

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Nama : Nisa Uzufatul Jannah

NIM : 210511001

Kelas : K1

PRAKTIKUM 3

Buatlah masing-masing 2 contoh polymorphism statis (overload) dan polymorphism dinamis (overriding).

1. Overload1.py

```
class Vehicle:
    def __init__(self, distance, time):
        self.distance = distance
        self.time = time

    def calculate_speed(self):
        pass

class Car(Vehicle):
    def calculate_speed(self):
        return self.distance / self.time

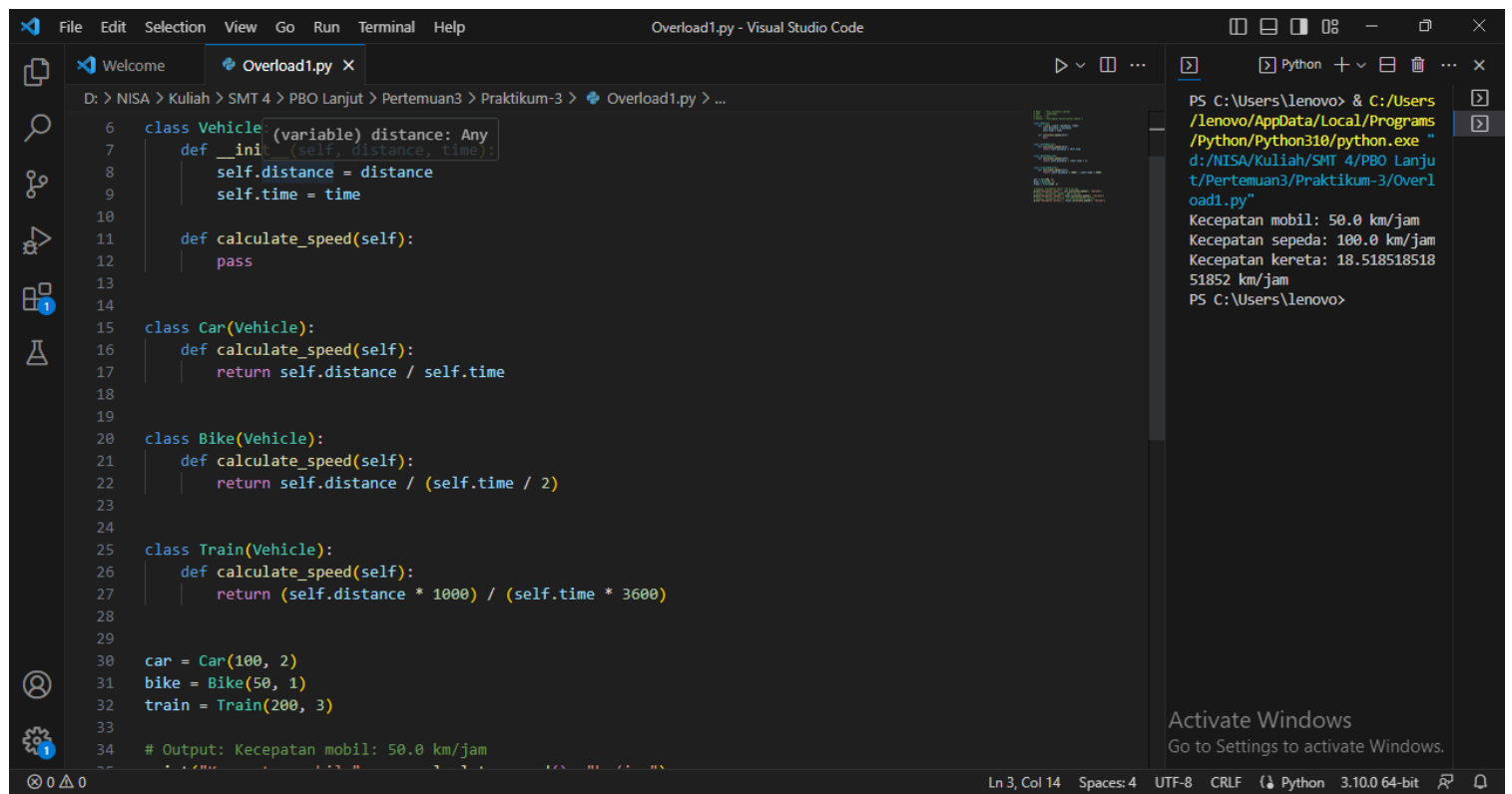
class Bike(Vehicle):
    def calculate_speed(self):
        return self.distance / (self.time / 2)

class Train(Vehicle):
    def calculate_speed(self):
        return (self.distance * 1000) / (self.time * 3600)

car = Car(100, 2)
bike = Bike(50, 1)
train = Train(200, 3)

# Output: Kecepatan mobil: 50.0 km/jam
print("Kecepatan mobil:", car.calculate_speed(), "km/jam")
# Output: Kecepatan sepeda: 100.0 km/jam
print("Kecepatan sepeda:", bike.calculate_speed(), "km/jam")
# Output: Kecepatan kereta: 18.51851851851852 km/jam
print("Kecepatan kereta:", train.calculate_speed(), "km/jam")
```

