PERTEMUAN KE - 3
METHOD

A. TUJUAN

1. Dapat memahami kegunaan dari method
2. Dapat membuat method tanpa return value
3. Dapat mebuat method dengan return value
4. Dapat membuat method Overloading

B. TEORI SINGKAT

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> ::= <tipe_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: setNama(), setAlamat().

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: getNama(), getAlamat().

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 drawInterger(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){
 .....
 }
}
```

C. PRAKTIK

Praktik 1 : membuat method mutator dan acessor

```
public class Calculation0{
 private int a,b;

 //method mutator
 public void setA(int a) { this.a=a; }
 public void setB(int b) { this.b=b; }
 //method acessor
 public int getA()  { return a;}
 public int getB()  { return b;}

 public static void main(String args[]){
  Calculation0 obj=new Calculation0();
  obj.setA(10);
  obj.setB(20);
```

```
  System.out.println("==Data==");
  System.out.println("Nilai pertama"+ obj.getA());
  System.out.println("Nilai kedua"+ obj.getB());
  }
}
```

Praktik 2 : membuat method overloading

.

```
class Calculation{
 private int a,b,c;

//method overloading dengan perbedaan jumlah parameter
 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);
 }

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

Praktik 3 membuat method overloading

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

//method overloading dengan perbedaan tipe data parameter
  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);
 }
}
```

D. LATIHAN

- *Latihan diberikan oleh dosen pengampu pada saat praktikum.*
- *Dikerjakan di laboratorium pada jam praktikum*

E. TUGAS

- *Tugas diberikan oleh dosen pengampu pada akhir praktikum.*
- *Dikerjakan di rumah dan dilampirkan pada laporan.*