

JOBSHEET 7

OVERLOADING DAN OVERRIDING

Object Oriented Program



Febrian Arka Samudra

2341720066 / 10

STUDY PROGRAM D-IV INFORMATICS ENGINEERING

INFORMATICS TECHNOLOGY MAJOR

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, kec, Lowokwaru, Kota Malang, Jawa Timur 65141

- **Percobaan 1**

Class karyawan:

```
J Karyawan.java > Karyawan
1  public class Karyawan {
2      private String nama;
3      private String nip;
4      private String golongan;
5      private double gaji;
6
7      public void setNama(String nama){
8          this.nama = nama;
9      }
10
11     public void setNip(String nip){
12         this.nip = nip;
13     }
14
15     public void setGolongan(String golongan){
16         this.golongan = golongan;
17
18         switch (golongan.charAt(index:0)) {
19             case '1': this.gaji=5000000;
20                 break;
21
22             case '2': this.gaji=3000000;
23                 break;
24
25             case '3':this.gaji=2000000;
26                 break;
27
28             case '4':this.gaji=1000000;
29                 break;
30
31             case '5':this.gaji=750000;
32                 break;
33         }
34     }
35 }
36
37     public void setGaji(double gaji){
38         this.gaji = gaji;
39     }
40
41     public String getNama(){
42         return nama;
43     }
44
45     public String getNip(){
46         return nip;
47     }
48
49     public String getGolongan(){
50         return golongan;
51     }
52
53     public double getGaji(){
54         return gaji;
55     }
56 }
```

Class staff:

```
J Staff.java > Staff
1 public class Staff extends Karyawan {
2     private int lembur;
3     private double gajiLembur;
4
5     public void setLembur(int lembur){
6         this.lembur = lembur;
7     }
8
9     public int getLembur(){
10        return lembur;
11    }
12
13    public void setGajiLembur(double gajiLembur){
14        this.gajiLembur = gajiLembur;
15    }
16
17    public double getGajiLembur(){
18        return gajiLembur;
19    }
20
21    public double getGaji (int lembur,double gajiLembur){
22        return super.getGaji() + (lembur*gajiLembur);
23    }
24
25    public double getGaji(){
26        return super.getGaji() + (lembur*gajiLembur);
27    }
28
29    public void lihatInfo(){
30        System.out.println("NIP :"+this.getNip());
31        System.out.println("Nama :"+this.getNama());
32        System.out.println("Golongan :"+this.getGolongan());
33        System.out.println("Jumlah lembur :"+this.getLembur());
34        System.out.printf(format:"Gaji Lembur : %.0f\n",this.getGajiLembur());
35        System.out.printf(format:"Gaji : %.0f\n",this.getGaji());
36    }
37 }
```

Class Manager:

```

J Manager.java > Manager
1  public class Manager extends Karyawan{
2      private double tunjangan;
3      private String bagian;
4      private Staff st[];
5
6      public void setTunjangan(double tunjangan){
7          this.tunjangan = tunjangan;
8      }
9
10     public double getTunjangan(){
11         return tunjangan;
12     }
13
14     public void setBagian(String bagian){
15         this.bagian = bagian;
16     }
17
18     public String getBagian(){
19         return bagian;
20     }
21
22     public void setStaff(Staff st[]){
23         this.st = st;
24     }
25
26     public void viewStaff() {
27         int i;
28         System.out.println(x:"-----");
29         for(i=0;i<st.length;i++) {
30             st[i].lihatInfo();
31         }
32         System.out.println(x:"-----");
33     }

```

```

35  public void lihatInfo() {
36      System.out.println("Manager :" + this.getBagian());
37      System.out.println("NIF :" + this.getNip());
38      System.out.println("Nama :" + this.getNama());
39      System.out.println("Golongan :" + this.getGolongan());
40      System.out.printf(format:"Tunjangan : %.0f\n", this.getTunjangan());
41      System.out.printf(format:"Gaji : %.0f\n", this.getGaji());
42      System.out.println("Bagian :" + this.getBagian());
43      this.viewStaff();
44  }
45
46  public double getGaji() {
47      return super.getGaji() + tunjangan;
48  }
49  }

