



Modul Praktikum Pemrograman Berorientasi Objek

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

2022
Semester Ganjil

Modul 5. Class

Dalam paradigma *Object Oriented Programming*, kita memodelkan objek di dunia nyata menjadi sebagai sebuah class di dalam program. Class sendiri adalah cetak biru (*blue print*) dari objek yang akan dibuat. Sebuah class berisi gambaran objek secara umum.

Capaian Pembelajaran

1. Mahasiswa mampu membuat program yang menerapkan konsep Class
2. Mahasiswa mampu membuat program yang menerapkan *access modifier*
3. Mahasiswa mampu membuat program dengan membaca class diagram

Materi

Struktur Class

Sebuah class memiliki state/attribute, method, dan constructor. Sintaks untuk mendeklarasikan class adalah sebagai berikut:

```
access_modifier class <nama_class> {  
    attribute;  
    method;  
    constructor;  
}
```

Contoh:

```
class Dot {  
    int x = 5;  
    int y = 3;  
  
    public void cetakXY() {  
        System.out.println("x:" + x + " y:" + y);  
    }  
}
```

Object

Untuk mengakses atribut ataupun method dari sebuah class, anda harus menginisiasikan sebuah objek terlebih dahulu. Objek adalah variabel/atribut

yang dibuat berdasarkan class tertentu. Untuk membuat sebuah object kita dapat menggunakan syntax sebagai berikut :

NamaClass **namaObject** = **new** NamaClass(**params**)

Class Attribute

Pada bagian sebelumnya, penggunaan istilah "variabel" untuk x dalam contoh (seperti yang ditunjukkan di bawah). Ini sebenarnya adalah atribut dari class. Atau dapat dikatakan bahwa atribut class adalah variabel di dalam class. Biasanya atribut digunakan untuk menyimpan data dari objek yang akan dibuat. Misalnya class Mahasiswa akan memiliki atribut nama, npm, kelas, angkatan, dll. Contoh:

```
class Dot{
    int x = 5;
    int y = 3;
}
```

Class Dot diatas memiliki atribut x dan y yang diberi nilai 5 dan 3.

Untuk mengakses atribut dari kelas dapat menggunakan simbol (.) pada objek yang dibuat dari suatu class. Sebagai contoh:

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Dot titik = new Dot();
        System.out.println(titik.x);

    }
}

class Dot{
    int x = 5;
    int y = 3;
}
```

Objek titik dibuat dari class Dot, kemudian atribut x diakses menggunakan **titik.x**.

Atribut tersebut dapat diubah dan diganti dengan nilai lainnya. Atribut pada class tidak harus langsung diberi nilai. Tidak ada batasan jumlah atribut pada class. Perhatikan contoh berikut:

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Dot titik = new Dot();

        titik.x = 1;
        titik.y = 2;

        System.out.println(titik.x);
        System.out.println(titik.y);

    }
}

class Dot{
    int x = 5;
    int y;
}
```

Output:

```
1
2
```

Method

Method merupakan sekumpulan function yang digunakan untuk memanipulasi nilai dari atribut, menerima informasi dari object lain, dan mengirim informasi ke object lain. Struktur dari method adalah:

```
access_modifier <return_type> <nama_method> (parameter) {
    statement;
}
```

Dalam kasus class Dot diatas, kita bisa membuat suatu method untuk menampilkan nilai x dan y, yaitu:

```
public void cetakXY() {
    System.out.println("x:" + x + " y:" + y);
}
```

```
}
```

Untuk mengakses method yang ada di class sama dengan mengakses atribut, yaitu menggunakan . (titik)

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Dot titik = new Dot();
        titik.cetakXY();

    }
}

class Dot {
    int x = 5;
    int y = 3;

    public void cetakXY() {
        System.out.println("x:" + x + " y:" + y);
    }
}
```

Output:

