

Laporan Praktikum PBO - Modul 10

Nama: Gerald Eberhard

NIM: 105223002

I. Pendahuluan

Program ini merupakan simulasi sistem penyewaan kendaraan yang terdiri dari berbagai jenis, yaitu Bus, Truk, dan MobilPribadi. Setiap kendaraan memiliki atribut dasar seperti nama, nomor plat, dan biaya sewa. Dengan memanfaatkan konsep pewarisan (inheritance), interface, serta polymorphism, program ini menunjukkan bagaimana struktur hierarki kelas dapat diterapkan secara efisien dalam pengembangan perangkat lunak berbasis objek.

II. Variabel

No	Nama Variabel	Tipe Data	Fungsi
1	platNomor	String	Menyimpan nomor plat kendaraan
2	merk	String	Menyimpan merek kendaraan
3	tahunProduksi	int	Menyimpan tahun produksi kendaraan
4	hargaSewaPerHari	int	Menyimpan harga sewa per hari untuk kendaraan
5	kapasitas	double	Menyimpan kapasitas muatan (hanya untuk Truk)
6	mobil	MobilPribadi	Objek untuk menyimpan instance MobilPribadi
7	bus	Bus	Objek untuk menyimpan instance Bus
8	truk	Truk	Objek untuk menyimpan instance Truk
9	hari	int	Parameter untuk menghitung biaya sewa berdasarkan jumlah hari

III. Constructor dan Method

No	Nama Metode	Jenis Metode	Fungsi
1	Kendaraan(String, String, int)	Constructor	Menginisialisasi objek Kendaraan dengan platNomor, merk, dan tahunProduksi
2	tampilkanInfo()	Method (Procedural)	Menampilkan informasi kendaraan (platNomor, merk, tahunProduksi)

3	hitungBiayaSewa(int)	Method (Functional)	Menghitung biaya sewa berdasarkan jumlah hari (abstrak di Kendaraan)
4	perluSupir()	Method (Functional)	Menentukan apakah kendaraan memerlukan supir (abstrak di Kendaraan)
5	kapasitasMuatan()	Method (Functional)	Mengembalikan kapasitas muatan (hanya di Truk)
6	MobilPribadi(String, String, int)	Constructor	Menginisialisasi objek MobilPribadi dengan memanggil konstruktor Kendaraan
7	Bus(String, String, int)	Constructor	Menginisialisasi objek Bus dengan memanggil konstruktor Kendaraan
8	Truk(String, String, int)	Constructor	Menginisialisasi objek Truk dengan memanggil konstruktor Kendaraan
9	main(String[])	Method (Procedural)	Metode utama untuk menjalankan program dan menguji sistem

