

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 10



2411102441250

Malik Sabarullah Akbar

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

## Membuat Program Penyewaan Mobil

Pada pertemuan kali ini saya membuat system program penyewaan mobil dengan OOP. Class saya terdiri dari 4 class yaitu, Car, CarDealer, Customer, dan ShowCollection. Masing-masing class memiliki method-nya masing”. Berikut code pembuatan system penyewaan mobil.

### 1. Kelas Car



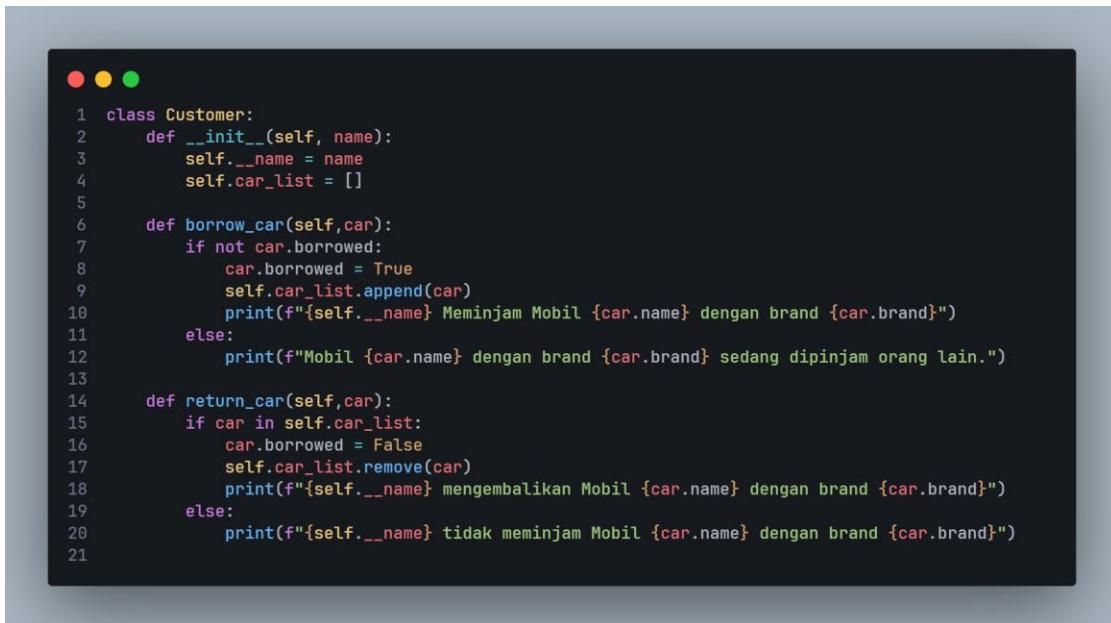
```

● ● ●
1  class Car:
2      def __init__(self, name, brand):
3          self.name = name
4          self.brand = brand
5          self.borrowed = False
6
7      def show_info(self):
8          status = "Borrowed" if self.borrowed else "Available"
9          return f"{'Mobil: {self.name}, Brand: {self.brand}, Status: {status}'}"
10

```

Pada kelas Car terdapat constructor yang menampung atribut public, dan method untuk menampilkan informasi terkait nama mobil, brand, dan status-nya.

### 2. Kelas Customer



```

● ● ●
1  class Customer:
2      def __init__(self, name):
3          self.__name = name
4          self.car_list = []
5
6      def borrow_car(self, car):
7          if not car.borrowed:
8              car.borrowed = True
9              self.car_list.append(car)
10             print(f"{self.__name} Meminjam Mobil {car.name} dengan brand {car.brand}")
11         else:
12             print(f"{'Mobil {car.name} dengan brand {car.brand} sedang dipinjam orang lain.'}")
13
14     def return_car(self, car):
15         if car in self.car_list:
16             car.borrowed = False
17             self.car_list.remove(car)
18             print(f"{self.__name} mengembalikan Mobil {car.name} dengan brand {car.brand}")
19         else:
20             print(f"{self.__name} tidak meminjam Mobil {car.name} dengan brand {car.brand}")
21

```

Pada kelas Customer juga memiliki constructor, tetapi terdapat attribute yang private dan juga public. Kelas ini juga memiliki method yaitu, borrow\_car ( method ini digunakan untuk meminjam mobil, dan merubah status borrowed ), dan return\_car ( method ini digunakan untuk mengembalikan, dan merubah status borrowed juga ). Method tersebut juga menampilkan informasi jika peminjaman atau pengembalian berhasil atau gagal.