```
x: 5 y: 3
```

Hubungan Class dan Object

Sederhananya, object adalah instance dari class. Seperti yang sudah dijelaskan di atas, bahwa class hanyalah sebuah blueprint, kerangka dasar. Dibutuhkan instansi pembuatan object untuk mengimplementasikan serta memberi identitas serta isi object tersebut.



Class dalam Satu File yang Sama

Pada bagian sebelumnya, class yang dibuat masih dalam satu file yang sama. Class dalam satu file dapat dibuat dibawah atau di atas class utama. Class tambahan yang dibuat dalam satu file yang sama tidak boleh menggunakan akses modifier 'public'.

Class dalam File Baru yang Berbeda

Sebelumnya class yang dibuat masih berada dalam satu file. Hal ini tidak menjadi masalah apabila class yang dibuat hanya sedikit, lain halnya jika class yang dimiliki sudah banyak dan panjang. Memisahkan class-class tersebut kedalam file yang berbeda bisa memudahkan pembagian program dan juga mempermudah maintenance. Coba untuk membuat file class baru dengan nama Dot.java yang berisi:

```
class Dot {
    int x = 5;
    int y = 3;

    public void cetakXY() {
        System.out.println("x:" + x + " y:" + y);
    }
}
```

Pastikan bahwa Dot.java berada dalam satu package/folder dengan class Main.

Access Modifier

Access modifier dapat dikatakan sebagai hak akses bagi class/method lain untuk menggunakan class/method tersebut.

Untuk class anda dapat menggunakan dua Access Modifiers berikut:

Modifier	Deskripsi
public	Class bertipe public dapat diakses dari class mana saja.
default	Class ini hanya dapat diakses dari class dalam package yang sama. Tipe ini digunakan saat Anda tidak menuliskan access modifier.

Access modifier untuk attributes, methods, dan constructors:

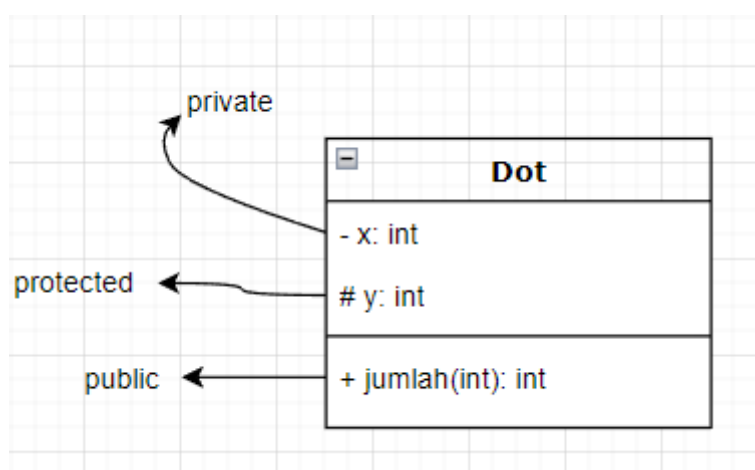
Modifier	Deskripsi
public	Access modifier bertipe public dapat diakses oleh class mana saja.
private	Hanya dapat diakses di dalam class yang dideklarasikan.
default	Class ini hanya dapat diakses oleh class dalam package yang sama. Tipe ini digunakan saat Anda tidak menuliskan access modifier.
protected	Dapat diakses dalam package yang sama dan subclass.

Simbol dalam Class Diagram

Dari ketiga access modifier yang dibahas, dalam penulisannya pada class diagram dapat mengacu pada tabel berikut.

Access modifier	Simbol
public	+
private	-
protected	#

Contoh class diagram untuk class Dot.



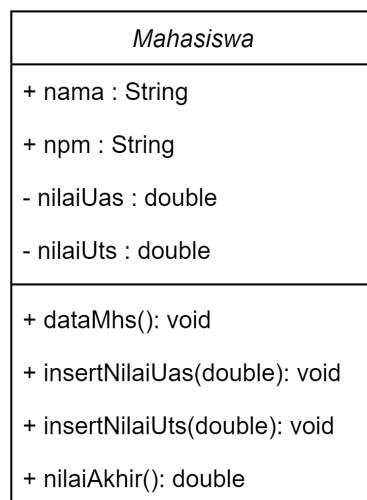
Langkah Praktikum

Membuat Project

1. Buka Netbeans
2. Buat Project baru dengan nama **Nama_Praktikum6**

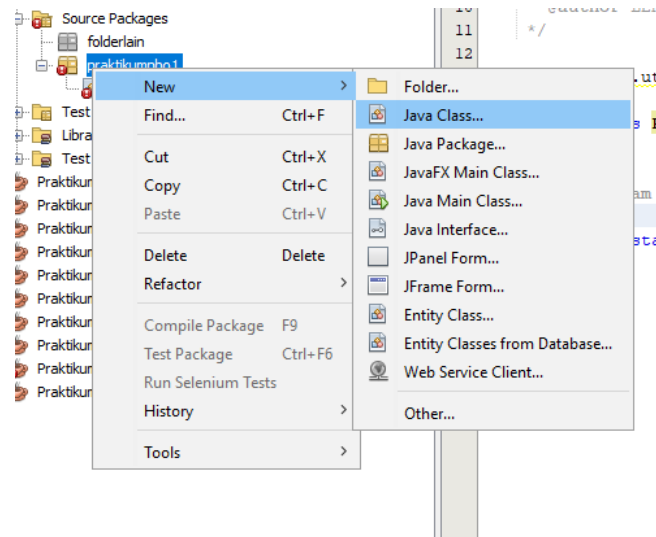
Studi Kasus

Alif sebagai asisten dosen PBO ingin mendata nilai mahasiswa yang ada di kelas nya. Untuk mempermudah pekerjaannya, Alif membuat program yang dapat menyimpan data mahasiswanya berdasarkan class diagram dibawah.

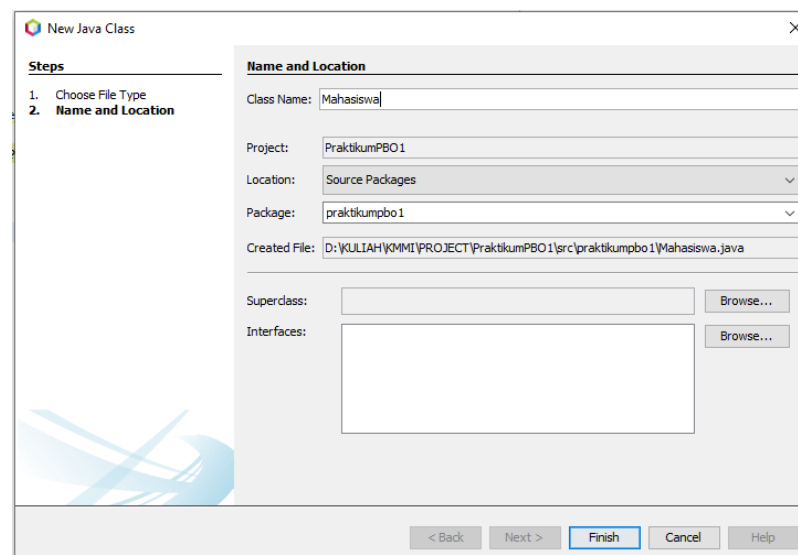


Membuat File Class Mahasiswa

1. Buat class baru dalam package yang sama dengan class Main



2. Beri nama class Mahasiswa



Let's Coding

1. Buatlah atribut nama dan npm sesuai dengan yang ada pada class diagram

