BAB 7
OVERLOADING DAN OVERRIDING

A. POKOK BAHASAN

- ⊕ Overloading
- ⊕ Overriding
- ⊕ Aturan tentang Overridden method

B. TUJUAN BELAJAR

Dengan Praktikum ini mahasiswa diharapkan dapat :

- ✓ Memahami tentang overloading
- ✓ Memahami tentang overriding
- ✓ Memahami aturan tentang overridden

C. DASAR TEORI

Overloading adalah suatu keadaan dimana ada beberapa method dengan nama yang sama pada suatu class tetapi dengan parameter yang berbeda (mempunyai implementasi dan return value). Tujuan dibuatnya overloading yaitu memudahkan pengguna method dengan fungsi yang hampir sama. Contoh :

```
public void print( String temp ){
System.out.println("Name:" + name);
System.out.println("Address:" +
address); System.out.println("Age:" +
age);
}

public void print(double eGrade, double mGrade, double
sGrade) System.out.println("Name:" + name);
System.out.println("Math Grade:" + mGrade);
System.out.println("English Grade:" + eGrade);
System.out.println("Science Grade:" + sGrade);
}
```

Overloading ini dapat terjadi pada class yang sama atau pada suatu parent class dan subclass-nya. Overloading mempunyai ciri-ciri sebagai berikut :

a. Nama Method harus sama
b. Daftar parameter harus berbeda
c. Return type boleh sama,juga boleh berbeda.

Overriding tidak sama dengan overloading, Overriding merupakan mekanisme dimana sebuah metode dapat dideklarasikan ulang pada kelas turunannya. Overriding mempunyai ciri-ciri sebagai berikut :

a. Nama Method harus sama
b. Daftar parameter harus sama
c. Return type harus sama

Berikut ini contoh terjadinya overriding dimana method RupiahVsDolar() pada class Sekarang meg-override method RupiahVsDolar() pada class Dulu.

```
class Dulu {
  public String RupiahVsDolar() {
    System.out.println("Rp 10.000");
  }
}
class Sekarang extends Dulu {
   public String RupiahVsDolar() {
    System.out.println("Rp 14.000");
  }
}
```

Method yang terkena override (overiden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-overide (overriding method).

D. PERCOBAAN

- **Percobaan 1**

Melakukan Overloading pada method.

Tulislah listing program berikut ini dan amati yang terjadi pada saat terjadinya overloading pada method.

```
class A {}
class B extends A {}
class C extends B {}
public class Overload
{
    void myOverload(A a)
    {
       System.out.println("Overload.myOverload(A a)");
    }
    void myOverload(B b)
    {
       System.out.println("Overload.myOverload(B b)");
    }
    public static void main(String[] args)
    {
        Overload o = new Overload();
        C c = new C();
        o.myOverload(c);
        /*
        *statement di atas akan menjalankan myOverload(B
b), karena
          *method tersebut lebih "dekat" dengan method
yang dicari
         *bila dibandingkan dengan myOverload(A a)
         */
    }}
```

- **Percobaan 2**

Melakukan Overloading pada method

Tulislah listing program berikut ini dan amati yang terjadi pada saat terjadinya overloading pada method.

```
public class Overload1
{
    void myMethod(short s)
    {
        System.out.println("short");
    }


    void myMethod(int i)
    {
        System.out.println("int");
    }
    void myMethod(long l)
    {
        System.out.println("long");
    }

    public static void main(String[] args)
    {
        byte b = 1;

        Overload1 o = new Overload1();
        o.myMethod(b);
        /*
         *statement di atas akan menghasilkan "short",
         *hal ini karena short lebih "dekat" dengan
           *byte bila dibandingkan dengan int ataupun
long.
         */
    }
}
```