

Nama : Rifki Fadilah

Kelas : R1

NIM : 210511011

PBO2 Praktikum3

Overload 1

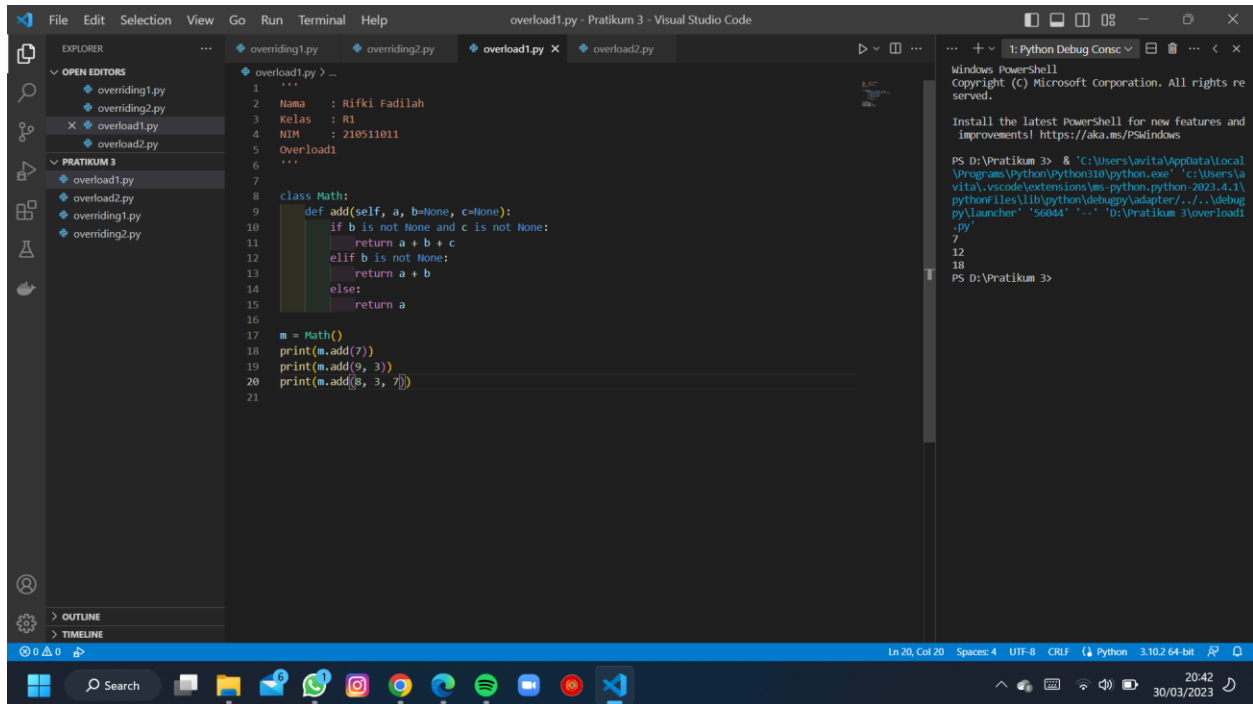
Script:

```
'''
Nama      : Rifki Fadilah
Kelas    : R1
NIM       : 210511011
Overload1
'''

class Math:
    def add(self, a, b=None, c=None):
        if b is not None and c is not None:
            return a + b + c
        elif b is not None:
            return a + b
        else:
            return a

m = Math()
print(m.add(7))
print(m.add(9, 3))
print(m.add(8, 3, 7))
```

Hasil Running Program



Overload 2

Script:

