



unindra
universitas indraprasta PGRI

Parameter

Konstruktor

PERTEMUAN 6

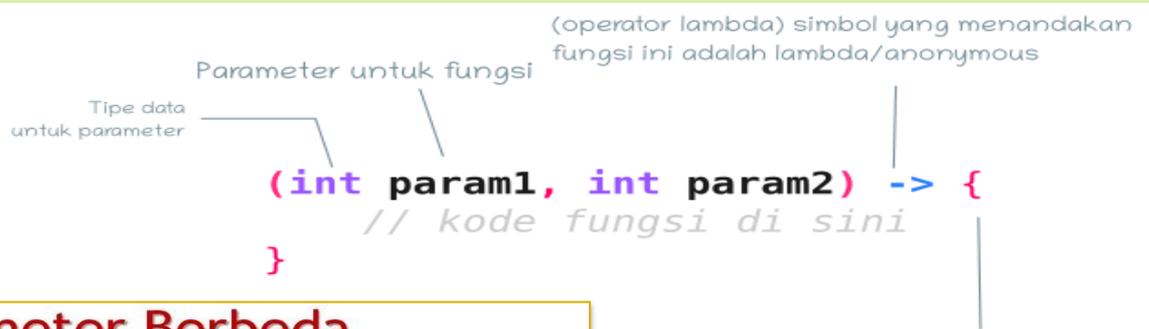
PARAMETER

- Parameter merupakan nilai yang bisa kita oper pada fungsi, seperti konstruktor atau method, parameter tersebut bisa kita gunakan jika nilai didalam fungsi tersebut ingin kita rubah, kita dapat menambahkan beberapa parameter dalam satu fungsi.
- Selain pada konstruktor, kita dapat menambahkan parameter pada method, parameter sangat diperlukan untuk memudahkan seorang pengembang untuk mengembangkan aplikasinya.

nama_class (type data property)

```
{  
    statement  
}
```

Sesuaikan nilai parameter dengan parameter yang sudah di set pada parameter (type data, property)



Jika fungsi anonymous hanya memiliki satu baris statement/ekspresi maka boleh tidak menggunakan kurung kurawal

Dengan Type Parameter Berbeda

```
public class Bidang {
```

```
    public int luas(int x, int y) {  
        return x * y;  
    }
```

Method ini hanya akan dijalankan jika dipanggil dengan menyertakan **2 parameter tipe integer**

```
    public double luas(int a, double t) {  
        return a * 0.5 * t;  
    }
```

Method ini hanya akan dijalankan jika dipanggil dengan menyertakan **1 parameter tipe integer dan 1 tipe double**

```
    public double luas(int d) {  
        double r = 0.5 * d;  
        return 3.14 * r * r;  
    }  
}
```

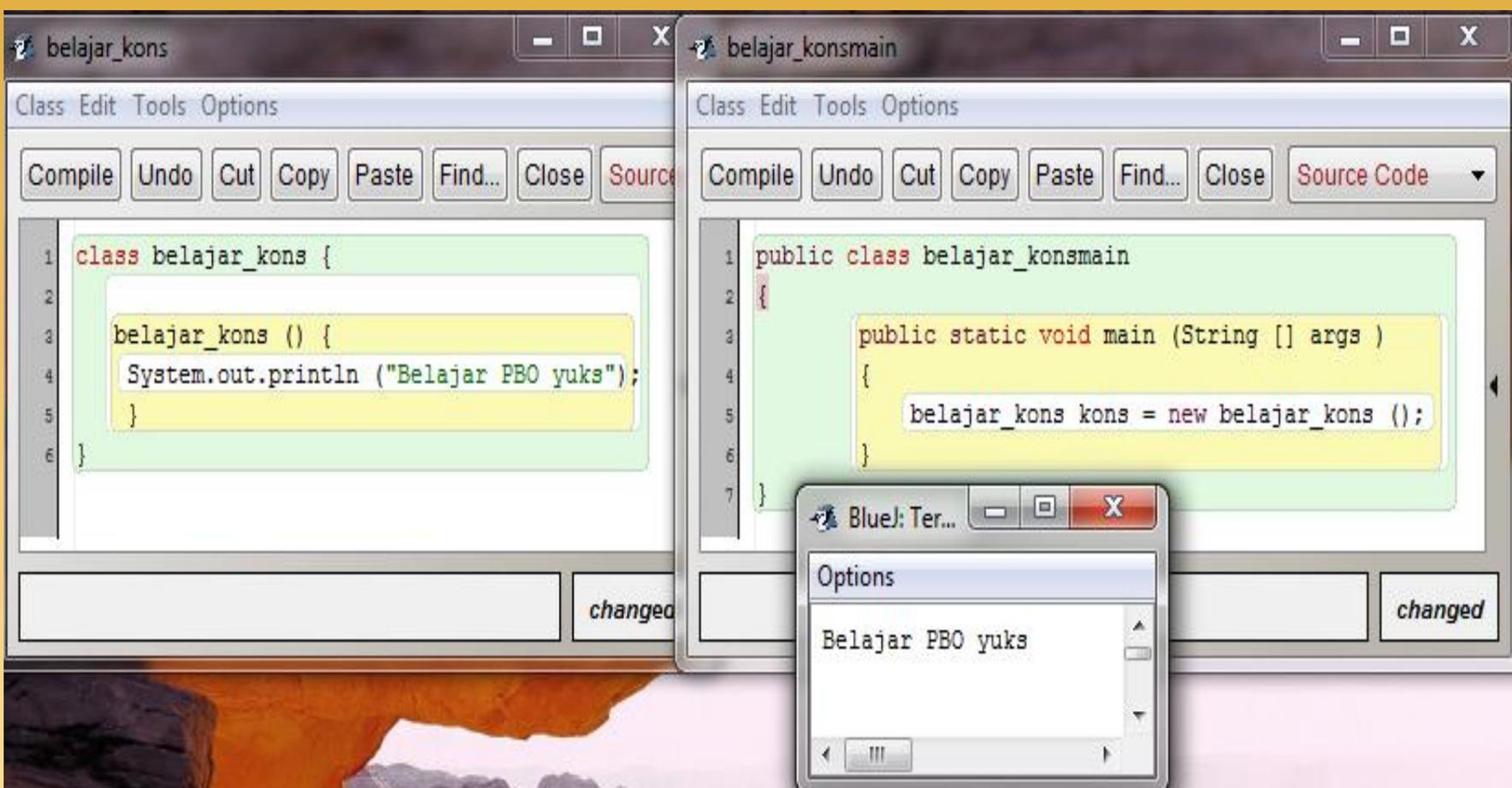
Method ini hanya akan dijalankan jika dipanggil dengan menyertakan **1 parameter tipe integer**