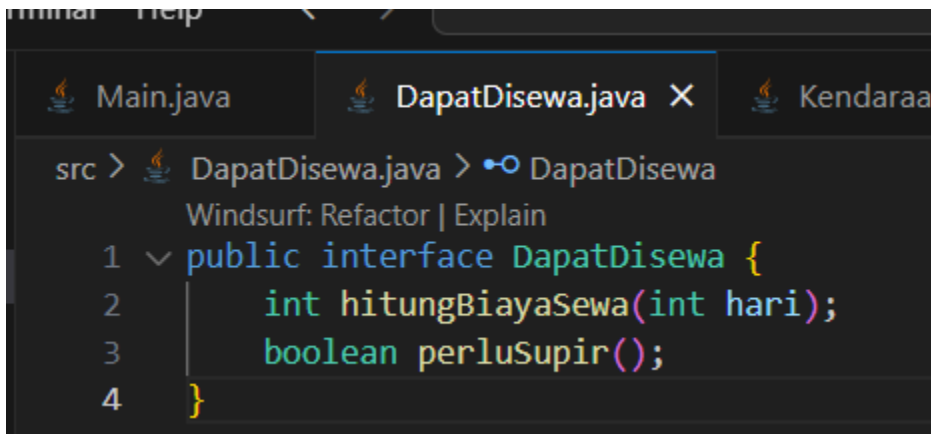
IV. Dokumentasi dan Pembahasan Code

1. Kendaraan.java

```
src > Main.java x DapatDisewa.java x Kendaraan.java x Muatan.java MobilPr
Windsurf: Refactor | Explain
1 public abstract class Kendaraan {
2     protected String platNomor;
3     protected String merk;
4     protected int tahunProduksi;
5
6     public Kendaraan(String platNomor, String merk, int tahunProduksi) {
7         this.platNomor = platNomor;
8         this.merk = merk;
9         this.tahunProduksi = tahunProduksi;
10    }
11
12    Windsurf: Refactor | Explain | Generate Javadoc | X
13    public void tampilkanInfo() {
14        System.out.println("Plat Nomor: " + platNomor);
15        System.out.println("Merk: " + merk);
16        System.out.println("Tahun Produksi: " + tahunProduksi);
17    }
18
19    public abstract int hitungBiayaSewa(int hari);
20    public abstract boolean perluSupir();
21 }
```

- **Fungsi Utama:** Mendefinisikan kelas abstrak Kendaraan sebagai dasar untuk semua jenis kendaraan dalam sistem sewa.
- **Struktur:**
 - Atribut: platNomor (String), merk (String), tahunProduksi (int) dengan akses protected agar dapat diwarisi.
 - Constructor: Kendaraan(String platNomor, String merk, int tahunProduksi) untuk inisialisasi atribut.
 - Method Konkret: tampilkanInfo() menampilkan informasi kendaraan (plat nomor, merek, tahun produksi) ke konsol.
 - Method Abstrak: hitungBiayaSewa(int hari) dan perluSupir() yang harus diimplementasikan oleh kelas turunan.
- **Logika:** Kelas ini menyediakan kerangka umum dengan pewarisan, memastikan semua kendaraan memiliki properti dasar dan metode yang konsisten.

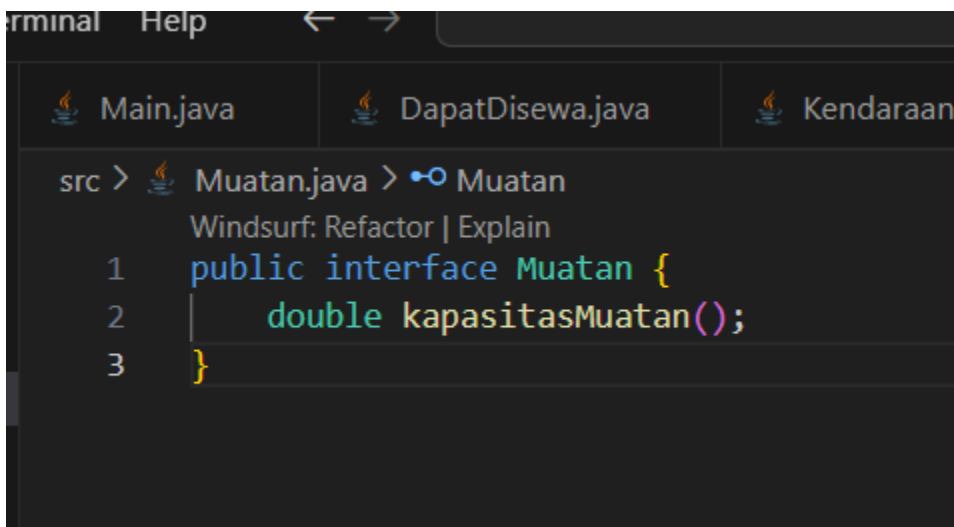
2. DapatDisewa.java



```
src > Main.java x DapatDisewa.java x Kendaraa
src > DapatDisewa.java > DapatDisewa
Windsurf: Refactor | Explain
1 public interface DapatDisewa {
2     int hitungBiayaSewa(int hari);
3     boolean perluSupir();
4 }
```

- **Fungsi Utama:** Mendefinisikan antarmuka DapatDisewa yang menetapkan kontrak untuk kendaraan yang dapat disewakan.
- **Struktur:**
 - Method: `hitungBiayaSewa(int hari)` (mengembalikan `int`) untuk menghitung biaya sewa berdasarkan jumlah hari.
 - Method: `perluSupir()` (mengembalikan `boolean`) untuk menentukan kebutuhan supir.
- **Logika:** Antarmuka ini memastikan semua kendaraan yang dapat disewakan memiliki logika untuk menghitung biaya dan mengecek kebutuhan supir, meningkatkan fleksibilitas sistem.

