

Kuis 1

Nama: Rizqi Bagus Andrean

Kelas: TI-1D

Absen: 25

```

1  class Pegawai{
2      String name;
3      int masaKerjaThn, gajiPokok;
4      double jamLembur;
5
6      public Pegawai(String name, int masaKerjaThn, int gajiPokok, double jamLembur){
7          this.name = name;
8          this.masaKerjaThn = masaKerjaThn;
9          this.gajiPokok = gajiPokok;
10         this.jamLembur = jamLembur;
11     }
12
13     public void printData() {
14         System.out.println("Nama : " + name);
15         System.out.println("Masa Kerja : " + masaKerjaThn + " tahun");
16         System.out.println("Gaji Pokok : " + gajiPokok);
17         System.out.println("Jam Lembur : " + jamLembur);
18         System.out.println("Gaji Lembur : " + hitungGajiLembur());
19         System.out.println("Tunjangan : " + hitungTunjangan());
20         System.out.println("Gaji Total : " + Double.toString(hitungGajiTotal()));
21         System.out.println();
22     }
23
24     public double hitungGajiLembur() {
25
26         if (masaKerjaThn < 5) {
27             return 0.05 * gajiPokok * jamLembur;
28         }else{
29             return 0.1 * gajiPokok * jamLembur;
30         }
31     }
32
33     public double hitungGajiTotal() {
34         return gajiPokok + hitungGajiLembur() + hitungTunjangan();
35     }
36
37     public double hitungTunjangan() {
38         if (masaKerjaThn < 5) {
39             return 0.25 * gajiPokok;
40         }else{
41             return 0.5 * gajiPokok;
42         }
43     }
44
45     public static void main(String[] args) {
46         Pegawai[] pegawais = new Pegawai[6];
47
48         pegawais[0] = new Pegawai("Budi", 2, 2000000, 5);
49         pegawais[1] = new Pegawai("Tono", 3, 3000000, 6);
50         pegawais[2] = new Pegawai("Adi", 4, 4000000, 7);
51         pegawais[3] = new Pegawai("Rina", 5, 5000000, 8);
52         pegawais[4] = new Pegawai("Joko", 6, 6000000, 9);
53         pegawais[5] = new Pegawai("Ani", 7, 7000000, 10);
54
55         int totalGajiG5 = 0;
56         int jmlPegawaiG5 = 0;
57
58         int indexGajiLemburTertinggi = 0;
59
60         for (int i = 0; i < pegawais.length; i++) {
61             if (pegawais[i].masaKerjaThn < 5) {
62
63                 if (pegawais[i].hitungGajiLembur() > pegawais[indexGajiLemburTertinggi].hitungGajiLembur()) {
64                     indexGajiLemburTertinggi = i;
65                 }
66             }else{
67                 totalGajiG5 += pegawais[i].hitungGajiTotal();
68                 jmlPegawaiG5++;
69             }
70         }
71
72         System.out.println("Rata-rata gaji pegawai dengan masa kerja lebih dari 5 tahun adalah " + totalGajiG5 / jmlPegawaiG5);
73         System.out.println();
74         System.out.println("Pegawai dengan gaji lembur tertinggi masa kerja dibawah 5 tahun adalah: ");
75         pegawais[indexGajiLemburTertinggi].printData();
76
77     }
78 }

```

Penjelasan:

```
class Pegawai{
    String name;
    int masaKerjaThn, gajiPokok;
    double jamLembur;

    public Pegawai(String name, int masaKerjaThn, int gajiPokok, double jamLembur){
        this.name = name;
        this.masaKerjaThn = masaKerjaThn;
        this.gajiPokok = gajiPokok;
        this.jamLembur = jamLembur;
    }
}
```

1.

Pertama membuat kelas dengan nama Pegawai dan memiliki attribute seperti name, masa kerja, gaji pokok, dan jam lembur

Lalu membuat constructor, dengan parameter name, masaKerja, gajiPokok dan jamLembur. Constructor digunakan untuk memasukkan data ke dalam attribute saat membuat object.

Disini menggunakan this karena nama dari attribute dan parameter dari constructornya sama.

```
public void printData() {
    System.out.println("Nama : " + name);
    System.out.println("Masa Kerja : " + masaKerjaThn + " tahun");
    System.out.println("Gaji Pokok : " + gajiPokok);
    System.out.println("Jam Lembur : " + jamLembur);
    System.out.println("Gaji Lembur : " + hitungGajiLembur());
    System.out.println("Tunjangan : " + hitungTunjangan());
    System.out.println("Gaji Total : " + Double.toString(hitungGajiTotal()));
    System.out.println();
}
```

2.

Fungsi printData bertipe void digunakan untuk menampilkan data dari pegawai.

```
public double hitungGajiLembur() {
    if (masaKerjaThn < 5) {
        return 0.05 * gajiPokok * jamLembur;
    } else {
        return 0.1 * gajiPokok * jamLembur;
    }
}
```

3.

Fungsi hitungGajiLembur memiliki return type berupa double, di dalam fungsi ini menghitung gajiLembur, dimana jika masaKerja yang diambil dari attribute masaKerjaThn kurang dari 5 tahun mereturn **0.05 * gajiPokok * jamLembur**

Jika masaKerja lebih dari atau sama dengan 5 tahun maka mereturn **0.1 * gajiPokok * jamLembur**

```
public double hitungTunjangan() {
    if (masaKerjaThn < 5) {
        return 0.25 * gajiPokok;
    }else{
        return 0.5 * gajiPokok;
    }
}
```

4.

