RESPONSI PBO

NAMA : FIKRY MUMTAZ PRATAMA
NIM : H1D024106

SHIFT : SHIFT B

TANGGAL : 07/12/2025

JAWAB

Penjelasan Program Sistem Pembayaran E-Wallet

Program ini menggunakan konsep **interface** untuk membuat aturan umum bagi semua metode pembayaran.
Pada contoh ini, metode pembayaran yang digunakan adalah **E-Wallet** seperti OVO, Dana, atau GoPay.

---

1. Interface PaymentMethod

```
public interface PaymentMethod {
   void processPayment();
   String getPaymentDetails();
   double getTransactionFee();
   double getBalance();
}
```

Penjelasan

- Interface ini berisi **empat method abstract**.
- Semua class yang menggunakan interface ini **wajib** mengisi (mengimplementasikan) method-method tersebut.
- Interface berfungsi sebagai **kontrak**:
  siapa pun yang memakai PaymentMethod **harus bisa** memproses pembayaran, menampilkan detail, menghitung biaya transaksi, dan menampilkan saldo.

---

2. Class EWalletPayment

public class EWalletPayment implements PaymentMethod {

Class ini **mengimplementasikan** PaymentMethod. Artinya, class ini menyediakan versi lengkap dari semua method dalam interface.

Atribut

- namaLayanan → nama e-wallet seperti OVO atau Dana
- saldo → saldo yang tersedia

- nominalBayar → nominal pembayaran yang ingin dilakukan

Constructor

Constructor mengisi data awal e-wallet:

public EWalletPayment(String namaLayanan, double saldo, double nominalBayar)

Method processPayment()

- Menghitung total yang harus dibayar (nominal + biaya admin)
- Mengecek apakah saldo cukup
- Jika cukup → saldo dikurangi dan menampilkan pesan berhasil
- Jika tidak cukup → tampilkan pesan gagal

Method getPaymentDetails()

Mengembalikan informasi layanan yang dipakai.
Contoh: “Pembayaran dilakukan melalui OVO”

Method getTransactionFee()

Mengembalikan biaya admin, misal:

return 2000;

Method getBalance()

Mengembalikan saldo saat ini setelah transaksi dilakukan.

---

3. Class PaymentTest (main)

Di bagian ini, program dijalankan dari langkah pertama sampai selesai.

1. Membuat objek e-wallet

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

Artinya:

- Layanan: OVO
- Saldo awal: 150.000
- Transaksi: 50.000

2. Menampilkan saldo awal

payment.getBalance();

3. Memproses pembayaran

payment.processPayment();

Program:

- Menambah biaya admin → total 52.000
- Cek saldo → saldo cukup
- Kurangi saldo → menjadi 98.000

4. Menampilkan saldo setelah transaksi & detail

payment.getBalance();

payment.getPaymentDetails();

```
31
32      @Override  1 usage  new *
33      public double getTransactionFee() {
34          return 2000;
35      }
36
37      @Override  no usages  new *
38      public double getBalance() {
39          return saldo;
40      }
41  }
42
```

Pengayaan.txt | PaymentTest.java | PaymentMethod.java | EWalletPayment.java

```
1   public class EWalletPayment implements PaymentMethod {  no usages  new *
2
3       private String namaLayanan;  2 usages
4       private double saldo;  4 usages
5       private double nominalBayar;  3 usages
6
7       public EWalletPayment(String namaLayanan, double saldo, double nominalBayar) {  no usages  new *
8           this.namaLayanan = namaLayanan;
9           this.saldo = saldo;
10          this.nominalBayar = nominalBayar;
11      }
12
13      @Override  no usages  new *
14      public void processPayment() {
15          double totalBayar = nominalBayar + getTransactionFee();
16
17          System.out.println("Memproses pembayaran sebesar " + nominalBayar + "...");
18
19          if (saldo >= totalBayar) {
20              saldo -= totalBayar;
21              System.out.println("Pembayaran berhasil!");
22          } else {
23              System.out.println("Pembayaran gagal! Saldo tidak cukup.");
24          }
25      }
26
27      @Override  no usages  new *
28      public String getPaymentDetails() {
29          return "Pembayaran dilakukan melalui " + namaLayanan;
30      }
```

Pengayaan.txt PaymentTest.java PaymentMethod.java EWalletPayment.java

```java
public interface PaymentMethod {  1 usage  1 implementation  new *

    void processPayment();  no usages  1 implementation  new *

    String getPaymentDetails();  no usages  1 implementation  new *

    double getTransactionFee();  1 usage  1 implementation  new *

    double getBalance();  no usages  1 implementation  new *
}
```

Pengayaan.txt PaymentTest.java PaymentMethod.java EWalletPayment.java

```java
public class PaymentTest {

    public static void main(String[] args) {

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

        EWalletPayment payment = new EWalletPayment( namaLayanan: "OVO",  saldo: 150000,  nominalBayar: 50000);

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

        payment.processPayment();
        System.out.println();

        System.out.println("Sisa saldo: " + payment.getBalance());
        System.out.println("Detail Transaksi: " + payment.getPaymentDetails());
    }
}
```

PaymentTest

```
"C:\Program Files\Eclipse Adoptium\jdk-21.0.7.6-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=56305" -Dfile.er
=== 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

Process finished with exit code 0
```