

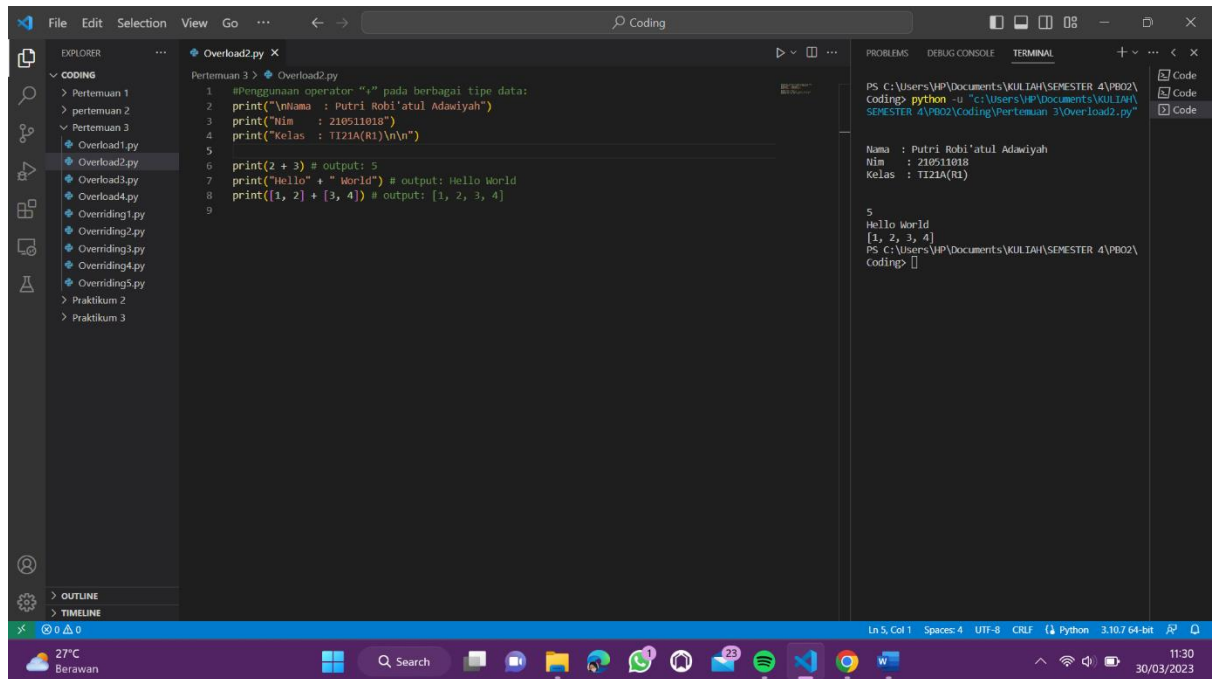
Nama : Putri Robi'atul Adawiyah

Nim : 210511018

Kelas : TI21A (R1)

Latihan 3 PBO2

1. Polymorphism statis (Overload)



```
File Edit Selection View Go ... Coding
```

EXPLORER

- CODING
 - Pertemuan 1
 - pertemuan 2
 - Pertemuan 3
 - Overload1.py
 - Overload2.py
 - Overload3.py
 - Overload4.py
 - Overriding1.py
 - Overriding2.py
 - Overriding3.py
 - Overriding4.py
 - Overriding5.py
 - Praktikum 2
 - Praktikum 3

Overload2.py

```
1 #Penggunaan operator "+" pada berbagai tipe data:
2 print("\nNama : Putri Robi'atul Adawiyah")
3 print("Nim : 210511018")
4 print("Kelas : TI21A(R1)\n\n")
5
6 print(2 + 3) # output: 5
7 print("Hello" + " World") # output: Hello World
8 print([1, 2] + [3, 4]) # output: [1, 2, 3, 4]
9
```

