

1. Alur kerja:

- Program dimulai dari main() di PaymentTest.
- Membuat objek EWalletPayment dengan nama layanan, nominal pembayaran, dan saldo awal.
- Menampilkan saldo awal menggunakan getBalance().
- Memanggil processPayment() untuk memproses pembayaran:
 - o Menghitung total biaya (nominal + biaya transaksi).
 - o Mengurangi saldo jika mencukupi; menampilkan status berhasil atau gagal.
- Menampilkan sisa saldo setelah pembayaran.
- Menampilkan detail transaksi menggunakan getPaymentDetails().

2. Fungsi:

- processPayment () → memproses pembayaran dan mengubah saldo.
- getPaymentDetails () → menampilkan informasi metode pembayaran.
- getTransactionFee () → mengembalikan biaya transaksi (2000).
- getBalance () → mengembalikan saldo saat ini.

```
public class PaymentTest {

    public static void main(String[] args) {

        System.out.println("=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===");

        EWalletPayment pembayaran = new EWalletPayment("OVO", 50000, 150000);

        System.out.println("Saldo awal: " + pembayaran.getBalance());

        pembayaran.processPayment();

        System.out.println("Sisa saldo: " + pembayaran.getBalance());

        System.out.println("Detail Transaksi: " + pembayaran.getPaymentDetails());
    }
}

// =====
// INTERFACE
// =====
interface PaymentMethod {
    void processPayment();
    String getPaymentDetails();
    double getTransactionFee();
    double getBalance();
}

// =====
// CLASS E-WALLET
```

```
// =====
class EWalletPayment implements PaymentMethod {

    private String namaLayanan;
    private double nominalPembayaran;
    private double saldo;

    public EWalletPayment(String namaLayanan, double nominalPembayaran, double
saldo) {
        this.namaLayanan = namaLayanan;
        this.nominalPembayaran = nominalPembayaran;
        this.saldo = saldo;
    }

    @Override
    public void processPayment() {
        System.out.print("Memproses pembayaran sebesar " + nominalPembayaran +
"...");

        double totalBiaya = nominalPembayaran + getTransactionFee();

        if (saldo >= totalBiaya) {
            saldo -= totalBiaya;
            System.out.println("Pembayaran berhasil!");
        } else {
            System.out.println("Pembayaran gagal! Saldo tidak cukup.");
        }
    }

    @Override
    public String getPaymentDetails() {
        return "Pembayaran dilakukan melalui " + namaLayanan;
    }

    @Override
    public double getTransactionFee() {
        return 2000;
    }

    @Override
    public double getBalance() {
        return saldo;
    }
}

```

3. Output:

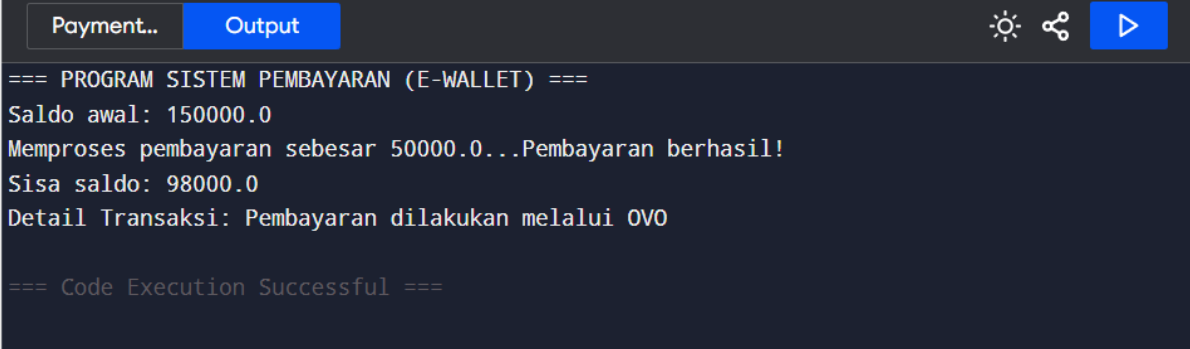
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===

Saldo awal: 150000.0

Memproses pembayaran sebesar 50000.0...Pembayaran berhasil!

Sisa saldo: 98000.0

Detail Transaksi: Pembayaran dilakukan melalui OVO

A screenshot of a code editor window. The window has a dark theme. At the top, there are two tabs: 'Payment..' and 'Output'. The 'Output' tab is active. To the right of the tabs are three icons: a sun, a share icon, and a play button. The main area of the editor shows the following text:

```
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===  
Saldo awal: 150000.0  
Memproses pembayaran sebesar 50000.0...Pembayaran berhasil!  
Sisa saldo: 98000.0  
Detail Transaksi: Pembayaran dilakukan melalui OVO  
  
=== Code Execution Successful ===
```