

# Object Oriented Programming

## Jobsheet 9

### Overloading and Overriding



Name: Azahra Salsabila

NIM: 2241720073

Class: 2I

No: 05

Major: Information Technology

Study Program: Informatic Engineering

## Experiment

- Karyawan

```
1 package semester3.jobsheet9;
2 public class Karyawan {
3     private String nama;
4     private String nip;
5     private String golongan;
6     private double gaji;
7
8     public void setNama(String nama) {
9         this.nama = nama;
10    }
11    public void setNip(String nip) {
12        this.nip = nip;
13    }
14    public void setGolongan(String golongan) {
15        this.golongan = golongan;
16        switch (golongan.charAt(0)) {
17            case '1': this.gaji = 5000000;
18                break;
19            case '2': this.gaji = 3000000;
20                break;
21            case '3': this.gaji = 2000000;
22                break;
23            case '4': this.gaji = 1000000;
24                break;
25            case '5': this.gaji = 750000;
26                break;
27        }
28    }
29    public void setGaji(double gaji) {
30        this.gaji = gaji;
31    }
32    public String getNama() {
33        return nama;
34    }
35    public String getNip() {
36        return nip;
37    }
38    public String getGolongan() {
39        return golongan;
40    }
41    public double getGaji() {
42        return gaji;
43    }
44 }
45
```

- Staff

```
1 package semester3.jobsheet9;
2 public class Staff extends Karyawan{
3     private int lembur;
4     private double gajiLembur;
5     public void setLembur(int Lembur) {
6         this.lembur = lembur;
7     }
8     public int getLembur() {
9         return lembur;
10    }
11    public void setGajiLembur(double gajiLembur) {
12        this.gajiLembur = gajiLembur;
13    }
14    public double getGajiLembur() {
15        return gajiLembur;
16    }
17    public double getGaji(int Lembur, double gajiLembur) {
18        return super.getGaji() + lembur * gajiLembur;
19    }
20    public double getGaji() {
21        return super.getGaji() + lembur * gajiLembur;
22    }
23    public void lihatInfo() {
24        System.out.println("NIP          : " + this.getNip());
25        System.out.println("Nama          : " + this.getNama ());
26        System.out.println("Golongan      : " + this.getGolongan ());
27        System.out.println("Jml Lembur    : " + this.getLembur());
28        System.out.printf("Gaji Lembur :%.0f\n", this.getGajiLembur());
29        System.out.printf("Gaji :%.0f\n", this.getGaji());
30    }
31 }
```

- Manager

```
1 package semester3.jobsheet9;
2 public class Manager extends Karyawan {
3     private double tunjangan;
4     private String bagian;
5     private Staff st[];
6
7     public void setTunjangan(double tunjangan) {
8         this.tunjangan = tunjangan;
9     }
10
11    public double getTunjangan() {
12        return tunjangan;
13    }
14
15    public void setBagian(String bagian) {
16        this.bagian = bagian;
17    }
18
19    public String getBagian() {
20        return bagian;
21    }
22
23    public void setStaff(Staff st[]) {
24        this.st = st;
25    }
26
27    public void viewStaff() {
28
29        int i;
30        System.out.println("-----");
31        for (i = 0; i < st.length; i++) {
32            st[i].lihatInfo();
33        }
34        System.out.println("-----");
35    }
36
37    public void lihatInfo() {
38        System.out.println("Manager :" + this.getBagian());
39        System.out.println("NIP :" + this.getNip());
40        System.out.println("Nama :" + this.getNama());
41        System.out.println("Golongan :" + this.getGolongan());
42        System.out.printf("Tunjangan :%.0f\n", this.getTunjangan());
43        System.out.printf("Gaji :.0f\n", this.getGaji());
44        System.out.println("Bagian :" + this.getBagian());
45
46        this.viewStaff();
47    }
48
49    public double getGaji() {
50        return super.getGaji() + tunjangan;
51    }
52 }
53
```

