**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 9**

***Overriding and Polymorphizm***

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Gerald Eberhard**

**(105223002)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**UNIVERSITAS PERTAMINA**

**2025**

1. **Pendahuluan**

Praktikum ini bertujuan untuk mengembangkan sistem pemesanan tiket transportasi sederhana menggunakan konsep pemrograman berorientasi objek (PBO) dalam bahasa pemrograman Java. Sistem ini dirancang untuk mengelola pemesanan tiket untuk tiga jenis transportasi: bus, kereta, dan pesawat. Setiap jenis transportasi memiliki informasi umum seperti nama, jumlah kursi, dan tujuan, serta perhitungan harga tiket yang berbeda berdasarkan jenis transportasi dan kelas layanan (Ekonomi, Bisnis, atau VIP). Konsep PBO yang diterapkan meliputi pewarisan, overriding, overloading, dan polimorfisme, yang memungkinkan pembuatan sistem yang modular, fleksibel, dan mudah dikembangkan. Tujuan utama praktikum ini adalah untuk memahami dan mengimplemen tasikan konsep-konsep tersebut dalam konteks nyata.

1. **Variabel**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Variabel** | **Tipe Data** | **Fungsi** |
| 1 | t | Transportasi | Variabel perulangan dalam loop for-each yang digunakan untuk mengakses setiap elemen dalam transportasiArray. |
| 2 | kelasLayanan | String | Variabel sementara untuk menyimpan nama kelas layanan (Ekonomi, Bisnis, VIP) yang digunakan dalam perhitungan harga tiket. |
| 3 | transportasiArray | Transportasi[] | Array yang menyimpan objek dari kelas Transportasi (termasuk Bus, Kereta, Pesawat) untuk implementasi polimorfisme. |
| 4 | pesawat | Transportasi[] | Objek dari kelas Pesawat yang mewakili transportasi pesawat dengan properti tertentu (nama, jumlahKursi, tujuan). |
| 5 | kereta | Transportasi[] | Objek dari kelas Kereta yang mewakili transportasi kereta dengan properti tertentu (nama, jumlahKursi, tujuan). |
| 6 | bus | Transportasi | Objek dari kelas Bus yang mewakili transportasi bus dengan properti tertentu (nama, jumlahKursi, tujuan). |
| 7 | Nama | String | Variabel yang diwarisi dari kelas Transportasi, menyimpan nama pesawat (misalnya, "Pesawat Air"). |
| 8 | jumlahKursi | int | Variabel yang diwarisi dari kelas Transportasi, menyimpan jumlah kursi yang tersedia pada pesawat (misalnya, 120). |
| 9 | tujuan | String | Variabel yang diwarisi dari kelas Transportasi, menyimpan tujuan pesawat (misalnya, "Medan"). |
| 10 | kelasLayanan | String | Parameter metode hitungHargaTiket(String kelasLayanan), menyimpan kelas layanan yang dipilih (Ekonomi, Bisnis, atau VIP). |
| 11 | basePrice | Double | Variabel lokal dalam metode hitungHargaTiket(String kelasLayanan), menyimpan harga dasar setelah override (150.000 untuk Pesawat). |
| 12 | nama | String | Menyimpan nama transportasi (misalnya, "Bus Trans"). |
| 13 | jumlahKursi | int | Menyimpan jumlah kursi yang tersedia pada transportasi. |
| 14 | tujuan | String | Menyimpan tujuan transportasi (misalnya, "Bandung"). |
| 15 | nama | String | Variabel yang diwarisi dari Transportasi, menyimpan nama bus (misalnya, "Bus Trans"). |
| 16 | jumlahKursi | int | Variabel yang diwarisi dari Transportasi, menyimpan jumlah kursi bus |
| 17 | tujuan | String | Variabel yang diwarisi dari Transportasi, menyimpan tujuan bus. |
| 18 | kelasLayanan | String | Parameter metode hitungHargaTiket(String kelasLayanan), menyimpan kelas layanan (Ekonomi, Bisnis, atau VIP). |
| 19 | basePrice | Double | Variabel lokal dalam metode hitungHargaTiket(String kelasLayanan), menyimpan harga dasar setelah override (110.000 untuk Bus). |
| 20 | nama | String | Variabel yang diwarisi dari Transportasi, menyimpan nama kereta (misalnya, "Kereta Cepat"). |
| 21 | jumlahKursi | int | Variabel yang diwarisi dari Transportasi, menyimpan jumlah kursi kereta. |
| 22 | tujuan | String | Variabel yang diwarisi dari Transportasi, menyimpan tujuan kereta. |
| 23 | kelasLayanan | String | Parameter metode hitungHargaTiket(String kelasLayanan), menyimpan kelas layanan (Ekonomi, Bisnis, atau VIP). |
| 24 | basePrice | double | Variabel lokal dalam metode hitungHargaTiket(String kelasLayanan), menyimpan harga dasar setelah override (120.000 untuk Kereta). |

