

# **PEMOGRAMAN JAVA**

Yoannita

**Overriding Method**

**Overloading Method**

# Method overriding

```
class A
{
    void tampil() {
        System.out.println("ini method class A");
    }
}
```

```
class B extends A
{
    void tampil() {
        System.out.println("ini method class B");
    }
}
```

```
class overriding
{
    public static void main(String args[])
    {
        B orang = new B();
        orang.tampil();
    }
}
```

- Untuk beberapa pertimbangan, terkadang class asal perlu mempunyai implementasi berbeda dari method yang khusus dari *superclass* tersebut.
- Oleh karena itulah, method overriding digunakan. *Subclass* dapat mengesampingkan method yang didefinisikan dalam *superclass* dengan menyediakan implementasi baru dari method tersebut. (subclass dapat meng-override fungsi yang didapat dari superclass)

# Method overloading

- Dalam *class* yang kita buat, kadangkala kita menginginkan untuk membuat ***method*** dengan nama yang sama namun mempunyai **fungsi yang berbeda menurut parameter yang digunakan**. Kemampuan ini dimungkinkan dalam pemrograman Java, dan dikenal sebagai *overloading method*.
- *Overloading method* mengizinkan sebuah *method* dengan nama yang sama namun memiliki parameter yang berbeda sehingga mempunyai implementasi dan *return value* yang berbeda pula. Daripada memberikan nama yang berbeda pada setiap pembuatan *method*, *overloading method* dapat digunakan pada operasi yang sama namun berbeda dalam implementasinya.

# Method overloading : Contoh

```
public class Bentuk {  
    ...  
    public void Gambar(int t1) {  
        ...  
    }  
    public void Gambar(int t1, int t2) {  
        ...  
    }  
    public void Gambar(int t1, int t2, int t3) {  
        ...  
    }  
    public void Gambar(int t1, int t2, int t3, int t4) {  
        ...  
    }  
}
```

# Method overloading

<u>return type</u>	<u>nama method</u>	<u>daftar parameter</u>
void	Gambar	(int t1)
void	Gambar	(int t1, int t2)
void	Gambar	(int t1, int t2, int t3)
void	Gambar	(int t1, int t2, int t3, int t4)
<hr/>	<hr/>	<hr/>
↓	↓	↓
sama	sama	berbeda

# Method overloading

```
class pegawai
```

```
{  
    void tampil(String nip, String nama, String jabatan) {  
        System.out.println("Tampil 3 variabel");  
        System.out.println("-----");  
        System.out.println("NIP    : " + nip);  
        System.out.println("Nama   : " + jabatan);  
        System.out.println("Jabatan : " + jabatan);  
        System.out.println();  
    }  
    void tampil(String nip, String nama) {  
        System.out.println("Tampil 2 variabel");  
        System.out.println("-----");  
        System.out.println("NIP    : " + nip);  
        System.out.println("Nama   : " + jabatan);  
        System.out.println();  
    }  
    void tampil(String nip) {  
        System.out.println("Tampil 1 variabel");  
        System.out.println("-----");  
        System.out.println("NIP    : " + nip);  
        System.out.println();  
    }  
}
```

```
class overloading {  
    public static void main(String args[]) {  
        pegawai pgw1, pgw2,pgw3;  
        pgw1 = new pegawai();  
        pgw2 = new pegawai();  
        pgw3 = new pegawai();  
  
        pgw1.tampil("101","Budi","Direktur");  
        pgw2.tampil("102");  
        pgw3.tampil("103","Desi");  
    }  
}
```