

# LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

## Pertemuan 7

Nama: Aditya Alfandy

NIM: H1D024103

### A. Alur Kerja Program

Program ini mensimulasikan sistem pembayaran elektronik sederhana menggunakan konsep Interface pada Java. Alur kerja program adalah sebagai berikut:

#### 1. Inisialisasi Objek:

Program dimulai di class PaymentTest (Main Class). Langkah pertama adalah membuat objek dari class EWalletPayment. Saat objek dibuat, constructor dipanggil untuk menginisialisasi atribut:

- Nama Layanan: "OVO"
- Saldo Awal: Rp 150.000
- Nominal Transaksi yang akan dilakukan: Rp 50.000

#### 2. Pengecekan Data Awal:

Program memanggil method `getBalance()` untuk menampilkan saldo awal pengguna ke layar konsol sebelum transaksi dilakukan.

#### 3. Pemrosesan Pembayaran:

Program memanggil method `processPayment()`. Di dalam method ini terjadi logika utama:

- Program menghitung total biaya = Nominal Transaksi + Biaya Admin (Fee).
- Program mengecek apakah saldo  $\geq$  total biaya.
- Jika Cukup: Saldo dikurangi total biaya, dan pesan "Pembayaran berhasil!" ditampilkan.
- Jika Tidak Cukup: Saldo tidak berubah, dan pesan "Pembayaran gagal!" ditampilkan.

#### 4. Pelaporan Akhir:

Setelah proses pembayaran selesai, program kembali menampilkan sisa saldo terbaru menggunakan `getBalance()` dan menampilkan detail layanan pembayaran menggunakan `getPaymentDetails()`.

### B. Fungsi dan Method yang Digunakan

Berikut adalah penjelasan fungsi (method) yang diimplementasikan dari interface `PaymentMethod` ke dalam class `EWalletPayment` sesuai spesifikasi<sup>1</sup>:

No	Nama Method	Tipe Return	Fungsi / Kegunaan
1.	processPayment()	void	Melakukan logika inti transaksi. Method ini memvalidasi apakah saldo mencukupi untuk membayar nominal transaksi ditambah biaya admin. Jika valid, saldo akan dikurangi secara otomatis.
2.	getPaymentDetails()	String	Mengembalikan informasi berupa <i>string</i> yang berisi nama layanan E-Wallet yang digunakan saat transaksi (contoh: "Pembayaran dilakukan melalui OVO").
3.	getTransactionFee()	double	Mengembalikan nilai biaya administrasi transaksi. Pada kasus ini, biaya diset sebesar <b>2000.0</b> .
4.	getBalance()	double	Mengembalikan nilai saldo terakhir yang tersimpan dalam atribut objek saat ini.

## C. Implementasi Kode (Source Code)

### 1. File: PaymentMethod.java (Interface)

```
public interface PaymentMethod {

    void processPayment();

    String getPaymentDetails();

    double getTransactionFee();

    double getBalance();

}
```

### 2. File: EWalletPayment.java (Class Implementation)

```
public class EWalletPayment implements PaymentMethod {
    private String namaLayanan;
    private double saldo;
    private double nominalTransaksi;

    public EWalletPayment(String namaLayanan, double saldo, double nominalTransaksi) {
        this.namaLayanan = namaLayanan;
        this.saldo = saldo;
        this.nominalTransaksi = nominalTransaksi;
    }

    @Override
    public double getTransactionFee() {
        return 2000.0;
    }
}
```

```

    }

    @Override
    public void processPayment() {
        double totalBiaya = nominalTransaksi + getTransactionFee();
        if (saldo >= totalBiaya) {
            saldo -= totalBiaya;
            System.out.println("Pembayaran berhasil!");
        } else {
            System.out.println("Pembayaran gagal! Saldo tidak cukup.");
        }
    }

    @Override
    public String getPaymentDetails() {
        return "Detail Transaksi: Pembayaran dilakukan melalui " + namaLayanan;
    }

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

```

### 3. File: PaymentTest.java (Main Class)

```

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

        EWalletPayment ovo = new EWalletPayment("OVO", 150000, 50000);

        System.out.println("Saldo awal: " + (int)ovo.getBalance());
        System.out.println("Memproses pembayaran sebesar 50000...");

        ovo.processPayment();

        System.out.println("Sisa saldo: " + (int)ovo.getBalance());
        System.out.println(ovo.getPaymentDetails());
    }
}

```

### D. Hasil Output Program

```
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===  
Saldo awal: 150000  
Memproses pembayaran sebesar 50000...  
Pembayaran berhasil!  
Sisa saldo: 98000  
Detail Transaksi: Pembayaran dilakukan melalui OVO
```