

PRAKTIKUM 2

PEMBUATAN CLASS DAN OBJEK

A. TUJUAN PEMBELAJARAN

1. Memahami mengenai konsep Class dan Objek
2. Mampu mengubah konsep Objek di dunia nyata menjadi objek dalam pemrograman.
3. Memahami cara membuat class.
4. Memahami cara mengakses data member dan method dari class.

B. DASAR TEORI

Class

Untuk mempelajari pemrograman Java terdapat beberapa hal yang harus dipelajari. Diantaranya adalah konsep class (kelas) dan obyek. Hal yang penting diketahui sehubungan dengan kelas dalam Java adalah mendeklarasikan sebuah kelas, membuat sebuah obyek, mendeklarasikan suatu atribut, mendeklarasikan suatu method dan mengakses anggota suatu obyek.

Class adalah template/blueprint/cetakan untuk membuat obyek. Class harus berasal dari entitas atau konsep dunia nyata. Obyek adalah hasil instansiasi dari sebuah kelas. Atribut atau variabel adalah identitas unik yang dimiliki dan melekat pada sebuah kelas. Method adalah fungsi atau tingkah laku atau kemampuan yang dimiliki oleh sebuah kelas untuk pengaksesan atribut atau tugas tertentu. Contoh kelas adalah : Mahasiswa, Vehicle, Tree, Fruit, Mobil, dan lain-lain. Dalam koding Java, untuk mendeklarasikan kelas Mahasiswa adalah sebagai berikut:

```
class Mahasiswa(){  
    // variabel  
    // method  
}
```

Class merupakan sebuah prototipe yang mendefinisikan variabel-variabel serta method-method secara umum. Dalam sebuah aplikasi, bisa terdiri dari hanya satu kelas ataupun beberapa kelas yang memiliki karakter atau kegunaan sendiri-sendiri.

Sebuah file sumber di Java disimpan dengan ekstensi .java. Sebuah file sumber dapat terdiri dari beberapa kelas, namun biasanya pada bahasa pemrograman Java sebuah file hanya terdiri dari satu kelas. Nama file sumber harus disimpan dengan nama yang sama dengan deklarasi nama kelas dalam program, misalnya Mahasiswa.java. Setelah dilakukan kompilasi maka pada Java akan dihasilkan sebuah file ".class" yang berisi bytecode dari setiap kelas. Jika sebuah file terdiri dari dua kelas maka setelah dikompilasi akan dihasilkan dua buah file ".class" yang nantinya akan dibaca oleh interpreter Java saat program dieksekusi.

Jadi komponen penyusun sebuah kelas ada dua bagian yaitu:

1. Variabel/properti/atribut

Variabel adalah karakteristik berupa data yang dimiliki oleh sebuah kelas. Dibagian ini dapat dideklarasikan atribut-atribut yang dibutuhkan kelas untuk menyelesaikan permasalahan. Sebagai contoh perhatikan kode dibawah ini:

```
String nama;
```

2. Method

Method merupakan tingkah laku atau kemampuan atau fungsi yang dimiliki oleh sebuah kelas. Method ini bisa berbentuk prosedur maupun fungsi. Contoh penulisannya perhatikan kode dibawah ini:

```
void daftarUlang(){
    System.out.println("Mahasiswa sedang daftar ulang");
}
```

Bila ditulis secara lengkap class Mahasiswa yang telang dibuat tadi, maka akan menjadi:

```
class Mahasiswa(){
    String nama;

    void daftarUlang(){
        System.out.println("Mahasiswa sedang daftar ulang");
    }
}
```

Format pendeklarasian class, variabel, dan method

a) Deklarasi class dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> class <nama_class> {
    [deklarasi_atribut]
    [deklarasi_konstruktor]
    [deklarasi_metode]
}
```

Contoh deklarasi class Siswa:

```
public class Siswa {
    ...
}
```

b) Deklarasi atribut dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <tipe> <nama_atribut> ;
```

Contoh:

```
public class Siswa {
    public int nrp;
    public String nama;
}
```

c) Deklarasi method dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <return_type> <nama_metode> ([daftar_argumen]) {
    [<statement>]
}
```

Contoh:

```
public class Siswa {
    public int nrp;
    public String nama;

    public void info() {
        System.out.println("Ini siswa PENS");
    }
}
```

Mengakses properti class

Properti milik kelas adalah variabel dan method. Untuk dapat mengakses properti milik kelas maka harus dibuat sebuah obyek terlebih dahulu. Untuk membuat sebuah obyek terdapat tiga bagian yaitu :

- Deklarasi : mendeklarasikan variable obyek
- Membuat Obyek dengan menggunakan keyword new
- Inisialisasi : Operator new diikuti dengan konstruktor, untuk menginisialisasi obyek baru.

Contoh untuk mendeklarasikan variable obyek adalah

