

MODUL 3 METHOD



CAPAIAN PEMBELAJARAN

Dapat membuat dan menggunakan berbagai method



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Textpad
2. JDK



DASAR TEORI

Method

Sebuah method menjelaskan behaviour dari sebuah object. Method juga dikenal sebagai fungsi atau prosedur. Pendeklarasian method biasa dituliskan seperti berikut ini :

```
<modifier> <returnType> <name>(<parameter>*) {  
    <statement>*  
}
```

dimana,

<modifier> dapat menggunakan beberapa *modifier* yang berbeda

<returnType> dapat berupa seluruh tipe data, termasuk *void*

<name> *identifier* atas *class*

<parameter> ::= <type_parameter> <nama_parameter>[,]

Method Tanpa Return Value

Jenis method ini ditandai dengan *return type* yang berupa *void* dan pada bagian *statement* tidak terdapat keyword **return**. Pada pemrograman berorientasi obyek method jenis ini digunakan untuk membuat method *mutator*. Nama untuk method *mutator* sebaiknya diawali dengan kata set, misalnya: setName(), setAddress().

Method Dengan Return Value

Jenis method ini ditandai dengan *return type* selain *void* dan pada bagian *statement* terdapat keyword **return**. Pada pemrograman berorientasi obyek method jenis ini digunakan untuk membuat method *acessor*, dimana method *acessor* fungsinya adalah untuk membaca/mendapatkan nilai suatu atribut. Nama untuk method *acessor* sebaiknya diawali dengan kata get, misalnya: getName(), getAddress().

Method Overloading

Bahasa java mendukung method *overloading*, java dapat membedakan beberapa *method* dengan nama yang sama di dalam sebuah kelas namun parameternya berbeda. Hal ini sangat menguntungkan karena memudahkan kita dalam mengingat nama method, bayangkan bila program pada class Gambar harus diberi nama drawInteger(int i), drawString(String s), drawDouble(double d). Method *overloading* dibedakan oleh jumlah dan jenis tipe data parameternya.

Contoh method overloading :

```
public class Gambar{
    public void draw(int i){
        .....
    }
    public void draw(String s){
        .....
    }
    public void draw(double d){
```

```

.....
}

public void draw(int i, double d){

.....
}
}

```



PRAKTIK

1. Praktik 1 (membuat method mutator dan method aksesori)

```

public class Mahasiswa{
    private String nim,nama;

    //method mutator
    public void setNim(String nim) {
        this.nim=nim;
    }
    public void setName(String nama) {
        this.nama=nama;
    }

    //method aksesori
    public String getNim() {
        return nim;
    }
    public String getName() {
        return nama;
    }

    public static void main(String args[]){
        Mahasiswa obj=new Mahasiswa();
        obj.setNim("175410001");
        obj.setName("Azkiya");

        System.out.println("==Data==");
        System.out.println("Nim   : "+ obj.getNim());
        System.out.println("Nama : "+ obj.getName());
    }
}

```

```
}
```

2. Praktik 2 (Memuat method overloading)

```
class Hitung{
    private int a,b,c,d;

    void tambah(int a,int b){
        System.out.println(a+b);
    }
    void tambah(int a,int b,int c){
        System.out.println(a+b+c);
    }
    void tambah(int a,int b,int c, int d){
        System.out.println(a+b+c+d);
    }

    public static void main(String args[]){
        Hitung obj=new Hitung();
        obj.tambah(10,10,10);
        obj.tambah(20,20);
        obj.tambah(20,5,10,20);
    }
}
```

3. Praktik membuat method overloading untuk dengan perbedaan tipe parameter

```
class Calculation2{
    private int a,b;
    private double c,d;

    void sum(int a,int b) {
        System.out.println(a+b);
    }
    void sum(double c,double d) {
        System.out.println(c+d);
    }

    public static void main(String args[]){
        Calculation2 obj=new Calculation2();
        obj.sum(10.5,10.5);
        obj.sum(20,20);
    }
}
```



LATIHAN

1. Modifikasi praktik 3 dengan menambahkan method sum dengan tipe parameter yang berbeda selanjutnya panggil melali main program !
2. Modifikasi praktik 1 dengan menambahkan method TampilkanData yang digunakan untuk menampilkan data nim dan nama !
3. Buatlah kelas lengkap dengan atribut dan oprasi untuk menentukan luas seitiga !



TUGAS

1. Buatlah program lengkap dengan atribut dan oprasi yang digunakan untuk meghitung nilai pangkat dari suatu bilangan !
2. Soal dar dosen pengampu



REFERENSI

Wagito, Sumiyatun, Fx. Henry Nugroho. 2015. Modul Praktikum Pemrograman Berorientasi Objek. STMIK AKAKOM. Yogyakarta