TERMINAL

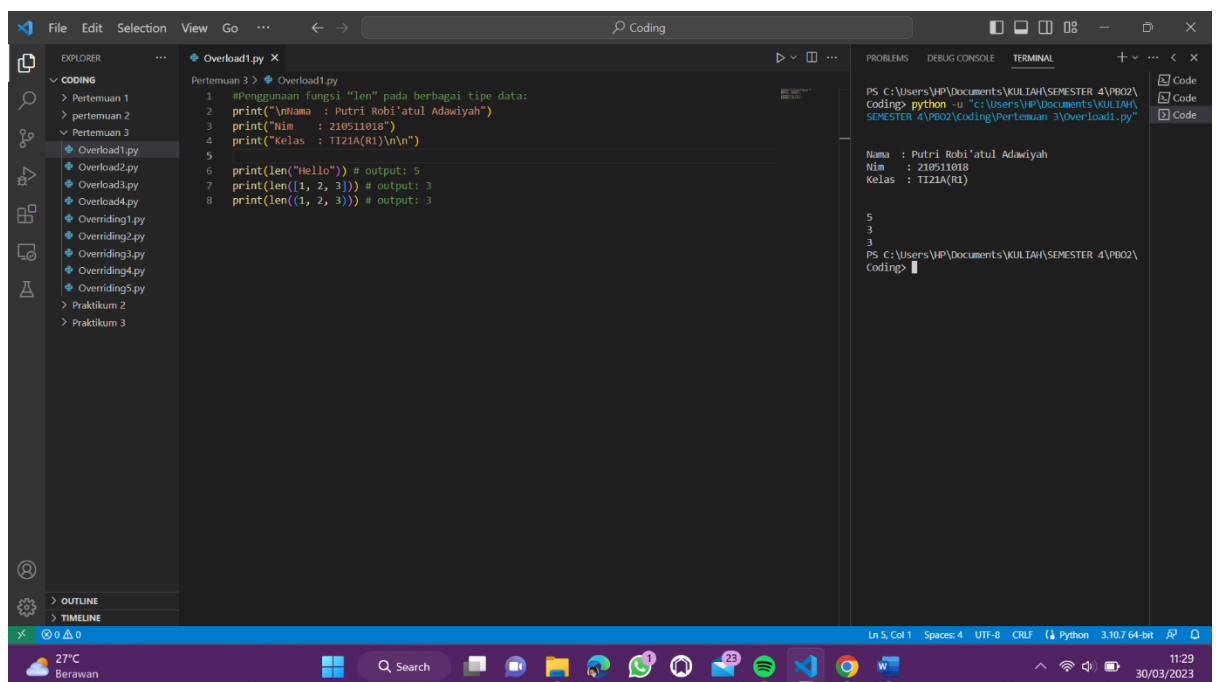
```
PS C:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding> python -u "c:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding\Pertemuan 3\Overload2.py"

Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

5
Hello World
[1, 2, 3, 4]
PS C:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding>
```

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit

27°C Berawan 11:30 30/03/2023



```
File Edit Selection View Go ... Coding
```

EXPLORER

- CODING
 - Pertemuan 1
 - pertemuan 2
 - Pertemuan 3
 - Overload1.py
 - Overload2.py
 - Overload3.py
 - Overload4.py
 - Overriding1.py
 - Overriding2.py
 - Overriding3.py
 - Overriding4.py
 - Overriding5.py
 - Praktikum 2
 - Praktikum 3

Overload1.py

```
1 #Penggunaan fungsi "len" pada berbagai tipe data:
2 print("\nNama : Putri Robi'atul Adawiyah")
3 print("Nim : 210511018")
4 print("Kelas : TI21A(R1)\n\n")
5
6 print(len("Hello")) # output: 5
7 print(len([1, 2, 3])) # output: 3
8 print(len((1, 2, 3))) # output: 3
```

