

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERTEMUAN 5
ENCAPSULATION & INHERITANCE



Disusun oleh:

Nama : Aditya Lucky Zulkarnaen
NIM : 24/537764/SV/24449
Kelas : PLB1
Dosen Pengampu : Margareta Hardiyanti, S.Kom., M.IM., M.Cs

PROGRAM STUDI D-IV TEKNOLOGI REKAYASA PERANGKAT LUNAK
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2025
PERTEMUAN 5

ENCAPSULATION & INHERITANCE

A. Penjelasan

1. Kantor.java

```
import java.util.ArrayList;
import java.util.List;
import java.text.NumberFormat;
import java.util.Locale;

public class Kantor {
    private List<Pegawai> daftarPegawai = new ArrayList<>();
    private static final NumberFormat currencyFormatter =
        NumberFormat.getInstance(new Locale("id", "ID"));

    public Kantor() {
        currencyFormatter.setGroupingUsed(true);
        currencyFormatter.setMaximumFractionDigits(0);
    }

    public void tambahPegawai(Pegawai pegawai) {
        daftarPegawai.add(pegawai);
    }

    public void hapusPegawai(String nama) {
        daftarPegawai.removeIf(p -> p.getNama().equalsIgnoreCase(nama));
    }

    public double hitungRataRataGaji() {
        if (daftarPegawai.isEmpty()) return 0;
        double totalGaji = daftarPegawai.stream().mapToDouble(Pegawai::getGaji).sum();
        return totalGaji / daftarPegawai.size();
    }
}
```

```

public void tampilkanNamaPegawai() {
    if (daftarPegawai.isEmpty()) {
        System.out.println("Tidak ada pegawai.");
    } else {
        System.out.println("Daftar Pegawai:");
        for (Pegawai p : daftarPegawai) {
            System.out.println("- " + p.getNama() + " (Gaji: Rp " +
formatCurrency(p.getGaji()) + ")");
        }
    }
}

private String formatCurrency(double amount) {
    return currencyFormatter.format(amount);
}
}

```

Kelas Kantor bertindak sebagai tempat penyimpanan dan pengelolaan pegawai. Semua pegawai akan didaftarkan dan dikelola dalam objek Kantor. Alur dalam Kantor.java :

- Menyimpan daftar pegawai dalam ArrayList<Pegawai>.
- Menambahkan pegawai saat dipanggil dari App.
- Menghapus pegawai berdasarkan nama.
- Menghitung rata-rata gaji dari semua pegawai yang terdaftar.
- Menampilkan daftar pegawai beserta gajinya.

2. Pegawai.java

```

public class Pegawai {
    String nama;
    int umur;
    double gaji;
    int lamaKerja;
}

```

```
public Pegawai(String nama, int umur, double gaji, int lamaKerja) {
    this.nama = nama;
    this.umur = umur;
    this.gaji = gaji;
    this.lamaKerja = lamaKerja;
}

public double getGaji() {
    return gaji;
}

public String getNama() {
    return nama;
}
}

class Direktur extends Pegawai {
    private static final double BONUS_DIREKTUR = 5000000;

    public Direktur(String nama, int umur, double gajiDasar, int lamaKerja) {
        super(nama, umur, gajiDasar * lamaKerja + BONUS_DIREKTUR, lamaKerja);
    }
}

class Manager extends Pegawai {
    public Manager(String nama, int umur, double gajiDasar, int lamaKerja) {
        super(nama, umur, gajiDasar * lamaKerja * 1.5, lamaKerja);
    }
}

class Supervisor extends Pegawai {
```

```

    public Supervisor(String nama, int umur, double gajiDasar, int lamaKerja) {
        super(nama, umur, gajiDasar * lamaKerja * 1.2, lamaKerja);
    }
}

class Operasional extends Pegawai {
    public Operasional(String nama, int umur, double gajiDasar, int lamaKerja) {
        super(nama, umur, gajiDasar * lamaKerja, lamaKerja);
    }
}

```

Kelas Pegawai berisi informasi dasar tentang pegawai, sementara anak kelasnya menentukan cara perhitungan gaji yang berbeda.

Alur dalam Pegawai.java :

- Kelas Pegawai sebagai superclass, menyimpan:
- nama, umur, gaji, lamaKerja.
- Metode getName() dan getGaji().
- Anak kelas (Direktur, Manager, Supervisor, Operasional) menentukan cara perhitungan gaji masing-masing.

