**TUGAS PERTEMUAN KE - 9**

**(PRAKTIKUM OOP 2024-2025)**

|  |  |
| --- | --- |
| **NAMA** | Muhammad Farid Fadillah |
| **NIM** | 2318018 |
| **KELAS** | C |
| **PEMBERI TUGAS** | ALRIZKY PUTRA DHANDI (2318116) |

## Tugas Rumah 1 :

Judul : Sistem Dealer Motor

Tema : Transportasi

*Source code Object Class/Abstact*(Dealermotor):

|  |
| --- |
| package BAB9.Interface;  public abstract class Dealermotor {      // Atribut private untuk enkapsulasi      private String merk, warna, tahunMasuk, tipeBahanBakar;      private int harga, jumlah;      // Konstruktor untuk inisialisasi objek      public Dealermotor(String merk, String warna, String tahunMasuk, String tipeBahanBakar, int harga, int jumlah) {          this.merk = merk;          this.warna = warna;          this.tahunMasuk = tahunMasuk;          this.tipeBahanBakar = tipeBahanBakar;          this.harga = harga;          this.jumlah = jumlah;      }      // Getter untuk setiap atribut      public String getMerk() { return merk; }      public String getWarna() { return warna; }      public String getTahunMasuk() { return tahunMasuk; }      public String getTipeBahanBakar() { return tipeBahanBakar; }      public int getHarga() { return harga; }      public int getJumlah() { return jumlah; }      // Setter untuk jumlah      public void setJumlah(int jumlah) { this.jumlah = jumlah; }      // Method untuk menampilkan data motor      public void tampilkanData() {          System.out.println("---- Data Motor Dealer ----");          System.out.println("Merk            : " + merk);          System.out.println("Warna           : " + warna);          System.out.println("Tahun Masuk     : " + tahunMasuk);          System.out.println("Bahan Bakar     : " + tipeBahanBakar);          System.out.println("Harga           : Rp " + harga);          System.out.println("Stok Tersedia   : " + jumlah);      }      // Overloading method: menampilkan data dengan peringatan jika stok minim      public void tampilkanData(int stokMinimal) {          tampilkanData();          if (jumlah < stokMinimal) {              System.out.println("\*\* Perhatian: Stok menipis! \*\*");          }      }      // Method untuk menambah stok      public void tambahStok(int tambahan) {          jumlah += tambahan;          System.out.println("Stok ditambah: " + tambahan + " unit. Total stok: " + jumlah);      }      // Method untuk mengurangi stok saat penjualan      public void kurangiStok(int jumlahTerjual) {          if (jumlahTerjual > jumlah) {              System.out.println("Stok tidak mencukupi!");          } else {              jumlah -= jumlahTerjual;              System.out.println("Terjual " + jumlahTerjual + " unit. Sisa stok: " + jumlah);          }      }      // Method abstract (akan diimplementasikan oleh subclass)      public abstract double hitungDiskon(int jumlahUnit);      public abstract void tampilkanPromo();      // Overload method untuk promo dengan nama bulan      public void tampilkanPromo(String bulan) {          System.out.println("Promo bulan " + bulan + " akan segera diumumkan!");      }  } |

Analisa:

*Class* Dealermotor.java adalah kelas induk yang berisi method-method untuk menampilkan data-data yang ada di sistem dealer motor ini. Di dalam kelas ini terdapat counstractor, overloading dan juga overloading.

*Source code Iterfacet*(PromoDealer):

|  |
| --- |
| public interface PromoDealer {      // Method untuk menampilkan promo      void tampilkanPromo();      // Method untuk menghitung diskon berdasarkan jumlah unit      double hitungDiskon(int jumlahUnit);  } |

Analisa:

Interface PromoDealer dirancang untuk mendefinisikan kontrak yang wajib diimplementasikan oleh kelas-kelas yang menyediakan fitur promosi dalam konteks dealer motor. Dengan mendeklarasikan method tampilkanPromo() dan hitungDiskon(int jumlahUnit), interface ini memastikan bahwa setiap kelas yang mengimplementasikannya memiliki kemampuan untuk menampilkan informasi promo secara konsisten dan menghitung diskon berdasarkan jumlah unit yang dibeli.

*Source code Object Class/Abstact*(Yamaha.java):

|  |
| --- |
| package BAB9.Interface;  public class Yamaha extends Dealermotor implements PromoDealer {      public Yamaha(String merk, String warna, String tahunMasuk, String bahanBakar, int harga, int jumlah) {          super(merk, warna, tahunMasuk, bahanBakar, harga, jumlah);      }      @Override      public void tampilkanPromo() {          System.out.println("\uD83D\uDD25 PROMO YAMAHA MINGGU INI \uD83D\uDD25");          System.out.println("- Diskon 7% untuk pembelian 2 unit atau lebih");          System.out.println("- Bonus jaket exclusive");      }      @Override      public double hitungDiskon(int jumlahUnit) {          return (jumlahUnit >1) ? 0.07 : 0.0;      }  } |

Analisa:

Kelas Yamaha yang berada dalam package BAB9.Interface mengimplementasikan interface PromoDealer, sehingga kelas ini wajib menyediakan implementasi method tampilkanPromo() dan hitungDiskon(int jumlahUnit).