```
Point originOne;
```

Selanjutnya membuat dan menginisialisasi obyek dengan cara:

```
originOne = new Point(23, 94);
```

Berikut ini adalah contoh pengaksesan anggota-anggota dari class Siswa melalui sebuah obyek:

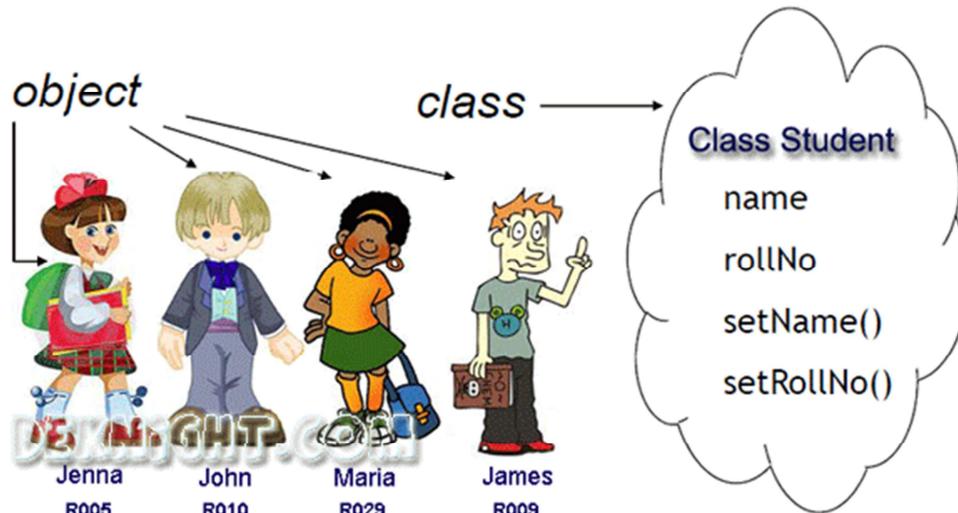
```
public class Siswa {
    public static void main(String args[]) {
        Siswa it=new Siswa();
        it.nrp=5;
        it.nama="Andi";
        it.info();
    }
}
```

Selanjutnya akan diberikan contoh sebuah class Siswa dan sebuah class lagi yang menggunakan class Siswa yaitu class Test. Class Siswa memiliki sebuah instance variabel yang bernama nrp, dan satu buah method yang bernama setNrp yang tidak memiliki return value sehingga dideklarasikan sebagai void. Method setNrp memiliki satu argumen atau parameter bernama i. Selanjutnya class Test memiliki main method yang berisi pembuatan sebuah obyek bernama anak. Selanjutnya obyek anak memanggil method setNrp dengan mengirimkan angka 5 yang digunakan untuk mengeset variabel nrp. Selanjutnya nilai nrp akan ditampilkan ke layar dengan memanggil method System.out.println().

```
public class Siswa {
    int nrp;
    public void setNrp(int i) {
        nrp=i;
    }
}

public class Test {
    public static void main(String args[]) {
        Siswa anak=new Siswa();
        anak.setNrp(5);
        System.out.println(anak.nrp);
    }
}
```

Ilustrasi hubungan antara class dan obyek dapat dilihat pada Gambar 1.1. Dimisalkan terdapat class bernama Student yang memiliki dua buah instance variabel yaitu name dan rollNo. Serta dua buah method yaitu setName() dan setRollNo(). Dari class Student tersebut maka dapat dibuat beberapa obyek. Dalam contoh ini dibuat 4 buah obyek yang masing-masing memiliki dan menyimpan variabelnya sendiri yaitu Jenna dengan rool No R005, John dengan rool No R010, Maria dengan rool No R029 dan James dengan rool No R009.



Gambar 1.1 Hubungan class dan obyek

Diambil dari <http://www.c4learn.com/java/java-class-concept/>

Konstruktor

Selain variabel dan method ada satu hal lagi yang dimiliki oleh sebuah kelas, yaitu konstruktor. Konstruktor ini adalah bagian program yang dipanggil pertama kali oleh program saat terjadi pembuatan obyek. Karena dipanggil pertama kali ketika obyek dibuat maka biasanya konstruktor digunakan untuk menginisialisasi variabel-variabel yang dimiliki oleh sebuah obyek.

Konstruktor harus memiliki nama yang sama dengan nama kelas. Konstruktor ini dipanggil pada saat proses instalasi kelas menjadi obyek. Kegunaan konstruktor adalah mengalokasikan ruang bagi sebuah kelas dalam obyek, memberikan nilai awal terhadap anggota data suatu obyek, membentuk tugas-tugas umum lainnya. Perlu diketahui bahwa konstruktor tidak mempunyai nilai balik (bahkan tanpa void).

Jadi konstruktor mempunyai ciri yaitu :

1. mempunyai nama yang sama dengan nama class
2. tidak mempunyai return type (seperti void, int, double dll)
3. memiliki argumen sebanyak 0..n

Contoh konstruktor:

```
public class Siswa {
    private int nrp;
    private String nama;
```

```

        public Siswa(int n, String m) {
            nrp=n;
            nama=m;
        }
    }

```

Jika tidak menuliskan kode konstruktor, maka secara otomatis kompiler akan menambahkan sebuah default constructor. Default constructor tidak memiliki argument dan tidak memiliki statement dalam body nya. Default constructor memiliki akses modifier yang sama dengan deklarasi kelasnya. Contoh konstruktor default adalah sebagai berikut:

```

public class Siswa{
    public Siswa(){ }
}

```

Sekali saja non default constructor dibuat/ditulis secara eksplisit, maka default konstruktor akan hilang. Contoh non default constructor:

```

public class Siswa{
    String nama;
    public Siswa(int n){
        this.mana = n;
    }
}

```

Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Hal ini disebut dengan overloading konstruktor. Contoh :

```

public class Siswa {
    private int nrp;
    private String nama;

    public Siswa(String m) {
        nrp=0;
        nama="";
    }

    public Siswa(int n, String m) {
        nrp=n;
        nama=m;
    }
}

```

C. TUGAS PENDAHULUAN

Membuat review mengenai Class dan Objek dan beri contoh 1 membuat class dan objek.

D. PERCOBAAN

Percobaan 1 : Mengakses data member suatu class.

Amati program dibawah ini:

```
public class Siswa {
    int nrp;
    public void setNrp(int i) {
        nrp=i;
    }
}

public class Test {
    public static void main(String args[]) {
        Siswa anak=new Siswa();
        anak.nrp = 5 ;
        System.out.println(anak.nrp);
    }
}
```

Percobaan 2 : Mengakses method suatu class.

Amati program dibawah ini:

```
public class Siswa {
    int nrp;
    public void setNrp(int i) {
        nrp=i;
    }
}

public class Test {
    public static void main(String args[]) {
        Siswa anak=new Siswa();
        anak.setNrp(5);
        System.out.println(anak.nrp);
    }
}
```

Percobaan 3 : Mengakses method suatu class.

Amati program dibawah ini:

```
public class Siswa {
    int nrp;
    String nama;

    public void setNrp(int i) {
        nrp=i;
    }

    public void setNama(String i) {
```

```

        nama=i;
    }
}

public class Test {
    public static void main(String args[]) {
        Siswa anak=new Siswa();
        anak.setNrp(5);
        anak.setNama("Budi");
        System.out.println(anak.nrp);
        System.out.println(anak.nama);
    }
}

```

Percobaan 4 : Membuat Class dan Konstruktor dengan parameter.

```

class Puppy{

    int puppyAge;

    public Puppy(String name){
        // This constructor has one parameter, name.
        System.out.println("Passed Name is : " + name );
    }

    public setAge( int age ){
        puppyAge = age;
    }

    public getAge( ){
        System.out.println("Puppy's age is : " + puppyAge );
        return puppyAge;
    }

    public static void main(String []args){
        Puppy myPuppy = new Puppy( "tommy" );

        myPuppy.setAge( 2 );

        myPuppy.getAge( );

        System.out.println("Variable Value : " + myPuppy.puppyAge );
    }
}

```

Percobaan 5 : Membuat class Point dan menghitung jarak antara dua Point. Buatlah fungsi utama untuk membuat dua Point dan menghitung jarak antara dua Point tersebut !

