



Bab 5

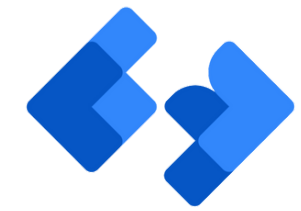
Overloading dan Overriding

Definisi Overloading



Method Overloading adalah sebuah kemampuan yang **membolehkan** sebuah class mempunyai 2 atau lebih method dengan **nama** yang **sama**. Dengan menggunakan method overloading , kita dapat membuat **beberapa** versi dari sebuah **method** dengan **statemen** yang **berbeda** berdasarkan **parameter** yang diberikan padanya. Jadi , ini akan membantu **menggunakan** kembali nama method yang sama untuk **tujuan** yang **berbeda**, dengan begitu method overloading akan memberikan **fleksibilitas** saat menggunakan sebuah **method**. Perbedaan parameter mencakup : **Jumlah** parameter , **tipe data** dari parameter tersebut atau urutan parameter .

Definisi Overloading



```
1 public class penjumlahan {
2
3     public void hitjumlah(int a, int b) {
4         System.out.println("Jumlah 2 angka =" + (a + b));
5     }
6
7     //overloading perbedaan jumlah parameter
8     public void hitjumlah(int a, int b, int c) {
9         System.out.println("Jumlah 3 angka =" + (a + b + c));
10    }
11
12    //overloading perbedaan tipe data parameter atau urutan parameter
13    public void hitjumlah(double a, int b) {
14        System.out.println("Jumlah 2 angka (double+int)= " + (a + b));
15    }
16 }
```

Output - PraktikumOOP_2118112 (run) ×

run:
Jumlah 2 angka (double+int)= 6.5
Jumlah 2 angka =14
Jumlah 3 angka =70
BUILD SUCCESSFUL (total time: 0 seconds)

Definisi Overriding



Overriding method adalah kemampuan dari subclass(child class) untuk **memodifikasi** method dari superclass, dengan cara **mendefinisikan** kembali method superclass-nya. Karena seringkali menemukan method yang tidak sesuai jika diterapkan pada subclass , dengan overriding subclass dapat menyesuaikan implementasi dari method yang diwarisi dari superclass sesuai kebutuhan subclass.

Aturan Overriding:

- 1.Parameter yang terdapat pada method Overriding di subclass harus sama dengan parameter yang terdapat pada parent class.
- 2.method pada subclass class yang akan di override, sebaiknya tidak menggunakan hak akses *private*, karena method pada subclass tidak boleh membatasi aksesibilitas yang lebih luas dari pada method pada parent class.

Class Diagram Overloading



Tunjangan
+ HitungTunjangan(int gajiDasar): double
+ HitungTunjangan(int gajiDasar,int tahunPengalaman): double
+ HitungTunjangan(int gajiDasar,int tahunPengalaman,String jenisPekerjaan): double

Implementasi Overloading



```
11 public class Tunjangan {
12
13     // Metode overload untuk menghitung tunjangan berdasarkan gaji dasar
14     public double hitungTunjangan(int gajiDasar) {
15         return gajiDasar * 0.1;
16     }
17
18     // Metode overload untuk menghitung tunjangan berdasarkan gaji dasar dan tahun pengalaman
19     public double hitungTunjangan(int gajiDasar, int tahunPengalaman) {
20         double tunjangan = gajiDasar * 0.1;
21         if (tahunPengalaman >= 5) {
22             tunjangan += 100000;
23         }
24         return tunjangan;
25     }
26
27     // Metode overload untuk menghitung tunjangan berdasarkan gaji dasar, tahun pengalaman, dan jenis pekerjaan
28     public double hitungTunjangan(int gajiDasar, int tahunPengalaman, String jenisPekerjaan) {
29         double tunjangan = gajiDasar * 0.1;
30         if (tahunPengalaman >= 5) {
31             tunjangan += 300000;
32         }
33         if (jenisPekerjaan.equalsIgnoreCase("manager")) {
34             tunjangan += 500000;
35         }
36         return tunjangan;
37     }
38 }
```

