

HANDS-ON LABS 3 PEMROGRAMAN BERORIENTASI OBJEK + LAB



Oleh:

Martogi Jekson C. Siagian 2105541054

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

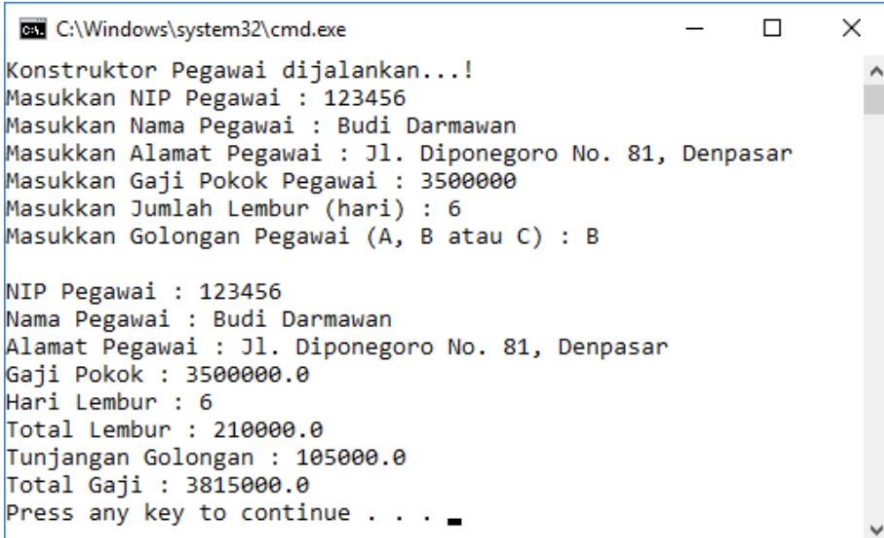
UNIVERSITAS UDAYANA

2023

Hands-On Labs 3

Tugas

1. Buatlah 2 buah kelas yang terdiri dari class Pegawai dan class Utama. Class Utama digunakan untuk memanggil class Pegawai yang berisi method main. Pada class Pegawai di dalamnya terdapat fungsi-fungsi asesor dan mutator yang digunakan untuk menyimpan dan merubah informasi dalam kelas tersebut. Ketentuan:
 - Seorang pegawai memiliki minimal atribut: NIP, nama, alamat, gaji pokok
 - Seorang pegawai memiliki gaji pokok, dan apabila seorang pegawai lembur, maka seorang pegawai akan mendapatkan uang lembur sebesar: jumlah hari lembur x 1% x gaji pokok dan apabila tidak lembur maka tidak mendapatkan uang lembur.
 - Seorang pegawai juga memiliki golongan. Setiap golongan mendapatkan tunjangan yang berbeda-beda. Berikut pembagian besarnya tunjangan setiap golongan.
 - Golongan A = 1,5% x gaji pokok
 - Golongan B = 3% x gaji pokok
 - Golongan C = 5% x gaji pokok
 - Buatlah agar user nantinya dapat menginputkan seluruh data yang diperlukan (input dari command line). Berikut ini adalah ilustrasi dari proses input dan output dari program.