```

class Point {
    public double x, y;

    Point(double x_value, double y_value) {
        x = x_value;
        y = y_value;
    }
}

```

```

public void clear() {
    this.x = 0;
    this.y = 0;
}
public double distance(Point that) {
    double xDiff = x - that.x;
    double yDiff = y - that.y;
    return Math.sqrt(xDiff * xDiff + yDiff * yDiff);
}
}

```

Percobaan 6 : Membuat array objek

```

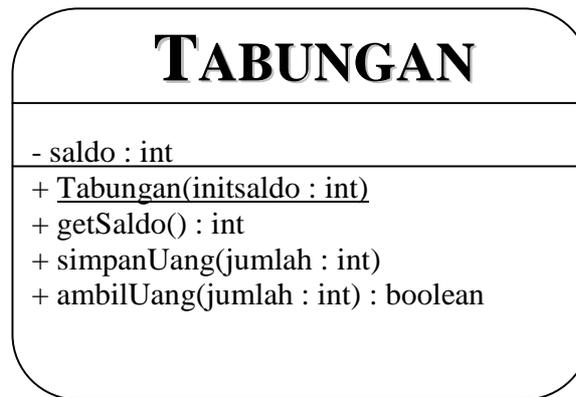
class Account{
    int a;
    int b;
    public void setData(int c,int d){
        a=c;
        b=d;
    }
    public void showData(){
        System.out.println("Value of a =" +a);
        System.out.println("Value of b =" +b);
    }
}

class ObjectArray{
    public static void main(String args[]){
        Account obj[] = new Account[2] ;
        obj[0] = new Account();
        obj[1] = new Account();
        obj[0].setData(1,2);
        obj[1].setData(3,4);
        System.out.println("For Array Element 0");
        obj[0].showData();
        System.out.println("For Array Element 1");
        obj[1].showData();
    }
}

```

E. LATIHAN

Latihan1 : Mengimplementasikan UML class diagram dalam program untuk class Tabungan



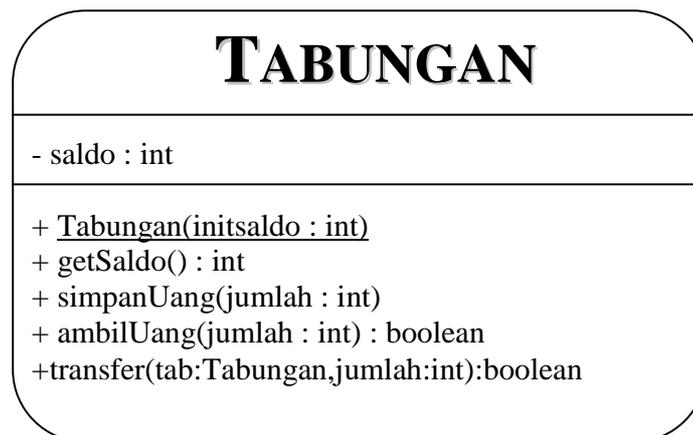
Gambar 2.4 Class Tabungan

Transformasikan class diagram diatas ke dalam bentuk program? Jalankan file TesTugas1.class. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```

Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000 ok
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000 ok
Jumlah uang yang diambil : 1600 gagal
Jumlah uang yang disimpan : 2000
Saldo sekarang = 3500
  
```

Latihan 2 : Menambahkan pada class Tabungan method transfer().



Gambar 2.4 Class Tabungan dengan tambahan method transfer()

Latihan 3 :

Buatlah class Lingkaran yang didalamnya terdapat variabel jari-jari. Buatlah method untuk menghitung luas dan keliling lingkaran. Selanjutnya buatlah sebuah obyek Lingkaran.

- Obyek lingkaran 1 dengan jari-jari 5.
- Obyek lingkaran 2 dengan jari-jari 7.5.
- Obyek lingkaran 3 dengan jari-jari random.

Hitung luas dan keliling lingkaran dari tiap obyek.

Latihan 4 :

Melanjutkan dari latihan 3. Buatlah class Lingkaran yang didalamnya terdapat variabel jari-jari dan Point untuk titik pusat. Tambahkan method pada class Lingkaran

```
public boolean bersinggungan(Lingkaran lingkaran){ }
```

mengembalikan nilai true jika bersinggungan dan bernilai false jika tidak bersinggungan.

Selanjutnya buatlah sebuah obyek Lingkaran.

- Obyek lingkaran 1 dengan jari-jari 2 dan titik pusat (2,2)
- Obyek lingkaran 2 dengan jari-jari 1 dan titik pusat (7,1)
- Obyek lingkaran 3 dengan jari-jari 1.5 dan titik pusat (4,1.5).

Dari 3 obyek tersebut, manakah obyek lingkaran yang bersinggungan ?

F. LAPORAN RESMI

Kerjakan hasil percobaan(D) dan latihan(E) di atas dan tambahkan analisa.