

# **LAPORAN**

## **IMPLEMENTASI PROGRAM OOP DAN CLASS DIAGRAM**

*Disusun Untuk Memenuhi Tugas*

*Mata Kuliah: Pemrograman Berorientasi Objek Praktik*

Dosen Pembimbing:  
Sri Wulandari, S.Kom., M.Cs.



Disusun oleh:  
Muhammad Zaki As Shidiqi  
5240411230

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2025**

## A. Tugas 1

### 1. Kode Program

```
class Menu:
    def __init__(self, nama, harga):
        self.nama = nama
        self.harga = harga
        self.tersedia = True

    def pesan(self):
        if self.tersedia:
            print(f'Menu "{self.nama}" telah dipesan.')
        else:
            print(f'Maaf, menu "{self.nama}" tidak tersedia.')

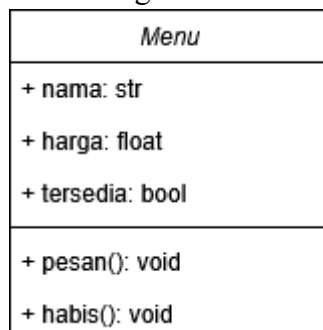
    def habis(self):
        if self.tersedia:
            self.tersedia = False
            print(f'Menu "{self.nama}" sekarang habis.')
        else:
            print(f'Menu "{self.nama}" sudah tidak tersedia.')

menu = Menu("Nasi Goreng", 20000)
menu.pesan()
menu.habis()
menu.pesan()
```

### 2. Output

```
Menu "Nasi Goreng" telah dipesan.
Menu "Nasi Goreng" sekarang habis.
Maaf, menu "Nasi Goreng" tidak tersedia.
```

### 3. Class Diagram



### 4. Penjelasan

Pada kode tersebut terdapat *class* bernama “Menu”, *class* ini memiliki 3 atribut yaitu nama, harga, dan tersedia. *Class* ini juga memiliki 2 *method* yaitu pesan yang

berfungsi untuk menampilkan pesan “Menu "<nama>" telah dipesan.” di terminal jika status tersedia adalah True. Selain itu, ada *method* habis yang berfungsi untuk mengubah status tersedia menjadi False dan menampilkan pesan “Menu "<nama>" sekarang habis.” di terminal.

Kemudian dilakukan pembuatan objek “menu” dari *class* Menu yang memiliki nama Nasi Goreng dan harga 20000. Setelah dilakukan pembuatan objek menu, kemudian dilakukan pemanggilan *method* pesan, *method* habis, dan kembali memanggil *method* pesan dari objek tersebut untuk melihat perubahan status ketersediaan.

## B. Tugas 2

### 1. Kode Program

```
class Komputer:
    def __init__(self, nama, pabrik, harga, jenis):
        self.nama = nama
        self.pabrik = pabrik
        self.harga = harga
        self.jenis = jenis

    def info(self):
        print(f"Komputer {self.nama} adalah jenis {self.jenis} yang diproduksi oleh {self.pabrik} dengan harga {self.harga}.")

class Processor(Komputer):
    def __init__(self, nama, pabrik, harga, jenis, core):
        super().__init__(nama, pabrik, harga, jenis)
        self.core = core

    def info(self):
        super().info()
        print(f"Jumlah Core: {self.core} Core.")

class RAM(Komputer):
    def __init__(self, nama, pabrik, harga, jenis, kapasitas):
        super().__init__(nama, pabrik, harga, jenis)
        self.kapasitas = kapasitas

    def info(self):
        super().info()
        print(f"Kapasitas RAM: {self.kapasitas} GB.")

class Hardisk(Komputer):
    def __init__(self, nama, pabrik, harga, jenis, kapasitas):
        super().__init__(nama, pabrik, harga, jenis)
        self.kapasitas = kapasitas
```

```

def info(self):
    super().info()
    print(f"Kapasitas Hard Disk: {self.kapasitas} TB.")

komputer = Komputer("Laptop XYZ", "Dell", 15000000, "Laptop")
komputer.info()

prosesor = Processor("Intel i7", "Intel", 5000000, "Prosesor",
4)
prosesor.info()

ram = RAM("Corsair Vengeance", "Corsair", 2000000, "RAM", 16)
ram.info()

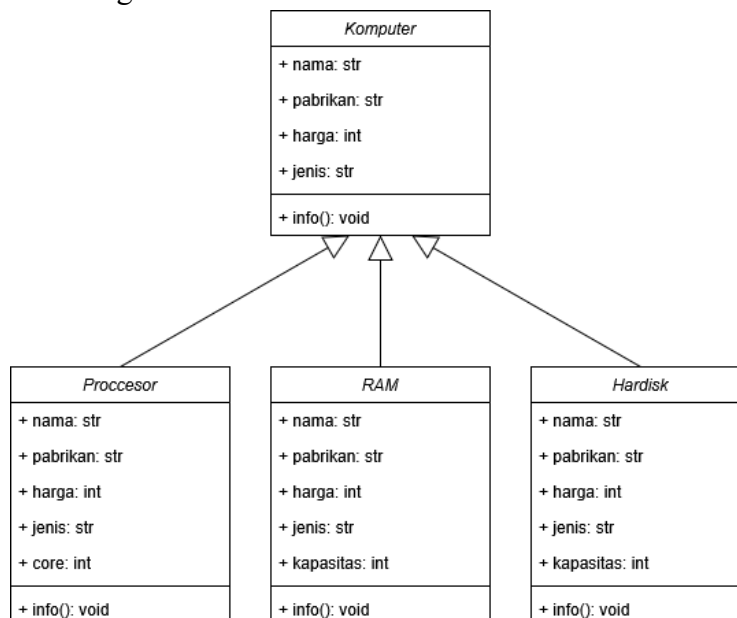
hardisk = Hardisk("Seagate Barracuda", "Seagate", 3000000, "Hard
Disk", 2)
hardisk.info()

