

## MODUL 8 METHOD



### CAPAIAN PEMBELAJARAN

---

Mahasiswa dapat memahami, membuat dan menyelesaikan kasus dengan menggunakan method tanpa parameter



### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

1. TextPad
2. JDK



### DASAR TEORI

---

#### 1. Definisi Method / Function

Fungsi digunakan untuk mempermudah didalam membuat sebuah program, terutama program yang besar dan banyak melakukan beberapa hal yang sama. Fungsi memiliki ciri-ciri sebagai berikut:

1. Memiliki nama dari fungsi tersebut.
2. Memiliki tugas spesifik tertentu.
3. Memiliki sekumpulan statement atau perintah untuk melakukan tugas tersebut.
4. Mengembalikan sebuah nilai kepada fungsi lain yang memanggil atau menggunakannya (jika perlu).

#### 2. Fungsi di Dalam Bahasa Java

Dalam bahasa java terdapat 2 macam fungsi yaitu :

1. Fungsi yang mengembalikan/menghasilkan nilai (non void function)
2. Fungsi tidak mengembalikan/menghasilkan nilai (void function)

### 3. Deklarasi Function

Pada dasarnya, cara mendeklarasikan fungsi di dalam bahasa Java serupa dengan bahasa C. Berikut ini adalah cara mendeklarasikan fungsi di dalam bahasa Java:

**<Modifier><Return type><function name><parameter list>**

Dimana :

**Modifier** : sebuah *access modifier*(*public, private, protected*), yang dapat dikombinasikan dengan tipe *modifier* lain (*static, final, abstract*).

*modifier static* digunakan untuk mendefinisikan *member class*, sehingga metode/fungsi dapat diakses langsung dengan nama kelas (Untuk *member/fungsi non static* maka dipanggil melalui *instant* atau dengan membuat objek terlebih dahulu)

**Return Type** : type nilai pengembalian dari fungsi (*int,String,boolean dll*)

**Function name** : nama fungsi yang nantinya akan di panggil di dalam program

**Parameter list** : adalah **nama dan tipe variable** yang akan digunakan untuk menyimpan nilai yang dibutuhkan oleh fungsi tersebut. Jika terdapat lebih dari 1 parameter, maka parameter ditulis dipisahkan dengan koma.

#### Method tanpa parameter

Method tanpa parameter adalah method yang tidak memiliki parameter sama sekali.

Contoh :

```
public class Method1{
    public static void buatGaris(){
        System.out.println("=====");
    }
    public static void main(String args[]){
        buatGaris();
    }
}
```

## Membuat dan memanggil fungsi

Fungsi didalam java dapat diletakkan disembarang tempat selama masih ada didalam class. Bisa sebelum program utama(public static void main) atau sesudahnya, Fungsi yang sudah dibuat dapat di panggil oleh fungsi yang lain atau dari dalam program utama.

Pemanggilan fungsi dilakukan dengan memanggil **nama fungsi** diikuti **parameternya (lihat contoh pada program)**.

### a. Fungsi yang menghasilkan nilai (*Non Void Function*)

Adalah fungsi yang ketika kembali ke program utamanya disertai dengan membawa suatu nilai.

Untuk mengembalikan nilai kedalam nama fungsi menggunakan perintah **return**.

### Contoh Program 1:

```
public class FungsiJava {  
    //membuat fungsi non void  
    static int jumlah2Bilangan(int a,int b){  
        return a+b;  
    }  
    public static void main(String[] args) {  
        // TODO code application logic here  
        int a=10, b=10;  
        //pemanggilan fungsi  
        System.out.println(jumlah2Bilangan(a,b));  
    }  
}
```

Output :



```
run:  
20  
BUILD SUCCESSFUL (total time: 0 seconds)
```

### b. Fungsi yang tidak menghasilkan nilai (void Function)

Sebuah fungsi tidak harus selalu mengembalikan nilai. Tipe dari fungsi yang tidak dapat mengembalikan nilai adalah **void**. Berikut adalah contoh fungsi yang tidak mengembalikan nilai, yaitu untuk menampilkan kata "Hello" ke layar sebanyak **n** kali.

### Contoh Program 2:

```

public class FungsiVoid {
    //membuat fungsi
    static void FungsiHello(int n) {
        int i;
        for (i = 0; i < n; i++) {
            System.out.printf("Hello\n");
        }
    }

    public static void main(String args[]){
        //pemanggilan fungsi
        FungsiHello(5);
    }
}

```

**Output :**

```

run:
Hello
Hello
Hello
Hello
Hello
Hello
BUILD SUCCESSFUL (total time: 0 seconds)

```



## RAKTIK

---

### Praktik 1. Menggunakan method tanpa parameter

```

public class Method1{
    public static void cetak(){
        System.out.println("STMIK AKAKOM");
    }
    public static void main(String args[]){
        cetak();
    }
}

```

Method diatas bersifat static dan bertipe void. Method yang bersifat static bisa langsung dipanggil dengan nama methodnya saja. Sebuah method juga bisa dipanggil lebih dari satu kali.

## Praktik 2. Pemanggilan method lebih dari sekali

```
public class Method2{
    public static void cetakKalimat(){
        System.out.println("Di dalam method kalimat");
    }

    public static void main(String args[]){
        cetakKalimat();
        System.out.println("Di dalam main");
        cetakKalimat();
    }
}
```

## Praktik 3. Menggunakan method dalam perulangan

```
public class Method3{
    public static void cetak(){
        System.out.println("STMIK AKAKOM");
    }
    public static void main(String args[]){
        for (int i = 0; i<10;i++)
            cetak();
    }
}
```

## Praktik 4. Method tanpa parameter dengan nilai balik (return type)

```
public class Method4{
    public static int jumlah(){
        int a = 8, b = 10;
        return (a + b);
    }

    public static void main(String args[]){
        System.out.println("Hasil pemanggilan method jumlah");
        System.out.println(jumlah());
    }
}
```

## Praktik 5. Memanggil method dengan menciptakan objek

```
public class Method5{
    public int jumlah(){
        int a = 8, b = 10;
        return (a + b);
    }

    public static void main(String args[]){
        Method5 obyek = new Method5();
    }
}
```

```
        System.out.println("Hasil pemanggilan method jumlah");  
        System.out.println(obyek.jumlah());  
    }  
}
```



## LATIHAN

---

1. Buatlah program dengan menggunakan method untuk menampilkan data pribadi Anda seperti NIK, Nama, Jenis kelamin, Umur, Alamat ! Panggil method tersebut melalui program utama !
2. Buat program dengan method untuk menghitung nilai maksimum, minimum dari dua buah bilangan.



## TUGAS

---

Buat sebuah aplikasi sederhana, gunakan menu dan method untuk menjalankan tiap-tiap menu.



## REFERENSI

---

1. Abdul Kadir, 2012, Algoritma & Pemrograman menggunakan Java, Penerbit Andi, Yogyakarta
2. Deitel H, Deitel P, 2012, Java How to Program, 9th edition, Prentice Hall.