

## Pert 7

### Alur kerja

1. Membuat interface PaymentMethod yang mendefinisikan kontrak fungsi pembayaran (processPayment, getPaymentDetails, getTransactionFee, getBalance).
2. Membuat class EWalletPayment yang mengimplementasikan semua method dari PaymentMethod dan menambahkan atribut khusus (nama layanan, saldo, nominal pembayaran).
3. Menambahkan logika pengecekan saldo di method processPayment untuk menentukan transaksi berhasil atau gagal.
4. Membuat class PaymentTest untuk membuat objek EWalletPayment lalu menguji seluruh method.
5. Program menampilkan rangkaian proses pembayaran, perubahan saldo, dan detail transaksi.

```
1 package pert7;
2
3 public class PaymentTest {
4     public static void main(String[] args) {
5         System.out.println("== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===");
6
7         double saldoAwal = 150000;
8         double nominal = 50000;
9
10        EWalletPayment transaksi = new EWalletPayment("OVO", saldoAwal, nominal);
11
12        System.out.println("Saldo awal: " + (long)transaksi.getBalance());
13
14        System.out.println("Memproses pembayaran sebesar " + (long)transaksi.getNominalBayar() + "...");
15        String hasil = transaksi.processPayment();
16        System.out.println(hasil);
17
18        System.out.println("Sisa saldo: " + (long)transaksi.getBalance());
19        System.out.println("Detail Transaksi: " + transaksi.getPaymentDetails());
20    }
21 }
```

State yang digunakan:

Dari interface PaymentMethod

- processPayment() → memproses transaksi, mengecek saldo, menentukan berhasil/gagal.
- getPaymentDetails() → mengembalikan informasi layanan e-wallet yang digunakan.
- getTransactionFee() → mengembalikan biaya transaksi.
- getBalance() → mengembalikan sisa saldo pengguna.

```
1 package pert7;
2
3 public interface PaymentMethod {
4     String processPayment();
5
6     String getPaymentDetails();
7
8     double getTransactionFee();
9
10    double getBalance();
11 }
```

Dari class EWalletPayment

- Constructor → mengisi nama layanan, saldo awal, dan nominal pembayaran.
- Override processPayment() → pengecekan saldo dan pemotongan saldo.
- Override getBalance() → mengembalikan saldo terbaru.
- Override getPaymentDetails() → memberikan detail transaksi e-wallet.

Hasil output

```
<terminated> PaymentTest [Java Application] C:\Users\M Javier Salman H\p2\pool\plugins\org.ec
==== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ====
Saldo awal: 150000
Memproses pembayaran sebesar 50000...
Pembayaran berhasil!
Sisa saldo: 98000
Detail Transaksi: Pembayaran dilakukan melalui OVO
```