

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 3
PEMOGRAMAN JAVA DASAR 2



Disusun Oleh:
Gerald Eberhard
(105223002)

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. Pendahuluan

Studi kasus yang menjadi dasar praktikum ini adalah perhitungan gaji karyawan di sebuah pabrik dengan sistem kerja tiga shift (pagi, siang, dan malam). Setiap karyawan memiliki total jam kerja yang berbeda dalam seminggu, dengan kemungkinan lembur jika melebihi 40 jam atau potongan gaji sebesar 10% jika kurang dari 30 jam. Selain itu, absensi tanpa alasan dikenakan potongan Rp100.000 per hari. Program ini dirancang untuk mencatat data karyawan (ID, nama, shift, jam kerja, dan hari absen), menghitung gaji berdasarkan tarif shift, lembur, dan potongan, serta menampilkan laporan gaji yang rapi.

Analisis yang mendasari pemilihan metode pemrograman meliputi kebutuhan untuk mengelola data berulang (banyak karyawan), pengambilan keputusan berdasarkan kondisi (shift, lembur, absen), dan validasi input pengguna. Oleh karena itu, metode yang dipilih meliputi:

- Struktur Kontrol (if-else dan switch-case): Untuk menentukan tarif gaji berdasarkan shift dan menghitung lembur/potongan.
- Perulangan (for loop dan do-while): Untuk memproses data beberapa karyawan dan memvalidasi input.
- Array: Untuk menyimpan data karyawan secara terstruktur.
- Konstanta (final): Untuk menjaga nilai tetap seperti tarif dan batas jam kerja.

Pemilihan metode ini sesuai dengan modul pemrograman Java dasar II, yang menekankan penggunaan struktur kontrol, perulangan, dan array untuk menyelesaikan masalah pemrograman praktis. Selain itu ada beberapa detail yang tidak diberitahukan seperti berapa harga pada shift pagi, shift siang, dan shift sore, serta apakah nilai dari masing-masing shift akan dianggap sama. Oleh karena itu saya simpulkan sendiri saja harganya untuk masing-masing shift.

II. Variabel

No	Nama Variabel	Tipe Data	Fungsi
1	idKaryawan	String	Menyimpan array ID karyawan untuk identifikasi unik.
2	namaKaryawan	String	Menyimpan array nama karyawan untuk informasi identitas.
3	shiftKaryawan	Int	Menyimpan array shift kerja (1=Pagi, 2=Siang, 3=Malam) untuk menentukan tarif.
4	jamKerja	Int	Menyimpan array total jam kerja per minggu untuk perhitungan gaji.
5	hariAbsen	Int	Menyimpan array jumlah hari absen untuk menghitung potongan gaji.
6	gajiTotal	double	Menyimpan array total gaji setelah perhitungan tarif, lembur, dan potongan.
7	TARIF_PAGI	Final Double	Konstanta untuk tarif per jam shift pagi (Rp 25.000).
8	TARIF_SIANG	Final Double	Konstanta untuk tarif per jam shift siang (Rp 30.000).
9	TARIF_MALAM	Final Double	Konstanta untuk tarif per jam shift malam (Rp 35.000)
10	TARIF_LEMBUR	Final Double	Konstanta untuk pengali tarif lembur (1.5 kali tarif normal).
11	POTONGAN_ABSEN	Final Double	Konstanta untuk potongan gaji per hari absen (Rp 100.000).
12	JAM_NORMAL	Final Double	Konstanta untuk batas jam kerja normal (40 jam).
13	JAM_MINIMUM	Final Double	Konstanta untuk batas minimum jam kerja (30 jam).
14	input	Scanner	Objek untuk membaca input dari pengguna.
15	jumlahKaryawan	Int	Variabel untuk menyimpan jumlah karyawan yang akan diinput.
16	i	Int	Variabel indeks untuk perulangan pengolahan data karyawan.
17	tarifPerJam	Double	Variabel sementara untuk menyimpan tarif per jam berdasarkan shift.
18	gajiNormal	Double	Variabel untuk menyimpan gaji dari jam kerja normal.
19	gajiLembur	Double	Variabel untuk menyimpan gaji dari jam lembur.
20	totalGaji	Double	Variabel sementara untuk menghitung total gaji sebelum disimpan.

