

TUGAS MATA KULIAH

PEMROGRAMAN LANJUT

PERUSAHAAN TAHAP 3

NAMA : ARSAL BIL HAKKI R. SAID
NIM : 245150400111057
KELAS : SI – E
GITHUB : <https://arsalbilhakki.github.io/Pemrograman-Lanjut---Perusahaan-Tahap-3/>

1. Link Github

[ArsalBilHakki/Pemrograman-Lanjut---Perusahaan-Tahap-3](https://arsalbilhakki.github.io/Pemrograman-Lanjut---Perusahaan-Tahap-3/)

2. Penjelasan Kode Program (Tabel)

Kelas Main

```
// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Main {
    // Mendeklarasikan kelas Main
    public static void main(String[] args) {
        // Mendeklarasikan metode Main sebagai titik masuk program
        Perusahaan perusahaan = new Perusahaan("PT. Jamu Tak Jemu");
        // Membuat objek (instansiasi) Perusahaan dengan nama perusahaan "PT. Jamu Tak Jemu"

        Karyawan karyawan1 = new Karyawan("Siti", 1000000, perusahaan);
        // Membuat objek Karyawan bernama karyawan1 dengan nama "Siti", gaji pokok 1000000,
        // dan referensi ke objek Perusahaan
        Karyawan karyawan2 = new Karyawan("Wati", 1000000, perusahaan);
        Karyawan karyawan3 = new Karyawan("Kartini", 800000, perusahaan);
        Karyawan karyawan4 = new Karyawan("Sarinah", 750000, perusahaan);
        Karyawan karyawan5 = new Karyawan("Sarinem", 750000, perusahaan);
```

```

Programmer programmer1 = new Programmer("Sopo", 1500000, perusahaan,
500000);
    // Membuat objek Programmer bernama programmer1 dengan nama "Sopo", gaji
    pokok 1500000,
    // referensi ke objek Perusahaan, dan bonus sebesar 500000
Programmer programmer2 = new Programmer("Jarwo", 1750000, perusahaan,
750000);

Manager manager1 = new Manager("Adit", 3500000, perusahaan, 1000000);
    // Membuat objek Manager bernama manager1 dengan nama "Adit", gaji pokok
    3500000,
    // referensi ke objek Perusahaan, dan insentif sebesar 1000000
Manager manager2 = new Manager("Denis", 2500000, perusahaan, 1000000);

karyawan1.info();
    // Memanggil metode info() pada objek karyawan1 untuk mencetak informasi
    karyawan
karyawan1.infoGaji();
    // Memanggil metode infoGaji() pada objek karyawan1 untuk mencetak total gaji
    karyawan
System.out.println();
    // Mencetak baris kosong
karyawan2.info();
    // Memanggil metode info() pada objek karyawan2 untuk mencetak informasi
    karyawan
karyawan2.infoGaji();
    // Memanggil metode infoGaji() pada objek karyawan2 untuk mencetak total gaji
    karyawan
System.out.println();
    // Mencetak baris kosong
karyawan3.info();
    // Memanggil metode info() pada objek karyawan3 untuk mencetak informasi
    karyawan
karyawan3.infoGaji();
    // Memanggil metode infoGaji() pada objek karyawan3 untuk mencetak total gaji
    karyawan
System.out.println();
    // Mencetak baris kosong
karyawan4.info();
    // Memanggil metode info() pada objek karyawan4 untuk mencetak informasi
    karyawan
karyawan4.infoGaji();
    // Memanggil metode infoGaji() pada objek karyawan4 untuk mencetak total gaji
    karyawan
System.out.println();
    // Mencetak baris kosong

