

Nama : Thoyib Abdillah Ardja
Nim : 210511026
Kelas : R1

PBO2 Latihan 4

Contoh 1

Script:

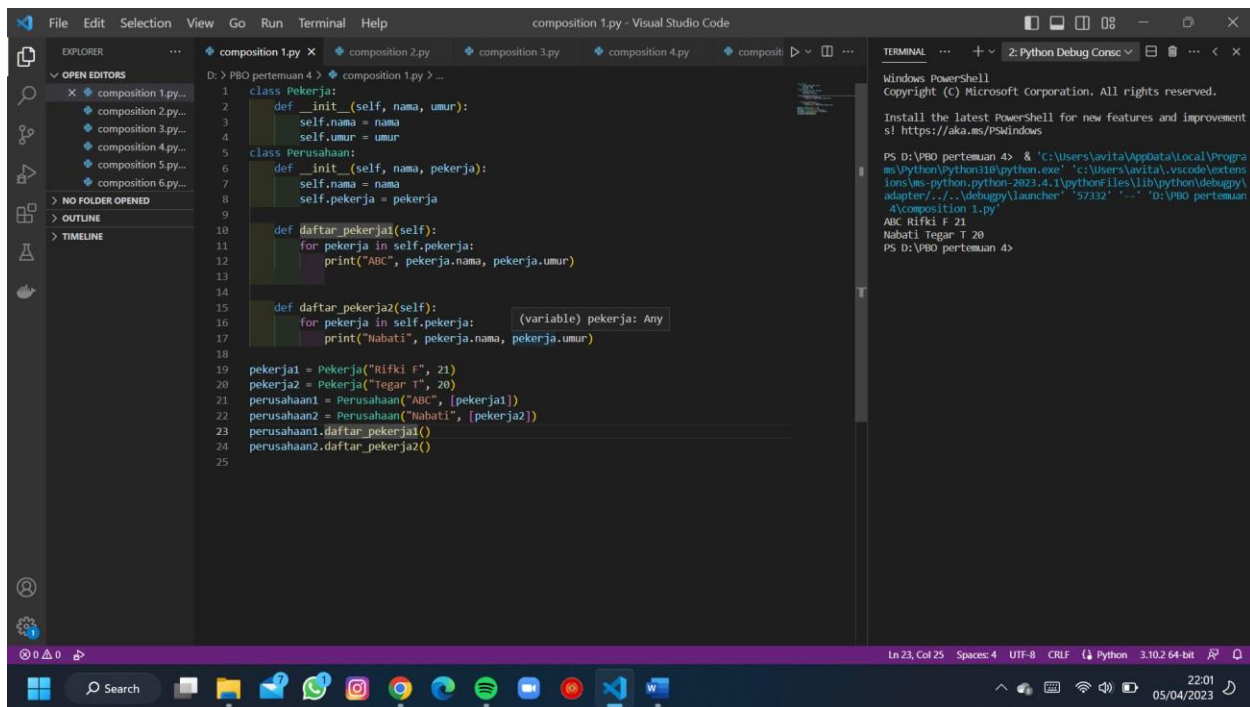
```
class Pekerja:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur
class Perusahaan:
    def __init__(self, nama, pekerja):
        self.nama = nama
        self.pekerja = pekerja

    def daftar_pekerja1(self):
        for pekerja in self.pekerja:
            print("ABC", pekerja.nama, pekerja.umur)

    def daftar_pekerja2(self):
        for pekerja in self.pekerja:
            print("Nabati", pekerja.nama, pekerja.umur)

pekerja1 = Pekerja("Rifki F", 21)
pekerja2 = Pekerja("Tegar T", 20)
perusahaan1 = Perusahaan("ABC", [pekerja1])
perusahaan2 = Perusahaan("Nabati", [pekerja2])
perusahaan1.daftar_pekerja1()
perusahaan2.daftar_pekerja2()
```

