



MODUL 10

FUNGSI 1

KOMPETENSI

Setelah menempuh materi ini, mahasiswa mampu:

1. Menguasai tentang konsep Fungsi dengan menggunakan bahasa pemrograman Java.
2. Menguasai cara pendeklarasian Fungsi dengan menggunakan bahasa pemrograman Java.
3. Menguasai cara pemanggilan Fungsi dengan menggunakan bahasa pemrograman Java.

MATERI DASAR

FUNGSI

Fungsi (function) adalah sejumlah intruksi yang dikelompokkan menjadi satu, berdiri sendiri, yang berfungsi untuk menyelesaikan suatu pekerjaan tertentu.

Jika menggunakan fungsi maka program dapat disusun secara lebih terstruktur (lebih modular) dan lebih efektif.

Modular: sekelompok statement yang berfungsi untuk menjalankan tugas tertentu, dikelompokkan sendiri dan dipisah, dengan diberikan nama tertentu, sehingga jika diperlukan dalam suatu program untuk menjalankan tugas tersebut, maka dapat memanggil nama statement tersebut.

Efektif: Jika tugas tersebut dilakukan dalam program berulang-ulang, maka tidak perlu dituliskan berulang-ulang, tapi yang dilakukan hanya cukup memanggil fungsi tersebut.

Deklarasi Fungsi

```
static TypeDataKembalian namaFungsi() {  
    // statement  
    // statement  
}
```

Keterangan:

- **Static** : Jenis fungsi yang dibuat bersifat static, agar dapat secara langsung di panggil di fungsi main yang juga bersifat static
- **TypeDataKembalian**: tipe data dari nilai yang dikembalikan (*output*) setelah fungsi dieksekusi
- **namaFungsi()**: nama fungsi yang dibuat



Contoh :

Pembuatan fungsi:

```
static void beriSalam() {  
    System.out.println("Halo! Selamat Pagi");  
}
```

Pemanggilan fungsi:

```
public static void main(String[] args) {  
    beriSalam();  
}
```

Fungsi dengan Parameter

- Parameter adalah variabel yang menampung nilai untuk diproses di dalam fungsi. Parameter berperan sebagai *input* untuk fungsi.
- Deklarasi:

```
static TipeDataKembalian namaFungsi(TipeData namaParameter,  
    TipeData namaParameterLain) {  
    // statement  
}
```

- **Parameter** : sebagai tempat utk data masukan yang akan diolah dalam fungsi. Banyaknya parameter menyesuaikan kebutuhan. Setiap parameter terdiri dari type data dan nama parameter (misal: int a, float b), sama persis seperti deklarasi variabel.
- Parameter ditulis di antara *parenthesis* (...) setelah nama fungsi.
- Bila terdapat lebih dari satu parameter, maka dipisah dengan tanda koma dan masing-masing parameter harus dideskripsikan tipe datanya.

Parameter Fungsi

- Fungsi memerlukan parameter ketika fungsi tersebut membutuhkan data yang asalnya dari luar fungsi untuk di olah dalam fungsi.
- Fungsi boleh tidak memiliki sama sekali parameter.
- Jumlah parameter fungsi yang bisa dimiliki fungsi menyesuaikan kebutuhan, dan tidak ada batasan maksimalnya.
- Pada saat deklarasi fungsi, penulisan parameter adalah dengan cara:
tipe_data nama _parameter.



Pembuatan fungsi dengan parameter:

```
static void beriUcapan(String ucapan) {  
    System.out.println(ucapan);  
}
```

Pemanggilan fungsi dan memberi nilai parameter:

```
String halo = "Hallo!";  
beriUcapan(halo);  
beriUcapan("Selamat datang di pemrograman Java");
```

Fungsi yang mengembalikan Nilai

- Sebuah fungsi yang dapat mengembalikan nilai *output* sehingga bisa diolah pada proses berikutnya.
- Pengembalian nilai pada fungsi menggunakan *keyword* return.
- Fungsi yang memiliki tipe data fungsi selain void yang memerlukan return. Fungsi void tidak memerlukan return.
- Nilai yang di-return-kan dari suatu fungsi harus sesuai dengan tipe data fungsi. Misalnya jika tipe data fungsi int, maka nilai yang di-return-kan harus nilai itu.
- Sebuah fungsi yang dapat mengembalikan nilai *output* sehingga bisa diolah pada proses berikutnya.
- Pengembalian nilai pada fungsi menggunakan *keyword* return.
- Deklarasi fungsi:

```
static TypeDataKembalian namaFungsi (TypeData namaParameter){  
    // statement  
    return variabelOutput;  
}
```

Pembuatan Fungsi dengan parameter dan return value:

```
static int luasPersegi(int sisi){  
    int luas = sisi * sisi;  
    return luas;  
}
```

Pemanggilan Fungsi dan memberi nilai parameter:

```
System.out.println("Luas Persegi dengan sisi 5 = " + luasPersegi(5));  
int luasan = luasPersegi(6);
```



