

Modul 2

Class dan Object

Tujuan

- Memahami konsep class & object.
- Dapat mengimplementasikan konsep class & object pada kasus sederhana.

2.1. Class dan Object

Class adalah cetak biru dari suatu objek riil di dunia. Penerapan objek riil direpresentasikan dalam bentuk class. Sebuah class mempunyai anggota yang terdiri dari atribut dan method. Atribut adalah semua field identitas yang kita berikan pada suatu class.

Contoh :

Class segiempat → object

Data : panjang, lebar

Object adalah gambaran dari entity, baik dunia nyata atau konsep dengan batasan-batasan dan pengertian yang tepat. Objek-objek ini kemudian juga dapat berupa gabungan dari beberapa objek yang lebih kecil.

Sebagai contoh, objek kubus mempunyai ciri-ciri khusus yang dimiliki oleh kubus. Misalnya mempunyai panjang, lebar, dan tinggi. Ciri-ciri khusus kubus tersebut harus menjadi identitas utama sebagai objek kubus.

Keyword “new” digunakan untuk menambahkan instansiasi/membuat sebuah object baru.

Contoh :

kubus hit

hit = new kubus()

2.2. Metode

Metode biasa kita kenal sebagai *function* dan *procedure*. Dikatakan fungsi bila method tersebut melakukan suatu proses dan mengembalikan suatu nilai (*return value*), dan dikatakan prosedur bila method tersebut hanya melakukan suatu proses dan tidak

mengembalikan nilai (*void*). Dalam OOP, method digunakan untuk memodularisasi program melalui pemisahan tugas dalam suatu class. Pemanggilan metode menspesifikasikan nama metode dan menyediakan informasi (parameter) yang diperlukan untuk melaksanakan tugasnya.

a. Metode tanpa nilai balik

Metode tanpa nilai balik harus dideklarasikan menggunakan kata kunci *void*, yang artinya tidak memiliki tipe kembalian.

Bentuk umum :

TipeAkses *void* NamaMetode (parameter)

```
{  
    //BadanMetode  
}
```

b. Metode dengan nilai balik

Bentuk umum :

TipeAkses TipeKembalian NamaMetode (parameter)

```
{  
    //BadanMetode  
    return nilai;  
}
```

Access Modifier ada 3 jenis yang umum:

1. *private*, akses terbatas pada class itu sendiri
2. *protected*, akses terbatas pada class itu sendiri atau dapat diturunkan ke class turunannya dan berganti akses ke *private*.
3. *public*, akses tidak terbatas pada dan dari luar bebas mengakses ke class tersebut.

2.3. Konstruktor

Konstruktor/Konstruktor adalah fungsi khusus yang sama dengan nama class dan tidak mempunyai tipe kembalian, tidak juga *void*. Kegunaan utama adalah untuk mendefinisikan suatu objek dari class. Secara default, konstruktor tidak ber-parameter.

Contoh :

a. Penggunaan Class dan Object

```
class Contoh2
{
    public double panjang;
    public double lebar;
}
//membuat kelas utama
class demokelas
{
    public static void main (String[ ] arg)
    {
        //deklarsi variabel lokal
        double luas;
        //instansiasi kelas segiempat
        Contoh2 se = new Contoh2();
        //inisialisasi nilai objek
        se.panjang = 8;
        se.lebar = 6;
        //hitung luas
        luas = se.panjang * se.lebar;
        //tampil luas
        System.out.println("luas segiempat = " +luas);
    }
}
```

b. Penggunaan Metode

```
class Contoh3
{
    public double panjang;
    public double lebar;
    public void cetakluas1()//definisi metode bentuk void
    {
        double luas; //variabel lokal didalam metode
        luas = panjang * lebar;
        System.out.println("luas segiempat = "+luas);
    }
    public double hitungluas()//definisi metode bentuk non
void
    {
        double luas; //variabel lokal didalam metode
        luas = panjang * lebar;
        return luas;
    }
}

//membuat kelas utama
class demokelas1
{
    public static void main (String[ ] arg)
    {
        double luas; //deklarsi variabel lokal
        Contoh3 se = new Contoh3 ();//instansiasi kelas
segiempat
        //inisialisasi nilai objek
        se.panjang = 8;
        se.lebar = 6;
        //panggil metode
        System.out.println("hasil luas dengan melakukakn
pemanggilan secara void");
        se.cetakluas1();
        System.out.println("hasil luas dengan melakukakn
pemanggilan secara non void");
        luas = se.hitungluas();
        System.out.println("Luas segiempat = " +luas);
    }
}
```