Hasil Running Program



Contoh 2

Script:

```
class Player:
    def __init__(self, name):
        self.name = name
        self.inventory = Inventory()
        print("Hero Layla")

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

class Inventory:
    def __init__(self):
        self.items = []

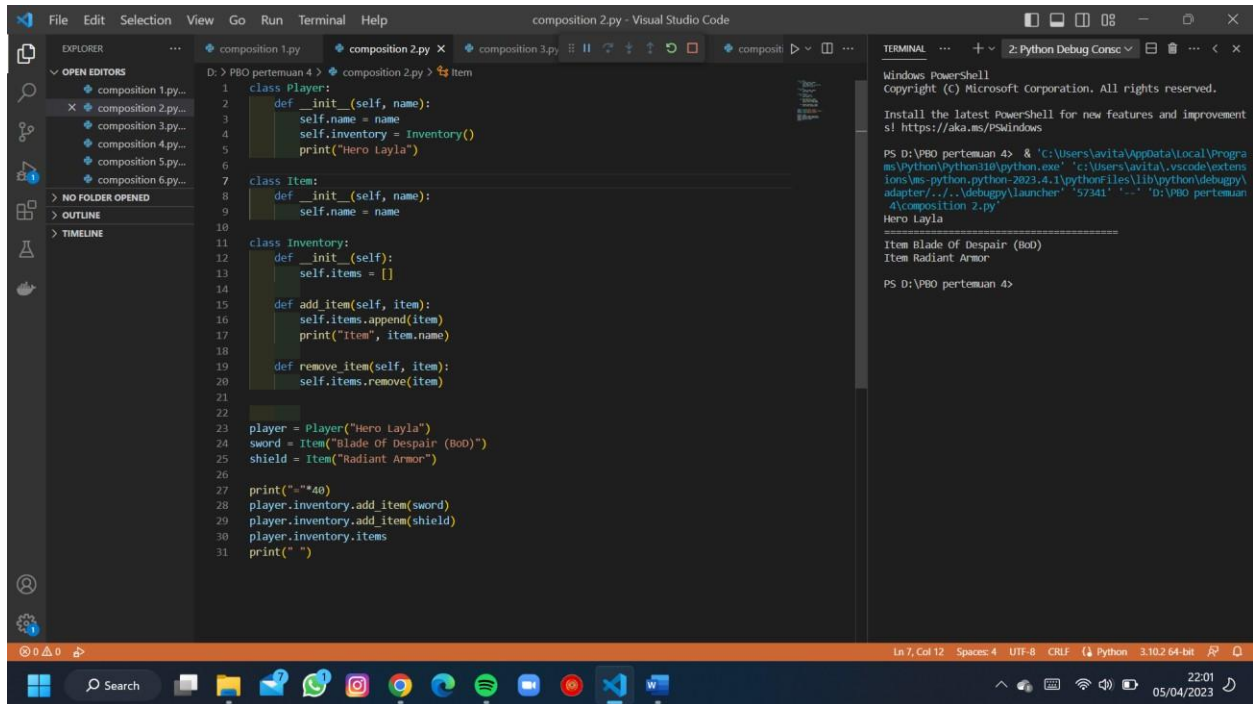
    def add_item(self, item):
        self.items.append(item)
        print("Item", item.name)

    def remove_item(self, item):
        self.items.remove(item)

player = Player("Hero Layla")
sword = Item("Blade Of Despair (BoD)")
shield = Item("Radiant Armor")

print("="*40)
player.inventory.add_item(sword)
player.inventory.add_item(shield)
player.inventory.items
print(" ")
```

Hasil Running Program



The screenshot displays the Visual Studio Code interface with a Python file named `composition 2.py` open. The code defines a `Player` class with an `inventory` attribute, an `Item` class, and an `Inventory` class. The `Player` class has an `__init__` method that takes a name and an `Inventory` object. The `Item` class has an `__init__` method that takes a name. The `Inventory` class has an `__init__` method, an `add_item` method, and a `remove_item` method. The program creates a `Player` object named `Hero Layla`, an `Item` object named `Blade Of Despair (BoD)`, and an `Item` object named `Radiant Armor`. It then adds the `Blade Of Despair (BoD)` and `Radiant Armor` items to the `Hero Layla` player's inventory and prints the inventory items.

```
1 class Player:
2     def __init__(self, name):
3         self.name = name
4         self.inventory = Inventory()
5         print("Hero Layla")
6
7 class Item:
8     def __init__(self, name):
9         self.name = name
10
11 class Inventory:
12     def __init__(self):
13         self.items = []
14
15     def add_item(self, item):
16         self.items.append(item)
17         print("Item", item.name)
18
19     def remove_item(self, item):
20         self.items.remove(item)
21
22
23 player = Player("Hero Layla")
24 sword = Item("Blade Of Despair (BoD)")
25 shield = Item("Radiant Armor")
26
27 print("~*~*~")
28 player.inventory.add_item(sword)
29 player.inventory.add_item(shield)
30 player.inventory.items
31 print(" ")
```

