

**NAMA: REVALDI ENZHA AGVIANDRY P**  
**NIM: H1D024094**  
**SHIFT KRS: B**  
**SHIFT BARU: A**

## **LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

### **pertemuan 7**

#### **1. Alur Kerja Program**

Interface digunakan untuk menentukan kerangka perilaku umum yang harus dimiliki oleh sebuah metode pembayaran, tanpa menentukan bagaimana implementasinya. Kelas `EWalletPayment` dibuat untuk mengimplementasikan interface `PaymentMethod` dan menyediakan fungsionalitas pembayaran menggunakan dompet digital.

Dalam kelas `PaymentTest`, program membuat sebuah objek pembayaran melalui e-wallet dan memproses transaksi dengan memanggil method `processPayment`. Program juga memanggil method `getPaymentDetails` untuk mendapatkan informasi mengenai metode pembayaran yang digunakan. Dengan cara ini, program menampilkan bagaimana interface dapat memberikan struktur yang sama untuk berbagai sistem pembayaran, meskipun implementasinya dapat berbeda.

#### **2. Fungsi dan Method yang Digunakan**

State (variabel):

- `nama`: Menyimpan nama e-wallet yang digunakan dalam transaksi.
- `jumlahSaldo`: Menyimpan jumlah saldo yang tersedia sebelum dan sesudah transaksi.
- `nominalTransaksi`: Menyimpan nilai transaksi yang akan diproses.

Method dan fungsinya:

- `processPayment()`: Memproses transaksi pembayaran dengan memeriksa saldo dan mengurangi saldo berdasarkan nominal transaksi ditambah biaya transaksi.
- `getPaymentDetails()`: Mengembalikan informasi singkat tentang metode pembayaran yang digunakan.
- `getTransactionFee()`: Mengambil nilai biaya transaksi yang ditentukan (dalam program ini selalu bernilai 2000).
- `getBalance()`: Mengambil nilai saldo e-wallet saat ini.
- (Interface `PaymentMethod`) method-method di atas merupakan kontrak yang wajib diimplementasikan oleh setiap kelas yang menggunakan interface tersebut.

### 3. Hasil Output Program

Saat program dijalankan dengan saldo awal 150000 dan nominal transaksi 50000, program menampilkan saldo awal terlebih dahulu. Kemudian transaksi diproses dengan menghitung apakah saldo mencukupi. Karena saldo lebih besar dari nominal transaksi, program menampilkan bahwa pembayaran berhasil dan menghitung sisa saldo setelah dikurangi nominal transaksi dan biaya transaksi. Program juga memanggil method `getPaymentDetails`, meskipun hasilnya tidak ditampilkan pada output utama.

output:

Saldo awal: 150000.0

Memproses pembayaran sebesar: 50000.0

Pembayaran berhasil!

sisa saldo: 98000.0