OUTPUT



The image shows a Visual Studio Code editor window with a Python file named `Overload1.py`. The code defines a base class `Vehicle` and three subclasses: `Car`, `Bike`, and `Train`. Each subclass overrides the `calculate_speed` method. The `Vehicle` class has an `__init__` method that takes `distance` and `time` as arguments. The `Car` class calculates speed as `distance / time`. The `Bike` class calculates speed as `distance / (time / 2)`. The `Train` class calculates speed as `(distance * 1000) / (time * 3600)`. The script creates instances of `Car`, `Bike`, and `Train` and prints their speeds.

```
6 class Vehicle:
7     def __init__(self, distance, time):
8         self.distance = distance
9         self.time = time
10
11     def calculate_speed(self):
12         pass
13
14
15 class Car(Vehicle):
16     def calculate_speed(self):
17         return self.distance / self.time
18
19
20 class Bike(Vehicle):
21     def calculate_speed(self):
22         return self.distance / (self.time / 2)
23
24
25 class Train(Vehicle):
26     def calculate_speed(self):
27         return (self.distance * 1000) / (self.time * 3600)
28
29
30 car = Car(100, 2)
31 bike = Bike(50, 1)
32 train = Train(200, 3)
33
34 # Output: Kecepatan mobil: 50.0 km/jam
```

The terminal output shows the execution of the script, displaying the calculated speeds for each vehicle:

```
PS C:\Users\lenovo> & C:/Users/lenovo/AppData/Local/Programs/Python/Python310/python.exe "d:/NISA/Kuliah/SMT 4/PBO Lanjut/Pertemuan3/Praktikum-3/Overload1.py"
Kecepatan mobil: 50.0 km/jam
Kecepatan sepeda: 100.0 km/jam
Kecepatan kereta: 18.51851851851852 km/jam
PS C:\Users\lenovo>
```

At the bottom of the terminal, there is a message: "Activate Windows. Go to Settings to activate Windows."