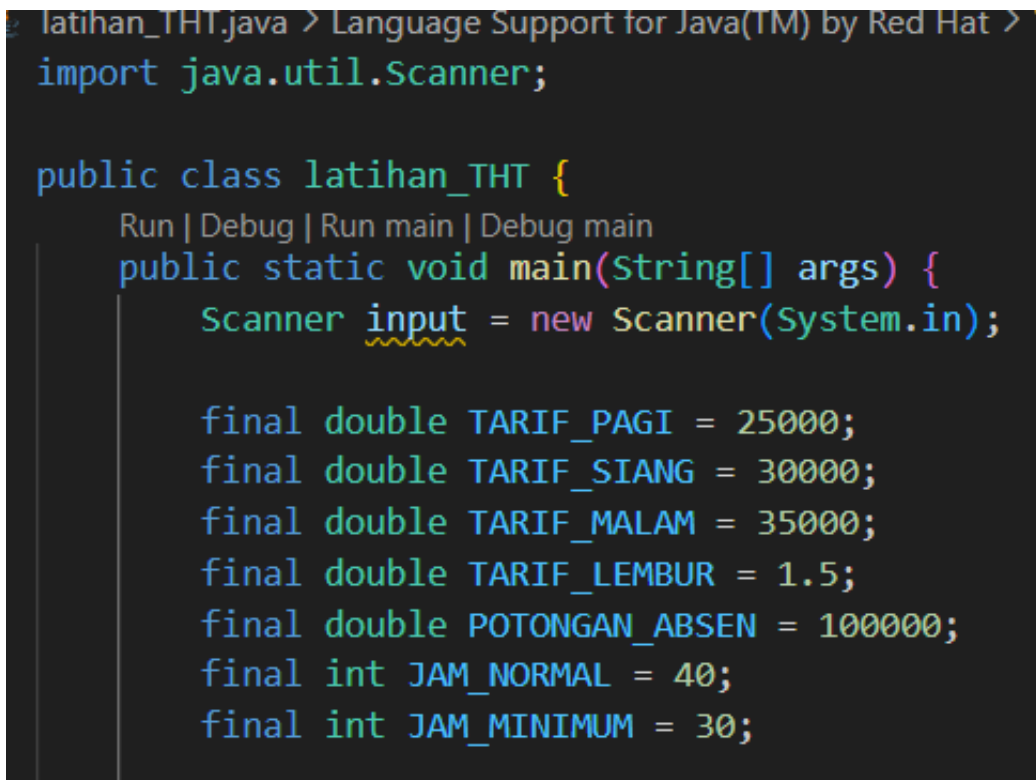
21.	shiftText	String	Variabel lokal tempat konversi nilai shift untuk kemudian akan di print
-----	-----------	--------	---

III. Constructor dan Method

Program ini merupakan versi yang sangat pemula jadi tidak ada konsep PBO yang diterapkan di dalam codingan ini sehingga hanya ada satu file dan satu method yaitu:

Nama Method	Jenis Method	Fungsi
main(String[] args)	Public Static	Fungsi utama yang menjalankan program, mengelola input data karyawan, perhitungan gaji, dan menampilkan laporan.

IV. Dokumentasi dan Pembahasan Code



```

latihan_THT.java > Language Support for Java(TM) by Red Hat >
import java.util.Scanner;

public class latihan_THT {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        final double TARIF_PAGI = 25000;
        final double TARIF_SIANG = 30000;
        final double TARIF_MALAM = 35000;
        final double TARIF_LEMBUR = 1.5;
        final double POTONGAN_ABSEN = 100000;
        final int JAM_NORMAL = 40;
        final int JAM_MINIMUM = 30;
    }
}

