

Nama : Yunan Faila Sofi

NIM : H1D024033

ShiftLama : G

ShiftBaru : C

Class Customer

package pert4;

```
public class Customer {
    // Gunakan 'protected' agar variabel dapat diakses langsung oleh subclass
    protected String namaLengkap;
    protected String nomorIdentitasCustomer;
    protected int totalBelanja;

    // Constructor untuk inisialisasi Customer
    public Customer(String namaLengkap, String nomorIdentitasCustomer, int totalBelanja) {
        this.namaLengkap = namaLengkap;
        this.nomorIdentitasCustomer = nomorIdentitasCustomer;
        this.totalBelanja = totalBelanja;
    }

    /**
     * Menampilkan informasi dasar pelanggan.
     */
    public void tampilkanInfo() {
        System.out.println(
            "Nama: " + this.namaLengkap +
            " | ID: " + this.nomorIdentitasCustomer +
            " | Total Belanja: Rp " + this.totalBelanja
        );
    }
}

/**
 * Subclass (Child Class) untuk pelanggan Member.
 * Mewarisi semua properti dan method dari Customer.
 */
class Member extends Customer {
    // Atribut tambahan khusus Member
}
```

```

private int poinReward;
private String levelKeanggotaan;

// Constructor untuk Member
public Member(String namaLengkap, String nomorIdentitasCustomer, int totalBelanja, int
poinReward, String levelKeanggotaan) {
    // Memanggil constructor dari parent class (Customer) menggunakan 'super'
    super(namaLengkap, nomorIdentitasCustomer, totalBelanja);

    // Inisialisasi atribut khusus Member
    this.poinReward = poinReward;
    this.levelKeanggotaan = levelKeanggotaan;
}

/**
 * Meng-override method tampilkanInfo() untuk menampilkan informasi detail Member.
 */
@Override
public void tampilkanInfo() {
    // 1. Panggil method tampilkanInfo() dari parent class (Customer)
    super.tampilkanInfo();

    // 2. Tampilkan data tambahan khusus Member
    System.out.println(
        "Poin Reward: " + this.poinReward +
        " | Level: " + this.levelKeanggotaan
    );
}
}

```

1. Alur Kerja Program (Workflow)

Program berjalan secara sekuensial (berurutan) dimulai dari class UjiCustomer. Berikut adalah langkah-langkah logisnya:

1. Inisialisasi Program: Program dimulai dari method main pada kelas UjiCustomer.
2. Instansiasi Objek Parent (Customer):
 - Program membuat objek customerBudi dengan tipe Customer.
 - Constructor Customer dijalankan untuk menyimpan data: Nama "Budi Santoso", ID "CST-001", dan Belanja 500.000.

3. Instansiasi Objek Child (Member):

- Program membuat objek memberSiti dengan tipe Member.
- Constructor Member dijalankan. Hal pertama yang dilakukan adalah memanggil `super(...)`. Ini mengirim data nama, ID, dan total belanja ke constructor Parent (Customer) untuk disimpan.
- Setelah itu, barulah atribut khusus (`poinReward` dan `levelKeanggotaan`) disimpan di kelas Member.

4. Eksekusi Method pada Objek Customer:

- Saat `customerBudi.tampilkanInfo()` dipanggil, program langsung menjalankan method yang ada di kelas Customer.

5. Eksekusi Method pada Objek Member (Polimorfisme):

- Saat `memberSiti.tampilkanInfo()` dipanggil, program menjalankan method yang ada di kelas Member (Override).
- Langkah unik: Di dalam method ini, perintah `super.tampilkanInfo()` dieksekusi terlebih dahulu. Artinya, program "meminjam" fungsi cetak milik Parent untuk menampilkan Nama dan ID.
- Setelah itu, barulah program mencetak baris kedua berisi Poin dan Level.

2. Fungsi dan Komponen yang Digunakan

Berikut adalah elemen teknis utama dalam kode pert4:

a) Kelas Customer (Superclass)

- `protected`: *Access modifier* pada variabel. Berbeda dengan `private`, `protected` mengizinkan variabel tersebut diakses langsung oleh kelas anak (Member) tanpa perlu `getter`, meskipun dalam kode ini akses dilakukan via method.
- Constructor Dasar: Menginisialisasi data umum yang dimiliki semua jenis pelanggan.

b) Kelas Member (Subclass)

- `extends Customer`: Menandakan bahwa Member adalah perluasan dari Customer.
- `super(parameter)` (di Constructor): Wajib dipanggil di baris pertama constructor anak. Fungsinya untuk memastikan variabel induk terisi sebelum variabel anak diisi.
- `super.tampilkanInfo()` (di Method): Ini adalah teknik Code Reusability. Daripada menulis ulang kode `System.out.println("Nama: " + ...)` yang sudah ada di Parent, kita cukup memanggil method milik Parent. Ini membuat kode lebih efisien dan tidak duplikat.

- `@Override`: Menandakan bahwa method ini memodifikasi perilaku method bernama sama dari kelas induk.
- c) Kelas UjiCustomer (Main Class)
Berfungsi sebagai *driver* atau pengendali utama untuk menciptakan objek dan menguji apakah data tersimpan serta tampil dengan benar sesuai tipe kelasnya (biasa atau member).

Class UjiCustomer

```
package pert4;
```

```
public class UjiCustomer {  
    public static void main(String[] args) {  
  
        // 1. Buat object Customer (Yunan)  
        Customer customerYunan = new Customer("Yunan Unsoed", "CST-001", 500000);  
  
        // 2. Buat object Member (Faila)  
        Member memberFaila = new Member("Faila Unsoed", "MBR-110", 1250000, 240,  
        "Gold");  
  
        // Tampilkan judul data pelanggan  
        System.out.println("=== DATA CUSTOMER BELANJAKU ===");  
  
        // --- Customer Biasa ---  
        System.out.println("Status: Customer Biasa");  
        customerYunan.tampilkanInfo();  
        System.out.println(); // Jarak baris  
  
        // --- Pelanggan Member ---  
        System.out.println("Status: Member");  
        memberFaila.tampilkanInfo();  
    }  
}
```

3. Hasil Output Program

```
=== DATA CUSTOMER BELANJAKU ===  
Status: Customer Biasa  
Nama: Yunan Unsoed | ID: CST-001 | Total Belanja: Rp 500000  
  
Status: Member  
Nama: Faila Unsoed | ID: MBR-110 | Total Belanja: Rp 1250000  
Poin Reward: 240 | Level: Gold  
  
Process finished with exit code 0
```

Analisis Output:

1. Yunan: Hanya mencetak satu baris karena menggunakan method standar Customer.
2. Faila: Mencetak dua baris.
 - Baris pertama ("Nama...", "ID...", "Total Belanja...") muncul karena perintah `super.tampilkanInfo()`.
 - Baris kedua ("Poin Reward...", "Level...") muncul dari perintah `System.out.println` tambahan yang ada di kelas Member.