3. Muatan.java

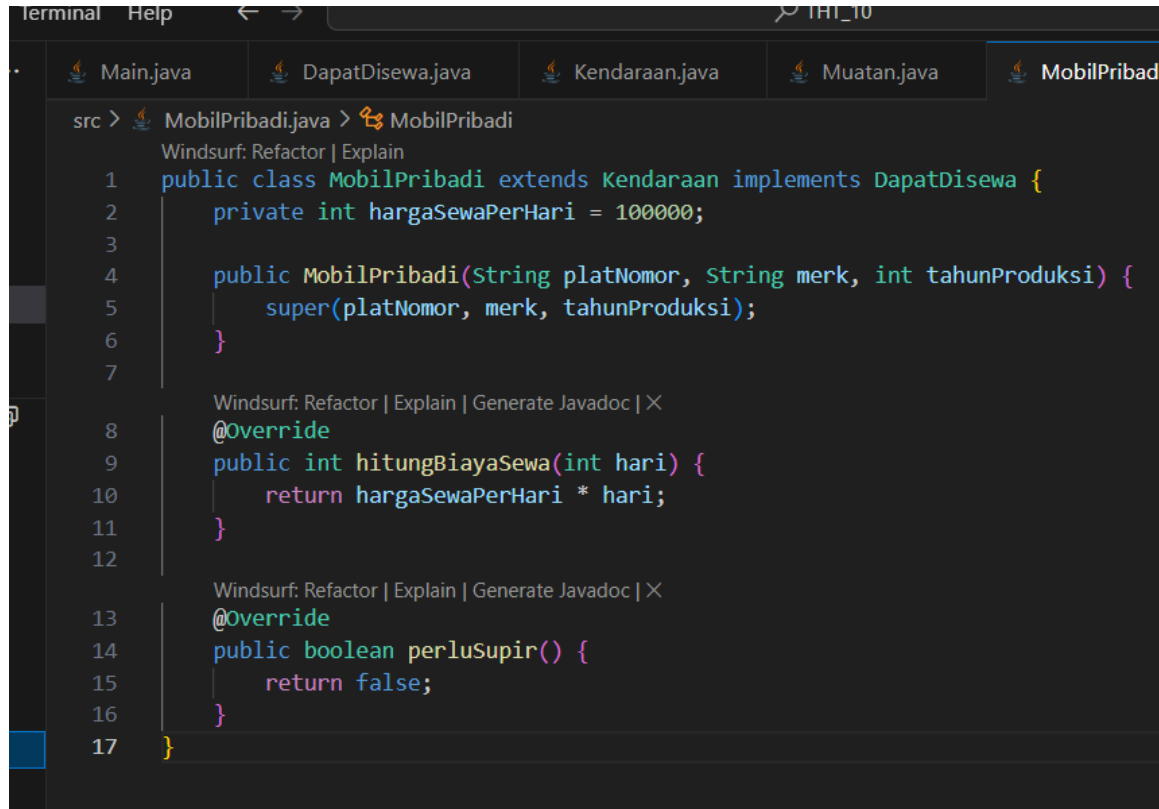


```
src > Main.java x DapatDisewa.java x Kendaraan.java
src > Muatan.java > Muatan
Windsurf: Refactor | Explain
1 public interface Muatan {
2     double kapasitasMuatan();
3 }
```

- **Fungsi Utama:** Mendefinisikan antarmuka Muatan untuk kendaraan yang memiliki kapasitas muatan, seperti truk.

- **Struktur:**
 - Method: kapasitasMuatan() (mengembalikan double) untuk mengembalikan kapasitas muatan dalam kg.
- **Logika:** Antarmuka ini memperluas fungsionalitas untuk kendaraan tertentu (misalnya, Truk) dengan atribut tambahan, mendukung polimorfisme.

