

Nama: Muhammad Hanif Jundin  
NIM: H1D024127  
Shift KRS: F  
Shift Baru: C

## Laporan Responsi 2 PBO - Pertemuan 7

### 1. Alur Kerja Program

1. Program dimulai dari method main() di kelas PaymentTest.
2. Dibuat objek EWalletPayment yang mewakili satu transaksi pembayaran menggunakan layanan e-wallet, nilai yang diisikan ke constructor antara lain nama layanan e-wallet, nominal pembayaran, dan saldo awal pengguna.
3. Program menampilkan judul "==== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ====" kemudian menampilkan saldo awal dengan memanggil getBalance(). Berikutnya, program menampilkan informasi nominal pembayaran dan memanggil processPayment() untuk memproses transaksi. Jika saldo cukup (termasuk biaya transaksi), maka saldo dikurangi dan mengembalikan pesan "Pembayaran berhasil!". Jika saldo tidak mencukupi, akan dikembalikan pesan "Pembayaran gagal! Saldo tidak cukup."
4. Setelah proses pembayaran, program menampilkan sisa saldo menggunakan getBalance(), lalu menampilkan detail transaksi dengan memanggil getPaymentDetails().

### 2. Method dan Fungsi yang Digunakan Interface PaymentMethod

- 1) void processPayment()  
memproses transaksi pembayaran dan menentukan apakah berhasil atau gagal.
- 2) String getPaymentDetails()  
mengembalikan informasi ringkas mengenai detail transaksi.
- 3) double getTransactionFee()  
mengembalikan nilai biaya transaksi (misalnya Rp 2000).
- 4) double getBalance()  
mengembalikan saldo pengguna saat ini.

## **Class EWalletPayment**

- 5) Constructor EWalletPayment(String namaLayanan, double nominal, double saldoAwal)  
mengisi nama layanan e-wallet, nominal pembayaran, dan saldo pengguna.
- 6) processPayment()  
mengecek apakah saldo cukup (saldo  $\geq$  nominal + fee), jika cukup saldo dikurangi dan pesan berhasil dikembalikan, jika tidak pesan gagal dikembalikan.
- 7) getPaymentDetails()  
mengembalikan kalimat seperti "Pembayaran dilakukan melalui [namaLayanan]."
- 8) getTransactionFee()  
mengembalikan nilai fee tetap, misalnya 2000.
- 9) Implementasi getBalance()  
mengembalikan saldo terkini setelah transaksi.

## **Class PaymentTest:**

- 10) main(String[] args)  
membuat objek EWalletPayment, menampilkan saldo awal, memproses pembayaran, menampilkan sisa saldo dan detail transaksi untuk menguji semua method.

### **3. 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
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** On the left, it lists several projects and files:
  - pert3
  - pert4
  - pert5
  - pert6
  - pert7** (selected):
    - EWalletPayment
    - PaymentMethod
    - PaymentTest** (highlighted in blue)
    - Pengayaan.txt
    - Responsi Praktikum PBO - Materi I
  - pert8
  - README.md
  - Responsi-2-main.iml
- Code Editor:** The main editor window displays the `PaymentTest.java` file content:

```
3  public class PaymentTest {
5      public static void main(String[] args) {
7          EWalletPayment pembayaran = new EWalletPayment("OVO", 50000, 150000);
9
10         System.out.println("Saldo awal: " + (int) pembayaran.getBalance());
12
13         pembayaran.processPayment();
15
16         System.out.println("Sisa saldo: " + (int) pembayaran.getBalance());
17         System.out.println("Detail Transaksi: " + pembayaran.getPaymentDetails());
18     }
}
```
- Terminal:** Below the code editor is a terminal window titled "PaymentTest" showing the execution output:

```
C:\Users\DELL\.jdks\openjdk-24.0.2+12-54\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8 ===
PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===
Saldo awal: 150000
Memproses pembayaran sebesar 50000...
Pembayaran berhasil!
Sisa saldo: 98000
Detail Transaksi: Pembayaran dilakukan melalui OVO
```