```
package praktikum6;

public class Mahasiswa {
    public String nama;
    public String npm;
}
```

2. Instansikan objek mahasiswa pada class Main

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args)
    {

        Mahasiswa mhs1 = new Mahasiswa();

    }

}
```

3. Beri nilai nama dan npm ke mhs1, dan tampilkan nilai tersebut

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Mahasiswa mhs1 = new Mahasiswa();

        mhs1.nama = "Naufal Anbial Falah";
        mhs1.npm = "2017051072";

        System.out.println("Nama : " + mhs1.nama);
        System.out.println("NPM : " + mhs1.npm);

    }

}
```

4. Kembali ke file Mahasiswa.java, dan buatlah atribut nilaiUas dan nilaiUts sesuai dengan class diagram

```
package praktikum6;

public class Mahasiswa {
    public String nama;
    public String npm;
    private double nilaiUas;
    private double nilaiUts;
}
```

5. Coba berilah nilai untuk nilaiUas dan nilaiUts, sama seperti memberi nilai pada nama dan npm sebelumnya

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Mahasiswa mhs1 = new Mahasiswa();

        mhs1.nama = "Naufal Anbial Falah";
        mhs1.npm = "2017051072";

        System.out.println("Nama :" + mhs1.nama);
        System.out.println("NPM  :" + mhs1.npm);

        mhs1.nilaiUas = 90;
        mhs1.nilaiUts = 90;

    }
}
```

6. Akan terjadi error seperti dibawah ini, ini dikarenakan nilaiUas dan nilaiUts memiliki access modifier private, sehingga tidak bisa diakses secara langsung dari class lain