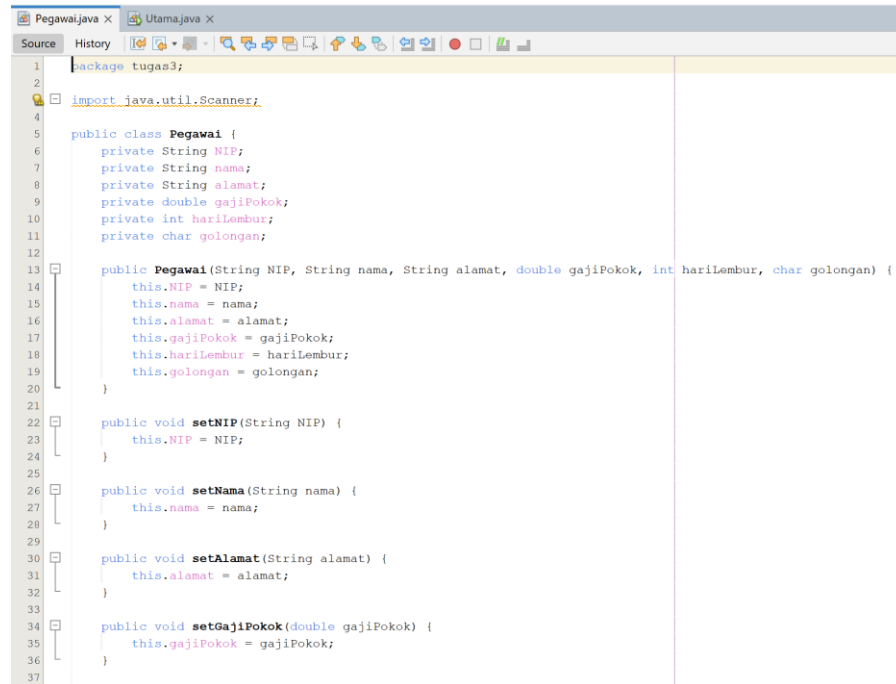
```
C:\Windows\system32\cmd.exe
Konstruktor Pegawai dijalankan...!
Masukkan NIP Pegawai : 123456
Masukkan Nama Pegawai : Budi Darmawan
Masukkan Alamat Pegawai : Jl. Diponegoro No. 81, Denpasar
Masukkan Gaji Pokok Pegawai : 3500000
Masukkan Jumlah Lembur (hari) : 6
Masukkan Golongan Pegawai (A, B atau C) : B

NIP Pegawai : 123456
Nama Pegawai : Budi Darmawan
Alamat Pegawai : Jl. Diponegoro No. 81, Denpasar
Gaji Pokok : 3500000.0
Hari Lembur : 6
Total Lembur : 210000.0
Tunjangan Golongan : 105000.0
Total Gaji : 3815000.0
Press any key to continue . . . █
```

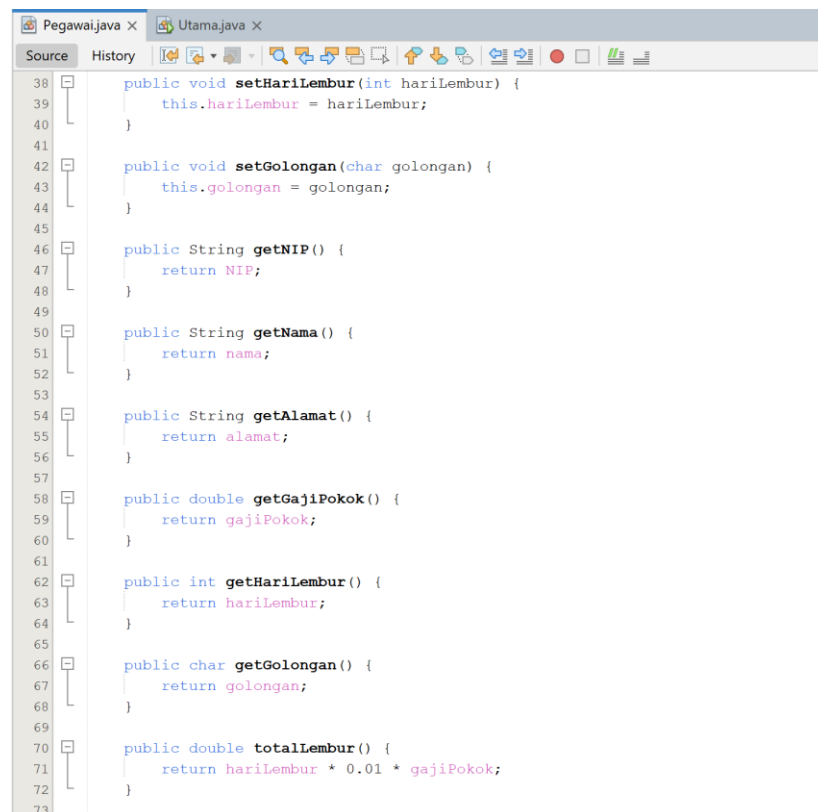
Jawab:

