

Nama: Aqsha Ramadhan Dimas Haryanto
NIM: H1D024089
SHIFT BARU: D
LAPORAN RESPONSI 2 PBO PERTEMUAN: 7

A. Alur kerja

1. Inisialisasi Objek (Setup)
 - Program membuat objek baru bernama ovo dari class EWalletPayment.
 - Constructor dipanggil dengan data: ("OVO", 50000, 150000).
 - Data ini disimpan ke dalam atribut private (namaLayanan, nominal, saldo) agar aman.
2. Pengecekan Awal (Pre-Transaction)
 - Program memanggil ovo.getBalance() untuk memastikan saldo awal tersimpan benar (Rp 150.000).
 - Program menampilkan nominal yang akan dibayar menggunakan ovo.getNominalPembayaran().
3. Proses Inti Pembayaran (Logic Execution)
 - Program memanggil method ovo.processPayment().
 - Di dalam method ini, terjadi perhitungan otomatis:
 - Program memanggil getTransactionFee() untuk mengambil biaya admin (Rp 2.000).
 - Menghitung Total Bayar = \$50.000 + 2.000 = 52.000\$.
 - Logika If-Else:
 - Jika Saldo (\$150.000\$) \geq Total Bayar (\$52.000\$) \rightarrow Kurangi Saldo.
 - Jika tidak \rightarrow Batalkan transaksi.
 - Method mengembalikan pesan string ("Pembayaran berhasil!").
4. Laporan Akhir (Post-Transaction)
 - Program memanggil kembali ovo.getBalance(). Karena transaksi berhasil, saldo yang tampil sekarang adalah sisa saldo (\$150.000 - 52.000 = 98.000\$).
 - Terakhir, method getPaymentDetails() dipanggil untuk konfirmasi nama layanan

B. Fungsi yang digunakan

Tugas ini berfokus pada Interface sebagai kontrak standar sistem pembayaran.

1. Interface (PaymentMethod)
 - Fungsi: Sebagai "Kontrak Kerja". File ini hanya berisi daftar method kosong (processPayment, getBalance, dll).
 - Tujuan: Menjamin bahwa aplikasi E-Wallet apa pun yang kita buat (OVO, Dana, Gopay) PASTI memiliki fitur pembayaran, cek saldo, dan cek biaya admin.
2. Keyword implements
 - Implementasi: public class EWalletPayment implements PaymentMethod
 - Fungsi: Menyatakan bahwa class EWalletPayment tunduk dan patuh pada kontrak PaymentMethod. Akibatnya, class ini WAJIB membuat isi kode untuk semua method yang ada di interface. Jika satu saja terlewat, program error.
3. Method Overriding (@Override)
 - Fungsi: Mengisi logika nyata ke dalam method yang sebelumnya kosong di interface.

- Di Interface, processPayment() hanya nama.
 - Di Class EWallet, processPayment() berisi rumus pengurangan saldo.
4. Encapsulation (private attributes)
- Implementasi: private double saldo;
 - Fungsi: Keamanan data. Saldo tidak bisa diubah sembarangan (misal: ovo.saldo = 0;) dari class Main. Saldo hanya bisa berubah melalui prosedur resmi yaitu method processPayment().
5. Hardcoded Return Value
- Implementasi: return 2000.0; pada method getTransactionFee().
 - Fungsi: Menyederhanakan logika biaya admin agar nilainya tetap (konstan) setiap kali transaksi dilakukan.

Hasil Output Program:

```
[Running] cd "d:\Responsi-2\pert7\" && javac PaymentTest.java && java PaymentTest
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===
Saldo awal: 150000
Memproses pembayaran sebesar 50000...
Pembayaran berhasil!
Sisa saldo: 98000
Detail Transaksi: Pembayaran dilakukan melalui OVO

[Done] exited with code=0 in 2.426 seconds
```