c. Penggunaan Konstruktor

```
import java.util.Scanner;
class Contoh4
{
    public double panjang;
    public double lebar;
    //defiisi konstruktor
    public Contoh4 ()
    {
        panjang = 8;
        lebar = 5;
    }
    //definisi metode
    public void isidata(double p, double l)
    {
        panjang = p;
        lebar = l;
    }
    public double hitungluas()
    {
        return panjang * lebar;
    }
    public void cetakluas(String s)
    {
        System.out.println(s + hitungluas());
    }
}
class demokelas2
{
    public static void main (String[ ] arg)
    {
        double a, b;
        Contoh4 sel;
        //buat objek pertama
        sel = new Contoh4();
        sel.cetakluas("luas pertama ");
        //ubah data
        sel.isidata(5, 10);
        sel.cetakluas("luas kedua : ");
        Scanner input = new Scanner (System.in);
        System.out.print("panjang : ");
        a = input.nextDouble();
        System.out.print("lebar : ");
        b = input.nextDouble();
        sel.isidata(a, b);
        sel.cetakluas("luas ketiga : ");
    }
}
```

2.4. Class Diagram

Pada pembuatan perangkat lunak berbasis Object Oriented, terdapat mekanisme untuk merancang perangkat lunak tersebut menggunakan UML (Unified Modeling Language). Salah satu bentuk perancangan dari UML adalah **Class Diagram**. Contoh class diagram diperlihatkan pada gambar di bawah ini.



Berdasarkan gambar di atas, terlihat bahwa class diagram merupakan sebuah bujur sangkar dengan 3 tingkatan, dengan konten di masing-masing tingkat sebagai berikut:

- Tingkat pertama: nama Class
- Tingkat kedua: nama atribut beserta tipe datanya
- Tingkat ketiga: nama method beserta keterangannya

a. Penulisan Atribut

Atribut dituliskan dengan aturan berikut:

visibility name: *type-expression* = *initial-value* { *property-string* }

Keterangan:

- Visibility: private (-), protected(#), dan public (+)
- Name: nama atribut
- Type-expression: tipe data dari atribut
- Initial-value: nilai awal atribut (jika ada)
- Property-string: aturan perancangan lainnya (jika ada)

Jika sebuah atribut bersifat final (merupakan konstanta), maka pada class diagram ditulis dengan huruf kapital. Jika atribut bersifat static, berikan *underline*.

b. Penulisan Method

Method dituliskan dengan aturan berikut:

visibility *name*(*parameter-list*) : *return-type-expression*{ *property-string*}

Keterangan:

- Visibility: private (-), protected(#), dan public (+)
- Name: nama method
- Parameter-list: penulisan parameter yang dilewatkan oleh method. Terdapat aturan lain untuk ini.
- Return -type-expression: tipe data yang dikembalikan oleh method
- Property-string: aturan perancangan lainnya (jika ada)

Aturan penulisan parameter-list:

kind *name*: *type-expression*= *default-value*

Keterangan:

- Kind: jenis parameter dari method. Terdapat 3 jenis parameter: in, out, inout. Khusus bahasa pemrograman Java, hanya ada tipe "in"/
- Name: nama parameter yang dilewatkan
- Type-expression: tipe data parameter yang dilewatkan
- Default-value: nilai awal parameter (jika ada).

Class diagram "Mahasiswa" di atas memiliki 2 konstruktor. Dengan kata lain, overloading constructor terjadi pada class "Mahasiswa". Sebuah class dapat memiliki lebih dari 1 konstruktor. Bahkan, sebuah class dapat tidak dituliskan konstruktornya. Secara default konstruktor kosong akan terbentuk jika tidak dituliskan konstruktor secara eksplisit pada sebuah class.

2.5. Tugas Praktikum

1. Conan merupakan anak SD penggemar novel detektif. Ia ingin membuat sebuah program untuk mendata setiap novel detektif yang ia miliki. Setiap novel memiliki judul, nama pengarang dan tahun terbit. Tapi Conan ingin mengetahui isi dari setiap novel, sehingga ia tahu deskripsi novel tersebut. Conan juga menginginkan informasi harga beli tercantum di program. Dikarenakan sewaktu-waktu ia ingin menjual kembali novelnya, terdapat mekanisme untuk menghitung harga jual novel. Rumus yang ia gunakan adalah “harga jual = harga beli – 20% * harga beli”.
 - a. Tentukanlah atribut dan method terlibat, gambarkan class diagramnya.
 - b. Buat programnya dengan membuat 3 objek berdasarkan class tersebut (data bebas), dan tampilkan informasi dari setiap buku tersebut. Lalu, tampilkan harga total beli buku serta harga total buku jika dijual.