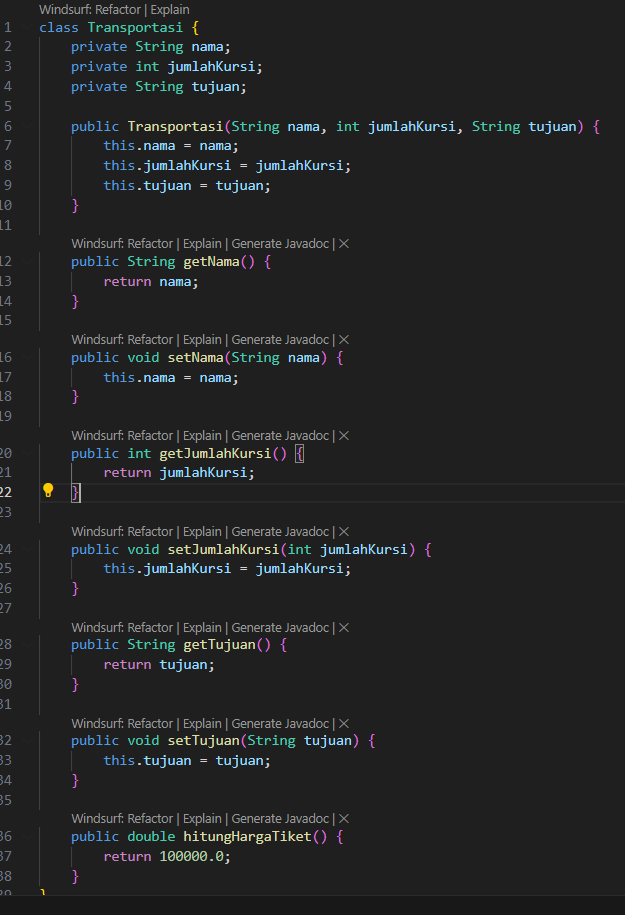
1. **Constructor dan Method**

|  |  |  |
| --- | --- | --- |
| **Nama Method** | **Jenis Method** | **Fungsi** |
| Transportasi(String nama, int jumlahKursi, String tujuan) | Constructor | Menginisialisasi objek Transportasi dengan properti nama, jumlahKursi, dan tujuan |
| getNama() | Method | Mengembalikan nilai nama. |
| setNama(String nama) | Method | Mengatur nilai nama. |
| getJumlahKursi() | Method | Mengembalikan nilai jumlahKursi. |
| setJumlahKursi(int jumlahKursi) | Method | Mengatur nilai jumlahKursi. |
| getTujuan() | Method | Mengembalikan nilai tujuan. |
| setTujuan(String tujuan) | Method | Mengatur nilai tujuan. |
| hitungHargaTiket() | Method | Mengembalikan harga tiket default (100.000). |
| @Override hitungHargaTiket() | Method | Mengembalikan harga tiket untuk Bus (100.000 + 10% = 110.000). |
| hitungHargaTiket(String kelasLayanan) | Method | Mengembalikan harga tiket berdasarkan kelas layanan (Ekonomi: +0%, Bisnis: +25%, VIP: +50%). |
| Bus(String nama, int jumlahKursi, String tujuan) | Constructor | Mengin isialisasi objek Bus dengan memanggil konstruktor Transportasi. |
| Kereta(String nama, int jumlahKursi, String tujuan) | Constructor | Mengin isialisasi objek Kereta dengan memanggil konstruktor Transportasi. |
| @Override hitungHargaTiket() | Method | Mengembalikan harga tiket untuk Kereta (100.000 + 20% = 120.000). |
| hitungHargaTiket(String kelasLayanan) | Method | Mengembalikan harga tiket berdasarkan kelas layanan (Ekonomi: +0%, Bisnis: +25%, VIP: +50%). |
| Pesawat(String nama, int jumlahKursi, String tujuan) | Cosntructor | Mengin isialisasi objek Pesawat dengan memanggil konstruktor Transportasi. |
| @Override hitungHargaTiket() | Method | Mengembalikan harga tiket untuk Pesawat (100.000 + 50% = 150.000). |
| hitungHargaTiket(String kelasLayanan) | Method | Mengembalikan harga tiket berdasarkan kelas layanan (Ekonomi: +0%, Bisnis: +25%, VIP: +50%). |
| main(String[] args) | Method | Metode utama untuk menjalankan program, membuat objek transportasi, dan menampilkan informasi serta harga tiket. |