```

Main Class:

```
J Utama.java > ...
1  public class Utama {
    Run | Debug
2      public static void main(String[] args) {
3
4          System.out.println(x:"Program Testing Class Manager & Staff");
5          Manager man[] = new Manager[2];
6          Staff staff1[] = new Staff[2];
7          Staff staff2[] = new Staff[3];
8
9          // Pembuatan Manager
10         man[0] = new Manager();
11         man[0].setNama(nama:"Tedjo");
12         man[0].setNip(nip:"101");
13         man[0].setGolongan(golongan:"1");
14         man[0].setTunjangan(tunjangan:5000000);
15         man[0].setBagian(bagian:"Administrasi");
16
17         man[1] = new Manager();
18         man[1].setNama(nama:"Atika");
19         man[1].setNip(nip:"102");
20         man[1].setGolongan(golongan:"1");
21         man[1].setTunjangan(tunjangan:2500000);
22         man[1].setBagian(bagian:"Pemasaran");
23
24         // Pembuatan Staff 1
25         staff1[0] = new Staff();
26         staff1[0].setNama(nama:"Usman");
27         staff1[0].setNip(nip:"0003");
28         staff1[0].setGolongan(golongan:"2");
29         staff1[0].setLembur(lembur:10);
30         staff1[0].setGajiLembur(gajiLembur:10000);
31
32         staff1[1] = new Staff();
33         staff1[1].setNama(nama:"Anugrah");
34         staff1[1].setNip(nip:"0005");
35         staff1[1].setGolongan(golongan:"2");
36         staff1[1].setLembur(lembur:10);
37         staff1[1].setGajiLembur(gajiLembur:55000);
38
39         man[0].setStaff(staff1);
40
41         // Pembuatan Staff 2
42         staff2[0] = new Staff();
43         staff2[0].setNama(nama:"Hendra");
44         staff2[0].setNip(nip:"0004");
45         staff2[0].setGolongan(golongan:"3");
46         staff2[0].setLembur(lembur:15);
47         staff2[0].setGajiLembur(gajiLembur:5500);
48
49         staff2[1] = new Staff();
50         staff2[1].setNama(nama:"Arie");
51         staff2[1].setNip(nip:"0006");
52         staff2[1].setGolongan(golongan:"4");
53         staff2[1].setLembur(lembur:5);
54         staff2[1].setGajiLembur(gajiLembur:100000);
```

```

56     staff2[2] = new Staff();
57     staff2[2].setNama(nama:"Mentari");
58     staff2[2].setNip(nip:"0007");
59     staff2[2].setGolongan(golongan:"3");
60     staff2[2].setLembur(lembur:6);
61     staff2[2].setGajiLembur(gajiLembur:20000);
62
63     man[1].setStaff(staff2);
64
65     man[0].lihatInfo();
66     man[1].lihatInfo();
67 }
68 }

```

Output:

```

PS D:\ArkaDocumentKuliah\Semester-3\Pe
T\AppData\Roaming\Code\User\workspaceS
Program Testing Class Manager & Staff
Manager :Administrasi
NIF :101
Nama :Tedjo
Golongan :1
Tunjangan : 5000000
Gaji : 10000000
Bagian :Administrasi
-----
NIP :0003
Nama :Usman
Golongan :2
Jumlah lembur :10
Gaji Lembur : 10000
Gaji : 3100000
NIP :0005
Nama :Anugrah
Golongan :2
Jumlah lembur :10
Gaji Lembur : 55000
Gaji : 3550000
-----

```

```

-----
Manager :Pemasaran
NIF :102
Nama :Atika
Golongan :1
Tunjangan : 2500000
Gaji : 7500000
Bagian :Pemasaran
-----

```

```

NIP :0004
Nama :Hendra
Golongan :3
Jumlah lembur :15
Gaji Lembur : 5500
Gaji : 2082500
NIP :0006
Nama :Arie
Golongan :4
Jumlah lembur :5
Gaji Lembur : 100000
Gaji : 1500000
NIP :0007
Nama :Mentari
Golongan :3
Jumlah lembur :6

