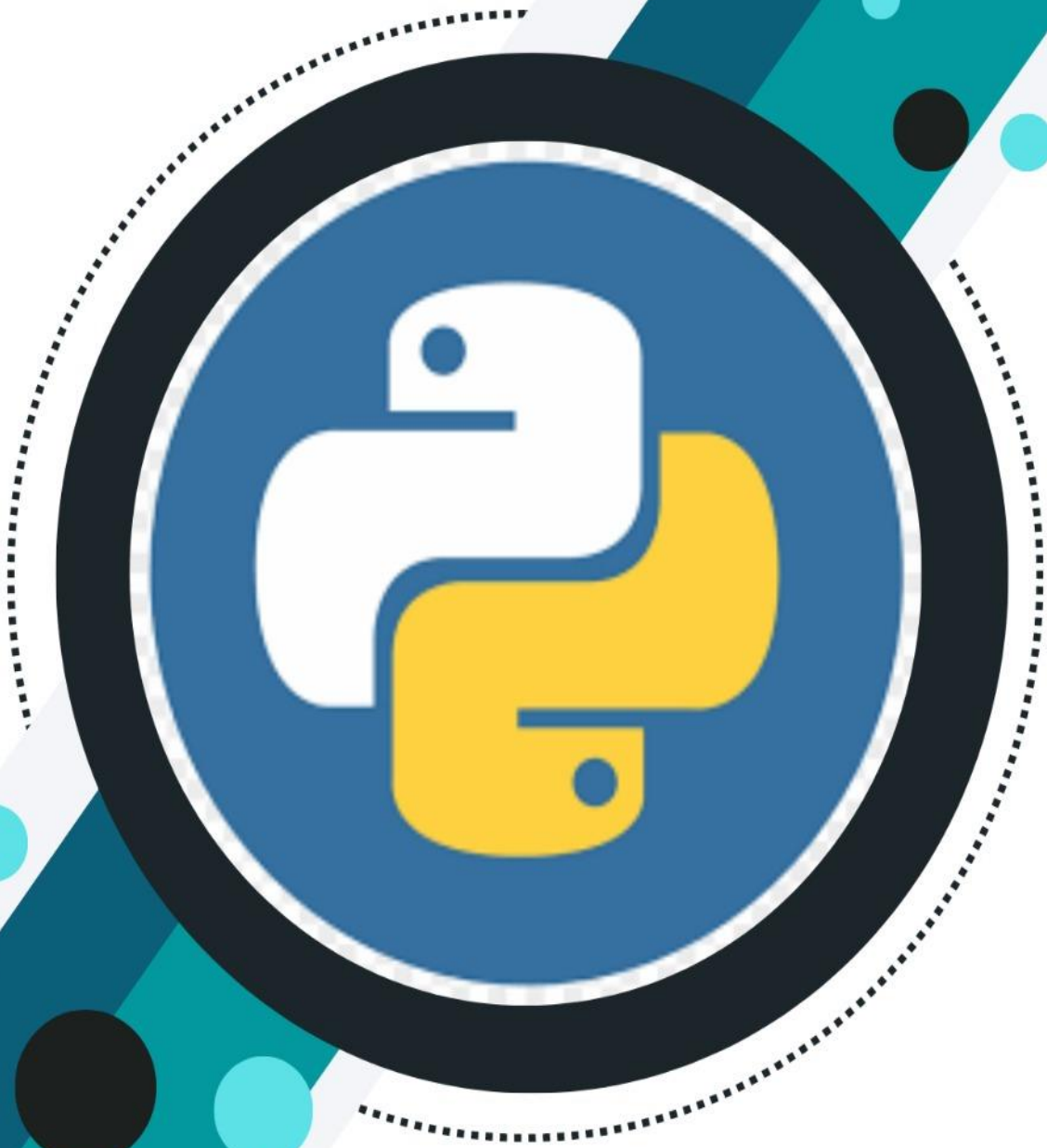


LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Nama : Zidan Khoirul Rizki

NIM : 210511049

Kelas : R2

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

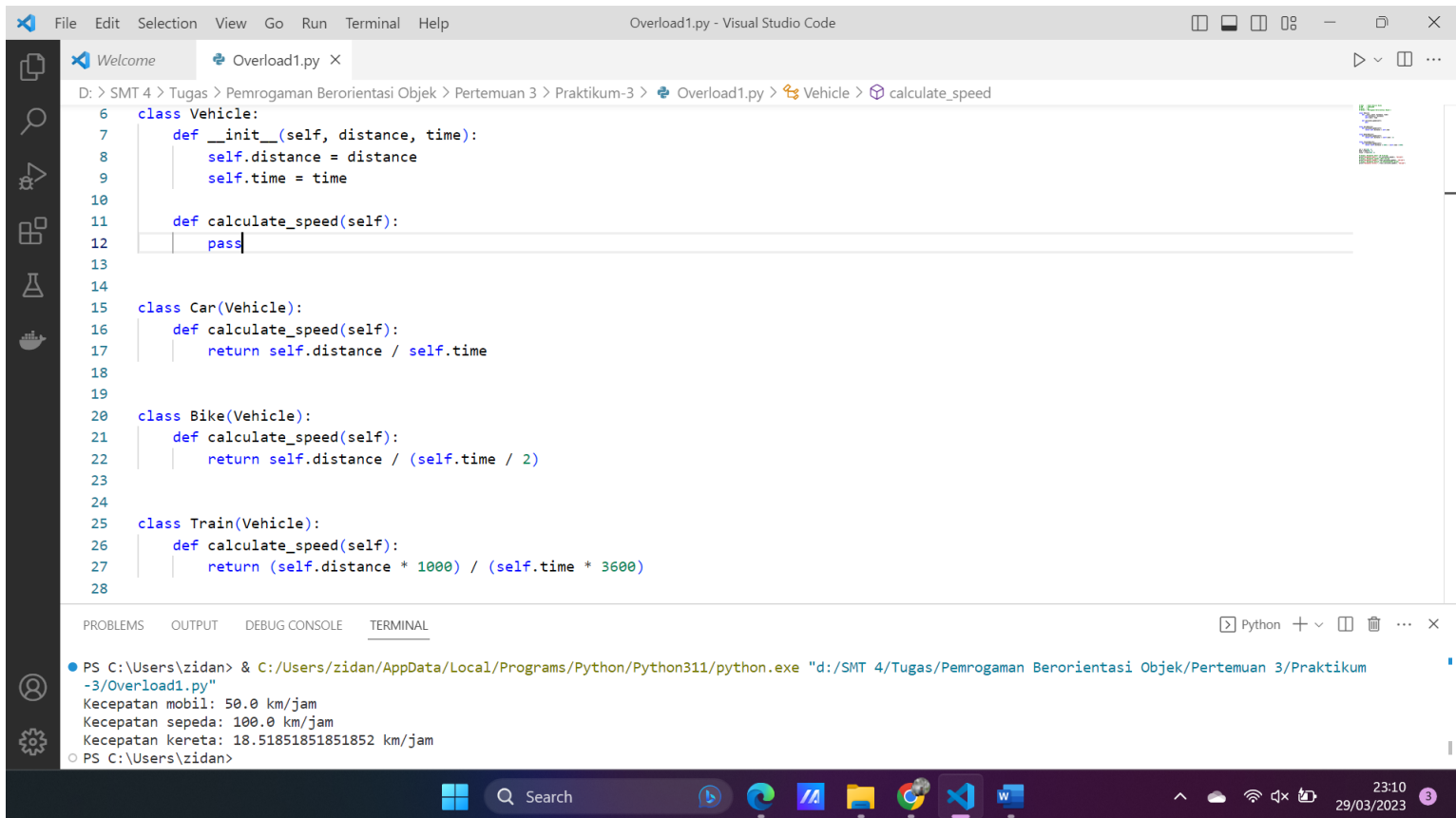
class Bike(Vehicle):
    def calculate_speed(self):
        return self.distance / (self.time / 2)

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 screenshot displays the Visual Studio Code interface. The main editor window shows a Python file named 'Overload1.py' with the following code:

```
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
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
PS C:\Users\zidan> & C:/Users/zidan/AppData/Local/Programs/Python/Python311/python.exe "d:/SMT 4/Tugas/Pemrograman Berorientasi Objek/Pertemuan 3/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\zidan>
```

The Windows taskbar at the bottom shows the time as 23:10 on 29/03/2023.