2. Overload2.py

```
class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def compute_salary(self):
        pass

class HourlyEmployee(Employee):
    def __init__(self, name, salary, hours):
        super().__init__(name, salary)
        self.hours = hours

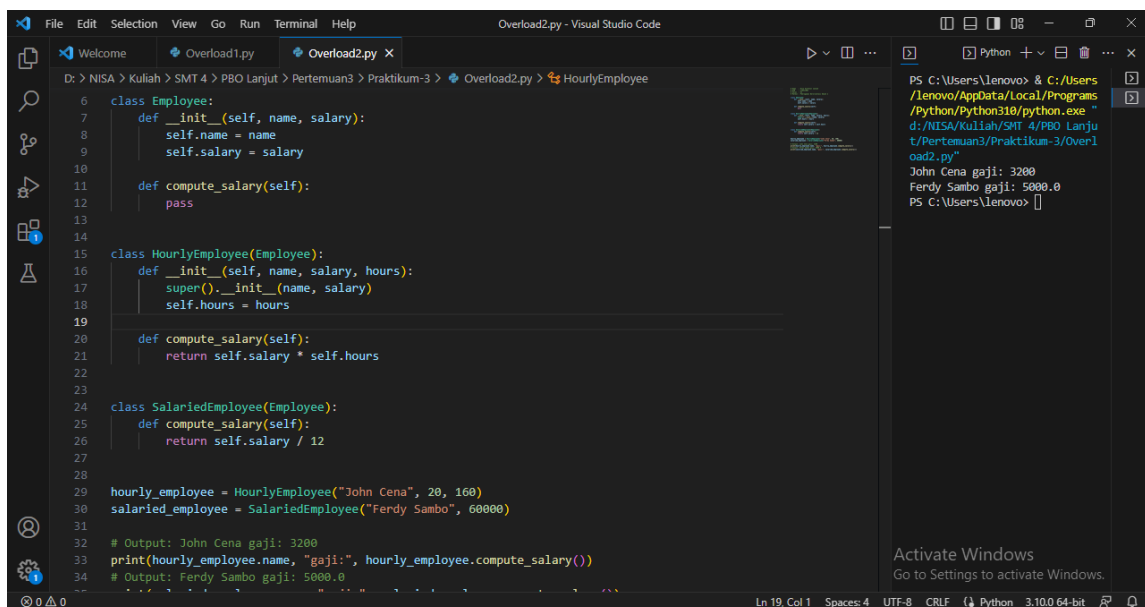
    def compute_salary(self):
        return self.salary * self.hours

class SalariedEmployee(Employee):
    def compute_salary(self):
        return self.salary / 12

hourly_employee = HourlyEmployee("John Cena", 20, 160)
salaried_employee = SalariedEmployee("Ferdly Sambo", 60000)

# Output: John Cena gaji: 3200
print(hourly_employee.name, "gaji:", hourly_employee.compute_salary())
# Output: Ferdly Sambo gaji: 5000.0
print(salaried_employee.name, "gaji:", salaried_employee.compute_salary())
```

