

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK
OVERLOADING DAN OVERRIDING



Disusun Oleh:

DIMAS SETYO NUGROHO (2341760188)

2C-SIB/10

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No. 9, Jattimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur
65141

Praktikum 1

1. Class Karyawan

```
OverloadOverRiding > praktikum1 > Karyawan.java > Karyawan > getGaji()
1  package OverloadOverRiding.primakulum1;
2
3  public class Karyawan {
4      private String nama;
5      private String nip;
6      private String golongan;
7      private double gaji;
8
9      // get dan set method
10     public void setName(String nama) {
11         this.nama = nama;
12     }
13
14     public void setNip(String nip) {
15         this.nip = nip;
16     }
17
18     public void setGolongan(String golongan) {
19         this.golongan = golongan;
20
21         // switch
22         switch (golongan.charAt(index:0)) {
23             case '1':
24                 this.gaji = 5000000;
25                 break;
26             case '2':
27                 this.gaji = 3000000;
28                 break;
29             case '3':
30                 this.gaji = 2000000;
31                 break;
32             case '4':
33                 this.gaji = 1000000;
34                 break;
35             case '5':
36                 this.gaji = 750000;
37                 break;
38         }
39     }
40
41     public void setGaji(double gaji)
42     {
43         this.gaji = gaji;
44     }
45     public String getName()
46     {
47         return nama;
48     }
49     public String getNip()
50     {
51         return nip;
52     }
53     public String getGolongan()
54     {
55         return golongan;
56     }
57     public double getGaji()
58     {
59         return gaji;
60     }
61 }
```

2. Class Manager

```

OverloadOverRiding > praktikum1 > Manager.java > Manager > setTunjangan(double)
1  package OverloadOverRiding.praktikum1;
2
3  public class Manager extends Karyawan{
4      // atribut
5      private double tunjangan;
6      private String bagian;
7      private Staff st[]; // array pada staff
8
9      // set dan get method
10     public void setTunjangan(double tunjangan)
11     {
12         this.tunjangan = tunjangan;
13     }
14     public double getTunjangan()
15     {
16         return tunjangan;
17     }
18     public void setBagian(String bagian)
19     {
20         this.bagian = bagian;
21     }
22     public String getBagian()
23     {
24         return bagian;
25     }
26     public void setStaff(Staff st[])
27     {
28         this.st = st;
29     }
30     public void viewStaff()
31     {
32         int i;
33         System.out.println(x:"-----");
34         // for looping untuk data dari st
35         for (i = 0; i < st.length; i++) {
36             st[i].lihatInfo();
37         }
38         System.out.println(x:"-----");
39     }

```

```

OverloadOverRiding > praktikum1 > Manager.java > Manager > viewStaff()
3  public class Manager extends Karyawan{
40     public void lihatInfo()
41     {
42         System.out.println("Manager : " + this.getBagian());
43         System.out.println("NIP : " + this.getNip());
44         System.out.println("Nama : " + this.getNama());
45         System.out.println("Golongan : " + this.getGolongan());
46         System.out.printf(format:"Tunjangan :%.0f\n",this.getTunjangan());
47         System.out.printf(format:"Gaji :%.0f\n",this.getGaji());
48         System.out.println("Bagian : " + this.getBagian());
49         System.out.println();
50         this.viewStaff();
51     }
52     // overloading
53     public double getGaji()
54     {
55         return super.getGaji() + tunjangan;
56     }
57 }

```

3. Class Staff

```

OverloadOverRiding > praktikum1 > Staff.java > Staff > lihatInfo()
1  package OverloadOverRiding.praktikum1;
2
3  public class Staff extends Karyawan {
4      // atribut
5      private int lembur;
6      private double gajiLembur;
7
8      public void setLembur(int lembur)
9      {
10         this.lembur = lembur;
11     }
12     public int getLembur()
13     {
14         return lembur;
15     }
16     public void setGajiLembur(double gajiLembur)
17     {
18         this.gajiLembur = gajiLembur;
19     }
20     public double getGajiLembur()
21     {
22         return gajiLembur;
23     }
24
25     // overloading method dari getGaji pada parent
26     public double getGaji(int lembur, double gajiLembur)
27     {
28         return getGaji() + lembur * gajiLembur;
29     }
30     // overriding dari kelas yang sama yang diambil dari parent
31     public double getGaji()
32     {
33         return super.getGaji() + lembur + gajiLembur;
34     }