Implementasi Overloading



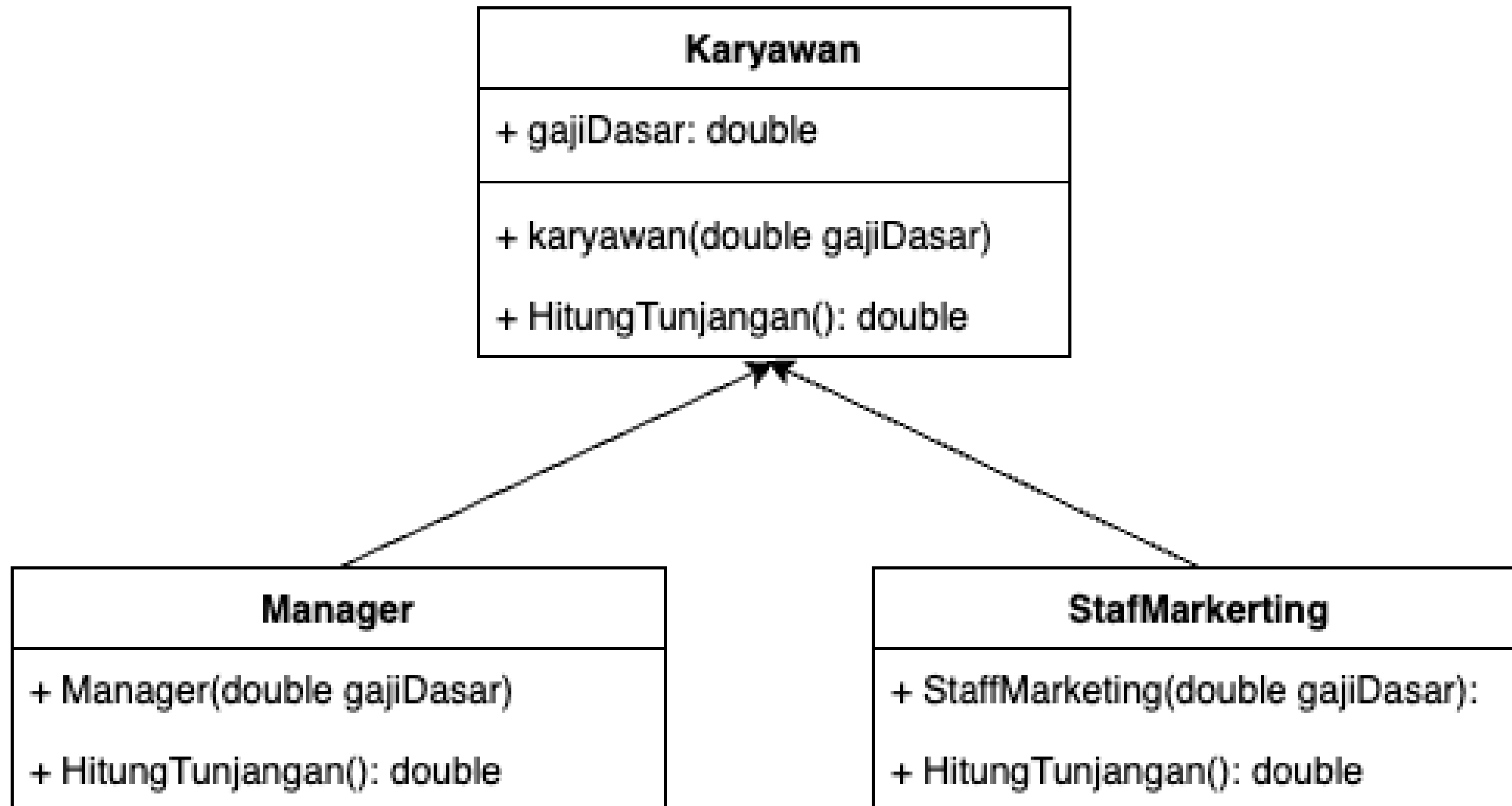
```
11 public class mainTunjangan {
12     public static void main(String[] args) {
13         Tunjangan hitTunjangan = new Tunjangan();
14
15         double tunjanganKaryawanBaru = hitTunjangan.hitungTunjangan( gajiDasar:4000000);
16         System.out.println("Tunjangan dari karyawan baru : " + tunjanganKaryawanBaru);
17
18         double tunjanganKaryawanLama = hitTunjangan.hitungTunjangan( gajiDasar:6000000, tahunPengalaman:7);
19         System.out.println("Tunjangan dari karyawan lama : " + tunjanganKaryawanLama);
20
21         double tunjanganjabatan = hitTunjangan.hitungTunjangan( gajiDasar:9000000, tahunPengalaman:3, jenisPekerjaan:"manager");
22         System.out.println("Tunjangan dari pengalaman dan jabatan :" + tunjanganjabatan);
23     }
24 }
```


Hasil Overloading

A screenshot of an IDE's output window. The window has a title bar with red, yellow, and green window control buttons. The title is "Output - PraktikumOOP_2118112 (run)" with a close button. On the left side of the output area, there are four icons: a green double arrow, a yellow double arrow, a red square, and a blue gear. The output text is as follows:

```
run:
Tunjangan dari karyawan baru : 400000.0
Tunjangan dari karyawan lama : 700000.0
Tunjangan dari pengalaman dan jabatan :1400000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```


Class Diagram Overriding



Implementasi Overriding



```
12 public class karyawan {  
13     public double gajiDasar;  
14  
15     public karyawan(double gajiDasar) {  
16         this.gajiDasar = gajiDasar;  
17     }  
18  
19     public double hitungTunjangan() {  
20         return gajiDasar * 0.1;  
21     }  
22 }
```

Implementasi Overriding



```
11 public class Manager extends karyawan {  
12  
13     public Manager(double gajiDasar) {  
14         super(gajiDasar);  
15     }  
16  
17     @Override  
18     public double hitungTunjangan() {  
19         return gajiDasar * 0.2 + 400000;  
20     }  
21 }
```

Implementasi Overriding



```
11 public class StaffMarketing extends karyawan{
12     public StaffMarketing(double gajiDasar) {
13         super(gajiDasar);
14     }
15
16     @Override
17     public double hitungTunjangan() {
18         return gajiDasar * 0.1 + 200000;
19     }
20 }
```

Implementasi Overriding



```
11 public class mainTunjanganKaryawan {  
12     public static void main(String[] args) {  
13         Manager manager = new Manager( gajiDasar:8000000);  
14         StaffMarketing staff = new StaffMarketing( gajiDasar:4500000);  
15  
16         System.out.println("Tunjangan Manager: " + manager.hitungTunjangan());  
17         System.out.println("Tunjangan Staff Marketing: " + staff.hitungTunjangan());  
18     }  
19 }
```

Hasil Overriding

A screenshot of an IDE's output window. The window has a title bar with red, yellow, and green window control buttons. The title is "Output - PraktikumOOP_2118112 (run) x". On the left side of the output area, there are four icons: a green double arrow, a yellow double arrow, a red square, and a small robot icon. The output text is as follows:

```
run:  
Tunjangan Manager: 2000000.0  
Tunjangan Staff Marketing: 650000.0  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```



"Open NetBeans, and let's write code Java"