1. **Dokumentasi dan Pembahasan Code**Program ini terdiri dari empat kelas utama: Transportasi, Bus, Kereta, Pesawat, dan Main. Berikut adalah penjelasan rinci tentang implementasi kode:

* Kelas Transportasi

Kelas Transportasi adalah kelas induk yang menyediakan kerangka dasar untuk se mua jenis transportasi. Properti nama, jumlahKursi, dan tujuan dideklarasikan sebagai private untuk menerapkan enkapsulasi, dengan metode getter dan setter untuk mengakses dan mengubah nilai properti. Metode hitungHargaTiket() mengembalikan harga tiket default sebesar 100.000.



* Kelas Bus, Kereta, dan Pesawat

Ketiga kelas ini adalah subclass dari Transportasi, masing-masing mengoverride metode hitungHargaTiket() untuk menghitung harga tiket sesuai aturan:

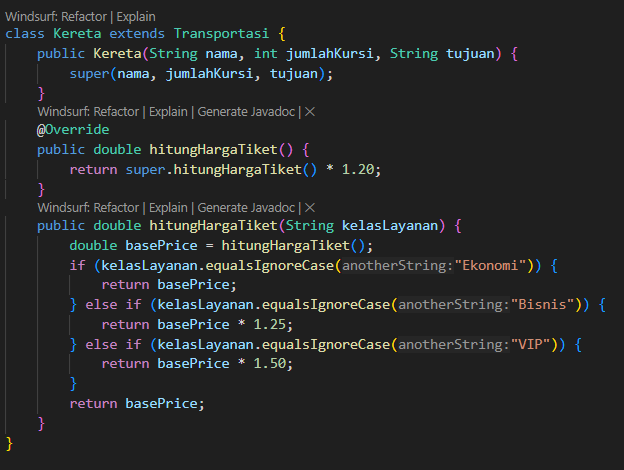
• Bus: Menambah 10% dari harga default (100.000\*1.10=110.000).

• Kereta: Menambah 20% dari harga default (100.000 \* 1.20 = 120.000). • Pesawat: Menambah 50% dari harga default (100.000 \* 1.50 = 150.000). Selain itu, setiap subclass memiliki metode overload hitungHargaTiket(String kelasLayanan) yang menyesuaikan harga berdasarkan kelas layanan: • Ekonomi: +0%. • Bisnis: +25%. • VIP: +50%. Berikut adalah contoh implementasi untuk kelas Bus:

Sebuah gambar berisi teks, cuplikan layar, software

Konten yang dihasilkan AI mungkin salah.

Berikut kelas Kereta:



Berikut Kelas Pesawat:   
Sebuah gambar berisi teks, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

* Kelas Main

Kelas Main berfungsi untuk menjalankan program. Objek dari Bus, Kereta, dan Pesawat dibuat dan disimpan dalam array bertipe Transportasi[]. Array ini digunakan untuk menunjukkan polimorfisme, di mana metode hitungHargaTiket() dipanggil secara di namis pada setiap objek. Untuk metode overload, pemeriksaan jenis (instanceof) dan casting digunakan untuk memanggil metode yang sesuai.

Sebuah gambar berisi teks, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

1. **Kesimpulan**

Praktikum ini berhasil menunjukkan penerapan konsep pewarisan, overriding, overloading, dan polimorfisme dalam pemrograman berorientasi objek menggunakan Java. Sistem pemesanan tiket transportasi yang dibuat memungkinkan pengelolaan tiket untuk bus, kereta, dan pesawat dengan struktur kode yang terorganisir dan modular. Pewarisan memungkinkan penggunaan kembali kode dari kelas induk Transportasi, semen tara overriding dan overloading memberikan fleksibilitas dalam perhitungan harga tiket.

Polimorfisme memungkinkan penanganan objek secara seragam melalui array, sehingga mempermudah pengembangan dan perawatan sistem. Praktikum ini memberikan pemahaman mendalam tentang konsep PBO dan penerapannya dalam kasus nyata. Sekian dan terima kasih.

1. **Daftar Pustaka**

Modul 9: Java Inheritence (Pewarisan) 2