

**LAPORAN PRAKTIKUM  
PEMOGRAMAN BERORIENTASI OBJEK  
MODUL III  
PEMROGRAMAN JAVA DASAR II**



**Disusun Oleh:  
Rippun Pirdasari 105223027**

**LABORATORIUM JARINGAN KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS SAINS DAN ILMU KOMPUTER  
UNIVERSITAS PERTAMINA  
2025**

## 1.Pendahuluan

Program ini dirancang untuk menghitung gaji karyawan di sebuah pabrik yang menerapkan sistem kerja dalam tiga shift, yaitu pagi, siang, dan malam. Setiap karyawan memiliki jumlah jam kerja yang bervariasi dalam seminggu, sehingga penghitungan gaji harus mempertimbangkan beberapa faktor seperti jumlah jam kerja, lembur, dan ketidakhadiran.

Gaji pokok karyawan dihitung berdasarkan tarif per jam yang berbeda untuk setiap shift kerja. Jika seorang karyawan bekerja lebih dari 40 jam dalam seminggu, maka jam kerja yang melebihi batas tersebut akan dihitung sebagai lembur dengan upah tambahan sebesar 1,5 kali dari tarif normal. Sebaliknya, jika karyawan bekerja kurang dari 30 jam dalam seminggu, maka gaji yang diterima akan dipotong sebesar 10% sebagai konsekuensi dari kurangnya jam kerja.

Selain itu, program juga memperhitungkan jumlah hari absen seorang karyawan. Jika karyawan tidak masuk kerja tanpa alasan yang jelas, maka mereka akan dikenakan potongan gaji sebesar Rp100.000 untuk setiap hari absen. Program ini juga memastikan bahwa data yang dimasukkan valid, dengan membatasi jumlah jam kerja maksimal dalam seminggu hingga 168 jam (jumlah jam dalam satu minggu penuh) dan jumlah hari absen maksimal 7 hari.

Melalui program ini, data karyawan yang meliputi ID, nama, shift kerja, total jam kerja, dan jumlah hari absen akan dicatat. Setelah semua data dimasukkan, program akan menghitung total gaji yang berhak diterima oleh karyawan setelah mempertimbangkan tarif per jam, lembur, serta potongan akibat absen atau kurangnya jam kerja. Output akhir dari program adalah laporan gaji karyawan yang mencantumkan informasi lengkap mengenai total penghasilan mereka dalam satu minggu kerja.

## II.Variabel

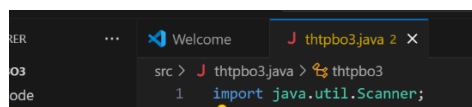
NO	Nama Variabel	Tipe Data	Fungsi
1.	Karyawanid	int	Menyimpan id karyawan dalam bentuk angka
2.	Jumlah Absen	int	Menyimpan jumlah hari karyawan tidak masuk kerja.
3.	Total Jam	double	Menyimpan jumlah jam kerja karyawan dalam seminggu.
4.	upahperjam	double	Menyimpan nilai upah per jam untuk setiap shift kerja

5.	Total gaji	double	Menyimpan total gaji yang dihitung berdasarkan jam kerja, lembur, dan potongan absen.
6.	Jam tambahan	double	Menyimpan jumlah jam lembur jika jam kerja lebih dari 40 jam.
7.	Upah lembur	double	Menyimpan tarif lembur yang dihitung berdasarkan upah per jam.
8.	Karyawan nama	string	Menyimpan nama karyawan dalam bentuk teks.
9.	Jadwal kerja	string	Menyimpan shift kerja karyawan (pagi, siang, malam).
10.	Scanner	Scanner	Objek untuk membaca input dari pengguna

### III. Constructor dan Method

NO	Nama Method	Jenis method	Fungsi
1.	Karyawan()	Constructor default (tanpa parameter)	Menginisialisasi objek Karyawan dengan nilai awal default.
2.	Karyawan(int id, String nama, String shift, double jam, int absen)	Constructor parameterized (dengan parameter)	Menginisialisasi objek Karyawan dengan data karyawan seperti ID, nama, shift, jam kerja, dan jumlah absen.
3.	main(String[] args)	static, void	Metode utama program yang menangani input dan output.
4.	hitungGaji()	double	Menghitung gaji berdasarkan jam kerja, lembur, dan absen.
5.	tampilkanGaji()	Non-static	Menampilkan laporan gaji karyawan.