- Utama

```
1 package semester3.jobsheet9;
2 public class Utama {
3     public static void main(String[] args) {
4         System.out.println("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("Tedjo");
12        man[0].setNip("101");
13        man[0].setGolongan("1");
14        man[0].setTunjangan(5000000);
15        man[0].setBagian("Administrasi");
16
17        man[1] = new Manager();
18        man[1].setNama("Atika");
19        man[1].setNip("102");
20        man[1].setGolongan("1");
21        man[1].setTunjangan(2500000);
22        man[1].setBagian("Pemasaran");
23
24        staff1[0] = new Staff();
25        staff1[0].setNama("Usman");
26        staff1[0].setNip("0003");
27        staff1[0].setGolongan("2");
28        staff1[0].setLembur(10);
29        staff1[0].setGajiLembur(10000);
30
31        staff1[1] = new Staff();
32        staff1[1].setNama("Anugrah");
33        staff1[1].setNip("0005");
34        staff1[1].setGolongan("2");
35        staff1[1].setLembur(10);
36        staff1[1].setGajiLembur(55000);
37        man[0].setStaff(staff1);
38
39        staff2[0] = new Staff();
40        staff2[0].setNama("Hendra");
41        staff2[0].setNip("0004");
42        staff2[0].setGolongan("3");
43        staff2[0].setLembur(15);
44        staff2[0].setGajiLembur(5500);
45
46        staff2[1] = new Staff();
47        staff2[1].setNama("Arie");
48        staff2[1].setNip("0006");
49        staff2[1].setGolongan("4");
50        staff2[1].setLembur(5);
51        staff2[1].setGajiLembur(100000);
52
53        staff2[2] = new Staff();
54        staff2[2].setNama("Mentari");
55        staff2[2].setNip("0007");
56        staff2[2].setGolongan("3");
57        staff2[2].setLembur(6);
58        staff2[2].setGajiLembur(20000);
59        man[1].setStaff(staff2);
60
61        // cetak informasi dari manager + staffnya
62        man[0].lihatInfo();
63        man[1].lihatInfo();
64    }
65 }
```

- Hasil run

```

Program Testing Class Manager & Staff
Manager :Administrasi
NIP :101
Nama :Tedjo
Golongan :1
Tunjangan :5000000
Gaji :.0f
Bagian :Administrasi
-----
NIP      :0003
Nama     :Usman
Golongan :2
Jml Lembur :10
Gaji Lembur :10000
Gaji :3100000
NIP      :0005
Nama     :Anugrah
Golongan :2
Jml Lembur :10
Gaji Lembur :55000
Gaji :3550000
-----
Manager :Pemasaran
NIP :102
Nama :Atika
Golongan :1
Tunjangan :2500000
Gaji :.0f
Bagian :Pemasaran

```

```

-----
NIP      :0004
Nama     :Hendra
Golongan :3
Jml Lembur :15
Gaji Lembur :5500
Gaji :2082500
NIP      :0006
Nama     :Arie
Golongan :4
Jml Lembur :5
Gaji Lembur :100000
Gaji :1500000
NIP      :0007
Nama     :Mentari
Golongan :3
Jml Lembur :6
Gaji Lembur :20000
Gaji :2120000
-----

```

1. Screenshot program code

```

1  package semester3.jobsheet9;
2  public class PerkalianKu {
3      void perkalian(int a, int b) {
4          System.out.println(a * b);
5      }
6      void perkalian(int a, int b, int c) {
7          System.out.println(a * b * c);
8      }
9      Run | Debug
10     public static void main(String[] args) {
11         PerkalianKu objek = new PerkalianKu();
12         objek.perkalian(a:25, b:43);
13         objek.perkalian(a:34, b:23, c:56);
14     }

```

Screenshot run java

```

1075
43792

```

## Questions

- 1) Dari source coding diatas terletak dimanakah overloading?
    - Overloading terjadi pada method perkalian yang didefinisikan dua kali dengan jumlah parameter yang berbeda.
  - 2) Jika terdapat overloading ada berapa jumlah parameter yang berbeda?
    - Jumlah parameter yang berbeda pada overloading di atas adalah:  
Method pertama memiliki 2 parameter (int a, int b).  
Method kedua memiliki 3 parameter (int a, int b, int c).
2. Screenshot program code

```
1  package semester3.jobsheet9;
2  public class PerkalianKu {
3      void perkalian(int a, int b) {
4          System.out.println(a * b);
5      }
6      void perkalian(double a, double b) {
7          System.out.println(a * b);
8      }
9      public static void main(String[] args) {
10         PerkalianKu objek = new PerkalianKu();
11         objek.perkalian(a:25, b:43);
12         objek.perkalian(a:34.56, b:23.7);
13     }
14 }
```

Screenshot run java

```
1075
819.072
```

## Questions

- 1) Dari source coding diatas terletak dimanakah overloading?
  - Overloading terjadi pada method perkalian, yang didefinisikan dua kali dengan jumlah parameter yang sama, tetapi tipe data parameter yang berbeda.
- 2) Jika terdapat overloading ada berapa tipe parameter yang berbeda?
  - Terdapat satu tipe parameter yang berbeda pada overloading di atas, yaitu tipe data parameter:  
Method pertama memiliki parameter dengan tipe data int.  
Method kedua memiliki parameter dengan tipe data double.

