

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 10
ABSTRACT & INTERFACE



Disusun Oleh:
Florence Regis Lake
105223006

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. PENDAHULUAN

Sebagai seorang pengembang perangkat lunak, telah ditugaskan membuat sebuah program Java berbasis konsol untuk memenuhi kebutuhan dari sebuah sistem sewa kendaraan. Dengan kondisi sekarang ini, dimana proses penyewaan kendaraan masih terbilang rumit, dibangunlah sistem yang mampu melakukan penyewaan atas kendaraan mobil pribadi, bus, dan truk.

Beberapa fitur yang diinginkan oleh perpustakaan tersebut agar dapat melengkapi sistem penyewaan kendaraan adalah, menerima data seperti plat nomor dari kendaraan itu, merk kendaraan, dan tahun produksi. Sistem juga mampu menghitung biaya sewa kendaraan setiap harinya dan menentukan apakah kendaraan tersebut memerlukan supir atau tidak. Untuk kendaraan truk, harus dihitung juga kapasitas muatan yang mampu ditampung kendaraan itu.

II. VARIABEL

No	Nama Variabel	Tipe data	Fungsi
1	platNomor	public String	Attribut dari class Kendaraan untuk menyimpan plat nomor dari kendaraan
2	merk	public String	Attribut dari class Kendaraan untuk menyimpan merk dari kendaraan
3	tahunProduksi	public int	Attribut dari class Kendaraan untuk menyimpan tahun produksi dari kendaraan
4	bus	Kendaraan = new Bus	Sebagai object dari class Bus yang meng-extend class Kendaraan
5	mobilPribadi	Kendaraan = new MobilPribadi	Sebagai object dari class MobilPribadi yang meng-extend class Kendaraan
6	truk	Kendaraan = new Truk	Sebagai object dari class Truk yang meng-extend class Kendaraan

III. CONSTRUCTOR DAN METHOD

No	Nama Method	Jenis Method	Fungsi
1	Kendaraan	constructor	Menginisiasi nilai pada atribut di class Kendaraan
2	MobilPribadi	constructor	Menginisiasi nilai pada atribut di class Kendaraan melalui class MobilPribadi
3	Bus	constructor	Menginisiasi nilai pada atribut di class Kendaraan melalui class Bus
4	Truk	constructor	Menginisiasi nilai pada atribut di class Kendaraan melalui class Truk
5	tampilkanInfo	procedural	Method yang nantinya akan di-override oleh class lainnya
6	hitungBiayaSewa	functional	Method yang nantinya akan di-override oleh class lainnya
7	perluSupir	functional	Method yang nantinya akan di-override oleh class lainnya
8	kapasitasMuatan	functional	Method yang nantinya akan di-override oleh class lainnya

IV. DOKUMENTASI DAN PEMBAHASAN CODE

3.1. Parent class

Parent class pada program ini adalah class Kendaraan yang merupakan sebuah abstract class. Dengan class ini, semua class lainnya seperti class Bus, MobilPribadi, dan Truk akan melakukan proses extend pada class ini yang nantinya menjadikan ketiga class tersebut menjadi child class dari class Kendaraan ini.

```
1 public abstract class Kendaraan {
2     public String platNomor;
3     public String merk;
4     public 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 double hitungBiayaSewa(int hari);
20    public abstract boolean perluSupir();
21 }
```

Gambar 3.1 parent class kendaraan

3.2. Child class

Terdapat tiga child class pada program ini, yaitu class MobilPribadi, class Bus, dan class Truk. Ketiga child class itu memiliki constructornya masing-masing dimana dalam constructor tersebut menggunakan syntax “super(platNomor, merk, tahunProduksi)” agar nilai dari variabel tersebut akan disimpan kedalam atribut pada parent class nya yaitu class Kendaraan. Juga method yang dimiliki di parent class akan diturunkan juga ke child class.

```
1 public class MobilPribadi extends Kendaraan implements DapatSewa{
2     public MobilPribadi(String platNomor, String merk, int tahunProduksi) {
3         super(platNomor, merk, tahunProduksi);
4     }
5
6     Windsurf: Refactor | Explain | Generate Javadoc | X
7     public void tampilkanInfo() {
8         super.tampilkanInfo();
9     }
10
11    Windsurf: Refactor | Explain | Generate Javadoc | X
12    public double hitungBiayaSewa(int hari) {
13        return 300000 * hari;
14    }
15
16    Windsurf: Refactor | Explain | Generate Javadoc | X
17    public boolean perluSupir() {
18        return false;
19    }
20 }
```

Gambar 3.2 child class MobilPribadi

```

1 public class Bus extends Kendaraan implements DapatSewa{
2     public Bus(String platNomor, String merk, int tahunProduksi) {
3         super(platNomor, merk, tahunProduksi);
4     }
5
6     Windsurf: Refactor | Explain | Generate Javadoc | X
7     public void tampilkanInfo() {
8         super.tampilkanInfo();
9     }
10
11    Windsurf: Refactor | Explain | Generate Javadoc | X
12    public double hitungBiayaSewa(int hari) {
13        return 650000 * hari;
14    }
15
16    Windsurf: Refactor | Explain | Generate Javadoc | X
17    public boolean perluSupir() {
18        return true;
19    }
20 }