### IV. Dokumentasi dan Pembahasan Code



```

src > J thtpbo3.java > thtpbo3
ode  1  import java.util.Scanner;

```

dalam program Java digunakan untuk mengimpor kelas Scanner dari paket java.util, yang memungkinkan program membaca input dari berbagai sumber, seperti keyboard, file, atau stream lainnya. Dengan menggunakan Scanner, program dapat menerima input dari pengguna secara langsung saat runtime, sehingga meningkatkan interaktivitas dan fleksibilitas aplikasi.

```
public class thtpbo3{  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print(s:"Masukkan ID: ");  
        int karyawanID = scanner.nextInt();  
        scanner.nextLine();  
        System.out.print(s:"Masukkan nama: ");  
        String karyawanNama = scanner.nextLine();  
        System.out.print(s:"Shift kerja: ");  
        String jadwalKerja = scanner.nextLine();  
        System.out.print(s:"Total jam kerja dalam seminggu: ");  
        double totalJam = scanner.nextDouble();  
        System.out.print(s:"Jumlah hari absen: ");  
        int jumlahAbsen = scanner.nextInt();  
    }  
}
```

Program ini menggunakan metode main, yang merupakan titik awal eksekusi dalam Java. Di dalam metode main, terdapat objek Scanner yang dideklarasikan dengan new Scanner(System.in), yang berfungsi untuk membaca input dari pengguna melalui keyboard. Program meminta beberapa input dari pengguna menggunakan System.out.print(), seperti ID karyawan (int karyawanID), nama karyawan (String karyawanNama), shift kerja (String jadwalKerja), total jam kerja dalam seminggu (double totalJam), dan jumlah hari absen (int jumlahAbsen). Untuk menangani jenis data yang berbeda, program menggunakan metode nextInt(), nextDouble(), dan nextLine() dari objek Scanner. Ada juga pemanggilan scanner.nextLine() setelah membaca nextInt() untuk menghindari masalah dengan buffer input. Secara keseluruhan, kode ini berfungsi untuk mengumpulkan data dari pengguna yang nantinya dapat digunakan untuk perhitungan gaji atau keperluan lainnya.

```
3   public class thtpbo3{  
4       public static void main(String[] args) throws Exception {  
19  
20           if (totalJam < 0 || totalJam > 168 || jumlahAbsen < 0) {  
21               System.out.println(x:"Error: Input tidak valid");  
22               return;  
23           }  
}
```

pemeriksaan validasi input menggunakan struktur percabangan if. Kode ini berada di dalam metode main dari kelas thtpbo3. Kondisi dalam if memeriksa apakah nilai totalJam kurang dari 0, lebih dari 168, atau jika jumlahAbsen kurang dari 0. Jika salah satu dari kondisi tersebut terpenuhi, maka program akan mencetak pesan kesalahan **"Error: Input tidak valid"** menggunakan System.out.println(). Namun, terdapat kesalahan sintaks pada pemanggilan System.out.println(x:"Error: Input tidak valid"), karena parameter x: tidak valid dalam Java. Seharusnya cukup System.out.println("Error: Input tidak valid"). Setelah mencetak pesan

kesalahan, perintah `return;` digunakan untuk menghentikan eksekusi program lebih awal.

```
double upahPerJam;
switch (jadwalKerja) {
    case "pagi":
        upahPerJam = 60000;
        break;
    case "siang":
        upahPerJam = 65000;
        break;
    case "malam":
        upahPerJam = 70000;
        break;
    default:
        System.out.println(x:"Error: Shift kerja tidak valid");
        return;
}
```