4. MobilPribadi.java



```

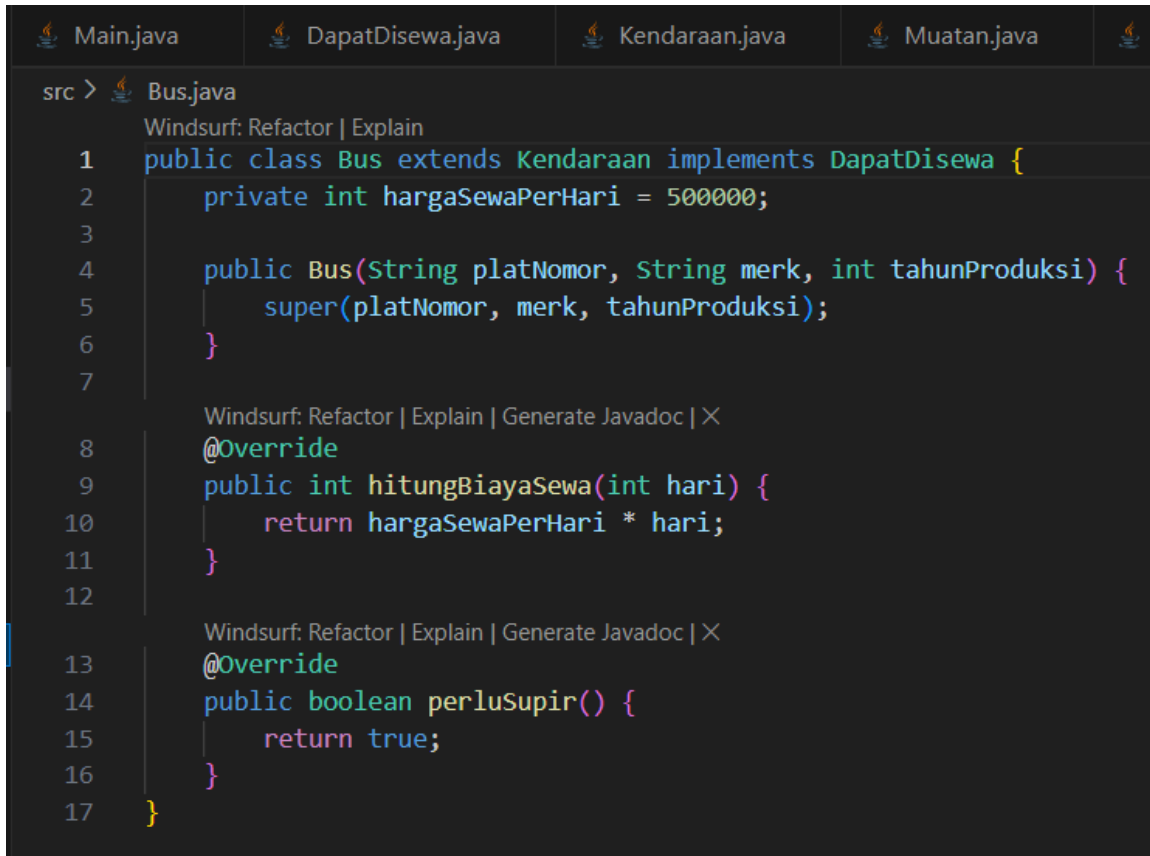
src > MobilPribadi.java > MobilPribadi
Windsurf: Refactor | Explain
1 public class MobilPribadi extends Kendaraan implements DapatDisewa {
2     private int hargaSewaPerHari = 100000;
3
4     public MobilPribadi(String platNomor, String merk, int tahunProduksi) {
5         super(platNomor, merk, tahunProduksi);
6     }
7
8     Windsurf: Refactor | Explain | Generate Javadoc | X
9     @Override
10    public int hitungBiayaSewa(int hari) {
11        return hargaSewaPerHari * hari;
12    }
13
14    Windsurf: Refactor | Explain | Generate Javadoc | X
15    @Override
16    public boolean perluSupir() {
17        return false;
18    }
19 }

```

- **Fungsi Utama:** Mengimplementasikan kelas MobilPribadi sebagai turunan dari Kendaraan dengan kemampuan sewa.
- **Struktur:**
 - Atribut: hargaSewaPerHari (int) dengan nilai default 100.000.
 - Constructor: MobilPribadi(String platNomor, String merk, int tahunProduksi) memanggil konstruktor Kendaraan.
 - Method:
 - hitungBiayaSewa(int hari) mengembalikan hargaSewaPerHari * hari.

