6. Method Java

By : Mahendar Dwi Payana, S.ST., M.T

Prasyarat :

1. Pengenalan Algoritma & Flowchart
2. Pengenalan Java
3. Variabel dan Implementasi
4. Tipe Data dan Implementasi

Referensi :

1. Buku "Pemrograman Bororientasi Objek. Teori dan Implementasi Java"
2. Pengarang : Raden Budiarto
3. Tahun : 2018
4. Jenis : E-Book - ebook play.books.com

6.1 Pengenalan Method

Method adalah sebuah blok kode yang berisi perintah-perintah yang dapat digunakan kembali. Method dapat digunakan untuk mengelompokkan perintah-perintah yang sering digunakan. ***Method*** juga dapat digunakan untuk mengelompokkan perintah-perintah yang berhubungan dengan suatu objek.

Pada Dasarnya ***Method*** adalah bagian kode yang akan dipanggil oleh program utama. ***Method*** dapat dipanggil berulang-ulang sesuai dengan kebutuhan. ***Method*** juga dapat menerima input dan menghasilkan output. ***Method*** dapat dipanggil dari ***Method*** lainnya. Hampir semua bahasa pemrograman mengimplementasinya ***Method***. Salah satu keuntungan menggunakan ***method*** selain diatas adalah keuntungan dalam proses *maintanance* program.

Selain itu method terdapat dua jenis yaitu ***build-in method*** dan ***user-defined method***. *Build-in method* adalah method yang sudah disediakan oleh bahasa pemrograman. Sedangkan *user-defined method* adalah method yang dibuat oleh user sesuai dengan kebutuhan.

6.2 Implementasi Method

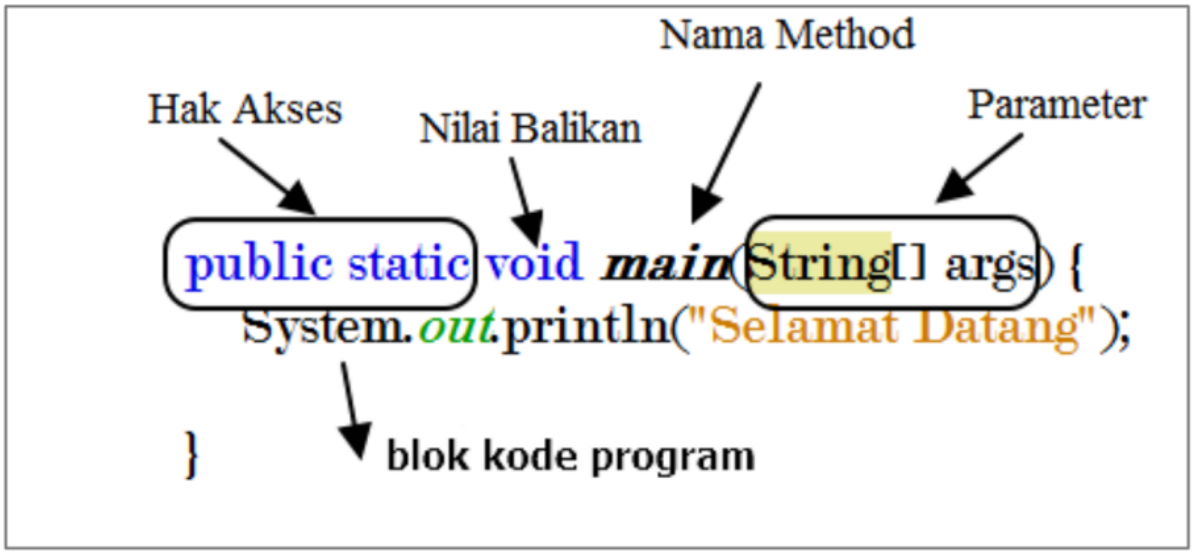
Untuk menggunakan ***Metode*** kita harus mengikut format penulisan yang sudah ditentukan oleh **JAVA**. Format penulisan ***Method*** pada **JAVA** adalah sebagai berikut.

```
modifier returnType nameOfMethod (Parameter List) {
   // method body
}
```