struktur kontrol switch untuk menentukan nilai upah per jam berdasarkan shift kerja karyawan. Variabel `jadwalKerja` diperiksa dalam beberapa case, yaitu "pagi", "siang", dan "malam". Jika `jadwalKerja` bernilai "pagi", maka `upahPerJam` akan diatur menjadi 60.000, jika "siang", maka `upahPerJam` bernilai 65.000, dan jika "malam", maka `upahPerJam` akan menjadi 70.000. Masing-masing kasus diakhiri dengan perintah `break;`, yang berfungsi untuk menghentikan eksekusi lebih lanjut dalam switch. Jika `jadwalKerja` tidak sesuai dengan ketiga opsi yang tersedia, maka blok default akan dieksekusi. Dalam default, program akan mencetak pesan "Error: Shift kerja tidak valid" menggunakan `System.out.println()`. Namun, terdapat kesalahan sintaks dalam pemanggilan `System.out.println(x:"Error: Shift kerja tidak valid");`, karena `x:` bukan bagian dari sintaks yang benar dalam Java. Seharusnya, perintah yang benar adalah `System.out.println("Error: Shift kerja tidak valid");`. Setelah mencetak pesan kesalahan, perintah `return;` digunakan untuk menghentikan eksekusi program lebih lanjut, mencegah penggunaan nilai `upahPerJam` yang tidak valid.

```
9         }
10
11         double totalGaji = totalJam * upahPerJam;
12         if (totalJam > 40) {
13             double jamTambahan = totalJam - 40;
14             double upahLembur = upahPerJam * 1.5;
15             totalGaji += jamTambahan * upahLembur;
16         } else if (totalJam < 30) {
17             totalGaji *= 0.9;
18         }
19
20         double potongan = jumlahAbsen * 100000;
21         totalGaji -= potongan;
```

Perhitungan gaji diawali dengan mengalikan total jam kerja (`totalJam`) dengan upah per jam (`upahPerJam`) dan menyimpannya dalam variabel `totalGaji`. Selanjutnya, terdapat kondisi `if` yang memeriksa apakah karyawan bekerja lebih dari 40 jam dalam seminggu. Jika iya, maka jumlah jam

kerja yang melebihi 40 jam dihitung sebagai jam lembur (jamTambahan), dan tarif upah lembur dihitung dengan mengalikan upahPerJam sebesar 1.5. Gaji tambahan dari lembur kemudian ditambahkan ke totalGaji. Selain itu, jika seorang karyawan bekerja kurang dari 30 jam dalam seminggu, maka total gaji dikurangi sebesar 10% dengan operasi  $\text{totalGaji} *= 0.9$ . Setelah menghitung gaji berdasarkan jam kerja, program juga menghitung potongan gaji akibat ketidakhadiran (jumlahAbsen). Potongan ini diperoleh dengan mengalikan jumlah hari absen dengan Rp100.000 dan disimpan dalam variabel potongan. Terakhir, total gaji dikurangi dengan nilai potongan tersebut menggunakan operasi  $\text{totalGaji} -= \text{potongan}$ , yang menyesuaikan gaji akhir setelah memperhitungkan jumlah hari absen karyawan.

```
System.out.println(x:"\n===== Laporan Gaji Karyawan =====");
System.out.println("ID Karyawan      : " + karyawanID);
System.out.println("Nama Karyawan    : " + karyawanNama);
System.out.println("Shift Kerja      : " + jadwalKerja);
System.out.println("Total Jam Kerja   : " + totalJam);
System.out.println("Jumlah Hari Absen : " + jumlahAbsen);
System.out.println("Total Gaji        : Rp" + totalGaji);

scanner.close();
}
```

mencetak laporan gaji karyawan ke layar. Output laporan diawali dengan teks pemisah "===== Laporan Gaji Karyawan =====" untuk memberikan tampilan yang lebih rapi. Selanjutnya, beberapa perintah `System.out.println` digunakan untuk menampilkan informasi penting mengenai karyawan, seperti ID karyawan (`karyawanID`), nama karyawan (`karyawanNama`), shift kerja (`jadwalKerja`), total jam kerja (`totalJam`), jumlah hari absen (`jumlahAbsen`), dan total gaji yang diterima (`totalGaji`).

Format tampilan dibuat dengan menambahkan tanda titik dua (:) agar lebih mudah dibaca. Total gaji juga dicetak dengan tambahan "Rp" agar menunjukkan bahwa nilai tersebut dalam mata uang rupiah. Setelah seluruh informasi ditampilkan, program memanggil `scanner.close()` untuk menutup objek Scanner, yang sebelumnya digunakan untuk menerima input dari pengguna. `Scanner.close` adalah praktik yang baik untuk menghindari kebocoran sumber daya dalam program.