



PEMROGRAMAN JAVA

Yoannita, S.Kom



- **Class & Method sederhana**

“We have a hunger of the mind which asks for knowledge of all around us, and the more we gain, the more is our desire; the more we see, the more we are capable of seeing.”

(Maria Mitchell)

Method

- Method atau metode adalah fungsi yang didefinisikan di dalam kelas dan beroperasi pada instance dari kelas tersebut.
- Pada contoh latihan-latihan sebelumnya, kita hanya memiliki satu method, yaitu method main(). Di dalam Java, kita dapat mendefinisikan banyak method yang akan kita panggil dari method yang berbeda.
- Karakteristik dari method :
 1. dapat mengembalikan satu nilai atau tidak sama sekali
 2. dapat diterima beberapa parameter yang dibutuhkan atau tidak ada parameter sama sekali. Parameter bisa juga disebut sebagai argumen dari fungsi
 3. setelah method telah selesai dieksekusi, dia akan kembali pada method yang memanggilnya.

Method

mengapa kita butuh untuk membuat banyak method? Mengapa kita tidak menuliskan semua kode pada sebuah method?

- Hal ini karena penyelesaian masalah yang sangat efektif adalah memecah masalah-masalah tersebut menjadi beberapa bagian. Kita juga dapat melakukan hal ini di Java dengan membuat method untuk mengatasi bagian tertentu dari masalah.
- Sebuah permasalahan dapat dipecah-pecah menjadi beberapa bagian kecil. Hal ini sangat baik sekali untuk membuat program yang sangat besar.

Contoh program

```
//namafile : contohmethodsederhana.java
```

```
class ContohMethodSederhana
```

```
{  
    public static void main(String[] args)  
    {  
        DemoMethod dm = new DemoMethod();  
        dm.cetakUcapan();  
    }  
}
```

Memanggil method cetakUcapan yang dipunyai class DemoMethod

```
class DemoMethod
```

```
{  
    void cetakUcapan()  
    {  
        System.out.println ("Selamat Datang");  
    }  
}
```

Contoh program


//namafile : ContohMethodParameter.java

```
class ContohMethodParameter
{
    public static void main(String[] args)
    {
        System.out.println("Contoh penggunaan method");
        DemoMethod dm = new DemoMethod();

        dm.cetakParameter("tolong cetak tulisan ini");

        String tulis = "aplikasi";
        dm.cetakParameter(tulis);
    }
}

class DemoMethod
{
    void cetakParameter(String teks)
    {
        System.out.println(teks);
    }
}
```



Contoh program dengan 2 parameter

```
//namafile : ContohMethodParameter.java
```

```
class ContohMethodParameter
```

```
{  
    public static void main(String[] args)  
    {  
        System.out.println("Contoh penggunaan method");  
  
        DemoMethod dm = new DemoMethod();  
  
        String nama = "June";  
        String umur = 19;  
        dm.cetakParameter(nama, umur );  
    }  
}
```

Urutan parameter harus sama

```
class DemoMethod
```

```
{  
    void cetakParameter(String teks, int angka)  
    {  
        System.out.println("nama: " + teks + " umur: " + angka);  
    }  
}
```

RETURN

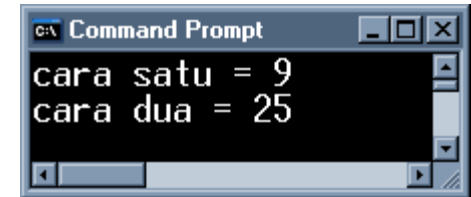
- digunakan dalam method (subroutine), Dengan menggunakan pernyataan return, maka eksekusi program akan kembali ke baris program yang memanggil method (subroutine) tersebut.



Contoh program : Return

```
// nama file : return1.java
```

```
class return1{  
    public static int hitung(int a){  
        return(a*a);  
    }  
}
```



```
public static void main (String args[]){  
    int b = hitung(3);  
    System.out.println("cara satu = " + b);  
    System.out.println("cara dua = " + hitung(5));  
}  
}
```


Contoh program : Return

```
// nama file : return2.java
class return1{
    public static String tampil(String a){
        return (a+" to Java Programming");
    }
}

public static void main (String args[]){
    String a = tampil("Welcome");
    System.out.println(a);
}
}
```