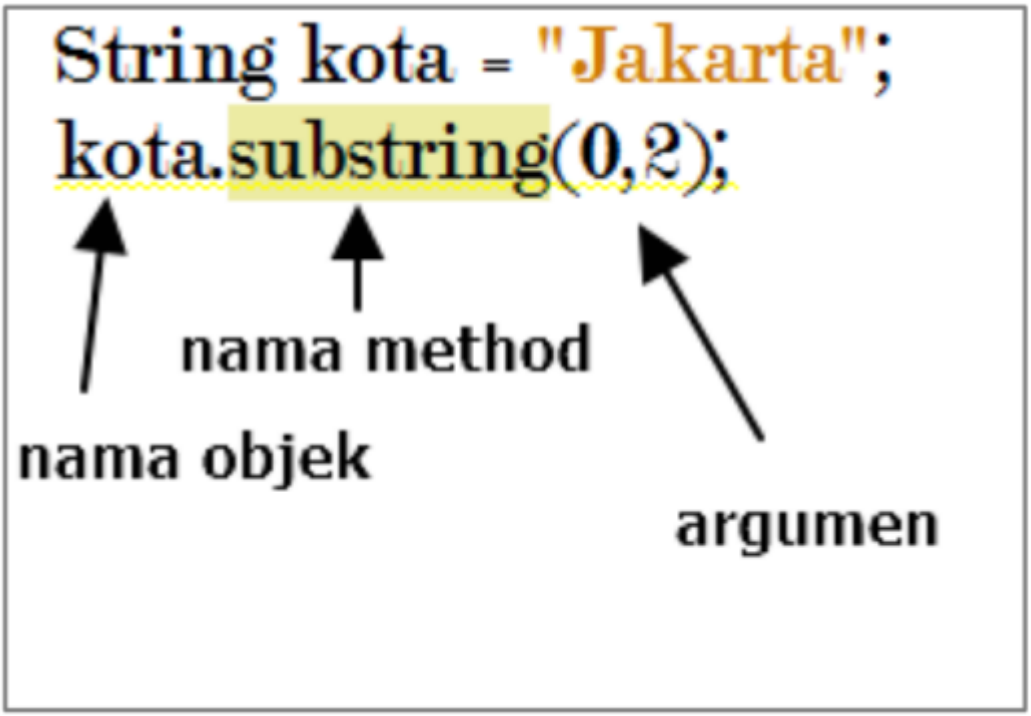
Penjelasan dari format di atas adalah sebagai berikut.

1. **Modifier** : Sebuah ***Method*** dapat memiliki akses yang berbeda: `public`, `private`, `protected`, `default`. Dan seterusnya.
2. **Return Type** : Sebuah ***Method*** dapat mengembalikan nilai. Nilai yang dikembalikan harus memiliki tipe data yang sesuai dengan tipe data yang telah ditentukan. Misalnya kita ingin mengembalikan nilai `int` maka kita harus menggunakan `int` pada ***Method***.
3. **Method Name** : Nama dari ***Method*** yang akan digunakan.
4. **Parameter List** : Daftar parameter yang akan digunakan pada ***Method***. Parameter adalah nilai yang akan digunakan pada ***Method***.
5. **Method Body** : Bagian dari ***Method*** yang berisi perintah-perintah yang akan dijalankan.

Berikut adalah ilustrasi dari format penulisan ***Method*** pada **JAVA**.



Setelah ***Method*** dibuat maka format pemanggilan adalah seperti pada Gambar dibawah ini.