KONSTRUKTOR

- Konstruktur ialah sebuah method yang namanya sama persis dengan nama class-nya.
- Konstruktur sendiri berfungsi untuk memberikan nilai awal pada sebuah class ketika class tersebut dibuat dalam bentuk objek pada class lain.
- Konstruktur bisa memiliki parameter bisa juga tidak.
- Dalam konstruktur, tidak boleh pakai keyword void.

- Metode ini di panggil secara otomatis oleh java ketika **new** dipakai untuk menciptakan instance kelas.
- Telah tersirat pada pembahasan sebelumnya, konstruktor sangatlah penting pada pembentukan sebuah *object*. Konstruktor adalah *method* dimana seluruh inisialisasi *object* ditempatkan.
- Konstruktor tidak memiliki *return value*.
- Suatu konstruktor tidak dapat diwariskan (inherited), harus menggunakan keyword `super()`
- Pada onstructor yang di overload pada class, maka anda bisa menggunakan `this()`

CONTOH KONSTRUKTOR TANPA PARAMETER



CONTOH KONSTRUKTOR dengan PARAMETER

The image displays two IDE windows and a terminal window. The left window, titled 'belajar_konsmainP', contains the following code:

```
1 public class belajar_konsmainP
2 {
3     public static void main (String [] args )
4     {
5         belajar_konsparameter kons = new belajar_konsparameter ("ANDI", 15 );
6         kons.info ();
7     }
8 }
```

The right window, titled 'belajar_konsparameter', contains the following code:

```
1 class belajar_konsparameter
2 {
3     String nama;
4     int umur;
5
6     belajar_konsparameter (String nama1,int umur1)
7     {
8         this.nama = nama1;
9         this.umur = umur1;
10    }
11
12    public void info ()
13    {
14        System.out.println ("NAMA : " + this.nama );
15        System.out.println ("UMUR : " + this.umur + " Tahun" );
16    }
17 }
```

The terminal window, titled 'Blue: Terminal Window - PBO...', shows the output of the program:

```
Options
NAMA : ANDI
UMUR : 15 Tahun
```

CONTOH PROGRAM KONSTRUKTOR dan PARAMETER

```
class mhs{
    String npm,nama;
    int tugas,uts,uas;

    mhs(String npm,String nama){
        this.npm=npm;
        this.nama=nama;
    }
    mhs(int tugas, int uts, int uas){
        this.tugas=tugas;
        this.uts=uts;
        this.uas=uas;
    }
    double total(){
        return (0.2*tugas)+(0.3*uts)+(0.5*uas);
    }
    void tampil(){
        System.out.println("NPM Mhs : "+this.npm);
        System.out.println("Nama Mhs : "+this.nama);
    }
    void tampil2(){
        System.out.println("Nilai Tugas : "+this.tugas);
        System.out.println("Nilai UTS : "+this.uts);
        System.out.println("Nilai UAS : "+this.uas);
        System.out.println("Nilai Akhir : "+this.total());
    }
}
```

CONTOH

```
public class Nilai{  
    public static void main(String[] args) {  
        mhs mahasiswa=new mhs("001", "salman");  
        mhs mahasiswa2=new mhs(80,90,90);  
        mahasiswa.tampil();  
        mahasiswa2.tampil2();  
    }  
}
```

HASIL :

```
NPM Mhs : 001  
Nama Mhs : salman  
Nilai Tugas : 80  
Nilai UTS : 90  
Nilai UAS : 90  
Nilai Akhir : 88.0
```

Latihan

Method dan Konstruktor dan Parameter

1. Berikan penjelasan Anda simpulan dari apa dan kegunaan Method void dan return dalam Pemrograman Berorientasi Objek ?
2. Berikan penjelasan Anda simpulan dari apa dan kegunaan Konstruktor dan Parameter dalam Pemrograman Berorientasi Objek ?
3. Buatlah program class yang isinya sesuai dengan ketentuan :
 - a. NPM Ganjil buatlah dengan tema Penggajian
 - b. NPM Genap buatlah dengan tema Koperasi
 - c. Tiap mahasiswa membuat minimal 3 method (void dan non-void)dengan masing – masing method mendeklarasikan minimal 3 atribut
 - d. Gunakan access specifier/modifier dalam atributnya
 - e. Gunakan method konstruktor dengan paramater dan tanpa parameter