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

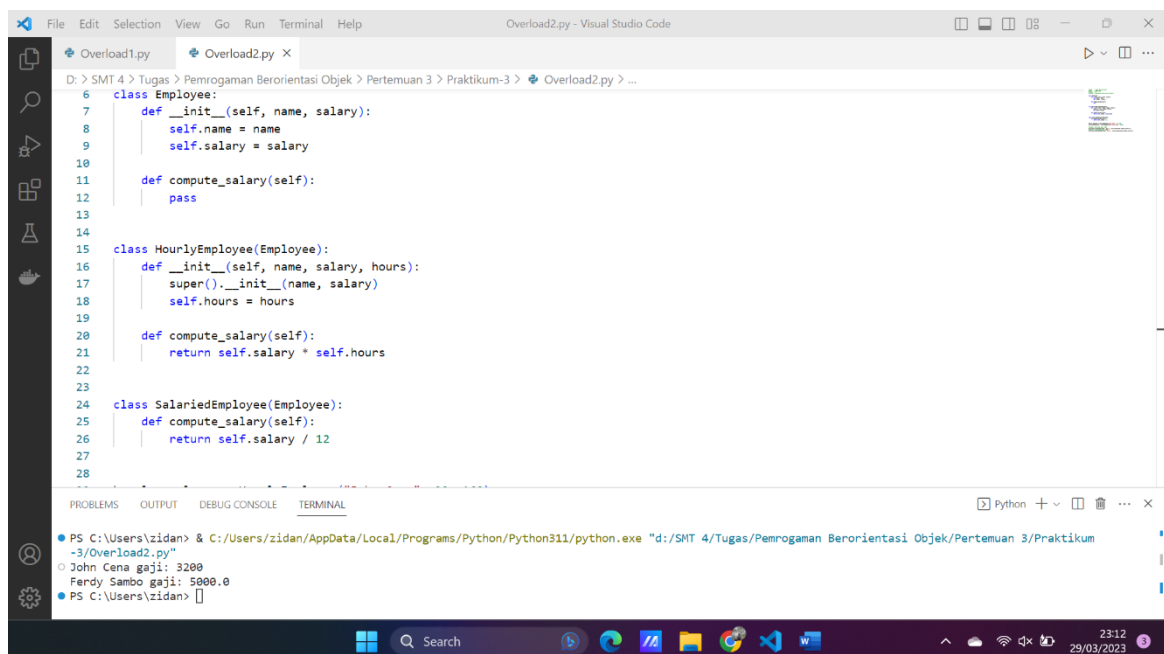
    def compute_salary(self):
        return self.salary * self.hours

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

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

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

OUTPUT



The screenshot shows the Visual Studio Code interface with the file 'Overload2.py' open. The code is identical to the one in the previous block. The terminal at the bottom displays the output of the program:

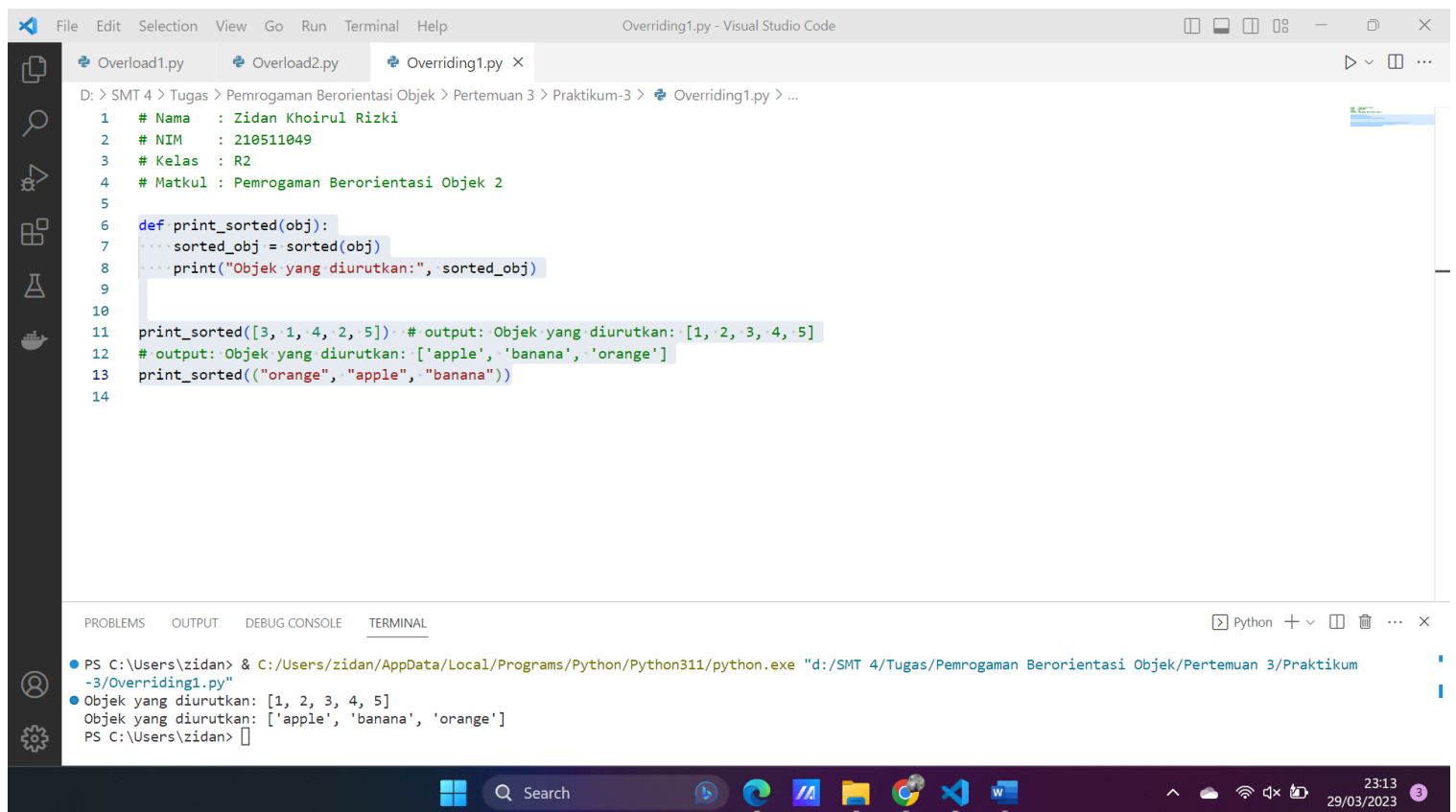
```
PS C:\Users\zidan> & C:/Users/zidan/AppData/Local/Programs/Python/Python311/python.exe "d:/SMT 4/Tugas/Pemrograman Berorientasi Objek/Pertemuan 3/Praktikum -3/Overload2.py"
John Cena gaji: 3200
Ferdy Sambo gaji: 5000.0
PS C:\Users\zidan>
```

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



The screenshot displays the Visual Studio Code interface. The editor window shows the file `Overriding1.py` with the following code:

```
1 # Nama : Zidan Khoirul Rizki  
2 # NIM : 210511049  
3 # Kelas : R2  
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
```

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

```
PS C:\Users\zidan> & C:/Users/zidan/AppData/Local/Programs/Python/Python311/python.exe "d:/SMT 4/Tugas/Pemrograman Berorientasi Objek/Pertemuan 3/Praktikum  
-3/Overriding1.py"  
Objek yang diurutkan: [1, 2, 3, 4, 5]  
Objek yang diurutkan: ['apple', 'banana', 'orange']  
PS C:\Users\zidan>
```

The Windows taskbar at the bottom indicates the date and time as 23:13 on 29/03/2023.