TERMINAL

```
PS C:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding> python -u "c:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding\Pertemuan 3\Overload1.py"

Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

5
3
3
PS C:\Users\VP\Documents\KULIAH\SEMESTER 4\PBO2\Coding>
```

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.7 64-bit

27°C Berawan 11:29 30/03/2023

The screenshot shows the Visual Studio Code editor with a file named `Overload3.py` open. The file explorer on the left shows a project structure with folders `Pertemuan 1`, `Pertemuan 2`, `Pertemuan 3`, and `Praktikum 2`, `Praktikum 3`. The `Overload3.py` file is selected. The code in the editor is as follows:

```
1 #Penggunaan fungsi "max" pada berbagai tipe data:
2 print("\nNama : Putri Robi'atul Adawiyah")
3 print("Nim : 210511018")
4 print("Kelas : TI21A(R1)\n\n")
5
6 print(max(2, 5)) # output: 5
7 print(max([1, 2, 3])) # output: 3
8 print(max("Hello")) # output: "o"
```

The terminal on the right shows the command `python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding\Pertemuan 3\Overload3.py"` and its output:

```
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

5
3
o
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding>
```

The screenshot shows the Visual Studio Code editor with a file named `Overload4.py` open. The file explorer on the left shows a project structure with folders `Pertemuan 1`, `Pertemuan 2`, `Pertemuan 3`, and `Praktikum 2`, `Praktikum 3`. The `Overload4.py` file is selected. The code in the editor is as follows:

```
1 #Penggunaan metode "sort" pada berbagai tipe data:
2 print("\nNama : Putri Robi'atul Adawiyah")
3 print("Nim : 210511018")
4 print("Kelas : TI21A(R1)\n\n")
5
6 a = [3, 1, 2]
7 a.sort()
8 print(a) # output: [1, 2, 3]
9
10 b = ["c", "a", "b"]
11 b.sort()
12 print(b) # output: ["a", "b", "c"]
```

The terminal on the right shows the command `python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding\Pertemuan 3\Overload4.py"` and its output:

```
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

[1, 2, 3]
['a', 'b', 'c']
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding>
```

2. Polymorphism dinamis (Overriding)

The screenshot shows the Visual Studio Code editor with a file named `Overriding2.py` open. The Explorer sidebar on the left shows a project structure with folders `Pertemuan 1`, `Pertemuan 2`, and `Pertemuan 3`. Under `Pertemuan 3`, there are files `Overload1.py`, `Overload2.py`, `Overload3.py`, `Overload4.py`, `Overriding1.py`, and `Overriding2.py`. The `Overriding2.py` file is selected and its content is displayed in the main editor. The code defines two classes: `Rectangle` and `Circle`, both inheriting from a base class `Shape`. The `Rectangle` class has attributes `width` and `height`, and the `Circle` class has an attribute `radius`. Both classes have an `area` method. The `main` function creates instances of both classes and prints their areas. The output is shown in the terminal on the right.

```
4 print("\nNama : Putri Robi'atul Adawiyah")
5 print("Nim : 210511018")
6 print("Kelas : TI21A(R1)\n\n")
7
8 class Rectangle(Shape):
9     def __init__(self, width, height):
10         self.width = width
11         self.height = height
12     def area(self):
13         return self.width * self.height
14
15 class Circle(Shape):
16     def __init__(self, radius):
17         self.radius = radius
18     def area(self):
19         return 3.14 * self.radius ** 2
20
21 shapes = [Rectangle(3, 4), Circle(5)]
22 for shape in shapes:
23     print(shape.area())
24
```

Terminal output:

```
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding> python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding\Pertemuan 3\Overriding2.py"
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

12
78.5
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding>
```