```

```
karyawan5.info();  
    // Memanggil metode info() pada objek karyawan5 untuk mencetak informasi  
karyawan  
    karyawan5.infoGaji();  
    // Memanggil metode infoGaji() pada objek karyawan5 untuk mencetak total gaji  
karyawan  
    System.out.println();  
    // Mencetak baris kosong  
  
    programmer1.info();  
    // Memanggil metode info() pada objek programmer1 untuk mencetak informasi  
programmer  
    programmer1.infoGaji();  
    // Memanggil metode infoGaji() pada objek programmer1 untuk mencetak total  
gaji programmer  
    System.out.println();  
    // Mencetak baris kosong  
    programmer2.info();  
    // Memanggil metode info() pada objek programmer2 untuk mencetak informasi  
programmer  
    programmer2.infoGaji();  
    // Memanggil metode infoGaji() pada objek programmer2 untuk mencetak total  
gaji programmer  
    System.out.println();  
    // Mencetak baris kosong  
  
    manager1.info();  
    // Memanggil metode info() pada objek manager1 untuk mencetak informasi  
manager  
    manager1.infoGaji();  
    // Memanggil metode infoGaji() pada objek manager1 untuk mencetak total gaji  
manager  
    System.out.println();  
    // Mencetak baris kosong  
    manager2.info();  
    // Memanggil metode info() pada objek manager2 untuk mencetak informasi  
manager  
    manager2.infoGaji();  
    // Memanggil metode infoGaji() pada objek manager2 untuk mencetak total gaji  
manager  
    System.out.println();  
    // Mencetak baris kosong  
    }  
}
```

Kelas Karyawan

```
// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Karyawan {
    // Mendeklarasikan kelas Karyawan sebagai kelas dasar untuk karyawan

    private String nama;
    private double gajiPokok;
    private Perusahaan perusahaan;
    // Menginstansiasi atribut nama, gajiPokok, dan perusahaan sebagai atribut private

    public Karyawan(String nama, double gajiPokok, Perusahaan perusahaan) {
        // Mendeklarasikan konstruktor Karyawan untuk menginisialisasi atribut nama,
        gajiPokok, dan perusahaan

        setNama(nama);
        // Memanggil metode setNama untuk memvalidasi dan mengatur nilai atribut
        nama
        this.gajiPokok = gajiPokok;
        // Mengisi atribut gajiPokok dengan nilai dari parameter gajiPokok
        this.perusahaan = perusahaan;
        // Memberikan referensi ke objek perusahaan
    }

    public String getNama() {
        // Mendeklarasikan metode getNama untuk mengembalikan nilai atribut nama

        return nama;
        // Mengembalikan nilai atribut nama
    }

    public void setNama(String nama) {
        // Mendeklarasikan metode setNama untuk memvalidasi dan mengatur nilai atribut
        nama

        if (nama.length() < 4) {
            // Seleksi kondisi untuk memastikan nama memiliki minimal 4 karakter
            throw new IllegalArgumentException("Nama karyawan setidaknya terdiri atas
            4 karakter.");
            // Jika nama kurang dari 4 karakter, lemparkan exception
        }
    }
}
```

```

    this.nama = nama;
    // Mengisi atribut nama dengan nilai dari parameter nama
}

public double getGajiPokok() {
    // Mendeklarasikan metode getGajiPokok untuk mengembalikan nilai atribut
    gajiPokok

    return gajiPokok;
    // Mengembalikan nilai atribut gajiPokok
}

public Perusahaan getPerusahaan() {
    // Mendeklarasikan metode getPerusahaan untuk mengembalikan nilai atribut
    perusahaan

    return perusahaan;
    // Mengembalikan nilai atribut perusahaan
}

public double hitungGaji() {
    // Mendeklarasikan metode hitungGaji untuk menghitung total gaji karyawan

    return gajiPokok;
    // Mengembalikan nilai gajiPokok sebagai total gaji
}

public void infoGaji() {
    // Mendeklarasikan metode infoGaji untuk mencetak nama dan total gaji karyawan

    System.out.println("Nama: " + nama + ", Total Gaji: " +
Util.formatRupiah(hitungGaji()));
    // Mencetak nama dan total gaji karyawan,
    // yang diformat ke mata uang Rupiah menggunakan Util.
}

public void info() {
    // Mendeklarasikan metode info untuk mencetak informasi karyawan

    System.out.println("Nama: " + nama);
    // Mencetak nama karyawan
    System.out.println("Gaji Pokok: " + Util.formatRupiah(gajiPokok));
    // Mencetak gaji pokok dalam format mata uang Rupiah menggunakan Util.
    System.out.println("Nama Perusahaan: " +
perusahaan.getNAMA_PERUSAHAAN());