Untuk membuat program tersebut saya telah membuat 2 class yaitu class Pegawai & class Utama yang dimana kelas Pegawai berisi tentang methodnya dan kelas Utama untuk memanggil class Pegawai

❖ Kode class Pegawai



```
1 package tugas3;
2
3 import java.util.Scanner;
4
5 public class Pegawai {
6     private String NIP;
7     private String nama;
8     private String alamat;
9     private double gajiPokok;
10    private int hariLembur;
11    private char golongan;
12
13    public Pegawai(String NIP, String nama, String alamat, double gajiPokok, int hariLembur, char golongan) {
14        this.NIP = NIP;
15        this.nama = nama;
16        this.alamat = alamat;
17        this.gajiPokok = gajiPokok;
18        this.hariLembur = hariLembur;
19        this.golongan = golongan;
20    }
21
22    public void setNIP(String NIP) {
23        this.NIP = NIP;
24    }
25
26    public void setName(String nama) {
27        this.nama = nama;
28    }
29
30    public void setAlamat(String alamat) {
31        this.alamat = alamat;
32    }
33
34    public void setGajiPokok(double gajiPokok) {
35        this.gajiPokok = gajiPokok;
36    }
37 }
```



```
38 public void setHariLembur(int hariLembur) {
39     this.hariLembur = hariLembur;
40 }
41
42 public void setGolongan(char golongan) {
43     this.golongan = golongan;
44 }
45
46 public String getNIP() {
47     return NIP;
48 }
49
50 public String getName() {
51     return nama;
52 }
53
54 public String getAlamat() {
55     return alamat;
56 }
57
58 public double getGajiPokok() {
59     return gajiPokok;
60 }
61
62 public int getHariLembur() {
63     return hariLembur;
64 }
65
66 public char getGolongan() {
67     return golongan;
68 }
69
70 public double totalLembur() {
71     return hariLembur * 0.01 * gajiPokok;
72 }
73 }
```

```
Pegawai.java X  Utama.java X
Source History
74 public double tunjanganGolongan() {
75     double tunjangan = 0;
76     switch (golongan) {
77         case 'A':
78             tunjangan = gajiPokok * 0.015;
79             break;
80         case 'B':
81             tunjangan = gajiPokok * 0.03;
82             break;
83         case 'C':
84             tunjangan = gajiPokok * 0.05;
85             break;
86     }
87     return tunjangan;
88 }
89
90 public double totalGaji() {
91     return gajiPokok + totalLembur() + tunjanganGolongan();
92 }
93
94 public void printInfo() {
95     System.out.println("\n");
96     System.out.println("NIP Pegawai : " + NIP);
97     System.out.println("Nama Pegawai : " + nama);
98     System.out.println("Alamat Pegawai : " + alamat);
99     System.out.println("Gaji Pokok : " + gajiPokok + ".0");
100    System.out.println("Hari Lembur : " + hariLembur);
101    System.out.println("Total Lembur : " + totalLembur() + ".0");
102    System.out.println("Tunjangan Golongan : " + tunjanganGolongan() + ".0");
103    System.out.println("Total Gaji : " + totalGaji() + ".0");
104 }
105 }
106 }
```

❖ Kode class Utama

```
Pegawai.java X  Utama.java X
Source History
1 package tugas3;
2
3 import java.util.Scanner;
4
5 public class Utama {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         System.out.println("Konstruktor Pegawai dijalankan...!");
9
10        System.out.print("Masukkan NIP Pegawai: ");
11        String NIP = scanner.nextLine();
12
13        System.out.print("Masukkan Nama Pegawai: ");
14        String nama = scanner.nextLine();
15
16        System.out.print("Masukkan Alamat Pegawai: ");
17        String alamat = scanner.nextLine();
18
19        System.out.print("Masukkan Gaji Pokok Pegawai: ");
20        double gajiPokok = scanner.nextDouble();
21
22        System.out.print("Masukkan Hari Lembur Pegawai: ");
23        int hariLembur = scanner.nextInt();
24
25        System.out.print("Masukkan Golongan Pegawai (A, B, atau C): ");
26        char golongan = scanner.next().charAt(0);
27
28        Pegawai pegawai = new Pegawai(NIP, nama, alamat, gajiPokok, hariLembur, golongan);
29        pegawai.printInfo();
30    }
31 }
32 }
```