The terminal output shows the execution of the program, displaying the player's name and the items in the inventory:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows

PS D:\PBO pertemuan 4> & "C:\Users\avita\AppData\Local\Programs\Python\Python310\python.exe" "c:\Users\avita\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\python\debugpy\adapter\c:/z/.debugpy/launcher" "57341" "-" "D:\PBO pertemuan 4\composition 2.py"
Hero Layla
=====
Item Blade Of Despair (BoD)
Item Radiant Armor

PS D:\PBO pertemuan 4>
```

Contoh 3

Script:

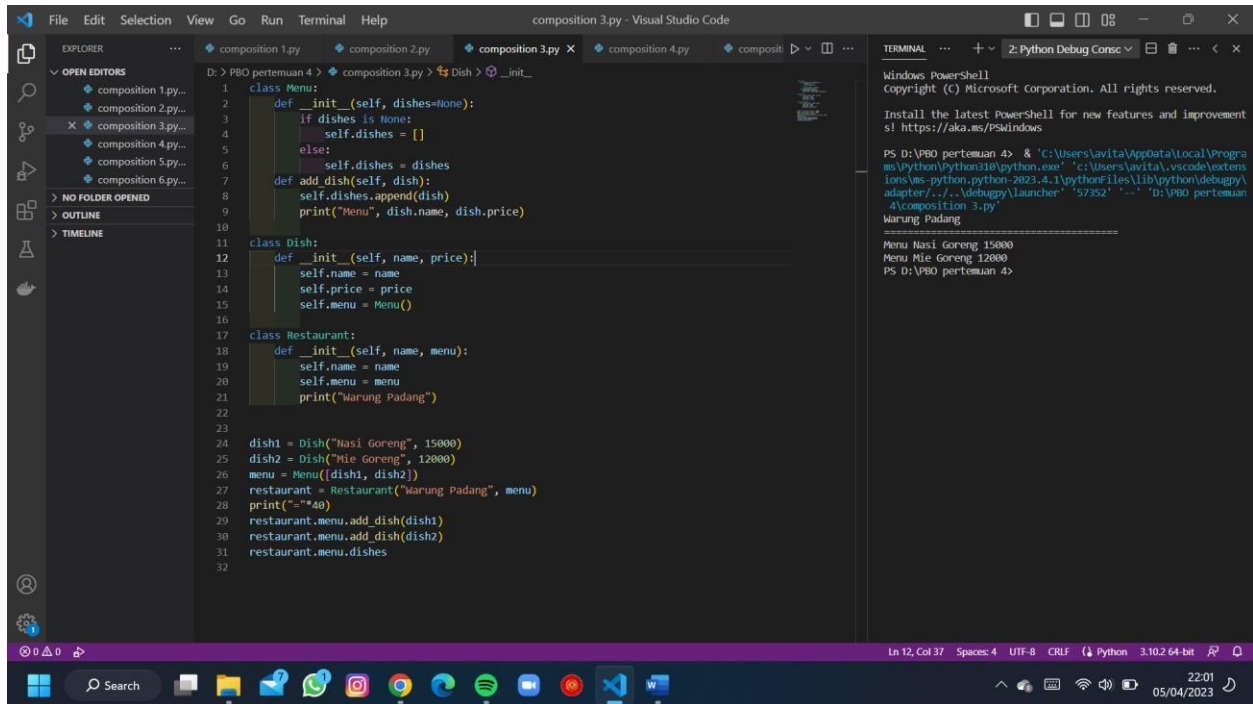
```
class Menu:
    def __init__(self, dishes=None):
        if dishes is None:
            self.dishes = []
        else:
            self.dishes = dishes
    def add_dish(self, dish):
        self.dishes.append(dish)
        print("Menu", dish.name, dish.price)