4. Overriding2.py

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

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

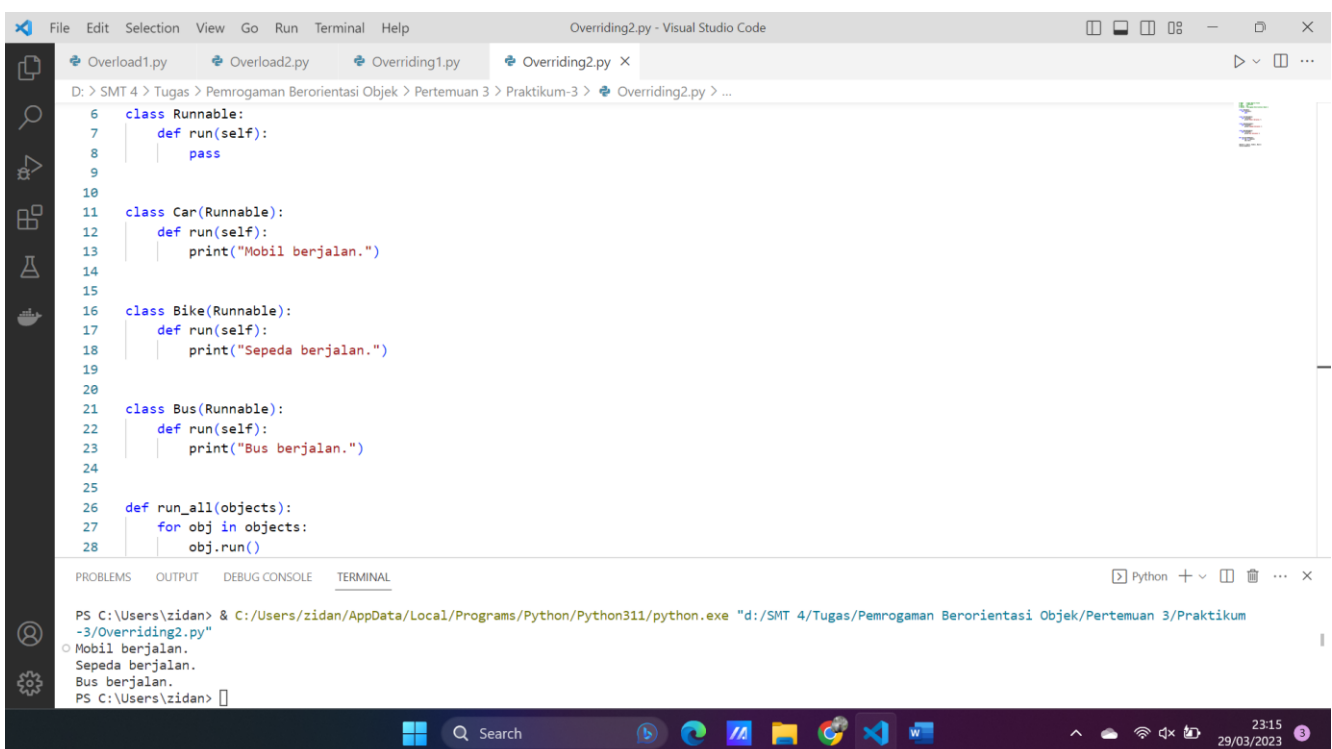
class Bike(Runnable):
    def run(self):
        print("Sepeda berjalan.")

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 editor with a file named 'Overriding2.py' open. The code defines a base class 'Runnable' 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 at the bottom shows the command to run the script and the resulting output: 'Mobil berjalan.', 'Sepeda berjalan.', and 'Bus berjalan.'.

```
File Edit Selection View Go Run Terminal Help Overriding2.py - Visual Studio Code
D:\> SMT 4 > Tugas > Pemrograman Berorientasi Objek > Pertemuan 3 > Praktikum-3 > Overriding2.py > ...

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()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + - 
PS C:\Users\zidan> & C:\Users\zidan\AppData\Local\Programs\Python\Python311\python.exe "d:/SMT 4/Tugas/Pemrograman Berorientasi Objek/Pertemuan 3/Praktikum
-3/Overriding2.py"
Mobil berjalan.
Sepeda berjalan.
Bus berjalan.
PS C:\Users\zidan>
```