FUNGSI 1

TIM PENGAJAR
DSAR PEMROGRAMAN
2019

KOMPTENSI

Setelah menempuh materi ini, mahasiswa hendaknya mampu:

1. Menguasai tentang konsep Fungsi
2. Menguasai cara pendeklarasian Fungsi
3. Menguasai cara pemanggilan Fungsi

DEFINISI 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.

DEFINISI FUNGSI

- **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 ...(1)

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

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

Fungsi dengan Parameter ...(2)

- **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 paramter 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.

Fungsi dengan Parameter ...(3)

- **Contoh:**

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...(1)

- 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.

Fungsi yang mengembalikan Nilai...(1)

- 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 (TipeData namaParameter){
// statement
return variabelOutput;
}
```

Fungsi yang mengembalikan Nilai...(2)

- **Contoh**

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... (1)

- Variabel Lokal: vriabel 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.

SCOPE OF VARIABLE... (2)

```
public class Fungsi1 {

    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();

    }

}
```

Variabel Global

Variabel Lokal

Variabel Lokal

Method Pass by Value Vs Pass by Reference

- **Pass by Value** mengirimkan parameter berdasarkan nilai variabel asalnya yang akan dihubungkan terhadap paramater fungsi pemanggil.
- **Pass by Reference** mengirimkan parameter berdasarkan alamat dari nilai tertentu, maka dari itu bila ada nilai yang dirubah dari alamat asalnya maka akan terjadi perubahan juga terhadap nilai parameter yang di panggil.

Fungsi Pass by Value Vs Pass by Reference

```
public class PassbyValue {
    static void UbahNilai(int j){
        j=33;
    }

    public static void main(String[] args) {
        int i=10;
        System.out.println(i);
        UbahNilai(i);
        System.out.println(i);
    }
}
```

```
run:
10
10
```

```
public class PassbyRef {
    static void UbahArray (int[] arr){
        for (int i=0; i<arr.length; i++){
            arr[i]=i+50;
        }
    }

    public static void main(String[] args) {
        int[] umur = {10, 11, 12};

        for (int i = 0; i < umur.length; i++) {
            System.out.println(umur[i]);
        }

        UbahArray(umur);
        for (int i=0; i<umur.length; i++){
            System.out.println(umur[i]);
        }
    }

}
```

```
run:
10
11
12
50
51
52
```

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)
```

Dua Fungsi dapat saling mengCALL

```
public class FungsiCallFungsi {
    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;
        while (Z<50){
            x+=2;
            y+=2;
            Z=Hitung(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 80
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 1

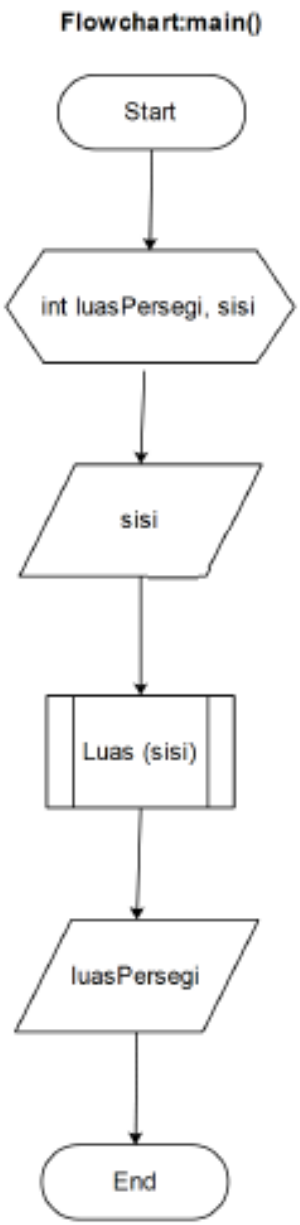
```
public class ContohVarargs1 {
    static void tampilIsi(int ...a){
        System.out.println("Jumlah parameter ada "+ a.length);
        System.out.println("isinya : ");

        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
    }

}
```

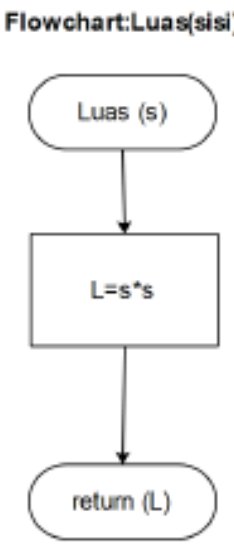
```
run:
Jumlah parameter ada 1
isinya :
Parameter ke-0 : 10

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

Contoh Flowchart Fungsi

- Buatlah flowchart untuk menghitung Luas sebuah persegi. Untuk menghitung Luas gunakan fungsi.

Contoh Flowchart Fungsi





Soal

1. Buatlah flowchart untuk menghitung luas permukaan dan volume balok. (luas permukaan dan volume balok menggunakan fungsi)!
2. Buatlah flowchart untuk menampilkan deret bilangan genap antara 1-100 . (deret yang ditampilkan menggunakan fungsi)
3. Buatlah flowchart untuk menginputkan nama dan nilai mahasiswa (menggunakan array) kemudian hitunglah nilai rata-rata kelas. Fungsi terdiri dari

a. Fungsi menampilkan data mahasiswa
b. Fungsi menghitung nilai rata-rata kelas