**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IX**

**OVERRIDING dan POLYMORPHISM**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Adri Hidayat 105223046**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**UNIVERSITAS PERTAMINA**

**2025**

1. **Pendahuluan**

Perkembangan teknologi dan kebutuhan hiburan digital mendorong pengembangan berbagai jenis permainan komputer, termasuk game berbasis teks yang sederhana namun menarik untuk dipelajari. Game berbasis teks memungkinkan interaksi tanpa grafis kompleks, sehingga cocok sebagai media pembelajaran dasar pemrograman dan logika game. Proyek ini bertujuan membuat game petualangan berbasis teks dengan fitur registrasi, login, serta mekanisme eksplorasi, pertarungan, dan pengelolaan status pemain menggunakan bahasa pemrograman Java. Dengan struktur modular yang memisahkan tanggung jawab tiap kelas, pengembangan dan pemeliharaan kode menjadi lebih terorganisir. Melalui proyek ini diharapkan dapat memahami konsep pemrograman berorientasi objek, interaksi input-output teks, serta penerapan logika permainan sederhana namun fungsional.

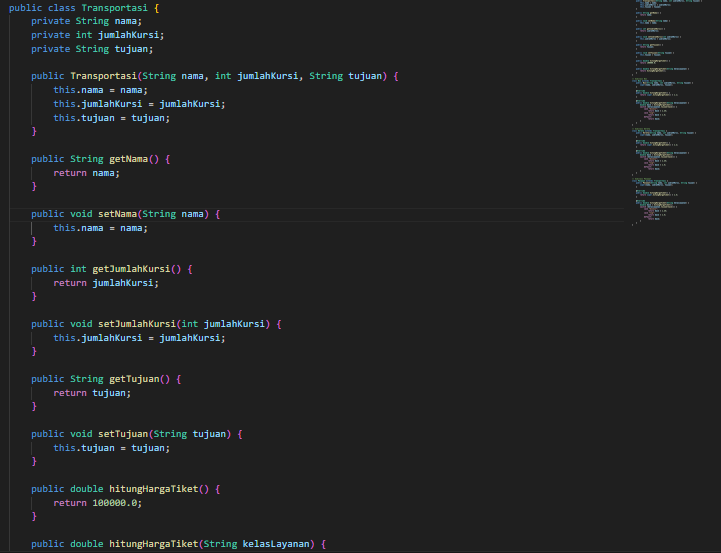
1. **Variabel**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Variabel | Tipe Data | Fungsi |
| 1 | nama | String | Menyimpan nama transportasi |
| 2 | jumlahKursi | int | Menyimpan jumlah kursi yang tersedia |
| 3 | tujuan | String | Menyimpan tujuan transportasi |
| 4 | daftarTransportasi | Transportasi[] | Array yang menyimpan objek transportasi (Bus, Kereta, Pesawat) |
| 5 | kelasLayanan | String[] | Array untuk menyimpan daftar kelas layanan (ekonomi, bisnis, VIP) |
| 6 | base | double | Menyimpan harga dasar sebelum ditambah biaya kelas layanan |

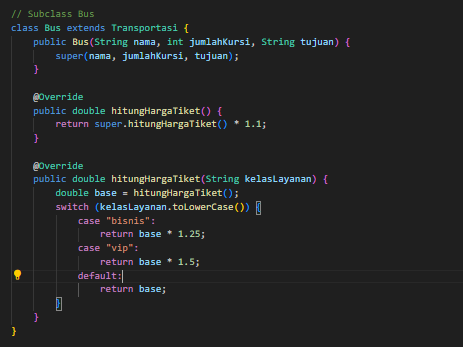
1. **Constructor dan Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Metode | Jenis Metode | Fungsi |
| 1 | Transportasi(...) | Konstruktor | Menginisialisasi objek Transportasi dengan nama, jumlah kursi, dan tujuan |
| 2 | getNama() | Getter | Mengambil nilai nama |
| 3 | setNama(String nama) | Setter | Mengatur nilai nama |
| 4 | getJumlahKursi() | Getter | Mengambil nilai jumlahKursi |
| 5 | setJumlahKursi(int jumlahKursi) | Setter | Mengatur nilai jumlahKursi |
| 6 | getTujuan() | Getter | Mengambil nilai tujuan |
| 7 | setTujuan(String tujuan) | Setter | Mengatur nilai tujuan |
| 8 | hitungHargaTiket() | Method | Menghitung harga tiket dasar (Rp 100.000) |
| 9 | hitungHargaTiket(String) | Overloaded Method | Menghitung harga tiket berdasarkan kelas layanan |
| 10 | Bus(...) | Konstruktor | Membuat objek Bus dengan parameter yang diwariskan |
| 11 | Kereta(...) | Konstruktor | Membuat objek Kereta dengan parameter yang diwariskan |
| 12 | Pesawat(...) | Konstruktor | Membuat objek Pesawat dengan parameter yang diwariskan |
| 13 | main(String[] args) | Main Method | Titik masuk program, menampilkan info transportasi dan harga tiket |

