

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL III
PEMROGRAMAN JAVA DASAR II



Disusun Oleh:

Florence Regis Lake 105223006

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA
2025

I. PENDAHULUAN

Sebagai seorang pengembang perangkat lunak, anda diminta untuk membuat program yang menghitung gaji karyawan di sebuah pabrik yang memiliki sistem kerja dalam tiga shift: pagi, siang, dan malam. Setiap karyawan memiliki jam kerja yang berbeda-beda dalam seminggu dan bisa saja mengalami lembur atau bahkan absen. Gaji dihitung berdasarkan jumlah jam kerja dalam seminggu, dengan tarif per jam yang berbeda tergantung pada shift kerja. Jika karyawan bekerja lebih dari 40 jam, maka kelebihan jam dihitung sebagai lembur dengan bayaran lebih tinggi. Sebaliknya, jika mereka bekerja kurang dari 30 jam dalam seminggu, gaji mereka akan dipotong sebesar 10%. Selain itu, jika karyawan tidak masuk kerja tanpa alasan, mereka akan dikenakan potongan gaji sebesar Rp100.000 per hari absen.

Berdasarkan kebutuhan yang tersebut, terdapat beberapa fitur yang diinginkan dalam program seperti mencatat data karyawan seperti ID, nama, shift kerja, total jam kerja dalam seminggu, serta jumlah hari absen. Program ini juga harus menghitung total gaji mereka setelah mempertimbangkan upah, lembur, dan potongan yang berlaku.

Sebagai pengembang, algoritma yang digunakan diharapkan dapat berupa perulangan untuk meminta data dari pengguna, lalu gunakan aturan yang sesuai untuk menghitung gaji. Pastikan program dapat menangani kesalahan input, seperti jika pengguna memasukkan jam kerja yang tidak masuk akal atau jumlah hari absen yang berlebihan. Setelah semua data dimasukkan, program harus menampilkan laporan gaji yang rapi dan mudah dibaca.

II. VARIABEL

No	Nama Variabel	Tipe data	Fungsi
1	jumlahKaryawan	int	Menyimpan value banyak karyawan
2	karyawanData	String	Array yang menyimpan data ID, nama, dan shift
3	dataAbsen	int	Array yang menyimpan data jam kerja, dan banyaknya absen
4	gajiPerJam	int	Sebagai inisiasi jumlah gaji per jam berdasarkan shift
5	gaji	double	Menyimpan akumulasi gaji yang diterima karyawan
6	jamLembur	int	Menghitung banyaknya jam lembur, dan mengakumulasikannya sebagai gaji tambahan
7	gajiList	double	Array yang menyimpan gaji setiap karyawan
8	I	int	Sebagai iterasi perulangan

III. DOKUMENTASI DAN PEMBAHASAN CODE

3.1. Variabel utama

Variabel utama yang digunakan pada program ini, seperti yang telah dijelaskan fungsinya pada bab sebelumnya dapat terlihat pada potongan code berikut ini:

```
7      int jumlahKaryawan;  
8  
9      System.out.print(s:"Masukkan jumlah karyawan: ");  
10     jumlahKaryawan = input.nextInt();  
11     input.nextLine();  
12  
13     String[][] karyawanData = new String[jumlahKaryawan][3];  
14     int[][] dataAbsen = new int[jumlahKaryawan][2];  
15     double[] gajiList = new double[jumlahKaryawan];  
16     int gajiPerJam;  
17     double gaji;  
18     int jamLembur;
```

Gambar 3.1 variabel

Variabel yang ditunjukkan merupakan variabel utama yang digunakan dalam program ini, namun terdapat juga beberapa variabel pendukung seperti variabel *i* yang bertipe data int sebagai iterasi perulangan.

3.2. Input pengguna

Sesuai dengan kebutuhan, program diminta untuk dapat menerima input data karyawan seperti *banyaknya karyawan*, *ID*, *nama*, *shift kerja dalam seminggu*, dan *jumlah hari absen*. Oleh karena itu, dibuatlah code yang dapat menanyakan informasi yang dibutuhkan, kemudian memungkinkan pengguna untuk memasukkan input yang sesuai.

```
9      System.out.print(s:"Masukkan jumlah karyawan: ");  
10     jumlahKaryawan = input.nextInt();  
11     input.nextLine();
```

Gambar 3.2 input jumlah karyawan

```
20     for (int i = 0; i < jumlahKaryawan; i++) {  
21         System.out.println("\nData Karyawan " + (i + 1) + ":");  
22         System.out.print(s:"ID: ");  
23         karyawanData[i][0] = input.nextLine();  
24         System.out.print(s:"Nama: ");  
25         karyawanData[i][1] = input.nextLine();  
26         System.out.print(s:"Shift (pagi/siang/malam): ");  
27         karyawanData[i][2] = input.nextLine();
```