- `perluSupir()` mengembalikan false (mobil pribadi tidak memerlukan supir).
- **Logika:** Kelas ini menyediakan implementasi spesifik untuk mobil pribadi dengan biaya tetap dan tanpa kebutuhan supir.

5. Bus.java

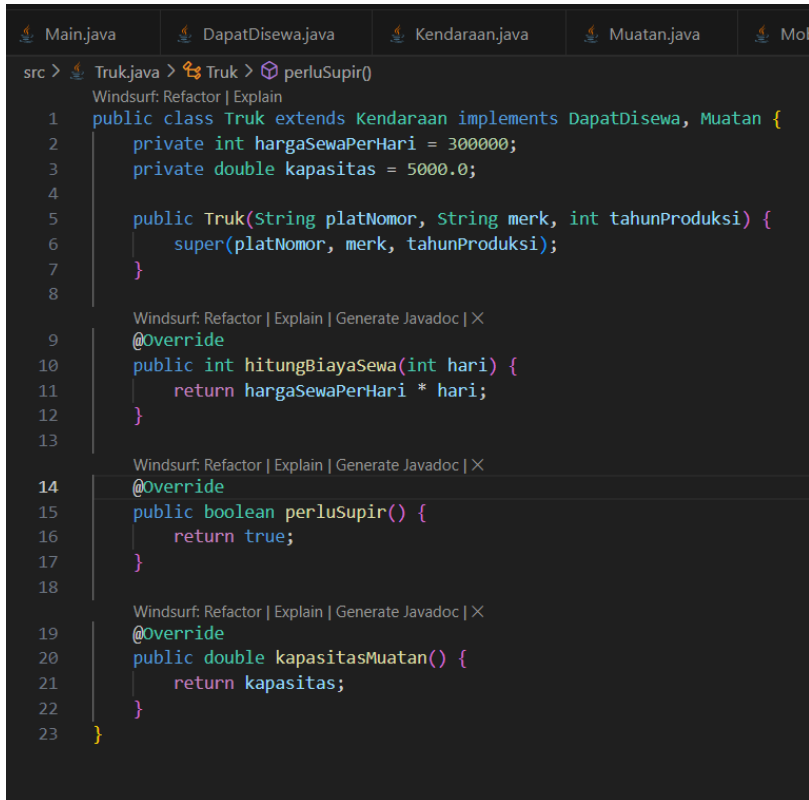


```
src > Bus.java
Windsurf: Refactor | Explain
1 public class Bus extends Kendaraan implements DapatDisewa {
2     private int hargaSewaPerHari = 500000;
3
4     public Bus(String platNomor, String merk, int tahunProduksi) {
5         super(platNomor, merk, tahunProduksi);
6     }
7
8     Windsurf: Refactor | Explain | Generate Javadoc | X
9     @Override
10    public int hitungBiayaSewa(int hari) {
11        return hargaSewaPerHari * hari;
12    }
13
14    Windsurf: Refactor | Explain | Generate Javadoc | X
15    @Override
16    public boolean perluSupir() {
17        return true;
18    }
19 }
```

- **Fungsi Utama:** Mengimplementasikan kelas Bus sebagai turunan dari Kendaraan dengan kemampuan sewa.
- **Struktur:**
 - Atribut: `hargaSewaPerHari` (int) dengan nilai default 500.000.
 - Constructor: `Bus(String platNomor, String merk, int tahunProduksi)` memanggil konstruktor Kendaraan.
 - Method:
 - `hitungBiayaSewa(int hari)` mengembalikan `hargaSewaPerHari * hari`.
 - `perluSupir()` mengembalikan true (bus memerlukan supir).

- **Logika:** Kelas ini dirancang untuk bus dengan biaya lebih tinggi dan kebutuhan supir, mencerminkan karakteristik nyata.