```

```

OverloadOverRiding > praktikum1 > Staff.java > Staff > getGaji()
3  public class Staff extends Karyawan {
35     // info method
36     public void lihatInfo()
37     {
38         System.out.println("NIP : " + this.getNip());
39         System.out.println("Nama : " + this.getNama());
40         System.out.println("Golongan : " + this.getGolongan());
41         System.out.println("Jml Lembur : " + this.getLembur());
42         System.out.printf(format:"Gaji Lembur :%.0f\n", this.getGajiLembur());
43         System.out.printf(format:"Gaji :%.0f\n", this.getGaji());
44         System.out.println();
45     }
46 }

```

4. Class Utama

```

OverloadOverRiding > praktikum1 > 📄 Utama.java > 🚀 Utama > 🏠 main(String[])
1  package OverloadOverRiding.praktikum1;
2
3  public class Utama {
4      public static void main(String[] args) {
5          System.out.println("Program Testing Class Manager & Staff");
6          Manager man[] = new Manager(2);
7          Staff staff1[] = new Staff(2);
8          Staff staff2[] = new Staff(3);
9
10         // pembuatan manager
11         man[0] = new Manager();
12         man[0].setNama(nama:"Tedjo");
13         man[0].setNip(nip:"101");
14         man[0].setGolongan(golongan:"1");
15         man[0].setTunjangan(tunjangan:5000000);
16         man[0].setBagian(bagian:"Administrasi");
17
18         man[1] = new Manager();
19         man[1].setNama(nama:"Atika");
20         man[1].setNip(nip:"102");
21         man[1].setGolongan(golongan:"1");
22         man[1].setTunjangan(tunjangan:2500000);
23         man[1].setBagian(bagian:"Penasaran");
24         // staff
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         man[0].setStaff(staff1);

```

```

OverloadOverRiding > praktikum1 > 📄 Utama.java > 🚀 Utama > 🏠 main(String[])
3  public class Utama {
4      public static void main(String[] args) {
40         staff2[0] = new Staff();
41         staff2[0].setNama(nama:"Hendra");
42         staff2[0].setNip(nip:"0004");
43         staff2[0].setGolongan(golongan:"3");
44         staff2[0].setLembur(lembur:15);
45         staff2[0].setGajiLembur(gajiLembur:5500);
46
47         staff2[1] = new Staff();
48         staff2[1].setNama(nama:"Arie");
49         staff2[1].setNip(nip:"0006");
50         staff2[1].setGolongan(golongan:"4");
51         staff2[1].setLembur(lembur:5);
52         staff2[1].setGajiLembur(gajiLembur:100000);
53
54         staff2[2] = new Staff();
55         staff2[2].setNama(nama:"Mentari");
56         staff2[2].setNip(nip:"0007");
57         staff2[2].setGolongan(golongan:"3");
58         staff2[2].setLembur(lembur:6);
59         staff2[2].setGajiLembur(gajiLembur:20000);
60         man[1].setStaff(staff2);
61
62         // cetak informasi dari manager + stafnya
63         man[0].lihatInfo();
64         man[1].lihatInfo();
65
66     }

```

5. Hasil running

```

Program Testing Class Manager & Staff
Manager : Administrasi
NIP : 101
Nama : Tedjo
Golongan : 1
Tunjangan :5000000
Gaji :10000000
Bagian : Administrasi

-----
NIP : 0003
Nama : Usman
Golongan : 2
Jml Lembur : 10
Gaji Lembur :10000
Gaji :3010010

NIP : 0005
Nama : Anugrah
Golongan : 2
Jml Lembur : 10
Gaji Lembur :55000
Gaji :3055010

-----
Manager : Penasaran
NIP : 102
Nama : Atika
Golongan : 1
Tunjangan :2500000
Gaji :7500000
Bagian : Penasaran

```

```

-----
NIP : 0004
Nama : Hendra
Golongan : 3
Jml Lembur : 15
Gaji Lembur :5500
Gaji :2005515

NIP : 0006
Nama : Arie
Golongan : 4
Jml Lembur : 5
Gaji Lembur :100000
Gaji :1100005

NIP : 0007
Jml Lembur : 6
Gaji Lembur :20000
Gaji :2020006

-----
PS D:\folder matkul\semester 3\Praktikum PBO\Praktikum_FolderPBO>

```

Praktikum Latihan

1. Class Perkalianku

```

OverloadOverRiding > latihan1 > Perkalianku.java > Perkalianku
1 package OverloadOverRiding.latihan1;
2
3 public class Perkalianku {
4     void perkalian(int a, int b)
5     {
6         System.out.println(a * b);
7     }
8
9     void perkalian(int a, int b, int c)
10    {
11        System.out.println(a * b * c);
12    }
13
14    Run | Debug
15    public static void main(String[] args) {
16        Perkalianku objek = new Perkalianku();
17        objek.perkalian(a:25, b:43);
18        objek.perkalian(a:34, b:23, 56);
19    }
20
21 1075
22 43792
23 PS D:\folder matkul\semester 3\Praktikum PBO\Praktikum_FolderPBO>

```

2. Pertanyaan

- Dari source coding diatas terletak dimanakah overloading?

