**LAPORAN PRAKTIKUM**

**Studi Kasus**



**Disusun Oleh:**

**Dzakir Tsabit Asy Syafiq (241511071)**

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**

**Politeknik Negeri Bandung**

**09/10/2025**

**STUDI KASUS 1 – ANOTHER TYPE OF EMPLOYEE (COMMISSION)**

**1. Analisis Masalah**

Studi kasus ini meminta kita untuk menambahkan satu jenis karyawan baru ke dalam hierarki kelas yang sudah ada. Karyawan baru ini adalah seorang pekerja per jam yang juga mendapatkan komisi berdasarkan total penjualan.

Kelas baru ini akan dinamai Commission dan harus memenuhi persyaratan berikut:

* Mewarisi (extends) kelas Hourly.
* Memiliki dua variabel instance tambahan: totalSales (tipe double) untuk total penjualan dan commissionRate (tipe double) untuk persentase komisi.
* Memiliki constructor yang menerima 6 parameter (5 parameter untuk constructor Hourly dan 1 untuk commissionRate). Constructor ini harus memanggil constructor kelas induknya (super()).
* Memiliki metode addSales(double totalSales) untuk menambahkan jumlah penjualan.
* Meng-override metode pay() untuk menghitung total gaji (gaji per jam + komisi). Setelah pembayaran, total penjualan harus di-reset menjadi 0.
* Meng-override metode toString() untuk menampilkan informasi tambahan, yaitu total penjualan.

**2. Implementasi Kode**

Berikut adalah implementasi kode yang diperlukan untuk menyelesaikan studi kasus ini.

**A. Kelas Commission.java**

Ini adalah kelas baru yang dibuat sesuai dengan spesifikasi. Kelas ini mewarisi Hourly dan menambahkan fungsionalitas untuk menghitung komisi.

// Nama File: Commission.java

public class Commission extends Hourly {

private double totalSales; // Total penjualan yang dibuat

private double commissionRate; // Tingkat komisi (dalam desimal, misal 0.20 untuk 20%)

/\*\*

\* Constructor untuk membuat objek Commission.

\* Memanggil constructor dari kelas induk (Hourly) dan menginisialisasi variabel komisi.

\*/

public Commission(String eName, String eAddress, String ePhone,

String socSecNumber, double rate, double commRate) {

// Memanggil constructor kelas induk (Hourly)

super(eName, eAddress, ePhone, socSecNumber, rate);

// Mengatur tingkat komisi

this.commissionRate = commRate;

this.totalSales = 0; // Inisialisasi total penjualan awal adalah 0

}

/\*\*

\* Metode untuk menambahkan total penjualan.

\*/

public void addSales(double totalSales) {

this.totalSales += totalSales;

}

/\*\*

\* Meng-override metode pay() dari kelas induk.

\* Menghitung total gaji = (gaji per jam) + (gaji komisi).

\*/

@Override

public double pay() {

// 1. Hitung gaji per jam dengan memanggil metode pay() dari kelas induk (Hourly)

double hourlyPayment = super.pay();

// 2. Hitung gaji dari komisi

double commissionPayment = totalSales \* commissionRate;

// 3. Reset total penjualan kembali ke 0 setelah dihitung

totalSales = 0;

// 4. Kembalikan total pembayaran

return hourlyPayment + commissionPayment;

}

/\*\*

\* Meng-override metode toString() untuk menambahkan informasi total penjualan.

\*/

@Override

public String toString() {

// Memanggil toString() dari kelas induk (Hourly) untuk mendapatkan info dasar

String result = super.toString();

// Menambahkan informasi total penjualan

result += "\nTotal Sales: " + totalSales;

return result;

}

}

**Pertanyaan:** The pay method must call the pay method of the parent class to compute the pay for hours worked then add to that the pay from commission on sales. (See the pay method in the Executive class.) The total sales should be set back to 0 (note: you don't need to set the hoursWorked back to 0—why not?)

**Jawaban:** Alasannya terletak pada **prinsip pewarisan dan pembagian tanggung jawab (separation of concerns)**.

1. **Tanggung Jawab Hourly**: Variabel hoursWorked adalah milik kelas Hourly. Metode pay() di dalam kelas Hourly sudah dirancang untuk melakukan dua hal: menghitung pembayaran berdasarkan payRate \* hoursWorked dan kemudian **me-reset hoursWorked menjadi 0**.
2. **Panggilan super.pay()**: Di dalam metode pay() kelas Commission, baris double hourlyPayment = super.pay(); dieksekusi. Panggilan super.pay() ini mengeksekusi seluruh kode yang ada di dalam metode pay() milik kelas induknya (Hourly). Artinya, proses penghitungan gaji per jam dan reset hoursWorked sudah **otomatis dilakukan** oleh kelas Hourly.
3. **Tanggung Jawab Commission**: Kelas Commission hanya bertanggung jawab atas variabel dan logika yang ia definisikan sendiri, yaitu totalSales dan commissionRate. Oleh karena itu, setelah komisi dihitung, adalah tugas metode pay() di Commission untuk me-reset totalSales ke 0.

Singkatnya, hoursWorked di-reset oleh Hourly.pay() dan totalSales di-reset oleh Commission.pay(). Ini adalah contoh desain yang baik di mana setiap kelas hanya mengelola state (variabel) miliknya sendiri

**4. Latihan dan Hasil**

Setelah memodifikasi Staff.java, program akan dijalankan dan menghasilkan output yang mencakup dua karyawan baru.

**Perhitungan Gaji Karyawan Komisi**

Berikut adalah rincian perhitungan manual untuk memverifikasi hasil program:

1. **Karyawan 1 (Budi)**
   * Tarif per jam: $6.25
   * Jam kerja: 35
   * Total penjualan: $400
   * Tingkat komisi: 20% (0.20)
   * **Perhitungan:**
     + Gaji Per Jam: 35×$6.25=$218.75
     + Gaji Komisi: $400×0.20=$80.00
     + **Total Pembayaran:** $218.75+$80.00=$298.75
2. **Karyawan 2 (Siti)**
   * Tarif per jam: $9.75
   * Jam kerja: 40
   * Total penjualan: $950
   * Tingkat komisi: 15% (0.15)
   * **Perhitungan:**
     + Gaji Per Jam: 40×$9.75=$390.00
     + Gaji Komisi: $950×0.15=$142.50
     + **Total Pembayaran:** $390.00+$142.50=$532.50