3. App.java

```

import java.util.Scanner;
import java.text.NumberFormat;
import java.util.Locale;

public class App {
    private static final NumberFormat currencyFormatter =
        NumberFormat.getInstance(new Locale("id", "ID"));

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Kantor kantor = new Kantor();
        int pilihan = 0;
    }
}

```

```

currencyFormatter.setGroupingUsed(true);
currencyFormatter.setMaximumFractionDigits(0);

do {
    tampilkanMenu();
    System.out.print("Masukkan pilihan: ");

    try {
        pilihan = Integer.parseInt(scanner.nextLine());

        switch (pilihan) {
            case 1:
                tambahPegawai(scanner, kantor);
                break;
            case 2:
                kantor.tampilkanNamaPegawai();
                break;
            case 3:
                System.out.println("Rata-rata      Gaji:      Rp      "      +
formatCurrency(kantor.hitungRataRataGaji()));
                break;
            case 4:
                hapusPegawai(scanner, kantor);
                break;
            case 5:
                System.out.println("Terima kasih telah menggunakan sistem.");
                break;
            default:
                System.out.println("Pilihan tidak valid. Silakan coba lagi.");
        }
    }
}

```

```

        } catch (NumberFormatException e) {
            System.out.println("Input tidak valid. Masukkan angka.");
        }

        System.out.println();
    } while (pilihan != 5);

    scanner.close();
}

private static String formatCurrency(double amount) {
    return currencyFormatter.format(amount);
}

private static void tampilkanMenu() {
    System.out.println("=== SISTEM MANAJEMEN PEGAWAI ===");
    System.out.println("1. Tambah Pegawai");
    System.out.println("2. Tampilkan Daftar Pegawai");
    System.out.println("3. Hitung Rata-rata Gaji");
    System.out.println("4. Hapus Pegawai");
    System.out.println("5. Keluar");
}

private static void tambahPegawai(Scanner scanner, Kantor kantor) {
    System.out.println("\n=== TAMBAH PEGAWAI ===");
    System.out.println("Pilih jenis pegawai:");
    System.out.println("1. Direktur");
    System.out.println("2. Manager");
    System.out.println("3. Supervisor");
    System.out.println("4. Operasional");
    System.out.print("Masukkan pilihan: ");

```

```
try {
    int jenisPegawai = Integer.parseInt(scanner.nextLine());

    if (jenisPegawai < 1 || jenisPegawai > 4) {
        System.out.println("Jenis pegawai tidak valid!");
        return;
    }

    System.out.print("Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Umur: ");
    int umur = Integer.parseInt(scanner.nextLine());

    System.out.print("Gaji Dasar: ");
    double gajiDasar = Double.parseDouble(scanner.nextLine());

    System.out.print("Lama Kerja (tahun): ");
    int lamaKerja = Integer.parseInt(scanner.nextLine());

    Pegawai pegawai = null;

    switch (jenisPegawai) {
        case 1:
            pegawai = new Direktur(nama, umur, gajiDasar, lamaKerja);
            break;
        case 2:
            pegawai = new Manager(nama, umur, gajiDasar, lamaKerja);
            break;
        case 3:
```



```

        pegawai = new Supervisor(nama, umur, gajiDasar, lamaKerja);
        break;
    case 4:
        pegawai = new Operasional(nama, umur, gajiDasar, lamaKerja);
        break;
    }

    kantor.tambahPegawai(pegawai);
    System.out.println("Pegawai " + nama + " berhasil ditambahkan!");
    System.out.println("Gaji: Rp " + formatCurrency(pegawai.getGaji()));

} catch (NumberFormatException e) {
    System.out.println("Input tidak valid. Masukkan angka yang sesuai.");
}
}

private static void hapusPegawai(Scanner scanner, Kantor kantor) {
    System.out.print("Masukkan nama pegawai yang akan dihapus: ");
    String nama = scanner.nextLine();

    kantor.hapusPegawai(nama);
    System.out.println("Pegawai dengan nama " + nama + " telah dihapus (jika ada).");
}
}

```

Kelas App menjalankan program utama dan berinteraksi dengan pengguna.

Alur dalam App.java :

- Membuat objek Kantor untuk menyimpan daftar pegawai.
- Menampilkan menu utama yang berisi pilihan:
- Tambah pegawai

- Tampilkan daftar pegawai
- Hitung rata-rata gaji
- Hapus pegawai
- Keluar
- Menangani input pengguna dan memanggil metode di Kantor.
- Meminta data pegawai saat menambahkan pegawai baru.