6.3 Contoh Program Method

Mari kita coba membuat sebuah contoh penggunaan ***Method*** pada **JAVA**. Pada contoh ini kita akan membuat sebuah ***Method*** yang akan menghitung luas persegi panjang. Berikut adalah contoh programnya.

```
public class Method {
    public static void main(String[] args) {
        int panjang = 10;
        int lebar = 5;
        int luas = hitungLuas(panjang, lebar); // Memanggil Method
        System.out.println("Luas Persegi Panjang = " + luas);
    }
    public static int hitungLuas(int panjang, int lebar) {
        int luas = panjang * lebar;
        return luas;
    }
}
```

Dari contoh diatas kita dapat melihat *method* `hitungLuas()` yang kemudian dipanggil kembali pada variabel `int luas = hitungLuas(panjang, lebar);`

6.4 Jenis Method

Pada **JAVA** terdapat 2 jenis ***Method*** berdasarkan nilai balik atau *return value* yaitu ***Method*** yang mengembalikan nilai dan ***Method*** yang tidak mengembalikan nilai. ***Method*** yang mengembalikan nilai disebut dengan ***Function***. Sedangkan ***Method*** yang tidak mengembalikan nilai disebut dengan ***Procedure***. Berikut adalah contoh ***Function*** dan ***Procedure*** pada **JAVA**.

```
public class Method {
    public static void main(String[] args) {
        // Memanggil Method
        hitungLuas(10, 5); // Memanggil Procedure
        int luas = hitungLuas(10, 5); // Memanggil Function
        System.out.println("Luas Persegi Panjang = " + luas);
    }
    public static void hitungLuas(int panjang, int lebar) {
        int luas = panjang * lebar;
        System.out.println("Luas Persegi Panjang = " + luas);
    }
    public static int hitungLuas(int panjang, int lebar) {
        int luas = panjang * lebar;
        return luas;
    }
}
```

Pada contoh diatas kita dapat melihat ***Method*** `hitungLuas()` yang kemudian dipanggil kembali pada variabel `int luas = hitungLuas(panjang, lebar);` dan `hitungLuas(10, 5);`

Pada `hitungLuas(10, 5);` kita dapat melihat bahwa ***Method*** tersebut tidak mengembalikan nilai. Sedangkan pada `int luas = hitungLuas(panjang, lebar);` kita dapat melihat bahwa ***Method*** tersebut mengembalikan nilai.

Latihan Pagi

1. Latihan 1

```
public class Main {
    public static void main(String[] args) {
        int luas = PersegiPanjang(5,3);
        int luaslain = PersegiPanjang(100,60);
        System.out.println("Luas dari yang anda input adalah : "+
luas);
        System.out.println("Luas dari persegi panjang lain adalah
: " + luaslain);
        System.out.println(namaMahasiswa("Budiawan"));
    }

    //method yang kita buat (user-defind method)
    public static int PersegiPanjang(int panjang, int lebar){
        int luas = panjang * lebar;
        return luas;
    }

    public static String namaMahasiswa(String nama){
        return nama;
    }
}
```

2. Latihan 2

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int panjang = 10;
        int lebar;

        System.out.print("Masukan nilai panjang : ");

        Scanner input = new Scanner(System.in);
```

```
        panjang = input.nextInt();

        System.out.print("Masukan nilai lebar : ");
        lebar = input.nextInt();

        int luas = PersegiPanjang(panjang,lebar);
        System.out.println("Luas dari yang anda input adalah : "+
luas);
        System.out.println(namaMahasiswa("Budiawan"));
        System.out.println("Luas linkaran adalah : " +
LuasLingkaran(13));
    }

    //method yang kita buat (user-defind method)
    public static int PersegiPanjang(int panjang, int lebar){
        int luas = panjang * lebar;
        return luas;
    }

    public static String namaMahasiswa(String nama){
        return nama;
    }

    public static double LuasLingkaran(int jariJari){
        double luas = 2 * Math.PI * jariJari;
        return luas;
    }
}
```

3. Latihan 3 (Perkalian 2 buah volume kubus)

```
public class Main {
    public static void main(String[] args) {
        int hasil = PerkalianKubus(jumlahDuaVolumeKubus(3,2),
jumlahDuaVolumeKubus(4,2));

        System.out.println("hasil dari perkalian dua kubus adalah
" + hasil);
    }

    public static int jumlahDuaVolumeKubus(int kubusa, int kususb)
{
        int volumeA = kubusa * kubusa * kubusa;
        int volumeB = kususb * kususb * kususb;
        int total = volumeA + volumeB;
        return total;
    }
```

```
    public static int PerkalianKubus(int c, int d){
        return c * d;
    }

}
```

<< Back

Next >>