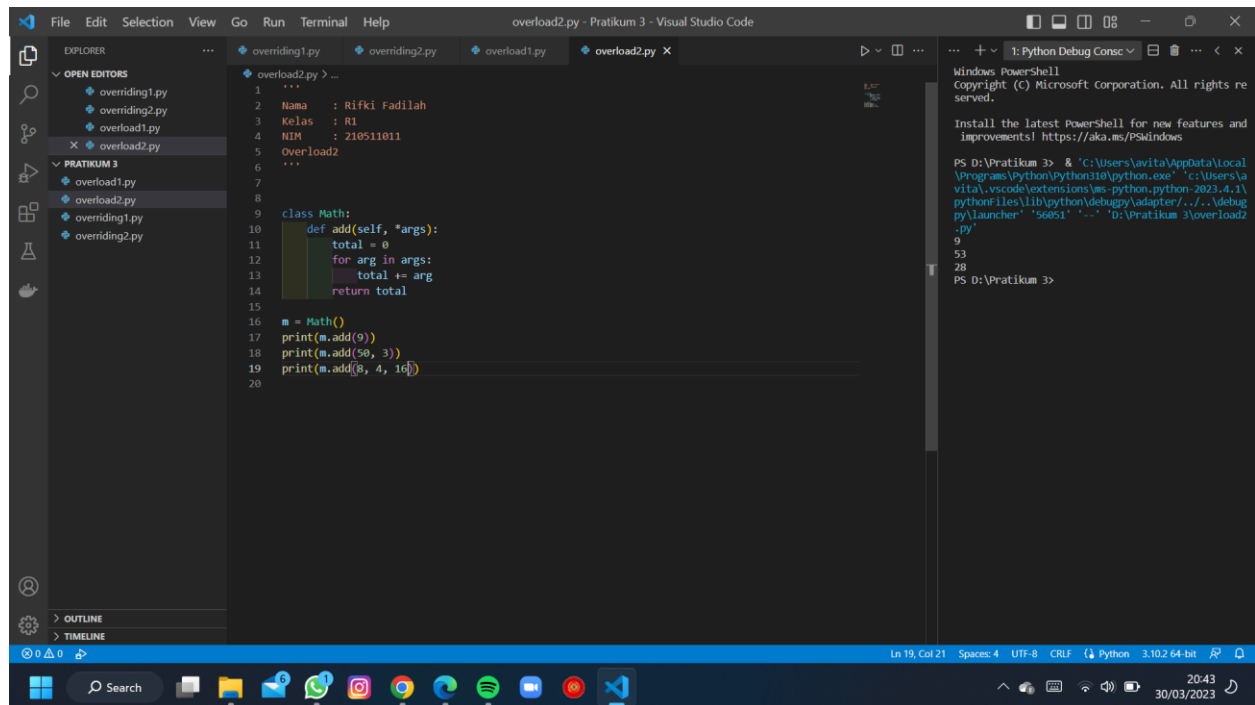
```
'''
Nama      : Rifki Fadilah
Kelas    : R1
NIM       : 210511011
Overload2

class Math:
    def add(self, *args):
        total = 0
        for arg in args:
            total += arg
```

```
        return total
```

```
m = Math()  
print(m.add(9))  
print(m.add(50, 3))  
print(m.add(8, 4, 16))
```

Hasil Running Program



Overriding 1

Script:

```
'''
Nama      : Rifki Fadilah
Kelas    : R1
NIM       : 210511011
Overriding1
'''

class Pekerja:
    def __init__(self, name):
        self.name = name

    def gaji(self):
        raise NotImplementedError

class Supervisor(Pekerja):
    def gaji(self):
        return ' Rp 3.000.000'

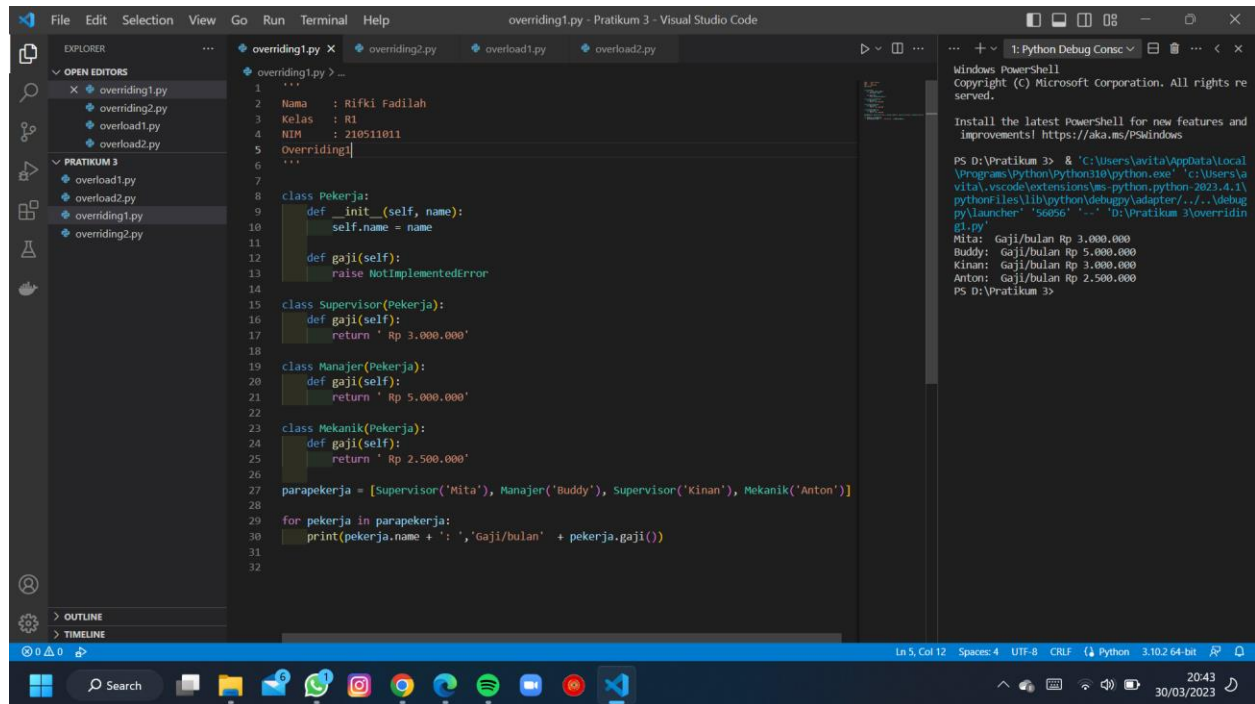
class Manajer(Pekerja):
    def gaji(self):
        return ' Rp 5.000.000'

class Mekanik(Pekerja):
    def gaji(self):
        return ' Rp 2.500.000'

parapekerja = [Supervisor('Mita'), Manajer('Buddy'), Supervisor('Kinan'),
Mekanik('Anton')]

for pekerja in parapekerja:
    print(pekerja.name + ': ', 'Gaji/bulan' + pekerja.gaji())
```

