

Laporan Pertemuan 4

Nama : [Al Jatsiya Profitar Taqwala]
NIM : [H1D024095]
Shift KRS : [B]
Shift Baru : [I]

1) Alur Kerja Program

Program dieksekusi secara berurutan mulai dari method main di kelas *UjiCustomer*. Berikut adalah langkah demi langkah apa yang terjadi saat program dijalankan:

a) Instansiasi Objek *Customer* (Parent):

- ▶ Baris *new Customer(...)* dijalankan.
- ▶ Constructor *Customer* dipanggil untuk menyimpan data dasar: "Budi Santoso", ID "CST-001", dan total belanja 500.000.

b) Instansiasi Objek *Member* (Child):

- ▶ Baris *new Member(...)* dijalankan.
- ▶ Langkah Kritis: Constructor *Member* pertama-tama memanggil *super(...)*. Ini "mengoper" data nama, ID, dan total belanja ke kelas induk (*Customer*) untuk disimpan.
- ▶ Setelah parent selesai, *Member* menyimpan atribut khususnya sendiri (*poin* dan *level*).

c) Eksekusi Output:

- ▶ Saat *cust1.tampilkanInfo()* dipanggil: Program menjalankan method standar milik kelas Customer.
- ▶ Saat *memb1.tampilkanInfo()* dipanggil: Program menjalankan method Override milik kelas *Member*.
 - Di dalamnya, ia memanggil *super.tampilkanInfo()* (mencetak nama/ID/belanja).
 - Kemudian ia mencetak baris tambahan (Poin & Level).

2) Penjelasan Fungsi

a) Inheritance (*extends*)

Kelas *Member* didefinisikan dengan *class Member extends Customer*.

- ▶ Fungsi: Menjadikan *Member* sebagai turunan dari *Customer*.
- ▶ Efek: *Member* otomatis mewarisi atribut *nama*, *id*, dan *totalBelanja*. Kita tidak perlu mendeklarasikan ulang variabel-variabel tersebut di dalam *Member*.

b) Access Modifier (*protected*)

Pada kelas *Customer*, atribut menggunakan *protected*.

- ▶ Fungsi: Memberikan hak akses kepada kelas turunannya (*Member*) untuk mengakses variabel ini secara langsung jika diperlukan. Jika menggunakan *private*, kelas *Member* tidak akan bisa melihat variabel tersebut secara langsung.

c) Penggunaan *super* (Dua Konteks)

Kode ini menggunakan kata kunci *super* untuk dua tujuan berbeda:

- ▶ Di Constructor (*super(...)*):
 - Digunakan untuk memanggil constructor milik parent. Ini memastikan bahwa atribut dasar (nama, id, belanja) diinisialisasi dengan benar oleh logika yang sudah ada di kelas *Customer* sebelum kelas *Member* menambahkan atribut tambahannya.
- ▶ Di Method (*super.tampilkanInfo()*):
 - Digunakan di dalam method *tampilkanInfo* milik *Member*.
 - Tujuannya adalah untuk "meminjam" kode cetak dari parent. Jadi, Member tidak perlu menulis ulang *System.out.println("Nama: " + ...)* lagi. Cukup panggil induknya, lalu tambahkan print khusus member.

d) Method Overriding (@Override)

- Kelas Member menulis ulang method tampilkanInfo().
 - Alasan: Pelanggan biasa dan Member memiliki kebutuhan tampilan data yang berbeda. Member perlu menampilkan data lebih lengkap (ada poin dan level).

3) Hasil Output

The screenshot shows a Java development environment with the following details:

- Project Structure:** The project is named "responsipbo2H1D024095". It contains several source files: "pert2", "pert3", "pert5", "pert4", ".idea", and "src". The file "pert4" is currently selected.
- Terminal Output:** The terminal window displays the execution of a class named "UjiCustomer". The output shows:
 - A header: "4. Gunakan super(...) pada constructor"
 - A note: "[Data Uji]"
 - Customer details:
 - Status: Customer Biasa
 - Nama: Budi Santoso | ID: CST-001 | Total Belanja: Rp 500000
 - Customer details:
 - Status: Member
 - Nama: Siti Aminah | ID: MBR-110 | Total Belanja: Rp 1250000
 - Poin Reward: 240 | Level: Gold
- IDE Interface:** The bottom left shows various icons for file operations like opening, saving, and deleting files.