Fungsi hitungTunjangan memiliki return type berupa double, didalam fungsi ini menghitung jumlah tunjangan, Dimana jika masaKerja yang diambil dari attribute masaKerjaThn kurang dari 5 tahun mereturn **0.25 * gajiPokok**

Jika masaKerja lebih dari samadengan 5 tahun maka mereturn **0.5 * gajiPokok**

```
public double hitungGajiTotal() {
    return gajiPokok + hitungGajiLembur() + hitungTunjangan();
}
```

5.

Fungsi hitungGajiTotal memiliki return type berupa double, fungsi ini digunakan untuk menghitung gaji total yang didapat dari perhitungan gajiPoko + gajiLembur + tunjangan

```
public static void main(String[] args) {
    Pegawai[] pegawais = new Pegawai[6];

    pegawais[0] = new Pegawai(name:"Budi", masaKerjaThn:2, gajiPokok:2000000, jamLembur:5);
    pegawais[1] = new Pegawai(name:"Tono", masaKerjaThn:3, gajiPokok:3000000, jamLembur:6);
    pegawais[2] = new Pegawai(name:"Adi", masaKerjaThn:4, gajiPokok:4000000, jamLembur:7);
    pegawais[3] = new Pegawai(name:"Rina", masaKerjaThn:5, gajiPokok:5000000, jamLembur:8);
    pegawais[4] = new Pegawai(name:"Joko", masaKerjaThn:6, gajiPokok:6000000, jamLembur:9);
    pegawais[5] = new Pegawai(name:"Ani", masaKerjaThn:7, gajiPokok:7000000, jamLembur:10);
}
```

6.

Fungsi Main. Didalam fungsi main dibuat array yang berisi instance dari kelas pegawai berjumlah 6.

Lalu isi array tersebut dengan meng instansiasi kelas pegawai dan dimasukkan ke masing-masing index dari array pegawai

```
int totalGajiG5 = 0;
int jmlPegawaiG5 = 0;

int indexGajiLemburTertinggi = 0;
```

7.

Membuat variable total gaji dan jumlah pegawai, variable ini nanti digunakan untuk mencari rata-rata gaji.

Variable index gaji lembur tertinggi digunakan untuk menyimpan index dari array pegawai yang memiliki gajilembur tertinggi

```
for (int i = 0; i < pegawais.length; i++) {
    if (pegawais[i].masaKerjaTahn < 5) {
        if (pegawais[i].hitungGajiLembur() > pegawais[indexGajiLemburTertinggi].hitungGajiLembur()) {
            indexGajiLemburTertinggi = i;
        }
    } else {
        totalGajiG5 += pegawais[i].hitungGajiTotal();
        jmlPegawaiG5++;
    }
}

System.out.println("Rata-rata gaji pegawai dengan masa kerja lebih dari 5 tahun adalah " + totalGajiG5 / jmlPegawaiG5);
System.out.println();
System.out.println(x:"Pegawai dengan gaji lembur tertinggi masa kerja dibawah 5 tahun adalah: ");
pegawais[indexGajiLemburTertinggi].printData();
}
```

8.

Lakukan perulangan. Didalam perulangan kita mengecek apakah pegawai dengan masa kerja lebih dari 5 tahun.

Jika lebih dari 5 tahun maka kita akan menyimpan total gaji karyawan tersebut dan ditambahkan ke variable totalGajiG5, lalu variable jmlPegawaiG5 juga akan ditambahkan 1.

Jika kurang dari 5 tahun, maka kita akan melakukan pengecekan jika gajiLemburnya lebih tinggi dari gajiLembur pegawai sebelumnya maka kita akan simpan index pegawai tersebut dei variable indexGajiLembur Tertinggi.

7.

```
System.out.println("Rata-rata gaji pegawai dengan masa kerja lebih dari 5 tahun adalah " + totalGajiG5 / jmlPegawaiG5);
System.out.println();
System.out.println(x:"Pegawai dengan gaji lembur tertinggi masa kerja dibawah 5 tahun adalah: ");
pegawais[indexGajiLemburTertinggi].printData();
}
```

Selanjutnya kita akan menampilkan rata-rata gaji pegawai yang masa kerja >5 tahun dengan membagi totalGajiG5 dengan jmlPegawai.

Lalu juga akan ditampilkan pegawai dengan masa kerja <5 tahun yang memiliki tunjangan tertinggi

Dengan memanggil fungsi prindata dari instance pegawai dari array pegawais dengan index yang telah disimpan pada variable indexGajiLembur tertinggi.

Hasil:

```
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\kuis 1> cd C:\Users\Acer\Tugas Kuliah\rs\Acer\AppData\Roaming\Code\User\workspaceStorage\141702cfb8c283d88245d5399b888d02\redhat.java\jdt_ws
Rata-rata gaji pegawai dengan masa kerja lebih dari 5 tahun adalah 14466666
```

Pegawai dengan gaji lembur tertinggi masa kerja dibawah 5 tahun adalah:

Nama : Adi

Masa Kerja : 4 tahun

Gaji Pokok : 4000000

Jam Lembur : 7.0

Gaji Lembur : 1400000.0

Tunjangan : 1000000.0

Gaji Total : 6400000.0

```
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\kuis 1> █
```