Hasil Running Program



The screenshot shows a Visual Studio Code editor with a Python file named `overriding1.py`. The code defines a base class `Pekerja` with attributes `Nama`, `Kelas`, and `NIM`. It also defines three subclasses: `Supervisor`, `Manajer`, and `Mekanik`, each with a `gaji` method. The `Supervisor` class raises a `NotImplementedError`. The `Manajer` class returns 'Rp 5.000.000'. The `Mekanik` class returns 'Rp 2.500.000'. The main code creates instances of these classes and prints their names and salaries.

```
1 '''
2 Nama : Rifki Fadilah
3 Kelas : R1
4 NIM : 210511011
5 Overriding1
6 '''
7
8 class Pekerja:
9     def __init__(self, name):
10         self.name = name
11
12     def gaji(self):
13         raise NotImplementedError
14
15 class Supervisor(Pekerja):
16     def gaji(self):
17         return 'Rp 3.000.000'
18
19 class Manajer(Pekerja):
20     def gaji(self):
21         return 'Rp 5.000.000'
22
23 class Mekanik(Pekerja):
24     def gaji(self):
25         return 'Rp 2.500.000'
26
27 parapekerja = [Supervisor('Mita'), Manajer('Buddy'), Supervisor('Kinan'), Mekanik('Anton')]
28
29 for pekerja in parapekerja:
30     print(pekerja.name + ': ' + 'Gaji/bulan' + pekerja.gaji())
31
32
```

The PowerShell terminal on the right shows the output of the program:

```
PS D:\Pratikum 3> & 'C:\Users\varita\AppData\Local\Programs\Python\Python110\python.exe' 'C:\Users\varita\vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\python\debugpy\adapter\..\debugpy\launcher' '56056' '-.' 'D:\Pratikum 3\overriding1.py'
Mita: Gaji/bulan Rp 3.000.000
Buddy: Gaji/bulan Rp 5.000.000
Kinan: Gaji/bulan Rp 3.000.000
Anton: Gaji/bulan Rp 2.500.000
PS D:\Pratikum 3>
```

