Laporan Pertemuan 3

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 *UjiKaryawan*. Berikut adalah langkah demi langkah apa yang terjadi saat program dijalankan:
   a) **Inisialisasi(Start):** Program mencetak Header "DATA KARYAWAN TECHMAJU".
   b) **Pembuatan Objek *Karyawan* (Parent):**
      - Objek *karyawan1* dibuat.
      - Constructor *Karyawan* dipanggil: Nama diisi "Budi Santoso", Gaji diisi 4 juta.
   c) **Output Karyawan Biasa:**
      - *karyawan1.tampilInfo()* dipanggil. Karena ini objek *Karyawan* biasa, ia menjalankan method versi sederhana (hanya menampilkan nama dan gaji).
   d) **Pembuatan Objek *Manajer* (Child):**
      - Objek *manajer1* dibuat.
      - Constructor Chaining: Constructor *Manajer* dipanggil. Hal pertama yang dilakukannya adalah memanggil *super(...)*. Ini mengirimkan data "nama" dan "gaji" ke constructor *Karyawan* (parent) untuk disimpan.
      - Setelah parent selesai, barulah *Manajer* menyimpan data khususnya sendiri, yaitu *tunjangan* (2.5 juta).
   e) **Output Manajer:**
      - *manajer1.tampilInfo()* dipanggil.
      - Program mendeteksi bahwa *manajer1* memiliki versi method *tampilInfo* sendiri (Override).
      - Maka, program menjalankan versi *Manajer* yang lebih lengkap (menghitung total gaji + tunjangan).

2) **Penjelasan Fungsi**
   a) **Inheritance (*extends*)**
      Kelas *Manajer* didefinisikan dengan *extends Karyawan*.
      - Artinya: *Manajer* otomatis memiliki semua variabel (*nama, gajiPokok*) dan method yang dimiliki *Karyawan*. Kita tidak perlu menulis ulang variabel tersebut di dalam kelas Manajer.
   b) **Access Modifier (*protected*)**
      Pada kelas *Karyawan*, variabel dideklarasikan sebagai: *protected String nama;*
      - Fungsi: *protected* mengizinkan variabel tersebut diakses langsung oleh kelas turunannya (subclass).
      - Jika *private*: Kelas *Manajer* tidak akan bisa membaca *super.nama* atau *super.gajiPokok*.
   c) **Keyword *super***
      Digunakan dalam dua konteks di kode ini:
      - *super(nama, gajiPokok)* di Constructor: Berfungsi "mengoper" tugas inisialisasi ke parent. Karena *Karyawan* sudah punya logika untuk menyimpan nama dan gaji, *Manajer* cukup meminjam logika tersebut tanpa menulis ulang *this.nama = nama.*
      - *super.nama* di Method: Mengambil nilai variabel *nama* yang tersimpan di kelas induk.
   d) **Method Overriding (*@Override*)**

- Situasi: Kelas *Karyawan* punya method *tampilInfo()*. Kelas *Manajer* juga punya method *tampilInfo()* dengan nama yang persis sama.
- Fungsi: *Manajer* "menimpa" (override) perilaku standar *Karyawan*.
  - Versi Karyawan: Hanya cetak Nama & Gaji.
  - Versi Manajer: Cetak Nama, Gaji, Tunjangan, dan Total Pendapatan.
- Logika Perhitungan: Di dalam *Manajer*, total gaji dihitung dengan menjumlahkan *super.gajiPokok* (warisan) + *this.tunjangan* (milik sendiri).

3) Hasil Output