

LAPORAN PERTEMUAN 4

NAMA : AHMAD FIKRI ZAKARIA
NIM : H1D024062
MATERI : INHERITANCE LANJUTAN & KEYWORD SUPER

A. Alur Kerja (Workflow)

Pada pertemuan ini, saya mengembangkan sistem pendataan pelanggan toko "BelanjaRek" yang menerapkan konsep pewarisan tingkat lanjut menggunakan *keyword* super. Program membedakan antara pelanggan biasa (*Customer*) dan pelanggan prioritas (*Member*) dengan rincian pengerjaan sebagai berikut:

1. Pembuatan Class Parent (Customer):

- Mendefinisikan class *Customer* dengan atribut dasar nama, id, dan totalBelanja menggunakan *access modifier* *protected*.
- Membuat *constructor* untuk inialisasi data awal pelanggan.
- Membuat method *tampilkanInfo()* untuk mencetak struk data pelanggan standar.

2. Pembuatan Class Child (Member):

- Membuat class *Member* yang mewarisi class *Customer* menggunakan sintaks *extends*.
- Menambahkan atribut spesifik poin dan level.
- Menerapkan **Constructor Chaining**: Menggunakan *super(...)* untuk mengirim data nama, ID, dan total belanja ke *constructor parent*.
- Menerapkan **Method Overriding**: Menggunakan *super.tampilkanInfo()* di dalam method *tampilkanInfo()* milik *Member* untuk memanfaatkan logika tampilan data yang sudah ada di *parent*.

3. Pengujian Program (UjiCustomer):

- Membuat objek dari class *Customer* dan *Member* dengan data uji yang berbeda.
- Memastikan bahwa objek *Member* dapat menampilkan data gabungan (data warisan *parent* + data pribadinya) dengan format yang benar.

B. Penjelasan Kode & Fungsi

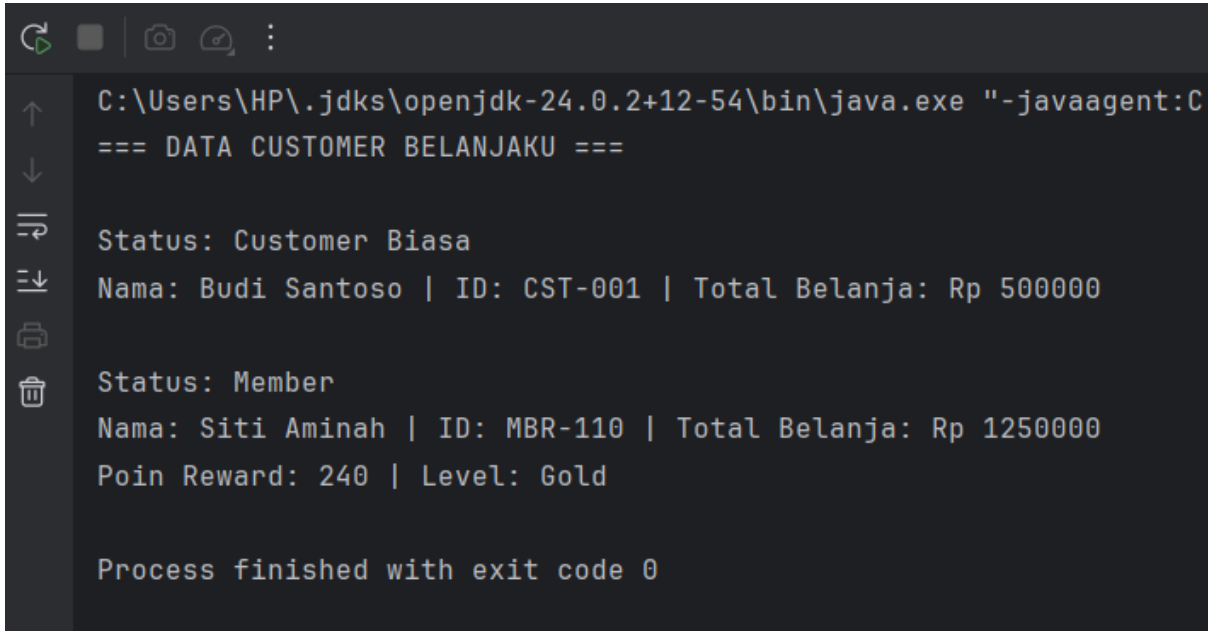
Berikut adalah analisis mendalam mengenai konsep teknis yang diterapkan dalam kode program:

1. **Inheritance (extends)** Class *Member* dideklarasikan dengan *public class Member extends Customer*. Ini menandakan bahwa *Member* adalah perluasan dari *Customer*. Dengan ini, semua atribut (nama, id, totalBelanja) secara otomatis dimiliki oleh objek *Member* tanpa perlu dideklarasikan ulang, mendukung prinsip efisiensi kode (*Code Reusability*).
2. **Keyword super pada Constructor** Pada baris pertama *constructor* *Member*, terdapat perintah *super(nama, id, totalBelanja)*. Kode ini berfungsi memanggil *constructor* milik class induk (*Customer*). Hal ini **wajib** dilakukan karena atribut dasar pelanggan berada di class induk, sehingga Java harus menginisialisasi bagian induknya terlebih dahulu sebelum menginisialisasi atribut tambahan (poin dan level) milik anak.
3. **Method Overriding dengan super** Method *tampilkanInfo()* pada class *Member* melakukan *Overriding* (penimpaan). Di dalamnya, perintah *super.tampilkanInfo()* digunakan untuk memanggil output standar (Nama & ID) dari class *Customer*. Setelah itu, barulah kode tambahan dijalankan untuk menampilkan Poin dan Level. Teknik ini mencegah duplikasi kode (*Redundancy*) karena kita tidak perlu menulis ulang perintah *System.out.println* untuk nama dan ID.

4. **Access Modifier protected** Atribut pada class Customer menggunakan modifier protected. Ini memberikan hak akses fleksibel di mana atribut tersebut tertutup bagi class luar yang tidak berkepentingan, namun tetap terbuka dan dapat diakses langsung oleh class turunannya (*subclass*) seperti Member jika diperlukan manipulasi data lebih lanjut.

C. Hasil Output Program

Berikut adalah tangkapan layar/teks hasil eksekusi program UjiPegguna.java:



```
C:\Users\HP\.jdk\openjdk-24.0.2+12-54\bin\java.exe "-javaagent:C
=== DATA CUSTOMER BELANJAKU ===

Status: Customer Biasa
Nama: Budi Santoso | ID: CST-001 | Total Belanja: Rp 500000

Status: Member
Nama: Siti Aminah | ID: MBR-110 | Total Belanja: Rp 1250000
Poin Reward: 240 | Level: Gold

Process finished with exit code 0
```