```

```

Gaji Lembur : 20000
Gaji : 2120000
-----

```

- Exercise

```

J Perkalianku.java > Perkalianku
1 public class Perkalianku {
2     public int a,b,c;
3
4     void perkalian(int a, int b){
5         this.a=a;
6         this.b=b;
7         System.out.println("Hasil perkalian dari " + a + " dan " + b + " adalah: " + (a * b));
8     }
9     void perkalian(int a, int b, int c){
10        this.a=a;
11        this.b=b;
12        this.c=c;
13        System.out.println("Hasil perkalian dari " + a + ", " + b + " dan " + c + " adalah: " + (a * b * c));
14    }
15
16    Run | Debug
17    public static void main(String args []){
18        Perkalianku objek = new Perkalianku();
19        objek.perkalian(a:25, b:43);
20        objek.perkalian(a:34, b:23, c:56);
21    }
22 }

```

```

Hasil perkalian dari 25 dan 43 adalah: 1075
Hasil perkalian dari 34, 23 dan 56 adalah: 43792

```

Output:

PS D:\ArkaDocumentKuliah\Semester-3\Pemrograman B

4.1 From the above source code, where is the overloading?

```
4      void perkalian(int a, int b){
5          this.a=a;
6          this.b=b;
7          System.out.println("Hasil perkalian dari " + a + " dan " + b + " adalah: " + (a * b));
8      }
```

```
9      void perkalian(int a, int b, int c){
10         this.a=a;
11         this.b=b;
12         this.c=c;
13         System.out.println("Hasil perkalian dari " + a + ", " + b + " dan " + c + " adalah: " + (a * b * c));
14     }
```

4.2 If there any overloading, how many parameters are different?

- The first perkalian method has 2 parameters:

```
4      void perkalian(int a, int b){
5          this.a=a;
6          this.b=b;
7          System.out.println("Hasil perkalian dari " + a + " dan " + b + " adalah: " + (a * b));
8      }
```

- The second perkalian method has 3 parameters:

```
9      void perkalian(int a, int b, int c){
10         this.a=a;
11         this.b=b;
12         this.c=c;
13         System.out.println("Hasil perkalian dari " + a + ", " + b + " dan " + c + " adalah: " + (a * b * c));
14     }
```

- The difference between the overloaded methods is 1 parameter. The first method accepts 2 parameters, while the second accepts 3.

•

```
J Perkalianku.java > Perkalianku
1  public class Perkalianku {
2
3      void perkalian(int a, int b) {
4          System.out.println(a * b);
5      }
6
7      void perkalian(double a, double b) {
8          System.out.println(a * b);
9      }
10
11      Run | Debug
12      public static void main(String[] args) {
13          Perkalianku objek = new Perkalianku();
14
15          objek.perkalian(a:25, b:43);
16          objek.perkalian(a:34.56, b:23.7);
17      }
```


4.3 From the above source code, where is the overloading?

```
3 void perkalian(int a, int b) {
4     System.out.println(a * b);
5 }

7 void perkalian(double a, double b) {
8     System.out.println(a * b);
9 }
```

4.4 If there any overloading, how many parameters are different?

- No Different

•

```
J Ikan.java > Ikan
1 public class Ikan {
2     public void swim(){
3         System.out.println(x:"Ikan bisa berenang");
4     }
5 }
```

```
J Piranha.java > Piranha
1 public class Piranha extends Ikan {
2     public void swim(){
3         System.out.println(x:"Piranha bisa makan daging");
4     }
5 }
```

```
J Fish.java > Fish
1 public class Fish {
2     Run | Debug
3     public static void main(String[] args) {
4         Ikan a = new Ikan();
5         Ikan b = new Piranha();
6
7         a.swim();
8         b.swim();
9     }
}
```

4.5 From the above source code, where is the overriding?

```
J Piranha.java > Piranha
1 public class Piranha extends Ikan {
2     public void swim(){
3         System.out.println(x:"Piranha bisa makan daging");
4     }
5 }
```

- class Piranha but extends it as an Ikan (as seen in the line `Ikan b = new Piranha();`), the overridden `swim()` method in the Piranha class will be called, not the one in the Ikan class.

So the overriding happens in the Piranha class with the `swim()` method

4.6 If there any overloading, how many parameters are different?

- In method overriding, the number of parameters must remain the same between the method in the superclass and the method in the subclass.
- method `swim()` in both the Ikan class and the Piranha class has no parameters. Therefore, the number of parameters is the same in both the superclass (Ikan) and the subclass (Piranha), meaning there are no differences in the number of parameters between the two methods.