```

- Bagian Pertama ada import java.util.Scanner untuk menggunakan library scanner sehingga bisa meminta input dari user. Kemudian ada deklarasi final atau kalo di c++ itu biasa disebut variabel konstan yang artinya tidak bisa diubah-ubah nilainya oleh operasi aritmatika apapun. Karena ini menggunakan java maka syntaksnya berubah menggunakan final diikuti dengan tipe data dan nama variabel serta nilai dari variabel tersebut.

```

System.out.println(x:"=== Program Perhitungan Gaji Karyawan ===");
System.out.print(s:"Masukkan jumlah karyawan: ");
int jumlahKaryawan = input.nextInt();
input.nextLine();

String[] idKaryawan = new String[jumlahKaryawan];
String[] namaKaryawan = new String[jumlahKaryawan];
int[] shiftKaryawan = new int[jumlahKaryawan];
int[] jamKerja = new int[jumlahKaryawan];
int[] hariAbsen = new int[jumlahKaryawan];
double[] gajiTotal = new double[jumlahKaryawan];

for (int i = 0; i < jumlahKaryawan; i++) {
    System.out.print("Masukkan ID karyawan ke-" + (i + 1) + ": ");
    idKaryawan[i] = input.next();
    System.out.print("Masukkan nama karyawan ke-" + (i + 1) + ": ");
    namaKaryawan[i] = input.next();
    System.out.print("Masukkan shift karyawan ke-" + (i + 1) + ": ");
    shiftKaryawan[i] = input.nextInt();
    System.out.print("Masukkan jam kerja karyawan ke-" + (i + 1) + ": ");
    jamKerja[i] = input.nextInt();
    System.out.print("Masukkan hari absen karyawan ke-" + (i + 1) + ": ");
    hariAbsen[i] = input.nextInt();
    System.out.print("Masukkan gaji karyawan ke-" + (i + 1) + ": ");
    gajiTotal[i] = input.nextDouble();
    input.nextLine();
}

```

- Berikutnya setelah mendeklarasikan variabel-variabel konstannya program akan berlanjut dengan meminta inputan jumlah karyawan yang ingin di data. Dan kemudian membuat array sebanyak jumlah karyawan untuk menyimpan ide, nama, shift, jam, berapa hari absen, dan gaji total.
- Kemudian, setelah ini program akan berlanjut ke dalam looping yang sangat panjang dimana hampir semua proses penghitungan akan berlangsung di dalam looping ini sehingga yang tersisa untuk dilakukan pada saat selesai melakukan looping adalah untuk melakukan print hasil dengan rapih

```

for (int i = 0; i < jumlahKaryawan; i++) {
    System.out.println("\nData Karyawan ke-" + (i + 1));
    System.out.print(s:"Masukkan ID Karyawan: ");
    idKaryawan[i] = input.nextLine();

    System.out.print(s:"Masukkan Nama Karyawan: ");
    namaKaryawan[i] = input.nextLine();

    do {
        System.out.print(s:"Masukkan Shift Kerja (1=Pagi, 2=Siang, 3=Malam): ");
        shiftKaryawan[i] = input.nextInt();
        if (shiftKaryawan[i] < 1 || shiftKaryawan[i] > 3) {
            System.out.println(x:"Shift tidak valid! Masukkan 1, 2, atau 3.");
        }
    } while (shiftKaryawan[i] < 1 || shiftKaryawan[i] > 3);

    do {
        System.out.print(s:"Masukkan Total Jam Kerja (0-70 jam): ");
        jamKerja[i] = input.nextInt();
        if (jamKerja[i] < 0 || jamKerja[i] > 70) {
            System.out.println(x:"Jam kerja tidak valid! Masukkan antara 0-70 jam.");
        }
    } while (jamKerja[i] < 0 || jamKerja[i] > 70);

    do {
        System.out.print(s:"Masukkan Jumlah Hari Absen (0-7 hari): ");
        hariAbsen[i] = input.nextInt();
        if (hariAbsen[i] < 0 || hariAbsen[i] > 7) {
            System.out.println(x:"Hari absen tidak valid! Masukkan antara 0-7 hari.");
        }
    } while (hariAbsen[i] < 0 || hariAbsen[i] > 7);
}

```

- Bagian berikut adalah melakukan looping untuk meminta inputan file oleh user. Pada kasus ini saya menggunakan do-while untuk memastikan inputan user telah sesuai. Mengapa do-while karena saya ingin user memasukkan inputannya terlebih dahulu kemudian akan melakukan pengecekan. Oleh karena itu do-while akan sangat efektif.

```

input.nextLine();
double tarifPerJam = 0;
switch (shiftKaryawan[i]) {
    case 1:
        tarifPerJam = TARIF_PAGI;
        break;
    case 2:
        tarifPerJam = TARIF_SIANG;
        break;
    case 3:
        tarifPerJam = TARIF_MALAM;
        break;
}