Overriding 2

Script:

```
'''
Nama : Rifki Fadilah
Kelas : R1
NIM : 210511011
Overriding2
'''

class Hero:
    def __init__(self, name):
        self.name = name

    def gaji(self):
        raise NotImplementedError

class Fighter(Hero):
```

```

    def gaji(self):
        return ' Exp Liner'

class Assassin(Hero):
    def gaji(self):
        return ' Junggler'

class Tank(Hero):
    def gaji(self):
        return ' Roamer'

class Marksman(Hero):
    def gaji(self):
        return ' Gold Liner'

class Mage(Hero):
    def gaji(self):
        return ' Mid Liner'

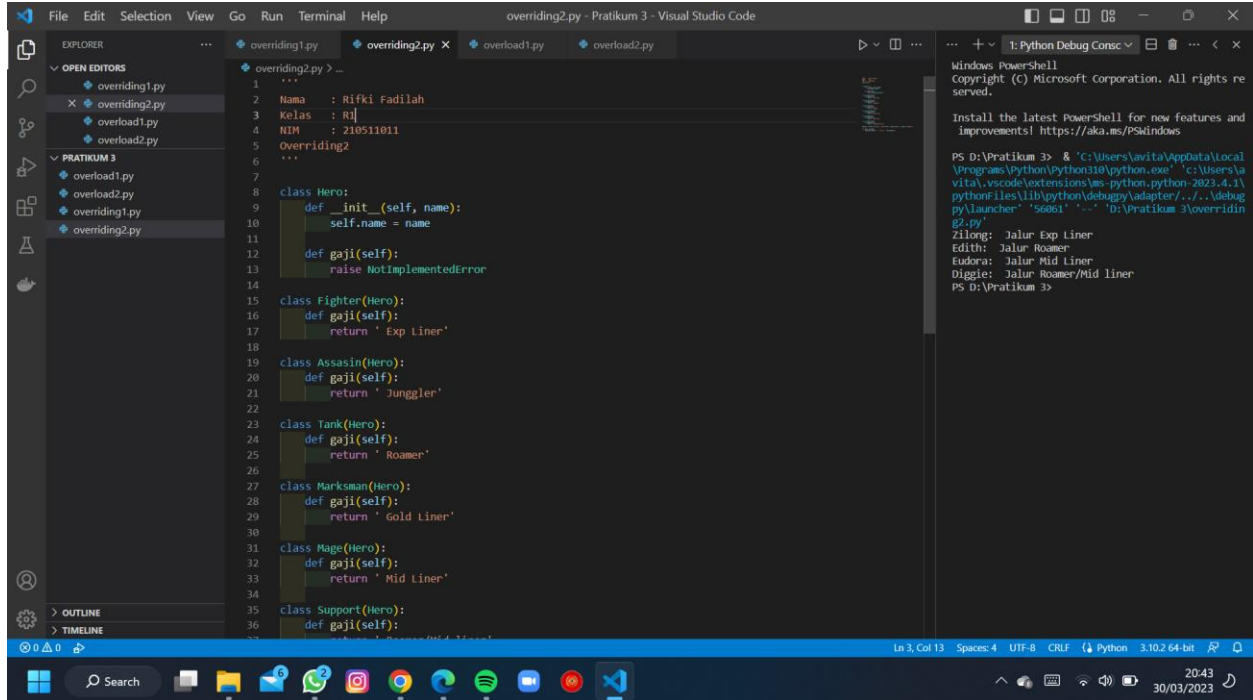
class Support(Hero):
    def gaji(self):
        return ' Roamer/Mid liner'

heroml = [Fighter('Zilong'), Tank('Edith'), Mage('Eudora'), Support('Diggie')]

for hero in heroml:
    print(hero.name + ': ', 'Jalur' + hero.gaji())

```

Hasil Running Program



The screenshot displays the Visual Studio Code interface with a Python file named `overriding2.py` open. The file contains a class hierarchy where `Hero` is the base class, and `Fighter`, `Assasin`, `Tank`, `Marksman`, `Mage`, and `Support` are subclasses. Each subclass overrides the `gaji` method to return a specific value. The `Hero` class has an `__init__` method that sets `self.name` and a `gaji` method that raises a `NotImplementedError`.

```
1 """
2 Nama : Rifki Fadilah
3 Kelas : RIJ
4 NIM : 210511011
5 Overriding2
6 """
7
8 class Hero:
9     def __init__(self, name):
10         self.name = name
11
12     def gaji(self):
13         raise NotImplementedError
14
15 class Fighter(Hero):
16     def gaji(self):
17         return 'Exp Liner'
18
19 class Assasin(Hero):
20     def gaji(self):
21         return 'Jungglar'
22
23 class Tank(Hero):
24     def gaji(self):
25         return 'Roamer'
26
27 class Marksman(Hero):
28     def gaji(self):
29         return 'Gold Liner'
30
31 class Mage(Hero):
32     def gaji(self):
33         return 'Mid Liner'
34
35 class Support(Hero):
36     def gaji(self):
```

The PowerShell terminal on the right shows the execution of a command to run the Python script, resulting in the following output:

```
PS D:\Pratikum 3> & 'C:\Users\varita\AppData\Local\Programs\Python\Python110\python.exe' 'C:\Users\varita\vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\python\debugpy\adapter\..\debugpy\launcher' '56061' '-.' 'D:\Pratikum 3\overriding2.py'
Zilong: Jalur Exp Liner
Edith: Jalur Roamer
Eudora: Jalur Mid Liner
Diggie: Jalur Roamer/Mid liner
PS D:\Pratikum 3>
```