpa return
69     def siapa(self):
70         print("Namaku adalah : " + self.name)
71
72     # method dengan argumen
73     def healthTambah(self, tambah):
74         self.health += tambah
75
76     # method dengan return
77     def getHealth(self):
78         return self.health
79
80
81 marvel1 = Marvel("Iron Man", 1000,900,800)
82 marvel2 = Marvel("Thor", 900,1000,900)
83 marvel3 = Marvel("Captain America", 800,700,600)
84
85 # pemanggilan method
86 marvel1.siapa()
87
88 #pemakaian method dengan argumen
89 marvel1.healthTambah(10)
90 print(marvel1.health)
91
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with a 'PYTHON' folder containing several Python files: modul1.py, modul2.py, modul3.py, modul4.py, modul5.py (selected), modul6.py, modul7.py, praktikum.py, and settings.json. The main editor displays the code for modul5.py. The code defines a 'Marvel' class with attributes 'name', 'health', 'attackPower', and 'armorNumber'. It includes methods for 'serang' (attack) and 'diserang' (be attacked). The code also creates instances of the 'Marvel' class for 'Iron Man' and 'Thor' and demonstrates the 'serang' method.

```
.vscode > praktikum > python > modul5.py > ...
92 # mengembalikan nilai dengan method
93 print(marvel1.getHealth())
94
95 class Marvel:
96
97     def __init__(self, name, health, attackPower, armorNumber):
98         # instance variable
99         self.name = name
100         self.health = health
101         self.attackPower = attackPower
102         self.armor = armorNumber
103
104     def serang(self, lawan):
105         print(self.name + " menyerang " + lawan.name)
106         lawan.diserang(self, self.attackPower)
107
108     def diserang(self, lawan, attackPower_lawan):
109         print(self.name + "diserang" + lawan.name)
110         attack_diterima = attackPower_lawan
111         print("Serang terasa : " + str(attack_diterima))
112         self.health -= attack_diterima
113         print("Darah " + self.name + " tersisa " + str(self.health))
114
115 ironman = Marvel("Iron Man",100,10,5)
116 thor = Marvel("Thor",95,15,10)
117
118 #ironman.serang()
119 ironman.serang(thor)
120 #print ("\n")
121 # #ironman.serang(thor)
122 #print("\n")
123 #thor.serang(ironman)
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with a 'PYTHON' folder containing several Python files: modul1.py, modul2.py, modul3.py, modul4.py, modul5.py (selected), modul6.py, modul7.py, praktikum.py, and settings.json. The main editor displays the code for modul5.py. The code defines a 'Marvel' class with attributes 'name', 'health', 'attackPower', and 'armorNumber'. It includes methods for 'serang' (attack) and 'diserang' (be attacked). The code also creates instances of the 'Marvel' class for 'Iron Man' and 'Thor' and demonstrates the 'serang' method.

```
python/modul5.py
Iron Man
1000
{'name': 'Iron Man', 'health': '1000'}
Iron Man
90
{'name': 'Captain America', 'health': 80, 'power': 5, 'armor': 70}
Hero Marvel dengan nama : Iron Man
1
Hero Marvel dengan nama : Thor
2
Hero Marvel dengan nama : Captain America
3
Namaku adalah :Iron Man
1010
1010
Iron Man menyerang Thor
ThordiserangIron Man
Hero Marvel dengan nama : Captain America
3
Namaku adalah :Iron Man
1010
1010
Iron Man menyerang Thor
ThordiserangIron Man
Serang terasa : 10
Darah Thor tersisa 85
PS C:\Users\USER\Desktop\python>
```