```

- Berikutnya ada switch-case untuk memberikan nilai tarif per jam untuk menghitung gaji karyawan. Berdasarkan pilihan shift 1, 2, atau 3 yang telah diminta sebelumnya kepada user.

```

double gajiNormal = 0;
double gajiLembur = 0;

if (jamKerja[i] > JAM_NORMAL) {
    gajiNormal = JAM_NORMAL * tarifPerJam;
    gajiLembur = (jamKerja[i] - JAM_NORMAL) * tarifPerJam * TARIF_LEMBUR;
} else {
    gajiNormal = jamKerja[i] * tarifPerJam;
}

double totalGaji = gajiNormal + gajiLembur;

if (jamKerja[i] < JAM_MINIMUM) {
    totalGaji = totalGaji * 0.9;
}

totalGaji -= hariAbsen[i] * POTONGAN_ABSEN;

gajiTotal[i] = (totalGaji < 0) ? 0 : totalGaji;
}

```

- Pada akhir looping akan dilakukan perhitungan lebih lanjut untuk mencari gaji total [i]. Pada awalnya akan digunakan variabel sementara yaitu total gaji. Total gaji sendiri di dapat dari perhitungan gaji lembur yang kemudian dijumlahkan dengan gaji normal.

```

System.out.println(x: "\n=== Laporan Gaji Karyawan ===");
System.out.printf(format: "%-10s %-20s %-10s %-15s %-10s %-15s\n", ...args: "ID", "Nama", "Shift", "Jam Kerja", "Absen", "Total Gaji");
System.out.println(x: "-----");

for (int i = 0; i < jumlahKaryawan; i++) {
    String shiftText = "";
    switch (shiftKaryawan[i]) {
        case 1: shiftText = "Pagi"; break;
        case 2: shiftText = "Siang"; break;
        case 3: shiftText = "Malam"; break;
    }

    System.out.printf(format: "%-10s %-20s %-10s %-15d %-10d Rp %-15.2f\n",
        idKaryawan[i], namaKaryawan[i], shiftText, jamKerja[i], hariAbsen[i], gajiTotal[i]);
}
input.close();

```

- Codingan pun diakhiri dengan menggunakan syntaks printf ” %-10s %-20s %-10s %-15s %-10s %-15s\n” untuk mengatur kerapihan dari tabel yang ingin di print. Printf bertujuan untuk mengeluarkan output dalam format tertentu. Arti dari masing-masing simbol pada syntaks yang digunakan untuk merapihkan tabel tersebut adalah sebagai berikut:
 - ✓ S= itu untuk string, d itu untuk integer, dan f itu untuk float.
 - ✓ % menandakan awal mula formating
 - ✓ - menunjukkan agar teks dibuat rata kiri
 - ✓ 10 itu menandakan lebar karakter
 - ✓ Khusus apabila tipe datanya float kita bisa menambahkan .2f dibelakangnya untuk menyingkronkan ketelitian berapa angka di belakang komanya. Jika mau 2 angka dibelakang koma cukup gunakan .2f

- Kemudian ada juga looping dengan for loop untuk mengubah nilai dari shiftkaryawan yang tadinya berupa angka menjadi huruf di translate sesuai dengan huruf yang diinputkan oleh user. Kemudian pada akhirnya setelah diterjemahkan maka shiftText akan di print beserta dengan id karyawan, nama karyawan, hari absen, jam kerja, dan total gaji.

V. Kesimpulan

Praktikum ini berhasil mengimplementasikan program perhitungan gaji karyawan menggunakan konsep pemrograman Java dasar seperti struktur kontrol (if-else, switch-case), perulangan (for, do-while), array, dan konstanta (final). Program memenuhi kebutuhan studi kasus dengan menghitung gaji berdasarkan shift, lembur, dan potongan absen, serta menampilkan laporan yang terstruktur. Penggunaan validasi input meningkatkan keandalan, meskipun masih bisa ditingkatkan dengan penanganan error lebih lanjut. Pengalaman ini memperkuat pemahaman tentang penerapan logika pemrograman dalam menyelesaikan masalah praktis. Sekian dan Terimakasih

VI. Daftar Pustaka

Modul 3: Pemograman Jawa Dasar II