5. Task

5.1 Overloading

Segitiga Class:

```
J Segitiga.java > Segitiga
1 public class Segitiga {
2     private int sudut;
3
4     public int totalSudut(int sudutA) {
5         sudut = 180 - sudutA;
6         return sudut;
7     }
8
9     public int totalSudut(int sudutA, int sudutB) {
10        sudut = 180 - (sudutA + sudutB);
11        return sudut;
12    }
13
14    public int keliling(int sisiA, int sisiB, int sisiC) {
15        return sisiA + sisiB + sisiC;
16    }
17
18    public double keliling(int sisiA, int sisiB) {
19        return Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
20    }
21
22    public int getSudut() {
23        return sudut;
24    }
25 }
```

Main Segitiga:

```
J MainSegitiga.java > ...
1  public class MainSegitiga {
    Run | Debug
2      public static void main(String[] args) {
3          Segitiga segitiga = new Segitiga();
4
5          int sudut1 = segitiga.totalSudut(sudutA:60);
6          System.out.println("Total sudut with one angle: " + sudut1);
7          System.out.println("Value of sudut field: " + segitiga.getSudut());
8
9          int sudut2 = segitiga.totalSudut(sudutA:60, sudutB:30);
10         System.out.println("Total sudut with two angles: " + sudut2);
11         System.out.println("Value of sudut field: " + segitiga.getSudut());
12
13         int keliling1 = segitiga.keliling(sisiA:3, sisiB:4, sisiC:5);
14         System.out.println("Perimeter of the triangle: " + keliling1);
15
16         double keliling2 = segitiga.keliling(sisiA:3, sisiB:4);
17         System.out.println("Hypotenuse: " + keliling2);
18     }
19 }
```

Output:

```
Total sudut with one angle: 120
Value of sudut field: 120
Total sudut with two angles: 90
Value of sudut field: 90
Perimeter of the triangle: 12
Hypotenuse: 5.0
PS D:\ArkaDocumentKuliah\Semeste
```

5.2 Overriding

Class Manusia:

```
J Manusia.java > Manusia
1 public class Manusia {
2     public void bernafas() {
3         System.out.println(x:"Manusia sedang bernafas.");
4     }
5
6     public void makan() {
7         System.out.println(x:"Manusia sedang makan.");
8     }
9 }
```

Class Mahasiswa:

```
J Mahasiswa.java > Mahasiswa
1 public class Mahasiswa extends Manusia {
2     @Override
3     public void makan() {
4         System.out.println(x:"Mahasiswa sedang makan mie instan.");
5     }
6
7     public void tidur() {
8         System.out.println(x:"Mahasiswa sedang tidur setelah belajar semalaman.");
9     }
10 }
```

Class dosen:

```
J Dosen.java > ...
1 public class Dosen extends Manusia{
2     @Override
3     public void makan() {
4         System.out.println(x:"Dosen sedang makan di kantin kampus.");
5     }
6
7     public void lembur() {
8         System.out.println(x:"Dosen sedang lembur menyiapkan materi kuliah.");
9     }
10 }
```

Main:

```
J MainTask2.java > MainTask2
1 public class MainTask2 {
2     public static void aktivitasMakan(Manusia manusia) {
3         manusia.makan();
4     }
5
6     Run | Debug
7     public static void main(String[] args) {
8         System.out.println(x:"Testing with Dosen object:");
9         Manusia dosen = new Dosen();
10        aktivitasMakan(dosen);
11        ((Dosen) dosen).lembur();
12
13        System.out.println(x:"\nTesting with Mahasiswa object:");
14        Manusia mahasiswa = new Mahasiswa();
15        aktivitasMakan(mahasiswa);
16        if (mahasiswa instanceof Mahasiswa) {
17            ((Mahasiswa) mahasiswa).tidur();
18        }
19    }
```

Output:

```
Testing with Dosen object:
Dosen sedang makan di kantin kampus.
Dosen sedang lembur menyiapkan materi kuliah.

Testing with Mahasiswa object:
Mahasiswa sedang makan mie instan.
Mahasiswa sedang tidur setelah belajar semalaman.
PS D:\ArkaDocumentKuliah\Semester-3\Pemrograman Berbasis Objek>
```

Github: <https://github.com/FebrianArkaSamudra/Semester-3/tree/main/Pemrograman%20Berbasis%20Objek/Week7>