6. Truk.java



```

src > Truk.java > Truk > perluSupir()
Windsurf: Refactor | Explain
1 public class Truk extends Kendaraan implements DapatDisewa, Muatan {
2     private int hargaSewaPerHari = 300000;
3     private double kapasitas = 5000.0;
4
5     public Truk(String platNomor, String merk, int tahunProduksi) {
6         super(platNomor, merk, tahunProduksi);
7     }
8
9     Windsurf: Refactor | Explain | Generate Javadoc | X
10    @Override
11    public int hitungBiayaSewa(int hari) {
12        return hargaSewaPerHari * hari;
13    }
14
15    Windsurf: Refactor | Explain | Generate Javadoc | X
16    @Override
17    public boolean perluSupir() {
18        return true;
19    }
20
21    Windsurf: Refactor | Explain | Generate Javadoc | X
22    @Override
23    public double kapasitasMuatan() {
24        return kapasitas;
25    }
26 }

```

- **Fungsi Utama:** Mengimplementasikan kelas Truk sebagai turunan dari Kendaraan dengan kemampuan sewa dan muatan.
- **Struktur:**
 - Atribut: hargaSewaPerHari (int) dengan nilai default 300.000, kapasitas (double) dengan nilai default 5000.0.
 - Constructor: Truk(String platNomor, String merk, int tahunProduksi) memanggil konstruktor Kendaraan.
 - Method:
 - hitungBiayaSewa(int hari) mengembalikan hargaSewaPerHari * hari.
 - perluSupir() mengembalikan true (truk memerlukan supir).
 - kapasitasMuatan() mengembalikan nilai kapasitas.
- **Logika:** Kelas ini mendukung fungsi tambahan untuk muatan, sesuai dengan antarmuka Muatan, dan mencerminkan kebutuhan supir.

7. Main.java

```
1 public class Main {
2     public static void main(String[] args) {
3         MobilPribadi mobil = new MobilPribadi(platNomor:"B1234AA", merk:"Toyota", tahunProduksi:2015);
4         System.out.println(x:"=== Mobil Pribadi ===");
5         mobil.tampilkanInfo();
6         System.out.println("Biaya sewa untuk 3 hari: " + mobil.hitungBiayaSewa(hari:3));
7         System.out.println("Perlu supir: " + mobil.perluSupir());
8
9         Bus bus = new Bus(platNomor:"B5678BB", merk:"Mercedes", tahunProduksi:2020);
10        System.out.println(x:"\n=== Bus ===");
11        bus.tampilkanInfo();
12        System.out.println("Biaya sewa untuk 2 hari: " + bus.hitungBiayaSewa(hari:2));
13        System.out.println("Perlu supir: " + bus.perluSupir());
14
15        Truk truk = new Truk(platNomor:"B9012CC", merk:"Hino", tahunProduksi:2018);
16        System.out.println(x:"\n=== Truk ===");
17        truk.tampilkanInfo();
18        System.out.println("Biaya sewa untuk 4 hari: " + truk.hitungBiayaSewa(hari:4));
19        System.out.println("Perlu supir: " + truk.perluSupir());
20        System.out.println("Kapasitas muatan: " + truk.kapasitasMuatan() + " kg");
21    }
22 }
```

- **Fungsi Utama:** Menguji sistem sewa kendaraan dengan membuat instance dari kelas turunan dan memanggil methodnya.
- **Struktur:**
 - Method: `main(String[] args)` sebagai entry point program.
 - Logika:
 - Membuat objek mobil (`MobilPribadi`), bus (`Bus`), dan truk (`Truk`) dengan parameter spesifik.
 - Memanggil `tampilkanInfo()`, `hitungBiayaSewa(3/2/4)`, `perluSupir()`, dan `kapasitasMuatan()` (untuk truk) untuk menampilkan hasil.
- **Logika:** Program ini mendemonstrasikan penggunaan polimorfisme dan pewarisan dengan memanggil method yang diimplementasikan secara berbeda di setiap kelas.

V. Kesimpulan

Program berhasil menerapkan prinsip-prinsip OOP seperti inheritance, interface, dan polymorphism. Setiap kelas dirancang modular dan reusable. Sistem ini fleksibel untuk dikembangkan, misalnya dengan menambahkan jenis kendaraan lain. Interface 'DapatDisewa' menjamin konsistensi perhitungan biaya sewa di seluruh kendaraan yang dapat disewa.

VI. Daftar Pustaka

[1]. Materi Praktikum Modul 10