The field Mahasiswa.nilaiUas is not visible

The field Mahasiswa.nilaiUts is not visible

7. Jangan panik, kembali ke file Mahasiswa.java dan buatlah method untuk memasukkan nilaiUts dan juga nilaiUas.

```
package praktikum6;

public class Mahasiswa {
    public String nama;
    public String npm;
    private double nilaiUas;
    private double nilaiUts;

    public void insertNilaiUas(double nilai){
        nilaiUas = nilai;
    }
    public void insertNilaiUts(double nilai){
        nilaiUts = nilai;
    }
}
```

8. Perbaiki error di class Main dengan memanggil method baru yang sudah dibuat, dan coba jalankan

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Mahasiswa mhs1 = new Mahasiswa();

        mhs1.nama = "Naufal Anbial Falah";
        mhs1.npm = "2017051072";

        mhs1.insertNilaiUas(90);
        mhs1.insertNilaiUts(90);

    }
}
```

9. Error berhasil diatasi, kita berhasil mengganti nilai hal ini bisa dilakukan dikarenakan method yang dibuat berada dalam satu kelas, dan access modifier function public, sehingga bisa diakses dari luar kelas. Coba akses nilaiUts dan nilaiUas, dan tampilkan.

```
package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Mahasiswa mhs1 = new Mahasiswa();

        mhs1.nama = "Naufal Anbial Falah";
        mhs1.npm = "2017051072";

        System.out.println("Nama : " + mhs1.nama);
        System.out.println("NPM : " + mhs1.npm);

        mhs1.insertNilaiUas(90);
        mhs1.insertNilaiUas(90);

        System.out.println("Nilai Uas : " + mhs1.nilaiUas);
        System.out.println("Nilai Uts : " + mhs1.nilaiUts);

    }
}
```

10. Akan terjadi error kembali, sama seperti saat kita mencoba memberi nilai secara langsung terhadap kedua atribut tersebut. Cobalah buatlah method untuk menampilkan data-data mahasiswa yang sudah dibuat dengan nama dataMhs.

```

package praktikum6;

public class Mahasiswa {
    public String nama;
    public String npm;
    private double nilaiUas;
    private double nilaiUts;

    public void insertNilaiUas(double nilai){
        nilaiUas = nilai;
    }
    public void insertNilaiUts(double nilai){
        nilaiUts = nilai;
    }

    public void dataMhs(){
        System.out.println("Nama      : " + nama);
        System.out.println("NPM      : " + npm);
        System.out.println("Nilai UAS : " + nilaiUas);
        System.out.println("Nilai UTS : " + nilaiUts);
    }
}

```

11. Panggil method tersebut di class Main

```

package praktikum6;

public class Praktikum6 {

    public static void main(String[] args) {

        Mahasiswa mhs1 = new Mahasiswa();

        mhs1.nama = "Naufal Anbial Falah";
        mhs1.npm = "2017051072";

        mhs1.insertNilaiUas(90);
        mhs1.insertNilaiUts(90);

        mhs1.dataMhs();
    }
}

```

12. Berdasarkan class diagram, kita belum memiliki method nilaiAkhir. Coba buat method tersebut untuk menghitung nilai akhir berupa rata-rata dari nilaiUts dan nilaiUas.

```
public double nilaiAkhir(){
    double jumlah = nilaiUts + nilaiUas;
    return jumlah/2;
}
```

13. Perbarui method dataMhs, sehingga nilaiAkhir juga bisa di tampilkan berbarengan dengan data lain

```
package praktikum6;

public class Mahasiswa {
    public String nama;
    public String npm;
    private double nilaiUas;
    private double nilaiUts;

    public void insertNilaiUas(double nilai){
        nilaiUas = nilai;
    }
    public void insertNilaiUts(double nilai){
        nilaiUts = nilai;
    }

    public void dataMhs(){
        System.out.println("Nama      : " + nama);
        System.out.println("NPM       : " + npm);
        System.out.println("Nilai UAS : " + nilaiUas);
        System.out.println("Nilai UTS : " + nilaiUts);
        System.out.println("Nilai Akhir : " + nilaiAkhir());
    }

    public double nilaiAkhir(){
        double jumlah = nilaiUts + nilaiUas;
        return jumlah/2;
    }
}
```

14. Coba tampilkan keseluruhan data mahasiswa

Tugas

Silakan lihat di vclass