OUTPUT



```
File Edit Selection View Go Run Terminal Help
Overload2.py - Visual Studio Code

Welcome Overload1.py Overload2.py x
D:\> NISA > Kuliah > SMT 4 > PBO Lanjut > Praktikum-3 > Overload2.py > HourlyEmployee

6 class Employee:
7     def __init__(self, name, salary):
8         self.name = name
9         self.salary = salary
10
11     def compute_salary(self):
12         pass
13
14
15 class HourlyEmployee(Employee):
16     def __init__(self, name, salary, hours):
17         super().__init__(name, salary)
18         self.hours = hours
19
20     def compute_salary(self):
21         return self.salary * self.hours
22
23
24 class SalariedEmployee(Employee):
25     def compute_salary(self):
26         return self.salary / 12
27
28
29 hourly_employee = HourlyEmployee("John Cena", 20, 160)
30 salaried_employee = SalariedEmployee("Ferdly Sambo", 60000)
31
32 # Output: John Cena gaji: 3200
33 print(hourly_employee.name, "gaji:", hourly_employee.compute_salary())
34 # Output: Ferdly Sambo gaji: 5000.0

PS C:\Users\lenovo> & C:\Users\lenovo\AppData\Local\Programs\Python\Python310\python.exe "d:/NISA/Kuliah/SMT 4/PBO Lanjut/Pertemuan3/Praktikum-3/Overload2.py"
John Cena gaji: 3200
Ferdly Sambo gaji: 5000.0
PS C:\Users\lenovo>

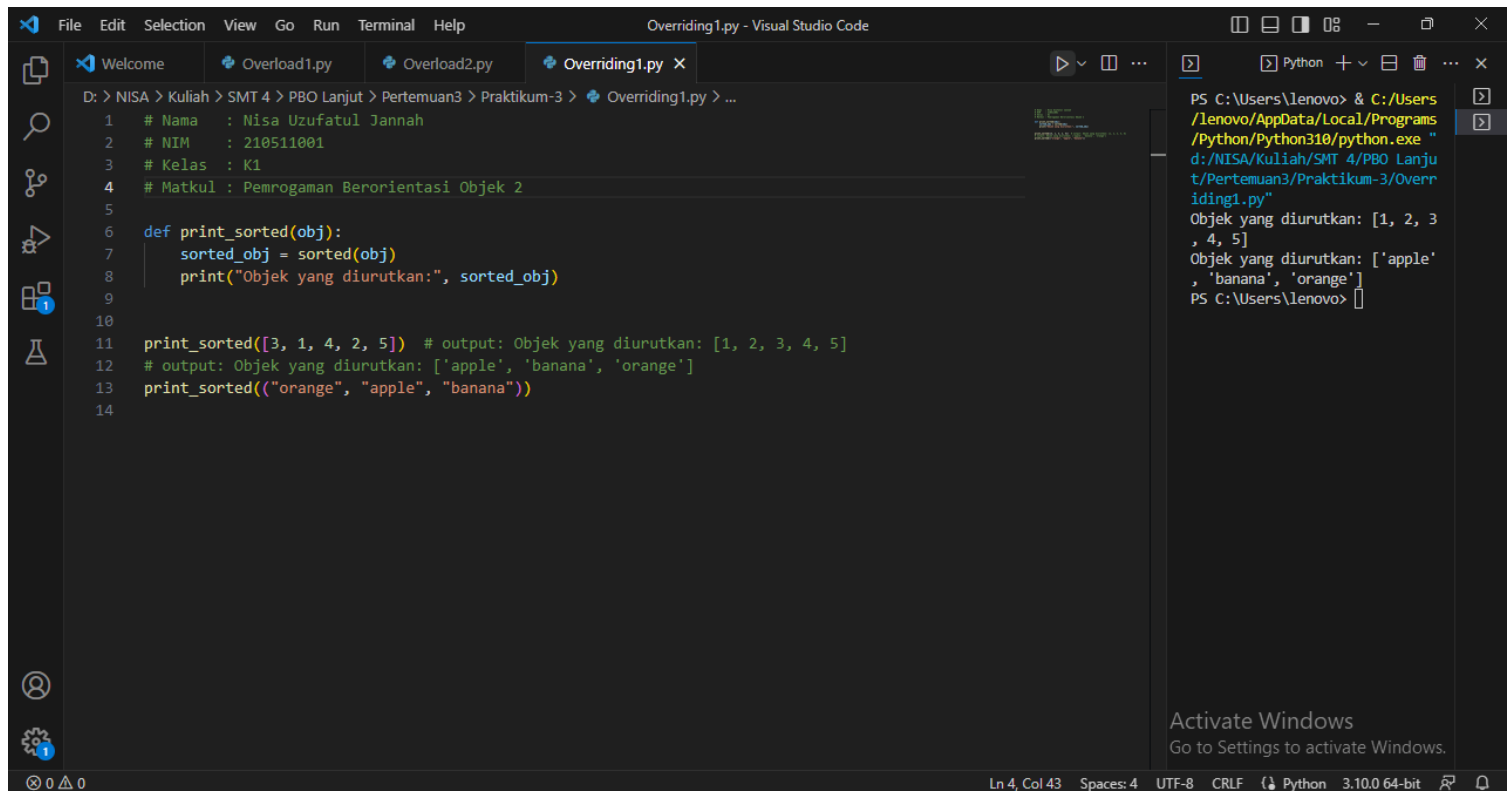
Activate Windows
Go to Settings to activate Windows.
```