Jawaban:

Overloading terletak pada metode perkalian dari Class Perkalianku. Dimana terdapat dua metode perkalian dengan nama yang sama, tetapi memiliki jumlah parameter yang berbeda.

- Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Jawaban:

Terdapat 2 jumlah parameter yang berbeda dari masing-masing method dimana method perkalian 1 terdapat 2 parameter int a dan int b sedangkan method perkalian 2 terdapat 3 parameter yaitu int a, int b, int c.

3. Class Perkalianku

```
OverloadOverRiding > latihan1 > Perkalianku.java > Perkalianku
1  package OverloadOverRiding.latihan1;
2
3  public class Perkalianku {
4      void perkalian(int a, int b)
5      {
6          System.out.println(a * b);
7      }
8
9      // void perkalian(int a, int b, int c)
10     // {
11     //     System.out.println(a * b * c);
12     // }
13
14     void perkalian(double a, double b)
15     {
16         System.out.println(a * b);
17     }
18
19     public static void main(String[] args) {
20         Perkalianku objek = new Perkalianku();
21         objek.perkalian(a:25, b:43);
22         objek.perkalian(a:34.56, b:23.7);
23     }
```

4. Pertanyaan

- a. Dari source coding diatas terletak dimanakah overloading?

Jawaban:

Overloading terletak pada metode perkalian. Dimana terdapat dua buah metode perkalian dengan nama yang sama, tetapi dengan tipe parameter yang berbeda.

- b. Jika terdapat overloading ada berapa tipe parameter yang berbeda?

Jawaban:

Terdapat 2 tipe parameter yang berbeda, dimana method perkalian pertama menggunakan tipe parameter berupa integer. Sedangkan tipe parameter pada method kedua menggunakan tipe data berupa double.

5. Class Ikan, Piranha dan Fish

```
OverloadOverRiding > latihan1 > Ikan.java > ...
1  package OverloadOverRiding.latihan1;
2
3  public class Ikan {
4      public void swim()
5      {
6          System.out.println(x:"Ikan bisa berenang");
7      }
8  }
```

```
OverloadOverRiding > latihan1 > Piranha.java > ...
1  package OverloadOverRiding.latihan1;
2
3  public class Piranha extends Ikan {
4      public void swim()
5      {
6          System.out.println(x:"Piranha bisa makan daging");
7      }
8  }
```

```
OverloadOverRiding > latihan1 > Fish.java > ...
1  package OverloadOverRiding.latihan1;
2
3  public class Fish {
4      public static void main(String[] args) {
5          Ikan a = new Ikan();
6          Ikan b = new Piranha();
7
8          a.swim();
9          b.swim();
10     }
11 }
```

Running code

```
Ikan bisa berenang
Piranha bisa makan daging
PS D:\folder matkul\semester 3\Praktikum PBO\Praktikum_FolderPBO>
```

6. Pertanyaan

- a. Dari source coding diatas terletak dimanakah overriding?

Jawaban:

Overriding terjadi pada kelas Piranha, dimana metode swim() dalam kelas Piranha override terhadap metode swim() yang ada di kelas induknya, yaitu Ikan. Kedua metode memiliki nama dan parameter yang sama, namun implementasinya berbeda.

- b. Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

Jawaban:

Overriding terjadi ketika sebuah metode dalam subclass memiliki definisi yang sama (nama dan parameter) dengan metode di superclass, tetapi dengan implementasi yang berbeda. Pada contoh kasus dalam hal ini yaitu superclass Ikan memiliki metode swim() yang menampilkan "Ikan bisa berenang". Sedangkan subclass Piranha: Memiliki metode swim() yang meng-override metode di kelas Ikan dan menampilkan "Piranha bisa makan daging".

