

**Nama : Al Kautsar Russetya Tamora**

**NIM : 2100018298**

**Kelas : F**

**URL Git :**

**Petunjuk: Realisasi Class Diagram dalam kode program anda dan tunjukkan realisasi tersebut berdasarkan beberapa hal berikut:**

**Catatan: jika diperlukan boleh gunakan perintah komentar di kode program untuk menjelaskan kode program dan tunjukkan kelas main untuk mensimulasikan realisasi.**

### **Komponen Dasar PBO**

- Daftar Objek, daftar property/state, daftar behavior

```
penyewa  
petugas  
sewa  
mobil_mpv  
mobil_suv
```

- Pembuatan Class (tidak perlu dengan isi class karena akan dijelaskan masing-masing)

```
public class penyewa  
public class petugas  
public class sewa  
public class mobil_mpv  
public class mobil_suv
```

- Variable (tipe data, access modifier, nama variable)

```
String nama_penyewa;  
String no_hp;  
String alamat;  
String nama_petugas;  
String no_hp;  
String kd_sewa;  
int harga;  
int lama_sewa;
```

- Method

- o Method berupa Function

```
public int total(int harga, int lama_sewa) {  
    System.out.println("-");  
    return 0;  
}
```

- o Method berupa Procedure

```

public void jenis_mobil1(){
    System.out.println("Anda akan menyewa mobil mpv jenis = ");
}
public void jenis_mobil2(){
    System.out.println("Anda akan menyewa mobil suv jenis = ");
}

```

o Method berupa Setter

```

public void setKd_sewa(String kd_sewa) {
    this.kd_sewa = kd_sewa;
}
public void setHarga(int harga) {
    this.harga = harga;
}
public void setLama_sewa(int lama_sewa) {
    this.lama_sewa = lama_sewa;
}
public void setName_petugas(String nama_petugas) {
    this.nama_petugas = nama_petugas;
}
public void setNo_hp(String no_hp) {
    this.no_hp = no_hp;
}
public void setName_penyewa(String nama_penyewa) {
    this.nama_penyewa = nama_penyewa;
}
public void setNo_hp(String no_hp) {
    this.no_hp = no_hp;
}
public void setAddress(String alamat) {
    this.alamat = alamat;
}
}

```

o Method berupa Getter

```

public String getName_penyewa() {
    return nama_penyewa;
}
public String getNo_hp() {
    return no_hp;
}
public String getAddress() {
    return alamat;
}
public String getName_petugas() {
    return nama_petugas;
}
public String getNo_hp() {
    return no_hp;
}
public String getKd_sewa() {
    return kd_sewa;
}
public int getHarga() {
    return harga;
}
public int getLama_sewa() {
    return lama_sewa;
}
}

```

- Method berupa Boolean properti

- Method berupa non-boolean properti

- Method dengan parameter

- Method dengan Argument Lists

- Method static

- Method final

## Konstruktor

- Constructor

```
public sewa(String kd_sewa) {  
    this.kd_sewa = kd_sewa;  
}
```

- Constructor Overloading

```
//Scanner petugas dan input data  
petugas ptgs = new petugas();  
Scanner Petugas = new Scanner(System.in);
```

- Instansiasi dengan menggunakan dua jenis konstruktor di kelas main

## Enkapsulasi dan access modifier

- Paket

- Praktek yang menunjukkan enkapsulasi dengan private di kelas main

- Praktek yang menunjukkan enkapsulasi dengan protected di kelas main

## Pewarisan

- Pembuatan perwarisan

```
public class mobil_mpv extends sewa {
    int harga;
    int lama_sewa;
    public mobil_mpv(String kd_sewa) {
        super(kd_sewa);
    }
}
public class mobil_suv extends sewa {
    int harga;
    int lama_sewa;
    public mobil_suv(String kd_sewa) {
        super(kd_sewa);
    }
}
```

- Pembuatan ancestor class atau kelas yang memiliki banyak kelas orang tua/kelas leluhur

- Praktek yang menunjukkan dan penjelasan mengapa harus dilakukan Pewarisan

## Polymorphisme

- Method Overloading

- Method Overriding

```
@Override
public void jenis_mobil2() {
    System.out.println(" 1. Fortuner, 800000/hari");
    System.out.println(" 2. Pajero, 900000/hari");
}

@Override
public int total(int harga, int lama_sewa) {
    return harga * lama_sewa;
}

@Override
public void jenis_mobil1() {
    System.out.println(" 1. Avanza, 250000/hari");
    System.out.println(" 2. X-pander, 300000/hari");
}

@Override
public int total(int harga, int lama_sewa) {
```

```
}  
    return harga * lama_sewa;  
}
```

- Overloading atau Overriding pada Method static

- Overloading atau Overriding pada Method final

- Overloading atau Overriding dengan perubahan return type

- Overloading atau Overriding dengan perubahan access modifier (public, protected, dan private)