3. Overriding1.py

```
def print_sorted(obj):  
    sorted_obj = sorted(obj)  
    print("Objek yang diurutkan:", sorted_obj)
```

```
print_sorted([3, 1, 4, 2, 5]) # output: Objek yang diurutkan: [1, 2, 3, 4, 5]  
# output: Objek yang diurutkan: ['apple', 'banana', 'orange']  
print_sorted(("orange", "apple", "banana"))
```

OUTPUT



```
File Edit Selection View Go Run Terminal Help  
Overriding1.py - Visual Studio Code  
Welcome Overload1.py Overload2.py Overriding1.py X  
D: > NISA > Kuliah > SMT 4 > PBO Lanjut > Pertemuan3 > Praktikum-3 > Overriding1.py > ...  
1 # Nama : Nisa Uzufatul Jannah  
2 # NIM : 210511001  
3 # Kelas : K1  
4 # Matkul : Pemrograman Berorientasi Objek 2  
5  
6 def print_sorted(obj):  
7     sorted_obj = sorted(obj)  
8     print("Objek yang diurutkan:", sorted_obj)  
9  
10  
11 print_sorted([3, 1, 4, 2, 5]) # output: Objek yang diurutkan: [1, 2, 3, 4, 5]  
12 # output: Objek yang diurutkan: ['apple', 'banana', 'orange']  
13 print_sorted(("orange", "apple", "banana"))  
14  
PS C:\Users\lenovo> & C:/Users/lenovo/AppData/Local/Programs/Python/Python310/python.exe "d:/NISA/Kuliah/SMT 4/PBO Lanjut/Pertemuan3/Praktikum-3/Overriding1.py"  
Objek yang diurutkan: [1, 2, 3, 4, 5]  
Objek yang diurutkan: ['apple', 'banana', 'orange']  
PS C:\Users\lenovo>  
Activate Windows  
Go to Settings to activate Windows.  
Ln 4, Col 43 Spaces: 4 UTF-8 CRLF Python 3.10.0 64-bit
```

