

# Pemrograman Berorientasi Objek

MODUL 4 OVERLOADING & OVERRIDING

ASISTEN LABORATORIUM

Tujuan Pembelajaran :

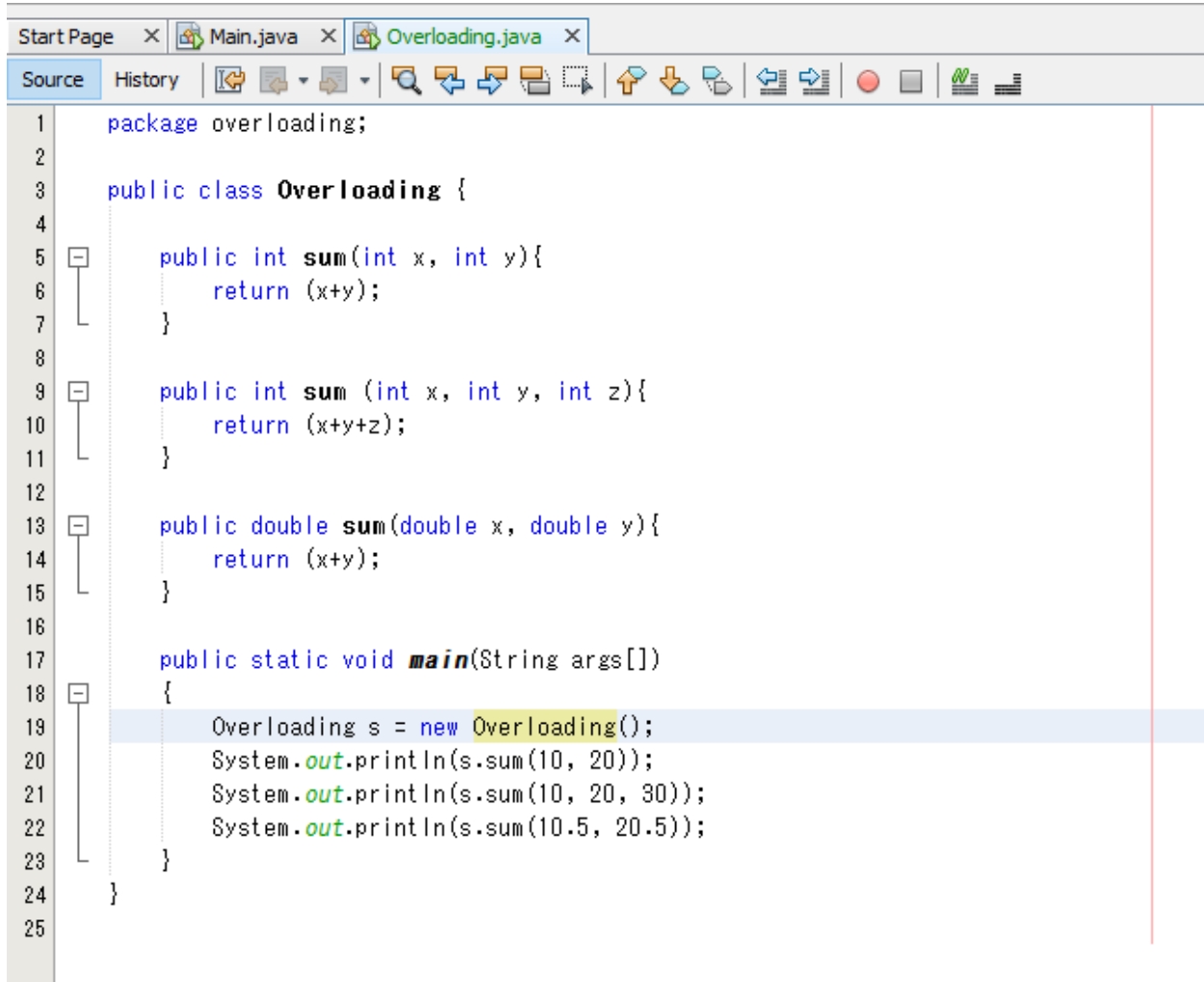
1. Mengetahui apa itu overloading
  2. Mengetahui apa itu overriding
  3. Mengetahui apa fungsi keyword final dan static
- 