The screenshot shows the Visual Studio Code editor with a file named `Overriding1.py` open. The Explorer sidebar on the left shows the same project structure as the first screenshot. The `Overriding1.py` file is selected and its content is displayed in the main editor. The code defines a class `Matematika` with an `add` method. The `main` function creates an instance of the class and calls the `add` method with different arguments. The output is shown in the terminal on the right.

```
1 print("\nNama : Putri Robi'atul Adawiyah")
2 print("Nim : 210511018")
3 print("Kelas : TI21A(R1)\n\n")
4
5 class Matematika:
6     def add(self, a, b):
7         return a + b
8     def add(self, a, b, c=0):
9         return a + b + c
10
11 mat = Matematika()
12 B = mat.add(5, 3, 4)
13 print(B)
14
15 mut = Matematika()
16 C = mut.add(7,3)
17 print(C)
```

Terminal output:

```
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding> python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding\Pertemuan 3\Overriding1.py"
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

12
10
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROJ2\Coding>
```

The screenshot shows a VS Code editor with a file named `Overriding3.py` open. The code defines a base class `Animal` with a `make_sound` method, and two subclasses: `Dog` and `Cat`, both overriding the `make_sound` method. It also includes a `Chihuahua` class that inherits from `Dog` and a `Siamese` class that inherits from `Cat`. The code instantiates objects of each class and calls the `make_sound` method for each. The terminal output shows the results of these calls.

```
1 print("\nNama : Putri Robi'atul Adawiyah")
2 print("Nim : 210511018")
3 print("Kelas : TI21A(R1)\n\n")
4
5 class Animal:
6     def make_sound(self):
7         print("The animal makes a sound")
8
9 class Dog(Animal):
10     def make_sound(self):
11         print("The dog barks")
12
13 class Cat(Animal):
14     def make_sound(self):
15         print("The cat meows")
16
17 class Chihuahua(Dog):
18     def make_sound(self):
19         print("The chihuahua yaps")
20
21 class Siamese(Cat):
22     def make_sound(self):
23         print("The Siamese purrs")
24     def animal_sound(anim):
25         anim.make_sound()
26
27 # Instantiate objects of each class
28 animal = Animal()
29 dog = Dog()
30 cat = Cat()
31 chihuahua = Chihuahua()
32 siamese = Siamese()
33
34 # Call the animal sound function for each object
35 animal.make_sound() # Output: The animal makes a sound
36 dog.make_sound() # Output: The dog barks
```

Terminal Output:

```
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding> python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding\Pertemuan 3\Overriding3.py"
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

The animal makes a sound
The dog barks
The cat meows
The chihuahua yaps
The Siamese purrs
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding>
```

The screenshot shows a VS Code editor with a file named `Overriding4.py` open. The code defines a base class `Vehicle` with an abstract method `start`, and two subclasses: `Car` and `Motorcycle`, both overriding the `start` method. It also includes a `Bus` class that inherits from `Vehicle`. The code instantiates objects of each class and calls the `start` method for each. The terminal output shows the results of these calls.

```
1 print("\nNama : Putri Robi'atul Adawiyah")
2 print("Nim : 210511018")
3 print("Kelas : TI21A(R1)\n\n")
4
5 from abc import ABC, abstractmethod
6
7 class Vehicle(ABC):
8     @abstractmethod
9     def start(self):
10         pass
11
12 class Car(Vehicle):
13     def start(self):
14         print("Starting car...")
15
16 class Motorcycle(Vehicle):
17     def start(self):
18         print("Starting motorcycle...")
19
20 class Bus(Vehicle):
21     def start(self):
22         print("Starting bus...")
23
24 vehicles = [Car(), Motorcycle(), Bus()]
25
26 for vehicle in vehicles:
27     vehicle.start()
```

Terminal Output:

```
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding> python -u "c:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding\Pertemuan 3\Overriding4.py"
Nama : Putri Robi'atul Adawiyah
Nim : 210511018
Kelas : TI21A(R1)

Starting car...
Starting motorcycle...
Starting bus...
PS C:\Users\HP\Documents\KULIAH\SEMESTER 4\PROG2\Coding>
```