class Dish:
    def __init__(self, name, price):
        self.name = name
        self.price = price
        self.menu = Menu()

class Restaurant:
    def __init__(self, name, menu):
        self.name = name
        self.menu = menu
        print("Warung Padang")

dish1 = Dish("Nasi Goreng", 15000)
dish2 = Dish("Mie Goreng", 12000)
menu = Menu([dish1, dish2])
restaurant = Restaurant("Warung Padang", menu)
print("="*40)
restaurant.menu.add_dish(dish1)
restaurant.menu.add_dish(dish2)
restaurant.menu.dishes
```

Hasil Running Program



The screenshot displays the Visual Studio Code interface with a Python file named `composition 3.py` open. The code defines a `Menu` class, a `Dish` class, and a `Restaurant` class. The `Menu` class has an `__init__` method that initializes a list of dishes and an `add_dish` method. The `Dish` class has an `__init__` method that takes a name and price. The `Restaurant` class has an `__init__` method that takes a name and a menu. The code creates two dishes, a menu, and a restaurant, and prints the menu.

```
1 class Menu:
2     def __init__(self, dishes=None):
3         if dishes is None:
4             self.dishes = []
5         else:
6             self.dishes = dishes
7     def add_dish(self, dish):
8         self.dishes.append(dish)
9         print("Menu", dish.name, dish.price)
10
11 class Dish:
12     def __init__(self, name, price):
13         self.name = name
14         self.price = price
15         self.menu = Menu()
16
17 class Restaurant:
18     def __init__(self, name, menu):
19         self.name = name
20         self.menu = menu
21         print("Warung Padang")
22
23
24 dish1 = Dish("Nasi Goreng", 15000)
25 dish2 = Dish("Mie Goreng", 12000)
26 menu = Menu([dish1, dish2])
27 restaurant = Restaurant("Warung Padang", menu)
28 print("-"*40)
29 restaurant.menu.add_dish(dish1)
30 restaurant.menu.add_dish(dish2)
31 restaurant.menu.dishes
32
```

The terminal output shows the execution of the program, displaying the menu and the restaurant name.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows

PS D:\PBO pertemuan 4> & 'C:\Users\avita\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\avita\.vscode\extensions\ms-python.python-2023.4.1\pythonFiles\lib\python\debugpy\adapter/c:/c:\Users\avita\AppData\Local\Programs\Python\Python310\python.exe' 'D:\PBO pertemuan 4\composition 3.py'
Warung Padang
=====
Menu Nasi Goreng 15000
Menu Mie Goreng 12000
PS D:\PBO pertemuan 4>
```

Contoh 4

Script:

```
class Song:
    def __init__(self, title, artist):
        self.title = title
        self.artist = artist

class Playlist:
    def __init__(self):
        self.songs = []
    def add_song(self, song):
        self.songs.append(song)
        print("Title", song.title)

class MediaPlayer:
    def __init__(self, playlist):
        self.playlist = playlist
        print("Play Music")

song1 = Song("Lose Yourself", "Eminem")
song2 = Song("Someone Like You", "Adele")
playlist = Playlist()
media_player = MediaPlayer(playlist)
print("="*40)
playlist.add_song(song1)
playlist.add_song(song2)
media_player.playlist.songs
```

Hasil Running Program

The image shows a Windows desktop with a Visual Studio Code editor window open. The editor is displaying a Python file named 'composition_4.py'. The code defines two classes: 'Song' and 'Playlist'. The 'Song' class has attributes 'title' and 'artist'. The 'Playlist' class has a 'songs' list and methods to add songs and print the playlist. The script creates two song objects, 'song1' and 'song2', and adds them to a 'playlist' object. Finally, it prints the title of the first song in the playlist, which is 'Lose Yourself'. The terminal window on the right shows the output of the script, which is 'Title Lose Yourself' followed by 'Title Someone Like You'. The Windows taskbar at the bottom shows various application icons and the system clock indicating 22:01 on 05/04/2023.

