

Laporan Praktikum Pemrograman Berorientasi Objek – Pertemuan 7

Dibuat oleh : NindyaAlif Romland

NIM : H1D024031

Shift Baru : I

Shift KRS : G

Deskripsi Program

Program ini mensimulasikan pembayaran menggunakan e-wallet dengan memanfaatkan interface sebagai kontrak perilaku. Interface PaymentMethod menentukan fungsi-fungsi dasar seperti memproses pembayaran, menampilkan detail, dan mengecek saldo. Class EWalletPayment mengisi semua fungsi tersebut, termasuk mengecek apakah saldo cukup dan menguranginya saat transaksi dilakukan. Saat program dijalankan, sistem memproses pembayaran dan menampilkan hasil transaksi beserta sisa saldo pengguna.

Alur Kerja Program

Program dimulai dengan membuat objek EWalletPayment yang memiliki saldo awal, lalu sistem memanggil method tampilkanDetail() untuk menampilkan informasi e-wallet. Ketika transaksi dilakukan, program memanggil prosesPembayaran(), lalu di dalamnya sistem mengecek apakah saldo mencukupi melalui cekSaldo(). Jika cukup, program menghitung biaya admin dengan hitungBiayaAdmin(), kemudian mengurangi saldo melalui kurangiSaldo(), dan menampilkan hasil akhir transaksi. Jika saldo tidak cukup, proses pembayaran dibatalkan dan pesan gagal ditampilkan. Program diakhiri dengan menampilkan saldo terbaru setelah transaksi.

Fungsi Yang Digunakan

- Interface
 - 1. processPayment()
Memproses pembayaran dan menentukan apakah transaksi berhasil atau gagal.
 - 2. getPaymentDetails()
Mengembalikan informasi detail transaksi (misalnya: “Pembayaran dilakukan melalui OVO”).
 - 3. getTransaction()
Mengembalikan biaya transaksi (misalnya Rp 2.000).
 - 4. getBalance()
Mengembalikan saldo e-wallet saat ini.

- Class
 1. processPayment()

Mengecek kecukupan saldo, mengurangi saldo jika cukup, dan menampilkan pesan "Pembayaran berhasil!" atau "Saldo tidak cukup!"
 2. getPaymentDetails()

Mengembalikan nama layanan e-wallet yang digunakan.
 3. getTransaction()

Mengembalikan biaya admin tetap (misalnya Rp 2.000).
 4. getBalance()

Mengembalikan nilai saldo terbaru.
- Main
 1. Main()

Membuat objek EwalletPayment, menampilkan saldo awal, memanggil processPayment(), menampilkan saldo akhir, dan menampilkan detail transaksi.

Hasil Output

```
== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ==
Saldo awal: 150000.0
Memproses pembayaran sebesar 50000.0...

Pembayaran berhasil!
Sisa saldo: 98000.0
Detail Transaksi: Pembayaran dilakukan melalui OVO

Process finished with exit code 0
```

```
public class EWalletPayment implements PaymentMethod { 2 usages
    private String layanan; 2 usages
    private double saldo; 5 usages
    private double nominal; 3 usages
    private final double fee = 2000; 2 usages

    public EWalletPayment(String layanan, double saldo, double nominal) { 1 usage
        this.layanan = layanan;
        this.saldo = saldo;
        this.nominal = nominal;
    }

    @Override 1 usage
    public void processPayment() {
        System.out.println("Memproses pembayaran sebesar " + nominal + "...");

        double total = nominal + fee;

        if (saldo >= total) {
            saldo -= total;
            System.out.println("\nPembayaran berhasil!");
            System.out.println("Sisa saldo: " + saldo);
        } else {
            System.out.println("\nPembayaran gagal! Saldo tidak cukup.");
        }
    }
}
```

```
@Override 1 usage
public String getPaymentDetails() {
    return "Pembayaran dilakukan melalui " + layanan;
}

@Override no usages
public double getTransactionFee() {
    return fee;
}

@Override 1 usage
public double getBalance() {
    return saldo;
}

}
```

```
public interface PaymentMethod { 1 usage 1 implementation
    void processPayment(); 1 usage 1 implementation
    String getPaymentDetails(); 1 usage 1 implementation
    double getTransactionFee(); no usages 1 implementation
    double getBalance(); 1 usage 1 implementation
}
```

```
public class PaymentTest {
    public static void main(String[] args) {
        System.out.println("==== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===");

        // buat objek e-wallet
        EWalletPayment pembayaran = new EWalletPayment(layanan, "OVO", saldo: 150000, nominal: 50000);

        // saldo awal
        System.out.println("Saldo awal: " + pembayaran.getBalance());

        // proses pembayaran
        pembayaran.processPayment();

        // tampilkan detail transaksi
        System.out.println("Detail Transaksi: " + pembayaran.getPaymentDetails());
    }
}
```