Gambar 3.3 input data karyawan

```

44     do {
45         System.out.print(s:"Total jam kerja dalam seminggu (0-80): ");
46         dataAbsen[i][0] = input.nextInt();
47         if (dataAbsen[i][0] < 0 || dataAbsen[i][0] > 80) {
48             System.out.println(x:"Jam kerja invalid! Harap masukkan angka antara 0-80.");
49         }
50     } while (dataAbsen[i][0] < 0 || dataAbsen[i][0] > 80);
51
52     do {
53         System.out.print(s:"Jumlah hari absen dalam seminggu (0-7): ");
54         dataAbsen[i][1] = input.nextInt();
55         if (dataAbsen[i][1] < 0 || dataAbsen[i][1] > 7) {
56             System.out.println(x:"Jumlah hari absen invalid! Harap masukkan angka antara 0-7.");
57         }
58     } while (dataAbsen[i][1] < 0 || dataAbsen[i][1] > 7);

```

Gambar 3.4 input total jam kerja dan jumlah hari absen

Pada input pada Gambar 3.2, program diharapkan dapat menerima input *banyaknya karyawan* yang nantinya akan digunakan untuk menentukan ukuran array pada variabel *karyawanData*, *dataAbsen*, dan *gajiList*. Untuk menerima input data karyawan, potongan code-nya dapat terlihat pada Gambar 3.3 dimana input tersebut dimasukan kedalam array *karyawanData*. Lalu untuk menerima input *shift kerja dalam seminggu* dan *jumlah hari absen* dapat terlihat pada Gambar 3.4 dan hasil input tersebut akan dimasukan kedalam variabel array *dataAbsen* dengan menggunakan sintaks do-while.

3.3. Pertimbangan upah

Terdapat beberapa hal yang dapat mempengaruhi gaji, seperti upah shift, lembur, dan potongan gaji dalam menghitung pertimbangan tersebut, maka dibuatlah potongan code berikut yang dapat memenuhi kebutuhan itu.

```

29     switch (karyawanData[i][2].toLowerCase()) {
30         case "pagi":
31             gajiPerJam = 50000;
32             break;
33         case "siang":
34             gajiPerJam = 55000;
35             break;
36         case "malam":
37             gajiPerJam = 60000;
38             break;
39         default:
40             System.out.println(x:"Shift invalid! Menggunakan shift pagi sebagai default.");
41             gajiPerJam = 50000;
42     }

```

Gambar 3.6 upah shift

Upah tersebut diberikan atas jadwal shift pegawai, dengan menggunakan sintaks switch-case dan jika input yang dimasukan tidak sesuai, maka shift akan diubah secara default ke jadwal pagi.

```

63     if (dataAbsen[i][0] > 40) {
64         jamLembur = dataAbsen[i][0] - 40;
65         gaji += jamLembur * gajiPerJam * 1.5;
66     }
67
68     if (dataAbsen[i][0] < 30) {
69         gaji *= 0.9;
70     }
71
72     gaji -= dataAbsen[i][1] * 100000;
73     gajiList[i] = gaji;

```

Gambar 3.7 pertimbangan upah

Pada Gambar 3.7 tersebut, upah yang dijadikan pertimbangan adalah jam *total jam kerja* yang disimpan pada array *dataAbsen[i][0]* dimana jika melebihi 40 jam dalam seminggu maka akan mendapatkan bonus, tetapi jika dibawah 30 jam seminggu maka akan dikenai potongan. Kemudian gaji akan dikurangi berdasarkan jadwal shift dikalikan 100000. Dan pada akhirnya gaji dari setiap karyawan disimpan pada array *gajiList[i]*.

3.4. Output

Permintaan terakhir yang ingin dipenuhi dari program adalah agar program mampu menampilkan laporan gaji dalam format yang rapi. Dengan adanya permintaan tersebut, maka program yang dapat menyesuaikan kebutuhan yang diinginkan.

```

76     System.out.println(x:"\nLaporan Gaji Karyawan:");
77     for (int i = 0; i < jumlahKaryawan; i++) {
78         System.out.printf(format:"ID: %s | Nama: %s | Shift: %s | Jam Kerja: %d | Absen: %d | Gaji: Rp%.2f\n",
79             karyawanData[i][0], karyawanData[i][1], karyawanData[i][2], dataAbsen[i][0], dataAbsen[i][1], gajiList[i]);
80     }

```

Gambar 3.8 code output

Dari potongan kode output yang telah ditampilkan pada Gambar 3.8, hasil dari code output tersebut nantinya dapat terlihat pada contoh sebagai berikut:

```

Masukkan jumlah karyawan: 1

Data Karyawan 1:
ID: 001
Nama: flo
Shift (pagi/siang/malam): pagi
Total jam kerja dalam seminggu (0-80): 32
Jumlah hari absen dalam seminggu (0-7): 0

Laporan Gaji Karyawan:
ID: 001 | Nama: flo | Shift: pagi | Jam Kerja: 32 | Absen: 0 | Gaji: Rp1600000.00

```

Gambar 3.9 contoh output

IV.KESIMPULAN

Berdasarkan permintaan dan kebutuhan yang ingin dicapai dan dilaksanakan oleh program, dan setelah terlihat hasil akhir atau output yang telah dibuat pada program ini. Maka dapat dikatakan bahwa program yang dibuat berhasil memenuhi kebutuhan yang diinginkan sesuai dengan permintaan tanpa adanya penambahan fitur maupun pengurangan fitur sehingga jalannya program dapat dilaksanakan secara efisien dan efektif.

