

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL IX
OVERRIDING dan POLYMORPHISM



Disusun Oleh:
Mohamad Dandung Sadat 105222033

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. Pendahuluan

Dalam pengembangan sistem perangkat lunak yang kompleks dan dinamis seperti sistem pemesanan tiket transportasi, pendekatan berorientasi objek memberikan struktur dan fleksibilitas yang sangat dibutuhkan. Paradigma Pemrograman Berorientasi Objek (PBO) memungkinkan kita untuk mendefinisikan entitas seperti Bus, Kereta, dan Pesawat sebagai objek yang memiliki atribut dan perilaku masing-masing, namun tetap dapat diproses dalam satu kesatuan melalui prinsip polymorphism.

Laporan ini membahas simulasi sistem transportasi yang dibangun dengan bahasa Java, dengan fokus pada penerapan method overriding, overloading, serta polymorphism. Dalam sistem ini, terdapat superclass Transportasi yang menjadi dasar bagi tiga subclass yaitu Bus, Kereta, dan Pesawat. Masing-masing subclass memiliki logika perhitungan harga tiket yang berbeda-beda, yang diterapkan melalui overriding method hitungHargaTiket().

Selain itu, overloading digunakan untuk memungkinkan pengguna memilih layanan berdasarkan kelas seperti Ekonomi, Bisnis, dan VIP, yang memengaruhi harga akhir tiket. Program juga memanfaatkan polymorphism agar ketiga jenis transportasi dapat dimasukkan ke dalam satu array bertipe Transportasi dan diproses secara seragam dalam satu loop. Pendekatan ini sejalan dengan prinsip OOP yaitu reusable dan extensible, menjadikan sistem lebih efisien dan mudah dikembangkan di masa depan.

II. Variabel

No	Nama Variabel	Tipe data	Fungsi
1	nama	String	Menyimpan nama dari transportasi
2	jumlahKursi	int	Menyimpan kapasitas kursi transportasi
3	tujuan	String	Menyimpan kota tujuan akhir perjalanan
4	hargaDefault	double	Nilai dasar tiket sebelum dikalikan berdasarkan jenis
5	daftarTransportasi	Transportasi[]	Array berisi objek-objek subclass Transportasi

III. Constructor dan Method

No	Nama Metode	Jenis Metode	Fungsi
1	Transportasi(String, int, String)	Constructor	Membuat objek transportasi umum
2	hitungHargaTiket()	Override	Menghitung harga default sesuai jenis transportasi
3	hitungHargaTiket(String)	Overload	Menghitung harga berdasarkan kelas layanan (ekonomi, bisnis, vip)
4	tampilkanInfo()	Procedural	Menampilkan informasi nama, jumlah kursi, dan tujuan
5	main(String[] args)	Main	Menguji seluruh fitur polymorphism dengan array objek Transportasi

IV. Dokumentasi dan Pembahasan Code

4.1. Class Transportasi

```
class Transportasi {  
    private String nama;  
    private int jumlahKursi;  
    private String tujuan;  
    protected double hargaDefault = 100000.0;  
}
```

Membuat superclass Transportasi sebagai dasar semua moda transportasi dengan atribut nama, jumlahKursi, dan tujuan.

```
public double hitungHargaTiket() {  
    return hargaDefault;  
}
```

Method default menghitung harga tiket awal, akan dioverride oleh subclass.

```
public double hitungHargaTiket(String kelasLayanan) {  
    return hitungHargaTiket();  
}
```

Method overload untuk menghitung harga berdasarkan kelas layanan.