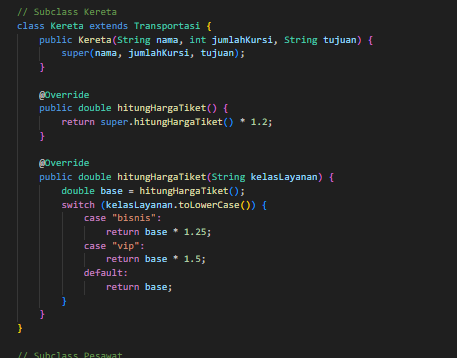
1. **Dokumentasi dan Pembahasan Code**

****

Kelas Transportasi adalah kelas induk yang merepresentasikan transportasi secara umum. Kelas ini memiliki tiga properti utama: nama, jumlahKursi, dan tujuan, yang semuanya bersifat private untuk menjaga prinsip enkapsulasi. Kelas ini menyediakan konstruktor, serta method getter dan setter untuk setiap atribut. Selain itu, kelas ini memiliki dua metode hitungHargaTiket(), satu tanpa parameter yang mengembalikan harga dasar Rp100.000, dan satu lagi versi overload dengan parameter kelasLayanan, yang dapat di-override oleh subclass untuk menyesuaikan harga berdasarkan jenis layanan.

****

Kelas Bus adalah subclass dari Transportasi yang mewakili jenis transportasi bus. Kelas ini meng-override metode hitungHargaTiket() untuk menambahkan biaya sebesar 10% dari harga dasar. Selain itu, kelas ini juga mengimplementasikan metode overload hitungHargaTiket(String kelasLayanan) yang akan menambah harga berdasarkan kelas: tidak ada tambahan untuk "ekonomi", 25% untuk "bisnis", dan 50% untuk "VIP". Perhitungan ini dilakukan dengan memanfaatkan hasil dari metode dasar yang telah dioverride.

****

Kelas Kereta merupakan turunan dari kelas Transportasi yang menggambarkan transportasi kereta. Override dilakukan pada metode hitungHargaTiket() untuk menambahkan 20% dari harga dasar. Seperti kelas Bus, kelas ini juga memiliki metode overload hitungHargaTiket(String kelasLayanan) untuk menentukan tambahan biaya berdasarkan kelas layanan. Fungsionalitas ini membuat harga tiket lebih fleksibel dan sesuai dengan layanan yang dipilih oleh pengguna.

****

Pesawat adalah subclass dari Transportasi yang merepresentasikan moda transportasi udara. Kelas ini meng-override metode hitungHargaTiket() dengan menambahkan 50% dari harga dasar. Dalam versi overload-nya, hitungHargaTiket(String kelasLayanan) memungkinkan perhitungan harga berdasarkan layanan yang dipilih, sama seperti kelas lain: "ekonomi" (tanpa tambahan), "bisnis" (tambahan 25%), dan "VIP" (tambahan 50%). Ini menjadikan kelas Pesawat fleksibel dalam mendukung berbagai kelas pelayanan.

****

Kelas Main merupakan titik masuk (entry point) dari program. Di dalamnya, dibuat array Transportasi[] yang menyimpan objek dari kelas Bus, Kereta, dan Pesawat. Dengan memanfaatkan konsep polimorfisme, program memanggil metode hitungHargaTiket() secara dinamis pada setiap objek dalam array. Kelas ini juga menampilkan informasi transportasi serta hasil perhitungan harga tiket untuk setiap jenis layanan: ekonomi, bisnis, dan VIP. Tujuan utama kelas ini adalah untuk mendemonstrasikan interaksi antar objek dan penggunaan polimorfisme dalam pewarisan kelas.

1. **Kesimpulan**

Dalam proyek ini telah dibuat sebuah sistem pemesanan tiket transportasi sederhana yang menerapkan konsep-konsep dasar Pemrograman Berorientasi Objek (OOP) seperti enkapsulasi, pewarisan, polimorfisme, dan overriding/overloading method. Kelas induk Transportasi digunakan sebagai dasar untuk menyimpan informasi umum dari setiap jenis transportasi seperti nama, jumlah kursi, dan tujuan. Tiga subclass yaitu Bus, Kereta, dan Pesawat dibuat untuk merepresentasikan masing-masing moda transportasi, dengan metode hitungHargaTiket() yang dioverride sesuai perhitungan harga masing-masing moda.

1. **Daftar Pustaka**

* Modul 9 Praktikum PBO