```

Gambar 3.3 child class Bus

```

1 public class Truk extends Kendaraan implements DapatSewa, Muatan {
2     private double kapasitas;
3
4     public Truk(String platNomor, String merk, int tahunProduksi, double kapasitas) {
5         super(platNomor, merk, tahunProduksi);
6         this.kapasitas = kapasitas;
7     }
8
9     Windsurf: Refactor | Explain | Generate Javadoc | X
10    public void tampilkanInfo() {
11        super.tampilkanInfo();
12        System.out.println("Kapasitas Muatan: " + kapasitas + " ton");
13    }
14
15    Windsurf: Refactor | Explain | Generate Javadoc | X
16    public double hitungBiayaSewa(int hari) {
17        return 2250000 * hari;
18    }
19
20    Windsurf: Refactor | Explain | Generate Javadoc | X
21    public boolean perluSupir() {
22        return true;
23    }
24
25    Windsurf: Refactor | Explain | Generate Javadoc | X
26    public double kapasitasMuatan() {
27        return kapasitas;
28    }
29 }

```

Gambar 3.4 child class Truk

Dari ketiga child class ini, mereka juga mengimplementasikan setiap method yang dimiliki pada parent class, class Kendaraan. Dengan begitu, saat dipanggil pada class Main seluruh method tersebut akan di-override yang memungkinkan untuk menjalankan method tersebut dan mendapatkan output menurut dari object class yang memanggilnya. Juga, ketiga child class ini melakukan proses extends dan implements dari class lainnya agar mendapatkan akses ke method pada class tersebut.

3.3. Interface

Terdapat dua interface yang dimiliki program ini, diantaranya adalah interface Muatan, dan interface DapatSewa. Masing-masing interface memiliki fungsinya sendiri, dimana interface Muatan memiliki method kapasitasMuatan(); dan interface DapatSewa memiliki method hitungBiayaSewa(int hari) dan perluSupir(). Dengan kedua interface tersebut, nantinya dapat digunakan oleh child class dalam implementasinya.

```
1 public interface DapatSewa {
2     public double hitungBiayaSewa(int hari);
3     public boolean perluSupir();
4 }
```

Gambar 3.5 interface DapatSewa

```
1 public interface Muatan {
2     public double kapasitasMuatan();
3 }
```

Gambar 3.6 interface Muatan

3.4. Main class

Pada class Main, diinisialisasilah object bus, mobilPribadi, dan truk. Saat diinisialisasi, ketiga object tersebut diberikan nilai sesuai dari atribut yang ada, dengan begitu dapat dilakukan uji coba hasil output dari program yang telah dibuat ini.

```
1 public class Main {
2     public static void main(String[] args) {
3         Kendaraan bus = new Bus(platNomor:"B1234CD", merk:"Mercedes", tahunProduksi:2020);
4         Kendaraan mobilPribadi = new MobilPribadi(platNomor:"B5678EF", merk:"Toyota", tahunProduksi:2019);
5         Kendaraan truk = new Truk(platNomor:"B9101GH", merk:"Isuzu", tahunProduksi:2018, kapasitas:5.0);
6
7         System.out.println();
8         bus.tampilkanInfo();
9         System.out.println("Biaya Sewa Bus untuk 3 hari: " + bus.hitungBiayaSewa(hari:3));
10        System.out.println("Perlu Supir: " + bus.perluSupir());
11        System.out.println();
12        mobilPribadi.tampilkanInfo();
13        System.out.println("Biaya Sewa Mobil Pribadi untuk 2 hari: " + mobilPribadi.hitungBiayaSewa(hari:2));
14        System.out.println("Perlu Supir: " + mobilPribadi.perluSupir());
15        System.out.println();
16        truk.tampilkanInfo();
17        System.out.println("Biaya Sewa Truk untuk 4 hari: " + truk.hitungBiayaSewa(hari:4));
18        System.out.println("Perlu Supir: " + truk.perluSupir());
19        System.out.println();
20    }
21 }
```

Gambar 3.7 class Main

```
Plat Nomor: B1234CD
Merk: Mercedes
Tahun Produksi: 2020
Biaya Sewa Bus untuk 3 hari: 1950000.0
Perlu Supir: true

Plat Nomor: B5678EF
Merk: Toyota
Tahun Produksi: 2019
Biaya Sewa Mobil Pribadi untuk 2 hari: 600000.0
Perlu Supir: false

Plat Nomor: B9101GH
Merk: Isuzu
Tahun Produksi: 2018
Kapasitas Muatan: 5.0 ton
Biaya Sewa Truk untuk 4 hari: 9000000.0
Perlu Supir: true
```

Gambar 3.8 output

V. KESIMPULAN

Berdasarkan permintaan dan kebutuhan yang ingin dicapai dan dilaksanakan oleh program, dan setelah terlihat hasil akhir atau output yang telah dibuat pada program ini. Maka dapat dikatakan bahwa program yang dibuat berhasil memenuhi kebutuhan yang diinginkan sesuai dengan permintaan tanpa adanya penambahan fitur maupun pengurangan fitur sehingga jalannya program dapat dilaksanakan secara efisien dan efektif.