4.2. Class Bus

```
class Bus extends Transportasi {  
    @Override  
    public double hitungHargaTiket() {  
        return super.hitungHargaTiket() * 1.10;  
    }  
}
```

Subclass Bus meng-override harga default menjadi 110%.

```
switch (kelasLayanan.toLowerCase()) {  
    case "ekonomi": return hargaDasarBus;  
    case "bisnis": return hargaDasarBus * 1.25;  
    case "vip": return hargaDasarBus * 1.50;  
}
```

Harga layanan ditentukan oleh kelas: ekonomi (0%), bisnis (+25%), VIP (+50%).

```
Transportasi[] daftarTransportasi = new Transportasi[3];  
daftarTransportasi[0] = new Bus(...);  
daftarTransportasi[1] = new Kereta(...);  
daftarTransportasi[2] = new Pesawat(...);
```

Polymorphism: array bertipe Transportasi dapat menampung semua objek sub

V. Kesimpulan

Berdasarkan hasil implementasi dan pengujian terhadap program sistem transportasi berbasis objek yang mencakup moda transportasi Bus, Kereta, dan Pesawat, dapat ditarik beberapa kesimpulan penting mengenai penerapan konsep Pemrograman Berorientasi Objek (PBO) dalam konteks simulasi nyata.

Pertama, program ini secara efektif mendemonstrasikan bagaimana pewarisan (inheritance) memungkinkan kita membuat struktur class yang efisien dan terorganisir. Superclass Transportasi berfungsi sebagai kerangka umum yang dapat diwarisi oleh berbagai jenis transportasi, sehingga kode dapat ditulis sekali namun digunakan berulang kali. Hal ini mendorong praktik code reusability dan mengurangi redundansi dalam kode program.

Kedua, penggunaan method overriding pada setiap subclass menunjukkan bahwa setiap moda transportasi memiliki logika perhitungan harga tiket yang unik. Bus, Kereta, dan Pesawat masing-masing meng-override method `hitungHargaTiket()` untuk menyesuaikan harga dasar sesuai jenis layanannya. Dengan pendekatan ini, program dapat memperlakukan semua transportasi dengan cara yang seragam, namun tetap menghormati perbedaan karakteristik internalnya.

Ketiga, method overloading juga dimanfaatkan untuk menangani variasi dalam permintaan pengguna, yakni ketika pengguna ingin memesan tiket berdasarkan kelas layanan seperti Ekonomi, Bisnis, atau VIP. Dengan menyediakan method `hitungHargaTiket(String kelasLayanan)`, sistem menjadi lebih fleksibel dalam menerima input, tanpa perlu membuat nama method baru yang membingungkan.

Selanjutnya, penerapan polymorphism adalah poin paling krusial dalam program ini. Dengan menyimpan semua objek transportasi ke dalam array bertipe `Transportasi[]`, program dapat memproses seluruh moda transportasi secara seragam meskipun objek yang dimasukkan memiliki implementasi method yang berbeda. Ini membuktikan bahwa polymorphism memegang peranan penting dalam menyederhanakan kompleksitas dan meningkatkan skalabilitas aplikasi.

Dari sisi praktik pengembangan, sistem ini menunjukkan bagaimana PBO tidak hanya mempermudah pengelolaan kode tetapi juga memungkinkan aplikasi dikembangkan lebih lanjut dengan cepat. Sebagai contoh, penambahan moda transportasi baru seperti Kapal Laut hanya memerlukan pembuatan subclass baru yang mewarisi Transportasi, tanpa perlu mengubah logika utama dalam `main()` atau struktur array.

Secara keseluruhan, studi kasus ini memperlihatkan bagaimana konsep-konsep inti PBO tidak hanya penting secara teori, tetapi juga sangat aplikatif dan relevan dalam membangun sistem perangkat lunak dunia nyata. Penggunaan inheritance, overriding, overloading, dan polymorphism dalam satu kerangka sistem menjadikan program ini contoh ideal dalam memahami dan menerapkan paradigma PBO pada level lanjut dan praktis.

VI. Daftar Pustaka

- Oracle Java Documentation -
<https://docs.oracle.com/javase/tutorial/java/index.html>
- GeeksForGeeks. (2022). Object-Oriented Programming in Java.