### 3. Screenshot program code

```
1 package semester3.jobsheet9;
2 public class Ikan {
3     public void swim() {
4         System.out.println(x:"Ikan bisa berenang");
5     }
6 }
7 class Piranha extends Ikan{
8     public void swim() {
9         System.out.println(x:"Piranha bisa makan daging");
10    }
11 }
12 class Fish {
13     Run | Debug
14     public static void main(String[] args) {
15         Ikan a = new Ikan();
16         Ikan b = new Piranha();
17         a.swim();
18         b.swim();
19     }
20 }
```

Screenshot run java

```
Ikan bisa berenang
Piranha bisa makan daging
```

### Questions

- 1) Dari source coding diatas terletak dimanakah overriding?
  - Overriding terjadi pada method swim yang didefinisikan di kelas Piranha, menggantikan method yang sama di kelas induk Ikan.
- 2) Jabarkanlah apabila sourcoding diatas jika terdapat overriding?
  - Method swim pada kelas Piranha mengoverride method swim pada kelas Ikan.
  - Ketika objek Piranha (objek b) dipanggil untuk memanggil method swim, akan dieksekusi method swim dari kelas Piranha, bukan dari kelas Ikan.
  - Ini menunjukkan bahwa perilaku method swim di kelas anak (Piranha) menggantikan perilaku method swim di kelas induk (Ikan).

### Assignment

#### 1. Overloading

Screenshot program code



```

1  package semester3.jobsheet9;
2  public class Segitiga {
3      private int sudut;
4
5      public int totalSudut(int sudutA) {
6          this.sudut = 180 - sudutA;
7          return this.sudut;
8      }
9
10     public int totalSudut(int sudutA, int sudutB) {
11         this.sudut = 180 - (sudutA + sudutB);
12         return this.sudut;
13     }
14
15     public int keliling(int sisiA, int sisiB, int sisiC) {
16         return sisiA + sisiB + sisiC;
17     }
18
19     public double keliling(int sisiA, int sisiB) {
20         double c = Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
21         return c;
22     }
23
24     Run | Debug
25     public static void main(String[] args) {
26         Segitiga segitiga = new Segitiga();
27         int totalSudut1 = segitiga.totalSudut(sudutA:60);
28         System.out.println("Total Sudut 1: " + totalSudut1);
29         int totalSudut2 = segitiga.totalSudut(sudutA:60, sudutB:30);
30         System.out.println("Total Sudut 2: " + totalSudut2);
31         int kelilingABC = segitiga.keliling(sisiA:3, sisiB:4, sisiC:5);
32         System.out.println("Keliling ABC: " + kelilingABC);
33         double kelilingAB = segitiga.keliling(sisiA:3, sisiB:4);
34         System.out.println("Keliling AB: " + kelilingAB);
35     }

```

Screenshot run java

```

Total Sudut 1: 120
Total Sudut 2: 90
Keliling ABC: 12
Keliling AB: 5.0

```

## 2. Overriding

Screenshot program code

```
1 package semester3.jobsheet9;
2 public class Manusia {
3     public void bernafas() {
4         System.out.println(x:"Manusia sedang bernafas");
5     }
6
7     public void makan() {
8         System.out.println(x:"Manusia sedang makan");
9     }
10 }
11
12 class Dosen extends Manusia {
13     public void makan() {
14         System.out.println(x:"Dosen sedang makan");
15     }
16
17     public void lembur() {
18         System.out.println(x:"Dosen sedang lembur");
19     }
20 }
21
22 class Mahasiswa extends Manusia {
23     public void makan() {
24         System.out.println(x:"Mahasiswa sedang makan");
25     }
26
27     public void tidur() {
28         System.out.println(x:"Mahasiswa sedang tidur");
29     }
30 }
```

```
1 package semester3.jobsheet9;
2 public class Main {
3     Run | Debug
4     public static void main(String[] args) {
5         Manusia manusia1 = new Dosen();
6         Manusia manusia2 = new Mahasiswa();
7         manusia1.bernafas();
8         manusia1.makan();
9         manusia2.bernafas();
10        manusia2.makan();
11    }
```

Screenshot run java

```
Manusia sedang bernafas
Dosen sedang makan
Manusia sedang bernafas
Mahasiswa sedang makan
```