### Overloading

Overloading adalah proses dimana adanya method atau fungsi yang berbeda-beda tetapi memiliki nama yang sama, tetapi penentu dari fungsi atau pembeda dari methodnya adalah dari jumlah parameter input atau tipe data dari parameter method tersebut.

```
class Overloading {  
  
    public void method(){  
        ...  
    }  
  
    public void method(int a){  
        ...  
    }  
  
    public void method(int a, int b){  
        ...  
    }  
  
    public void method(double a){  
        ...  
    }  
  
    public void method(double a, double b){  
        ...  
    }  
  
}
```

Contoh :



```

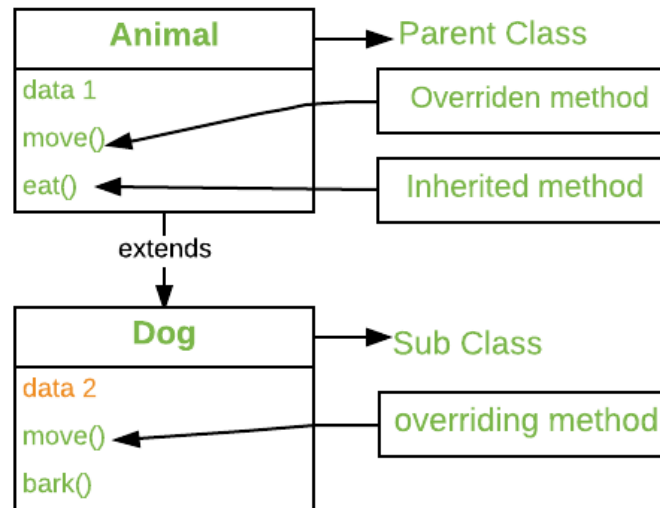
1  package overloading;
2
3  public class Overloading {
4
5      public int sum(int x, int y){
6          return (x+y);
7      }
8
9      public int sum (int x, int y, int z){
10         return (x+y+z);
11     }
12
13     public double sum(double x, double y){
14         return (x+y);
15     }
16
17     public static void main(String args[])
18     {
19         Overloading s = new Overloading();
20         System.out.println(s.sum(10, 20));
21         System.out.println(s.sum(10, 20, 30));
22         System.out.println(s.sum(10.5, 20.5));
23     }
24 }
25