```

## 2. Output

Komputer Laptop XYZ adalah jenis Laptop yang diproduksi oleh Dell dengan harga 15000000.  
 Komputer Intel i7 adalah jenis Prosesor yang diproduksi oleh Intel dengan harga 5000000.  
 Kecepatan prosesor: 3.5 GHz.  
 Komputer Corsair Vengeance adalah jenis RAM yang diproduksi oleh Corsair dengan harga 2000000.  
 Kapasitas RAM: 16 GB.  
 Komputer Seagate Barracuda adalah jenis Hard Disk yang diproduksi oleh Seagate dengan harga 3000000.  
 Kapasitas Hard Disk: 2 TB.

## 3. Class Diagram



## 4. Penjelasan

Pada kode tersebut terdapat 4 *class* yang pertama adalah “Komputer”, *class* ini memiliki 4 atribut yaitu nama, pabrikan, harga, dan jenis. *Class* ini juga memiliki 1

*method* yaitu info yang berfungsi untuk menampilkan pesan “Komputer <nama> adalah jenis <jenis> yang diproduksi oleh <pabrikan> dengan harga <harga>.” di terminal.

*Class* yang kedua adalah “Processor”, *class* ini memiliki 4 atribut warisan dari “Komputer” menggunakan *super()* yaitu nama, pabrikan, harga, dan jenis serta 1 atribut bukan warisan yaitu *core*. *Class* ini juga memiliki 1 *method* yaitu *info* yang berfungsi untuk menggunakan *method* *info* warisan dari “Komputer” dan menampilkan pesan “Jumlah Core: <core> Core.” di terminal. Selain itu, terdapat *class* “RAM” dan “Hardisk” yang juga mewarisi atribut dari “Komputer” menggunakan *super()*. *Class* “RAM” memiliki 1 atribut bukan warisan yaitu kapasitas dan *method* *info* untuk menampilkan pesan “Kapasitas RAM: <kapasitas> GB.”. Sedangkan *class* “Hardisk” memiliki 1 atribut bukan warisan yaitu kapasitas dan *method* *info* untuk menampilkan pesan “Kapasitas Hard Disk: <kapasitas> TB.”. Jadi nanti pada *class* turunan akan ada dua pesan yang muncul yaitu dari *method* *info* yang merupakan warisan dari “Komputer” dan perintah “print” pada *method* itu sendiri.

Kemudian dilakukan pembuatan objek “komputer” dari *class* Komputer yang memiliki nama Laptop XYZ, pabrikan Dell, harga 15000000, dan jenis Laptop. Setelah dilakukan pembuatan objek komputer, kemudian dilakukan pemanggilan *method* *info* dari objek tersebut. Adapun objek “prosesor” dari *class* “Processor” yang memiliki nama Intel i7, pabrikan Intel dengan core 4, objek “ram” dari *class* “RAM” dengan kapasitas 16, dan objek “hardisk” dari *class* “Hardisk” dengan kapasitas 2. Setelah dilakukan pembuatan objek-objek tersebut kemudian dilakukan pemanggilan *method* *info* dari masing-masing objek untuk menampilkan spesifikasinya.

Link repository Github: <https://github.com/muzaaqi/pemrograman-berorientasi-objek-praktik>