4. Overriding2.py

```
class Runnable:
    def run(self):
        pass

class Car(Runnable):
    def run(self):
        print("Mobil berjalan.")

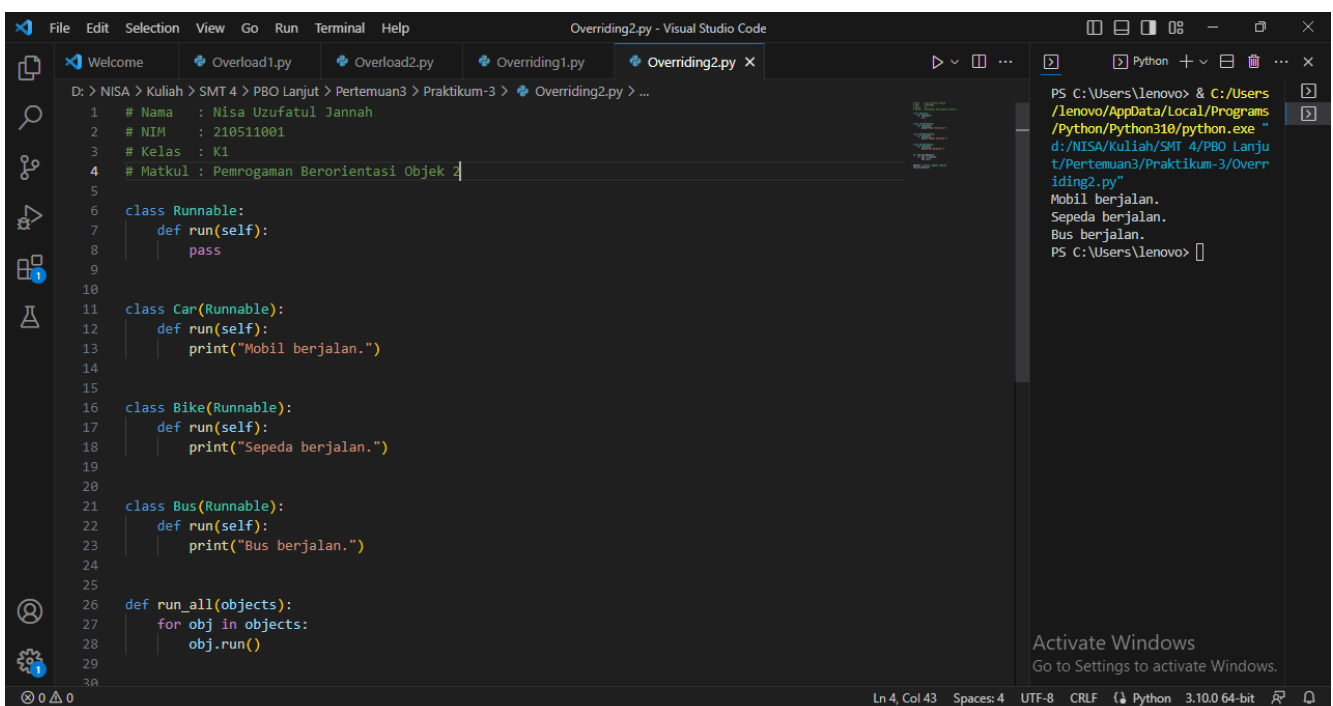
class Bike(Runnable):
    def run(self):
        print("Sepeda berjalan.")

class Bus(Runnable):
    def run(self):
        print("Bus berjalan.")

def run_all(objects):
    for obj in objects:
        obj.run()

objects = [Car(), Bike(), Bus()]
run_all(objects)
```

OUTPUT



The screenshot shows the Visual Studio Code interface with a file named `Overriding2.py` open. The code in the editor defines a `Runnable` base class with a `run` method, and three subclasses: `Car`, `Bike`, and `Bus`, each overriding the `run` method to print a specific message. A `run_all` function iterates over a list of these objects and calls their `run` methods. The terminal on the right shows the execution output: `Mobil berjalan.`, `Sepeda berjalan.`, and `Bus berjalan.`. The status bar at the bottom indicates the current position is at line 4, column 43.

```
File Edit Selection View Go Run Terminal Help
Overriding2.py - Visual Studio Code

D:\> NISA > Kuliah > SMT 4 > PBO Lanjut > Pertemuan3 > Praktikum-3 > Overriding2.py > ...
1 # Nama : Nisa Uzufatul Jannah
2 # NIM : 210511001
3 # Kelas : K1
4 # Matkul : Pemrograman Berorientasi Objek 4
5
6 class Runnable:
7     def run(self):
8         pass
9
10
11 class Car(Runnable):
12     def run(self):
13         print("Mobil berjalan.")
14
15
16 class Bike(Runnable):
17     def run(self):
18         print("Sepeda berjalan.")
19
20
21 class Bus(Runnable):
22     def run(self):
23         print("Bus berjalan.")
24
25
26 def run_all(objects):
27     for obj in objects:
28         obj.run()
29
30
```

PS C:\Users\lenovo> & C:/Users/lenovo/AppData/Local/Programs/Python/Python310/python.exe "d:/NISA/Kuliah/SMT 4/PBO Lanjut/Pertemuan3/Praktikum-3/Overriding2.py"

Mobil berjalan.
Sepeda berjalan.
Bus berjalan.
PS C:\Users\lenovo>

Activate Windows
Go to Settings to activate Windows.

Ln 4, Col 43 Spaces: 4 UTF-8 CRLF Python 3.10.0 64-bit