```

## Overriding

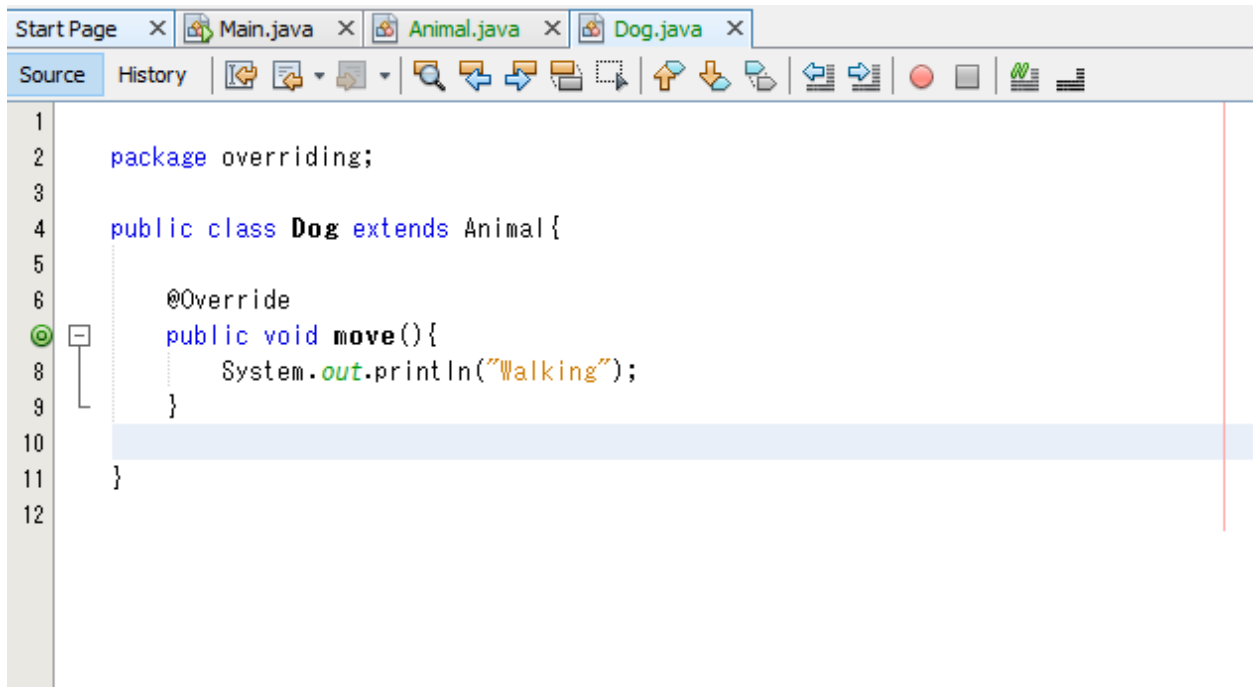
Fitur yang memungkinkan subclass atau kelas anak menyediakan implementasi spesifik dari metode yang sudah disediakan oleh super class atau kelas indukannya. Sehingga metode dalam subclass memiliki nama yang sama dengan method super classnya (parameter dan jenis return) tetapi memiliki proses yang berbeda dengan super classnya. Jadi dapat dibilang bahwa subclass menempa method dari superclassnya .

## MODUL 4 OVERLOADING & OVERRIDING



Contoh :

```
Start Page x Main.java x Animal.java x Dog.java x
Source History
1 package overriding;
2
3 public class Animal {
4
5     public String name;
6
7     public void move(){
8         System.out.println("Swimming");
9     }
10
11     public void eat(){
12         System.out.println("Eating");
13     }
14 }
15
16
```



```
1
2 package overriding;
3
4 public class Dog extends Animal{
5
6     @Override
7     public void move(){
8         System.out.println("Walking");
9     }
10
11 }
12
```

### Static Keyword

Static adalah non-akses modifier pada java yang bias digunakan :

- a. Blocks
- b. Variable
- c. Methods
- d. Nested class

Saat static member di deklarasikan, maka dapat diakses sebelum objek tersebut dibuatkan dan tanpa mereferensi dari sebuah objek. Contoh :

```
// Java program to demonstrate that a static member
// can be accessed before instantiating a class
class Test
{
    // static method
    static void m1()
    {
        System.out.println("from m1");
    }
}
```

```
public static void main(String[] args)
{
    // calling m1 without creating
    // any object of class Test
    m1();
}
```

### Final Keyword

Final digunakan pada konteks yang berbeda. Final adalah sebuah non-akses modifier yang dapat digunakan pada variable, method, dan kelas. Beberapa fungsi dari final

<b>Final Variable</b>	→	<b>To create constant variables</b>
<b>Final Methods</b>	→	<b>Prevent Method Overriding</b>
<b>Final Classes</b>	→	<b>Prevent Inheritance</b>

```
class Final {
    // a final variable
    final int THRESHOLD = 5;
    // a blank final variable
    final int THRESHOLD;
    // a final static variable PI
    static final double PI = 3.141592653589793;
    // a blank final static variable
    static final double PI;
}
```