*Source code Object Class/Abstact*(Honda.java):

|  |
| --- |
| package BAB9.Interface;  public class Honda extends Dealermotor implements PromoDealer {      public Honda(String merk, String warna, String tahunMasuk, String bahanBakar, int harga, int jumlah) {          super(merk, warna, tahunMasuk, bahanBakar, harga, jumlah);      }      @Override      public void tampilkanPromo() {          System.out.println("\uD83C\uDF89 PROMO HONDA BULAN INI \uD83C\uDF89");          System.out.println("- Diskon 10% untuk pembelian 2 unit atau lebih");          System.out.println("- Gratis servis & oli 3x");      }      @Override      public double hitungDiskon(int jumlahUnit) {          return (jumlahUnit > 1) ? 0.10 : 0.0;      }  } |

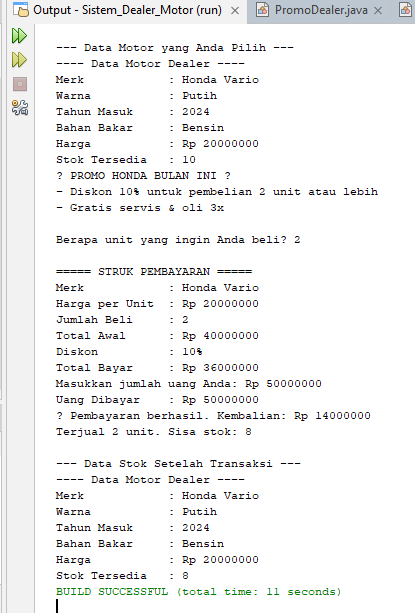
Analisa:

Kelas Honda dalam package BAB9.Interface mengimplementasikan interface PromoDealer, yang memaksa kelas ini untuk menyediakan metode tampilkanPromo() dan hitungDiskon(int jumlahUnit) sesuai kontrak interface.

*Source code Object Class/Abstact*(Main.java):

|  |
| --- |
| package BAB9.Interface;  import java.util.Scanner;  public class Main {      public static void main(String[] args) {          Scanner input = new Scanner(System.in);          // Membuat objek Honda dan Yamaha dengan tipe induk          Dealermotor honda = new Honda("Honda Vario", "Putih", "2024", "Bensin", 20000000, 10);          Dealermotor yamaha = new Yamaha("Yamaha NMAX", "Hitam", "2023", "Bensin", 30000000, 5);          System.out.println("===== SELAMAT DATANG DI DEALER MOTOR =====");          System.out.println("1. Honda Vario");          System.out.println("2. Yamaha NMAX");          System.out.print("Pilih motor yang ingin dibeli (1/2): ");          int pilihan = input.nextInt();          Dealermotor motorDipilih;          if (pilihan == 1) {              motorDipilih = honda;          } else if (pilihan == 2) {              motorDipilih = yamaha;          } else {              System.out.println("❌ Pilihan tidak valid.");              input.close();              return;          }          // Tampilkan detail dan promo motor          System.out.println("\n--- Data Motor yang Anda Pilih ---");          motorDipilih.tampilkanData();          motorDipilih.tampilkanPromo();          // Input jumlah beli          System.out.print("\nBerapa unit yang ingin Anda beli? ");          int jumlah = input.nextInt();          // Proses transaksi          PembayaranDealer transaksi = new PembayaranDealer(motorDipilih, jumlah);          transaksi.tampilkanTagihan();          // Input uang pembayaran          System.out.print("Masukkan jumlah uang Anda: Rp ");          int uang = input.nextInt();          transaksi.bayar(uang);          // Tampilkan stok setelah transaksi          System.out.println("\n--- Data Stok Setelah Transaksi ---");          motorDipilih.tampilkanData(3); // peringatan jika stok di bawah 3          input.close();      }  } |

Hasil Tampilan :



Gambar 9.1 Tampilan Hasil CetakDataOba

Analisa:

Kelas Main dalam package BAB9.Interface menunjukkan penerapan nyata dari konsep *polimorfisme* dan *interface* dalam program dealer motor. Dengan membuat objek Honda dan Yamaha menggunakan tipe induk Dealermotor, program mampu menangani berbagai jenis motor secara dinamis tanpa bergantung pada tipe konkret spesifik. Meskipun Dealermotor adalah kelas induk abstrak, metode tampilkanPromo() dan hitungDiskon() yang berasal dari interface PromoDealer tetap dapat dipanggil melalui objek motor yang dipilih, menunjukkan fleksibilitas polimorfisme melalui interface. Alur program juga memisahkan proses input, tampilan data, proses pembayaran, dan pengelolaan stok secara terstruktur, sehingga meningkatkan modularitas dan keterbacaan kode.

## Kesimpulan

* + - 1. *Interface* merupakan *prototype* atau *template* untuk sebuah *class*, deskripsi ini hampir serupa dengan *class* *abstrak*.
      2. Keuntungan membuat *interface* sendiri adalah menutupi kekurangan pada java yang hanya memperbolehkan satu kelas saja yang berhak mendapatkan warisan kelas induk (*extends*).
      3. Isi dari *interface* terdiri dari deklarasi metode yang bersifat *abstrak*, yaitu metode yang hanya didefinisikan tanpa isi (badan metode).