TUGAS

1. Overloading

Segitiga

```
OverloadOverRiding > Tugas > Segitiga.java > Segitiga > totalSudut(int)
1  package OverloadOverRiding.Tugas;
2
3
4  /**
5   * Segitiga
6   */
7  public class Segitiga {
8      // atribut
9      private int sudut;
10
11      public int getSudut() {
12          return sudut;
13      }
14
15      public void setSudut(int sudut) {
16          this.sudut = sudut;
17      }
18
19      public int totalSudut(int sudutA) {
20          sudut = 180 - sudutA;
21          return sudut;
22      }
23
24      public int totalSudut(int sudutA, int sudutB) {
25          sudut = 180 - (sudutA + sudutB);
26          return sudut;
27      }
28
29      public int keliling(int sisiA, int sisiB, int sisiC) {
30          return sisiA + sisiB + sisiC;
31      }
32
33      public double keliling(int sisiA, int sisiB) {
34          return Math.sqrt(Math.pow(sisiA, 2)) + Math.sqrt(Math.pow(sisiB, 2));
35      }
36  }
```

Main segitiga


```

OverloadOverRiding > Tugas > MainSegitiga.java > ...
1  package OverloadOverRiding.Tugas;
2
3  public class MainSegitiga {
4      public static void main(String[] args) {
5
6          Segitiga sgt = new Segitiga();
7
8          int sisiA = 9;
9          int sisiB = 4;
10         int sisiC = 16;
11
12         System.out.println();
13         System.out.print(s:"Sudut 1 sisi : ");
14         System.out.println(sgt.totalSudut(sisiA));
15         System.out.println();
16         System.out.print(s:"Sudut 2 sisi : ");
17         System.out.println(sgt.totalSudut(sisiA, sisiB));
18         System.out.println();
19         System.out.print(s:"KelAiling 3 sisi : ");
20         System.out.println(sgt.keliling(sisiA, sisiB, sisiC));
21         System.out.println();
22         System.out.print(s:"Keliling 2 sisi : ");
23         System.out.println(sgt.keliling(sisiA, sisiB));
24         System.out.println();
25     }
26 }

```

Hasil running

```

Sudut 1 sisi : 171
Sudut 2 sisi : 167
KelAiling 3 sisi : 29
Keliling 2 sisi : 13.0
PS D:\folder matkul\semester 3\Praktikum PBO\Praktikum FolderPBO>

```

2. Overriding

Parent Manusia

```

OverloadOverRiding > Tugas > Manusia.java > ...
1  package OverloadOverRiding.Tugas;
2
3  public class Manusia {
4      public void bernafas() {
5          System.out.println(x:"Manusia butuh bernafas");
6      }
7
8      public void makan() {
9          System.out.println(x:"Manusia butuh makan");
10     }
11 }

```

Child Dosen

```

OverloadOverRiding > Tugas > Dosen.java > Dosen
1  package OverloadOverRiding.Tugas;
2
3  public class Dosen extends Manusia {
4      public void lembur() {
5          System.out.println(x:"Dosen butuh lembur");
6      }
7
8      public void makan() {
9          System.out.println(x:"Dosen butuh makan");
10     }
11 }

```

Child Mahasiswa

```

OverloadOverRiding > Tugas > Mahasiswajava > Mahasiswa > makan()
1  package OverloadOverRiding.Tugas;
2
3  public class Mahasiswa extends Manusia {
4      public void tidur() {
5          System.out.println(x: "Mahasiswa butuh tidur");
6      }
7
8      public void makan() {
9          System.out.println(x: "Mahasiswa butuh makan");
10
11 }

```

Main Manusia

```

OverloadOverRiding > Tugas > MainManusia.java > MainManusia
1  package OverloadOverRiding.Tugas;
2
3  public class MainManusia {
4      public static void main(String[] args) {
5          // dynamic
6          Manusia manusia = new Manusia();
7          Manusia dos1 = new Dosen(); // disppatch
8          Manusia mhs1 = new Mahasiswa(); // dispatch
9          Dosen dosen = new Dosen();
10         Mahasiswa mahasiswa = new Mahasiswa();
11
12         // manusia
13         System.out.println(x: "\tManusia");
14         manusia.bernafas();
15         manusia.makan();
16         System.out.println();
17
18         System.out.println(x: "\tDosen");
19         dosen.lembur();
20         dos1.makan();
21         System.out.println();
22
23         System.out.println(x: "\tMahasiswa");
24         mahasiswa.tidur();
25         mhs1.makan();
26         System.out.println();
27     }
28 }

```

Hasil running

```

Manusia
Manusia butuh bernafas
Manusia butuh makan

Dosen
Dosen butuh lembur
Dosen butuh makan

Mahasiswa
Mahasiswa butuh tidur
Mahasiswa butuh makan

PS D:\folder matkul\semester 3\Praktikum PBO\Praktikum_FolderPBO>

```