```

```

    // Mencetak nama perusahaan dengan memanggil metode
    getNAMA_PERUSAHAAN dari kelas Perusahaan
}
}

```

Subkelas Manager

```

// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Manager extends Karyawan {
    // Mendeklarasikan kelas Manager sebagai subclass dari Karyawan

    private double insentif;
    // Menginstansiasi atribut insentif sebagai atribut private

    public Manager(String nama, double gajiPokok, Perusahaan perusahaan, double
    insentif) {
        // Mendeklarasikan konstruktor Manager untuk menginisialisasi atribut

        super(nama, gajiPokok, perusahaan);
        // Memanggil konstruktor superclass (Karyawan) untuk mengatur nilai atribut
        umum
        this.insentif = insentif;
        // Mengisi atribut insentif dengan nilai dari parameter insentif
    }

    @Override
    public double hitungGaji() {
        // Mendeklarasikan metode hitungGaji yang mengoverride metode superclass

        return getGajiPokok() + insentif;
        // Mengembalikan total gaji sebagai gaji pokok ditambah insentif
    }
}

```

Subkelas Programmer

```

// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//

```

```
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Programmer extends Karyawan {
// Mendeklarasikan kelas Programmer sebagai subclass dari Karyawan

    private double bonus;
    // Menginstansiasi atribut bonus sebagai atribut private

    public Programmer(String nama, double gajiPokok, Perusahaan perusahaan, double
bonus) {
// Mendeklarasikan konstruktor Programmer untuk menginisialisasi atribut

        super(nama, gajiPokok, perusahaan);
        // Memanggil konstruktor superclass (Karyawan) untuk mengatur nilai atribut
        umum
        this.bonus = bonus;
        // Mengisi atribut bonus dengan nilai dari parameter bonus
    }

    @Override
    public double hitungGaji() {
// Mendeklarasikan metode hitungGaji yang mengoverride metode superclass

        return getGajiPokok() + bonus;
        // Mengembalikan total gaji sebagai gaji pokok ditambah bonus
    }
}
}
```

Kelas Perusahaan

```
// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Perusahaan {
// Mendeklarasikan kelas Perusahaan untuk menyimpan informasi perusahaan

    private final String NAMA_PERUSAHAAN;
    // Menginstansiasi atribut NAMA_PERUSAHAAN sebagai atribut final dan private

    public Perusahaan(String NAMA_PERUSAHAAN) {
```

```

// Mendeklarasikan konstruktor Perusahaan untuk menginisialisasi atribut
NAMA_PERUSAHAAN

    this.NAMA_PERUSAHAAN = NAMA_PERUSAHAAN;
    // Mengisi atribut NAMA_PERUSAHAAN dengan nilai dari parameter
    NAMA_PERUSAHAAN
}

public String getNAMA_PERUSAHAAN() {
    // Mendeklarasikan metode getNAMA_PERUSAHAAN untuk mengembalikan nilai
    atribut NAMA_PERUSAHAAN

    return NAMA_PERUSAHAAN;
    // Mengembalikan nilai atribut NAMA_PERUSAHAAN
}
}

```

Kelas Util

```

// TUGAS PEMROGRAMAN LANJUT
// TAHAP 3
//
// NAMA MAHASISWA: ARSAL BIL HAKKI R. SAID
// NIM: 245150400111057
// KELAS: E
public class Util {
    // Mendeklarasikan kelas Util
    // yang menyediakan method static formatRupiah(double amount)
    // untuk memformat nilai ke format mata uang Rupiah.
    public static String formatRupiah(double amount) {
        // Mendeklarasikan metode formatRupiah untuk memformat nilai ke format mata
        uang Rupiah.
        return String.format("Rp%,.0f", amount).replace(",", ".");
        // Mengembalikan nilai dengan format mata uang Rupiah,
        // mengganti tanda koma dengan titik sesuai format lokal.
    }
}

```

3. Penjelasan Kode Program (Codesnap)

4. Diagram Kelas