

NAMA : MILA YUNITA
KELAS/NOMOR : TI-2C/12
NIM : 2041720027
MATA KULIAH : PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

JOBSHEET 7

OVERLOADING DAN OVERRIDING

4. Latihan

```
public class PerkalianKu {  
    void perkalian(int a, int b){  
        System.out.println(a * b);  
    }  
    void perkalian(int a, int b, int c){  
        System.out.println(a * b * c);  
    }  
    public static void main(String args []){  
        PerkalianKu objek = new PerkalianKu();  
        objek.perkalian(25, 43);  
        objek.perkalian(34, 23, 56);  
    }  
}
```

4.1 Dari source coding diatas terletak dimanakah overloading?

Jawab:

Overloading terletak pada 2 method dengan nama perkalian() yang memiliki parameter yang berbeda. Di bawah ini, yang diberi tanda merah, merupakan 2 method overloading.

```
6 package latihan_milay;  
7  
8 /**  
9  *  
10  * @author VivoBook  
11  */  
12 public class Perkalianku {  
13     void perkalian(int a, int b){  
14         System.out.println(a*b);  
15     }  
16  
17     void perkalian(int a, int b, int c){  
18         System.out.println(a * b * c);  
19     }  
20  
21     public static void main(String[] args) {  
22         Perkalianku objek = new Perkalianku();  
23  
24         objek.perkalian(25, 43);  
25         objek.perkalian(34, 23, 56);  
26     }  
27 }
```

NAMA : MILA YUNITA
KELAS/NOMOR : TI-2C/12
NIM : 2041720027
MATA KULIAH : PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

4.2 Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Jawab:

Dalam overloading yang ada di kode program diatas, terdapat satu parameter yang berbeda, yakni pada method perkalian() yang pertama, terdapat dua parameter yakni parameter a dan b yang bertipe data int. Sedangkan pada method perkalian kedua, terdapat tiga parameter yakni parameter a, b, dan c yang bertipe data int.

Potongan program untuk pertanyaan 4.3 dan 4.4

```
public class PerkalianKu {  
    void perkalian(int a, int b){  
        System.out.println(a * b);  
    }  
    void perkalian(double a, double b){  
        System.out.println(a * b);  
    }  
    public static void main(String args []){  
        PerkalianKu objek = new PerkalianKu();  
        objek.perkalian(25, 43);  
        objek.perkalian(34.56, 23.7);  
    }  
}
```

4.3 Dari source coding diatas terletak dimanakah overloading?

Jawab:

Berdasarkan source coding di atas, overloading terletak pada 2 method dengan nama perkalian() yang berada di dalam class PerkalianKu. Di bawah ini, yang diberi tanda merah, merupakan jenis method overloading.

NAMA : MILA YUNITA
KELAS/NOMOR : TI-2C/12
NIM : 2041720027
MATA KULIAH : PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

```
6 package latihan_milay;
7
8 /**
9  *
10  * @author VivoBook
11  */
12 public class Perkalianku {
13     void perkalian(int a, int b){
14         System.out.println(a*b);
15     }
16
17     void perkalian(double a, double b){
18         System.out.println(a * b);
19     }
20
21     public static void main(String[] args) {
22         Perkalianku objek = new Perkalianku();
23
24         objek.perkalian(25, 43);
25         objek.perkalian(34.56, 23.7);
26     }
27 }
```

4.4 Jika terdapat overloading ada berapa tipe parameter yang berbeda?

Jawab:

Dalam method overloading yang ada pada source program di atas, terdapat dua tipe parameter yang berbeda. Pada method perkalian() yang pertama, parameter a memiliki tipe data int sedangkan parameter a yang ada di method perkalian() yang kedua memiliki tipe data double. Kemudian parameter b yang ada di method perkalian() yang pertama memiliki tipe data int, sedangkan parameter b yang ada di method perkalian() yang kedua memiliki tipe data double.

NAMA : MILA YUNITA
KELAS/NOMOR : TI-2C/12
NIM : 2041720027
MATA KULIAH : PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

Potongan program untuk latihan nomor 4.5 dan 4.6

```
class Ikan{
    public void swim(){
        System.out.println("Ikan bisa berenang");
    }
}

class Piranha extends Ikan{
    public void swim(){
        System.out.println("Piranha bisa makan daging");
    }
}

public class Fish {
    public static void main(String[] args) {
        Ikan a = new Ikan();
        Ikan b = new Piranha();
        a.swim();
        b.swim();
    }
}
```

4.5 Dari source coding di atas terletak dimanakah overriding?

Jawab:

Pada source program di atas, overriding terletak pada method swim() yang ada di class Piranha. Class Piranha merupakan subchild dari class Ikan. Di bawah ini, yang diberi tanda merah merupakan method overriding.

```
6 package latihan_milay;
7
8 /**
9  *
10  * @author VivoBook
11  */
12 public class Ikan {
13     public void swim(){
14         System.out.println("Ikan bisa berenang");
15     }
16 }

6 package latihan_milay;
7
8 /**
9  *
10  * @author VivoBook
11  */
12 public class Piranha extends Ikan{
13     public void swim(){
14         System.out.println("Piranha bisa makan daging");
15     }
16 }
```

NAMA : MILA YUNITA
KELAS/NOMOR : TI-2C/12
NIM : 2041720027
MATA KULIAH : PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

4.6 Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

Jawab:

Dalam kode program di atas terdapat method overriding. Method overriding terjadi saat terdapat method dalam child yang memiliki nama dan parameter yang sama dengan method yang ada di class parentnya. Pada kode program tersebut, class Piranha merupakan child dari class Ikan. Di dalam class Piranha terdapat method swim() yang mana method swim() tersebut juga ada di dalam class Ikan. Method swim() yang ada di class Piranha dan class Ikan sama-sama memiliki nama method bernama swim(), tidak memiliki parameter, dan keduanya memiliki jenis kembalian null(void). Sehingga dapat disimpulkan bahwa method swim() yang ada pada class Ikan dan Piranha merupakan method overriding.

Method swim() yang ada di class Ikan terdapat perintah untuk menampilkan kalimat "Ikan bisa berenang". Sedangkan method swim() yang ada di class Piranha terdapat perintah untuk mencetak kalimat "Piranha bisa makan daging".