

# LAPORAN PROGRAM PERTEMUAN 7

## Alur Kerja Program

1. Program menjalankan file PaymentTest.java sebagai entry point.
2. Mencetak header ==== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ==.
3. Membuat objek pertama EWalletPayment dengan parameter:
  - o Nama penyedia: "OVO"
  - o Saldo awal: 150.000
  - o Nominal transaksi: 50.000
4. Sebelum Transaksi: Menampilkan data awal (saldo, nama penyedia, nominal, biaya transaksi).
5. Proses Pembayaran: Memanggil processPayment() untuk melakukan transaksi dengan menghitung total bayar (nominal + biaya transaksi).
6. Setelah Transaksi: Menampilkan detail transaksi lengkap menggunakan getPaymentDetails().
7. Uji Kasus Kedua: Membuat objek kedua untuk menguji transaksi dengan saldo tidak cukup:
  - o Nama penyedia: "Dana"
  - o Saldo awal: 10.000
  - o Nominal transaksi: 9.000
8. Menampilkan hasil transaksi kedua (diprediksi gagal karena biaya transaksi 2.000 membuat total menjadi 11.000 > saldo 10.000).

## Fungsi/Method yang Digunakan

- a. Interface PaymentMethod
  - Abstract Methods (wajib diimplementasikan):
    - o void processPayment(): Untuk memproses pembayaran.
    - o String getPaymentDetails(): Mengembalikan detail pembayaran sebagai String.
    - o double getTransactionFee(): Mengembalikan biaya transaksi.
    - o double getBalance(): Mengembalikan saldo pengguna.
- b. Class EWalletPayment (Implementasi PaymentMethod)
  - Atribut:
    - o namaPenyedia (String): Nama provider e-wallet.
    - o saldo (double): Saldo pengguna.
    - o nominalTransaksi (double): Jumlah yang akan dibayar.
    - o BIAYA\_TRANSAKSI (static final double): Biaya tetap sebesar 2000.0.
  - Constructor: EWalletPayment(String namaPenyedia, double saldoAwal, double nominalTransaksi)  
Menginisialisasi semua atribut dengan nilai yang diberikan.
  - Implementasi Interface Methods:
    - o processPayment():
      - Menghitung total bayar = nominal + biaya transaksi.
      - Mengecek kecukupan saldo.
      - Jika cukup: kurangi saldo dan tampilkan pesan sukses.
      - Jika tidak cukup: tampilkan pesan gagal.
    - o getPaymentDetails():
      - Mengembalikan string berisi detail lengkap transaksi.
    - o getTransactionFee():
      - Mengembalikan nilai konstan BIAYA\_TRANSAKSI.
    - o getBalance():
      - Mengembalikan saldo terkini.
  - Getter Tambahan (untuk testing):
    - o getNamaPenyedia(): Mengembalikan nama penyedia e-wallet.
    - o getNominalTransaksi(): Mengembalikan nominal transaksi.
- c. Class PaymentTest (Main Class)
  - Method main(String[] args):  
Menjalankan dua skenario pengujian:
    1. Transaksi berhasil (saldo mencukupi).
    2. Transaksi gagal (saldo tidak mencukupi).

## Output Program

```
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===

=== DATA SEBELUM TRANSAKSI ===
Saldo awal: 150000.0
Nama penyedia: OVO
Nominal transaksi: 50000.0
Biaya transaksi: 2000.0

Memproses pembayaran sebesar 50000.0...

Pembayaran berhasil!
Sisa saldo: 98000.0

=== DATA SETELAH TRANSAKSI ===
Detail Transaksi: Pembayaran dilakukan melalui OVO
Nominal: 50000.0
Biaya transaksi: 2000.0
Total yang harus dibayar: 52000.0

=== UJI TRANSAKSI DENGAN SALDO TIDAK CUKUP ===
Saldo awal: 10000.0
Memproses pembayaran sebesar 9000.0...
Pembayaran gagal! Saldo tidak cukup.
Dibutuhkan: 11000.0, Saldo tersedia: 10000.0
```