SCOPE OF VARIABLE

- **Variabel Lokal:** variabel yang dideklarasikan dalam suatu fungsi, dan hanya bisa diakses atau dikenali dari dalam fungsi itu sendiri.
- **Variabel Global:** Variabel yang dideklarasikan di luar blok fungsi, dan bisa diakses atau dikenali dari fungsi manapun.
- Variabel Global pada java di beri awalan static agar variabel tersebut bisa dipanggil langsung.

```
public class Fungsil {

    static int a = 10, b = 5;
    static double c;

    static int Kali() {
        int hasilKali = a * b;
        return hasilKali;
    }

    static void Tambah() {
        int hasilTambah = a + b;
        System.out.println("Hasil Tambah adalah " + hasilTambah);
    }

    public static void main(String[] args) {
        System.out.println("Hasil Kali adalah " + Kali());
        Tambah();
    }

}
```

Sebuah Fungsi dapat meng-CALL Fungsi Lain

```
public class FungsiCall {
    public static void main(String[] args) {
        int hasil=Hitung(5,2);
        System.out.println("Hasil Akhirnya adalah "+hasil);
    }

    static int Tambah(int x, int y){
        int Z=x+y;
        return Z;
    }

    static int Hitung (int c, int d){
        int E;
        c*=2;
        d*=2;
        E=Tambah(c,d);
        return E;
    }
}

run:
Hasil Akhirnya adalah 14
BUILD SUCCESSFUL (total time: 2 seconds)
```



Java Varargs (Variable Arguments)

- Varargs dipakai dengan cara menempatkan parameter-parameter dalam sebuah array dan array tsb yang akan menjadi parameter dari fungsi.
- Apabila tidak diketahui berapa jumlah dari parameter suatu fungsi dengan pasti, kita bisa menggunakan Variable Length Argument (Varargs).
- Deklarasi:

```
accessModifier methodName(datatype... arg) {  
    // method body  
}
```

Contoh:

```
public class ContohVarargs1 {  
    static void tampilIsi(int ...a){  
        System.out.println("Jumlah parameter ada "+ a.length);  
        System.out.println("isinya : ");  
  
        for(int i=0; i<a.length; i++){  
            System.out.println("Parameter ke-"+i+" : "+a[i]);  
        }  
  
        System.out.println();  
    }  
  
    public static void main(String args[]){  
        tampilIsi(10); // hanya ada satu parameter  
        tampilIsi(4,5,3); // ada 3 parameter  
    }  
}
```

run:

```
Jumlah parameter ada 1  
isinya :  
Parameter ke-0 : 10
```

```
Jumlah parameter ada 3  
isinya :  
Parameter ke-0 : 4  
Parameter ke-1 : 5  
Parameter ke-2 : 3
```

Percobaan 1

1. Buatlah sebuah project dan beri nama Greeting.java
2. Salin source code berikut:



```
public class greeting {  
    static void beriSalam(){  
        System.out.println("Halo! Selamat Pagi");  
    }  
    public static void main(String[] args){  
        beriSalam();  
    }  
}
```

3. Hasil running:

```
run:  
Halo! Selamat Pagi  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan2

1. Masih menggunakan projek pada percobaan 1. Edit program tersebut menjadi:

```
public class Greeting {  
    static void beriSalam(){  
        System.out.println("Halo! Selamat Pagi");  
    }  
    static void beriUcapan(String ucapan){  
        System.out.println(ucapan);  
    }  
    public static void main(String[] args){  
        beriSalam();  
        String salam = "Selamat datang di pemrograman Java";  
        beriUcapan(salam);  
    }  
}
```

4. Hasil running:

```
run:  
Halo! Selamat Pagi  
Selamat datang di pemrograman Java  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan 3

1. Buatlah sebuah project dan beri nama Hitung.java
2. Salin source code berikut:



```
public class Hitung {  
    static int kali(int angka1, int angka2){  
        int hasil = (angka1 + 10) % (angka2 + 19);  
        return hasil;  
    }  
    static int kurang(int angka1, int angka2){  
        angka1 = angka1 + 7;  
        angka2 = angka2 + 4;  
        int hasil = kali(angka1, angka2);  
        return hasil;  
    }  
    public static void main(String[] args){  
        Scanner input = new Scanner(System.in);  
        System.out.print("Masukkan nilai 1: ");  
        int nilai1 = input.nextInt();  
        System.out.print("Masukkan nilai 2: ");  
        int nilai2 = input.nextInt();  
        int hasil = kurang(nilai1, nilai2);  
        System.out.println("Hasil akhir adalah "+hasil);  
    }  
}
```

3. Hasil running:

```
run:  
Masukkan nilai 1: 5  
Masukkan nilai 2: 4  
Hasil akhir adalah 22  
BUILD SUCCESSFUL (total time: 6 seconds)
```