Contoh 5

Script:

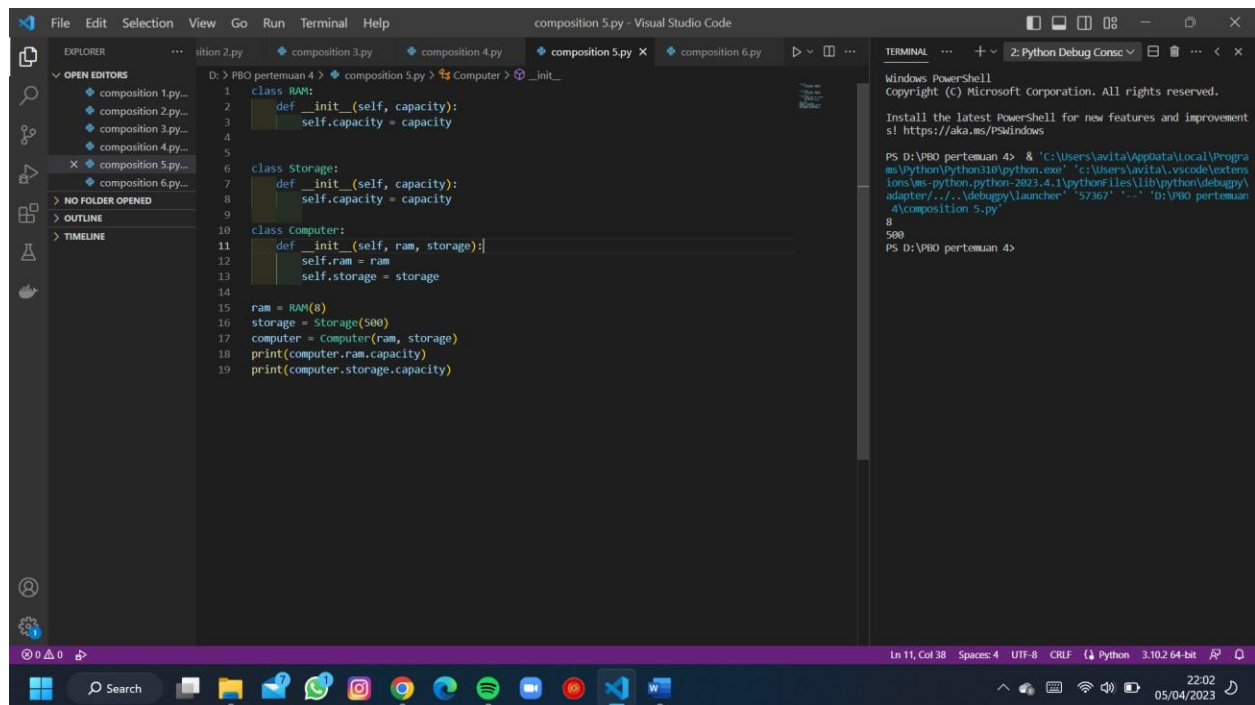
```
class RAM:
    def __init__(self, capacity):
        self.capacity = capacity

class Storage:
    def __init__(self, capacity):
        self.capacity = capacity

class Computer:
    def __init__(self, ram, storage):
        self.ram = ram
        self.storage = storage

ram = RAM(8)
storage = Storage(500)
computer = Computer(ram, storage)
print(computer.ram.capacity)
print(computer.storage.capacity)
```

Hasil Running Program



Contoh 6

Script:

```
class Wheel:
    def __init__(self, size):
        self.size = size
class Engine:
    def __init__(self, power):
        self.power = power
class Car:
    def __init__(self, wheels, engine):
        self.wheels = wheels
        self.engine = engine

wheel1 = Wheel(17)
wheel2 = Wheel(17)
wheel3 = Wheel(17)
wheel4 = Wheel(17)
engine = Engine(150)
car = Car([wheel1, wheel2, wheel3, wheel4], engine)
print(car.wheels[0].size)
```

Hasil Running Program