Berikut Penjelasan Mengenai kode tersebut

class Pegawai

Kode yang Saya buat adalah implementasi dari sebuah kelas Java yang disebut "Pegawai." Kelas ini digunakan untuk merepresentasikan data dan perilaku terkait pegawai, termasuk atribut-atribut seperti NIP, nama, alamat, gaji pokok, hari lembur, dan golongan. Berikut penjelasan setiap bagian kode tersebut:

- **Package dan Import**

```
1 package tugas3;
2
3 import java.util.Scanner;
```

Ini adalah pernyataan yang mendefinisikan paket (package) tempat kelas Pegawai berada (tugas3) dan mengimpor kelas Scanner dari pustaka java.util. Scanner digunakan untuk input dari pengguna, meskipun dalam kode ini, tidak ada penggunaan Scanner.

- **Deklarasi Kelas Pegawai**

```
5 public class Pegawai {
6     private String NIP;
7     private String nama;
8     private String alamat;
9     private double gajiPokok;
10    private int hariLembur;
11    private char golongan;
12 }
```

Ini adalah deklarasi kelas Pegawai yang memiliki atribut dan metode terkait dengan pegawai.

- **Atribut Kelas Pegawai**

Kelas Pegawai memiliki beberapa atribut (variabel) yang digunakan untuk menyimpan informasi pegawai. Atribut-atribut ini adalah NIP, nama, alamat, gajiPokok, hariLembur, dan golongan. Mereka adalah variabel-variabel privat, yang berarti hanya dapat diakses dalam kikelaskan Pegawai.

- **Konstruktor Pegawai**

```
13 public Pegawai(String NIP, String nama, String alamat, double gajiPokok, int hariLembur, char golongan) {  
14     this.NIP = NIP;  
15     this.nama = nama;  
16     this.alamat = alamat;  
17     this.gajiPokok = gajiPokok;  
18     this.hariLembur = hariLembur;  
19     this.golongan = golongan;  
20 }
```

Ini adalah konstruktor kelas Pegawai yang digunakan untuk menciptakan objek Pegawai. Ketika objek Pegawai diciptakan, atribut-atributnya diinisialisasi dengan nilai yang diberikan saat pembuatan objek.

- **Metode Setter dan Getter**

Ada sejumlah metode setter dan getter untuk mengakses dan mengubah nilai atribut-atribut Pegawai. Contohnya, setNIP, getNIP, setName, getName, dan seterusnya. Metode setter digunakan untuk mengubah nilai atribut, sedangkan metode getter digunakan untuk mengambil nilai atribut.

- **Metode totalLembur**

```
70 public double totalLembur() {  
71     return hariLembur * 0.01 * gajiPokok;  
72 }
```

Metode ini menghitung total lembur berdasarkan jumlah hari lembur dan gaji pokok pegawai. Nilai total lembur dihitung dengan rumus **hariLembur * 0.01 * gajiPokok**.

- **Metode tunjanganGolongan**

```
74 public double tunjanganGolongan() {  
75     double tunjangan = 0;  
76     switch (golongan) {  
77         case 'A':  
78             tunjangan = gajiPokok * 0.015;  
79             break;  
80         case 'B':  
81             tunjangan = gajiPokok * 0.03;  
82             break;  
83         case 'C':  
84             tunjangan = gajiPokok * 0.05;  
85             break;  
86     }  
87     return tunjangan;  
88 }
```

Metode ini menghitung tunjangan golongan berdasarkan golongan pegawai. Tunjangan golongan dihitung berdasarkan nilai atribut golongan, dengan nilai-nilai yang berbeda sesuai dengan golongan yang ditentukan.