### 3. Kelas ShowCollection



```
● ● ●
1 class ShowCollection:
2     def show_collection(self):
3         for car in self.car_collection:
4             print(car.show_info())
5
```

Kelas ini digunakan untuk inheritance pada kelas CarDealer nantinya. Kelas ini hanya memiliki method show\_collection digunakan untuk menampilkan informasi tiap mobil.

### 4. Kelas CardDealer



```
● ● ●
1 from .show_collection import ShowCollection
2
3 class CarDealer(ShowCollection):
4     def __init__(self):
5         self.car_collection = []
6
7     def add_car(self, car):
8         self.car_collection.append(car)
9
```

Kelas ini digunakan untuk menyimpan objek mobil ke dalam car\_collection agar bisa digunakan untuk meminjam mobil. Dan kelas ini juga mewarisi method yang ada di kelas ShowCollection. Dan CarDealer memiliki method penambahan mobil ke dalam car\_collection.

## 5. Operasi Utama

```
 1  from models.car import Car
 2  from models.car_dealer import CarDealer
 3  from models.customer import Customer
 4
 5  if __name__ == "__main__":
 6      mobil1 = Car("Civic", "Honda")
 7      mobil2 = Car("Mustang", "Ford")
 8      mobil3 = Car("Impreza", "Subaru")
 9
10     customer1 = Customer("Alice")
11
12     car_list = [mobil1, mobil2, mobil3]
13
14     for mobil in car_list:
15         dealer = CarDealer()
16         dealer.add_car(mobil)
17
18
19         print("\nKoleksi Mobil di Dealer:")
20         dealer.show_collection()
21
22         print()
23         customer1.borrow_car(mobil)
24
25         print("\nStatus Koleksi Mobil di Dealer Setelah Dipinjam:")
26         dealer.show_collection()
27
28         print()
29         customer1.return_car(mobil)
30         print("\nStatus Koleksi Mobil di Dealer Setelah Dikembalikan:")
31         dealer.show_collection()
32
33         print(20*"")
```

Program ini digunakan untuk melakukan operasi pembuatan object, melakukan Polymorphism, dan penggunaan method-method yang ada pada kelas atau lebih singkatnya operasi utama dari program system penyewaan mobil ini.

Output:

```
PS C:\Users\itsma\Documents\Tugas Kuliah\Informatika\PRATIKUM\Semester 3\OOP\p10> py .\main.py
```

```
Koleksi Mobil di Dealer:  
Mobil: Civic, Brand: Honda, Status: Available
```

```
Alice Meminjam Mobil Civic dengan brand Honda
```

```
Status Koleksi Mobil di Dealer Setelah Dipinjam:  
Mobil: Civic, Brand: Honda, Status: Borrowed
```

```
Alice mengembalikan Mobil Civic dengan brand Honda
```

```
Status Koleksi Mobil di Dealer Setelah Dikembalikan:  
Mobil: Civic, Brand: Honda, Status: Available
```

```
Koleksi Mobil di Dealer:  
Mobil: Mustang, Brand: Ford, Status: Available
```

```
Alice Meminjam Mobil Mustang dengan brand Ford
```

```
Status Koleksi Mobil di Dealer Setelah Dipinjam:  
Mobil: Mustang, Brand: Ford, Status: Borrowed
```

```
Alice mengembalikan Mobil Mustang dengan brand Ford
```

```
Status Koleksi Mobil di Dealer Setelah Dikembalikan:  
Mobil: Mustang, Brand: Ford, Status: Available
```

```
Koleksi Mobil di Dealer:  
Mobil: Impreza, Brand: Subaru, Status: Available
```

```
Alice Meminjam Mobil Impreza dengan brand Subaru
```

```
Status Koleksi Mobil di Dealer Setelah Dipinjam:  
Mobil: Impreza, Brand: Subaru, Status: Borrowed
```

```
Alice mengembalikan Mobil Impreza dengan brand Subaru
```

```
Status Koleksi Mobil di Dealer Setelah Dikembalikan:  
Mobil: Impreza, Brand: Subaru, Status: Available
```

```
PS C:\Users\itsma\Documents\Tugas Kuliah\Informatika\PRATIKUM\Semester 3\OOP\p10>
```

## **Kesimpulan**

Program ini dibuat untuk memperdalam terkait penggunaan OOP, khususnya pada Inheritance, Enkapsulasi, Polymorphism, dan Modularisasi. Saya menggunakan studi kasus ini karena mudah dibuat dan dipahami oleh karena itu saya membuat program system penyewaan mobil yang sederhana menggunakan OOP.