

Nama : Cahyo Dwi Putro	
NBI : 1961900333	
Kelas : C9	
Praktikum : Pengrograman Berorientasi Objek	
Tugas praktikum ke-2	

- Method overriding adalah sebuah method yang parent classnya diulang kembali oleh subclass. Aturan method overriding pada Java :
 - * parameter yang terdapat pada method overriding di subclass harus sama dengan parameter yang terdapat pada parent class.
 - * Aturan hak akses, hak akses method overriding di subclass tidak boleh lebih ketat dibandingkan dengan hak akses method pada parent class.

Contoh Sederhana :

```
public class contohoverride {
    public static void main(String[] args) {
        Mahasiswa mhs = new Mahasiswa();
        mhs.data();
    }
}

class Mahasiswa {
    public void data() {
        System.out.println("Mahasiswa");
    }
}
```

- Method overloading adalah kemampuan yang memperbolehkan sebuah class mempunyai 2 atau lebih method dengan nama yang sama, yang membedakannya adalah parameteranya. pada method overloading menunjuk :
 - * jumlah parameter.
 - * Tipe data dari parameter.
 - * urutan dari tipe data dan parameter.

method overloading juga dikaitkan dengan sebuah static polymorphism.

Contoh Sederhana :

Public class ContohOverloading {

```
public static void main(String[] args) {  
    Penggunaan c0 = new Penggunaan();  
    c0.jumlah(83, 32);  
    c0.jumlah(34, 20, 100);  
    c0.jumlah(42.10, 54);  
    c0.jumlah(28, 10.11);
```

3

class Penggunaan {

```
public void jumlah(int a, int b) {  
    System.out.println("jumlah 2 angka = " + (a+b));
```

3

// Overloading Perbedaan jumlah parameter

```
public void jumlah(int a, int b, int c) {  
    System.out.println("jumlah 3 angka = " + (a+b+c));
```

3

// Overloading perbedaan tipe data parameter (double + int)

```
public void jumlah(double a, int b) {  
    System.out.println("jumlah 2 angka = " + (a+b));
```

3

// Overloading perbedaan urutan tipe data parameter (int + double)

```
public void jumlah(int b, double a) {  
    System.out.println("jumlah 2 angka = " + (a+b));
```

3

3

3. Tabel jangkauan untuk masing-masing modifier:

modifier	class	package	subclass	world	keterjangan :
public	Y	Y	Y	Y	* Y artinya bisa diakses;
protected	Y	Y	Y	N	* N artinya tidak bisa diakses;
default	Y	Y	N	N	* Subclass artinya class anak;
private	Y	N	N	N	* world artinya seluruh package gpp;

- a. Default : class dan member hanya akan bisa diakses dari class itu sendiri dan package (class yang berada pada package orangnya).

contoh :

```
public class Pengjumlahan {  
    int jumlah(int a, int b) {  
        return a+b;  
    }  
}
```

b. public : membuat member dan class bisa diakses dari mana saja. contoh :

```
class Person {  
    public String name;  
    public changeName(String newname) {  
        this.name = newname;  
    }  
}
```

Pada class Person terdapat 2 member
yaitu : * atribut name
* method changeName

Kedua member tersebut bisa diakses
dari mana saja.

c. private : membuat member dan class hanya bisa diakses dari dalam class

itu sendiri. Sebagai contoh ada 2 point yang perlu diingat :

1. modifier private tidak bisa diberikan kepada class, enum, dan interface.
2. modifier private hanya bisa diberikan kepada member class.

Contoh :

```
class Person {  
    private String name;  
}
```

```
public void setName(name) {  
    this.name = name;  
}
```

```
public String getName() {  
    return this.name;  
}
```

d. protected : membuat member dan class hanya bisa diakses dari :

1. class itu sendiri
2. subclass atau class anak
3. package (class yang berada pada satu package dengannya)

Contoh :

```
public class Person {  
    protected String name;  
  
    public void setname(String name) {  
        this.name = name;  
    }  
  
    public String getname() {  
        return this.name;  
    }  
}
```

Pada Contoh disamping tiga memberikan modifier `protected` pada attribute `name`. Apabila dicoba diakses dari class yang di luar package dengannya maka tidak akan terjadi error dan apabila dicoba diakses dari luar package maka akan terjadi error karena attribute `name` telah diberikan modifier `protected`.

q. `Super()` : Jika suatu method mengoverride satu dari method superclass, kita bisa menggunakan method yang override tersebut melalui penggunaan keyword `super`. Selain dari itu juga dapat menggunakan keyword `super` untuk menyalin perintah suatu field yang tersenarai. Contoh :

```
public class ContohSuper {  
    public static void main(String[] args) {  
        Employee emp = new Employee();  
    }  
}  
  
class Person {  
    public Person() {  
        System.out.println("Eksesi konstruktor Person");  
    }  
}  
  
class Employee extends Person {  
    public Employee() {  
        super();  
        System.out.println("Eksesi konstruktor Employee");  
    }  
}
```