- **Metode totalGaji**

```
90 public double totalGaji() {  
91     return gajiPokok + totalLembur() + tunjanganGolongan();  
92 }
```

Metode ini menghitung total gaji pegawai dengan menggabungkan gaji pokok, tunjangan lembur, dan tunjangan golongan.

- **Metode printInfo**

```
94 public void printInfo() {  
95     System.out.println(" ");  
96     System.out.println("NIP Pegawai : " + NIP);  
97     System.out.println("Nama Pegawai : " + nama);  
98     System.out.println("Alamat Pegawai : " + alamat);  
99     System.out.println("Gaji Pokok: " + gajiPokok + ".0");  
100    System.out.println("Hari Lembur : " + hariLembur);  
101    System.out.println("Total Lembur : " + totalLembur() + ".0");  
102    System.out.println("Tunjangan Golongan : " + tunjanganGolongan() + ".0");  
103    System.out.println("Total Gaji: " + totalGaji() + ".0");  
104 }  
105 }
```

Metode ini digunakan untuk mencetak informasi lengkap tentang pegawai, termasuk NIP, nama, alamat, gaji pokok, jumlah lembur, dan total gaji.

class Utama

Kode ini adalah kelas Java dengan nama "class Utama" yang berisi metode main. Kode ini digunakan untuk mengambil input pengguna dari kelas Pegawai, menciptakan objek Pegawai, dan mencetak informasi pegawai berdasarkan input yang diberikan. Berikut adalah penjelasan mengenai kode tersebut:

- **Import Scanner dan Package**

Kode mengimpor kelas Scanner dari pustaka Java untuk mengambil input dari pengguna. Juga, itu mengimpor kelas Pegawai dari paket "tugas3," yang digunakan untuk membuat objek pegawai.

- **Metode 'main'**

```
public class Utama {  
    public static void main(String[] args) {
```

Ini adalah metode utama yang akan dijalankan saat program dimulai. Semua kode di dalam metode ini akan dieksekusi.

- **Membuat Objek Scanner**

```
7 | Scanner scanner = new Scanner(source: System.in);
```

Ini adalah deklarasi objek Scanner yang digunakan untuk mengambil input dari pengguna melalui keyboard (System.in).

- **Mengambil Input dari pengguna**

```
8 | System.out.println(x: "Konstruktor Pegawai dijalankan...!");
9 |
10 | System.out.print(s: "Masukkan NIP Pegawai: ");
11 | String NIP = scanner.nextLine();
12 |
13 | System.out.print(s: "Masukkan Nama Pegawai: ");
14 | String nama = scanner.nextLine();
15 |
16 | System.out.print(s: "Masukkan Alamat Pegawai: ");
17 | String alamat = scanner.nextLine();
18 |
19 | System.out.print(s: "Masukkan Gaji Pokok Pegawai: ");
20 | double gajiPokok = scanner.nextDouble();
21 |
22 | System.out.print(s: "Masukkan Hari Lembur Pegawai: ");
23 | int hariLembur = scanner.nextInt();
24 |
25 | System.out.print(s: "Masukkan Golongan Pegawai (A, B, atau C): ");
26 | char golongan = scanner.next().charAt(index: 0);
```

Kode berikut ini digunakan untuk mengambil berbagai informasi tentang pegawai dari pengguna melalui keyboard, Pengguna diminta untuk memasukkan NIP, nama, alamat, gaji pokok, hari lembur, dan golongan pegawai.

- **Membuat Objek Pegawai**

```
Pegawai pegawai = new Pegawai(NIP, nama, alamat, gajiPokok, hariLembur, golongan);
```

Setelah menerima input dari pengguna, kode ini menciptakan objek pegawai dari kelas Pegawai dengan menggunakan konstruktor Pegawai yang mengambil semua informasi yang dimasukkan oleh pengguna.

- **Mencetak Informasi Pegawai**

```
pegawai.printInfo();
```

Ini adalah perintah untuk mencetak informasi lengkap tentang pegawai yang telah dimasukkan pengguna, termasuk NIP, nama, alamat, gaji pokok, jumlah hari lembur, tunjangan golongan, dan

total gaji. Informasi ini dicetak dengan memanggil metode printInfo dari objek pegawai.