POLINDRA

Class dan Objek

Dr. Ir. Mohammad Yani, S.T., M.T., M.Sc

Pemrograman Prosedural vs PBO

- Unit dalam pemrograman procedural adalah **fungsi**, sedangkan unit dalam OOP adalah **class**
- Pemrograman procedural berkonsentrasi pada pembuatan fungsi-fungsi, sedangkan OOP memulai dari mengisolasi class, kemudian mencari/menggunakan metoda2 di dalamnya
- Pemrograman procedural memisahkan data program dari operasi2 yg memanipulasi datanya, sementara OOP focus pada keduanya

Konsep Class dan Objek

- “Class” merujuk pada **cetak biru (blue print)**. Class mendefinisikan variabel2 dan metoda2 pendukung Object
- “Object” adalah instan dari class. Tiap2 object memiliki sebuah class yg mendefinisikan data dan tingkah lakunya

Member Class

- **Sebuah class dapat memiliki tiga jenis member:**
  - ***fields***: variabel2 data yg mendefinisikan status class atau object
  - ***methods***: ***executable code*** dari class yg terbentuk dari beberapa *statements*. **Methods** dapat memanipulasi/merubah status object atau akses nilai dari member data
  - ***nested classes and nested interfaces***

Contoh Class

Sample class

```
class Pencil {
    public String color = “red”;
    public int length;
    public float diameter;

        public static long nextID = 0;

        public void setColor (String newColor) {
            color = newColor;
        }
}
```

Pendeklarasian Fields

- **Pendeklarasian Field**
  - Tipe data diikuti dengan nama field-nya
  - Tipe data primitive vs. Object reference
    - boolean, char, byte, short, int, long, float, double
  - Pendeklarasian field declarations dapat didahului dengan modifiers yang berbeda2
    - access control modifiers
    - static
    - final

Field Mofidier

- Access control modifiers
  - *private:* private members hanya dapat diakses oleh dirinya sendiri
  - *package:* package members dapat diakses oleh dirinya sendiri dan kelas lain yang ada dalam satu package
  - *protected:* protected members dapat diakses oleh dirinya sendiri, kelas lain dalam satu package, dan sub-class nya
  - *public:* public members dapat diakses oleh semua class

Pencil.java

```
public class Pencil {
    public String color = "red";
    public int length;
    public float diameter;
    private float price;

        public static long nextID = 0;

        public void setPrice (float
 newPrice) {
            price = newPrice;
        }
 }
```

CreatePencil.java

```
public class CreatePencil {
    public static void main (String args[]){
        Pencil p1 = new Pencil();
        p1.price = 0.5f;
        }
}
```

```
%> javac Pencil.java
%> javac CreatePencil.java
CreatePencil.java:4: price has private access in Pencil
      p1.price = 0.5f;
         ^
```

Field Mofidier (lanjutan)

- static
  - Hanya satu copy field static eksis, dibagi oleh object2 dari class tsb
  - Dapat diakses secara langsung dalam kelas dirinya sendiri
  - Akses dari luar harus didahuli oleh nama class spt berikut
  - ```
    System.out.println(Pencil.nextID);
    ```
  - Atau melalui object milik class
  - Dari luar class, field non-static harus diakses melalui referensi object
    ```

```
public class CreatePencil {
    public static void main (String args[]){
            Pencil p1 = new Pencil();
            Pencil.nextID++;
            System.out.println(p1.nextID);
            //Result? 1

            Pencil p2 = new Pencil();
            Pencil.nextID++;
            System.out.println(p2.nextID);
            //Result? 2

            System.out.println(p1.nextID);
            //Result? still 2!
    }
}
```

Note: this code is only for the purpose of showing the usage of static fields. It has POOR design!

Field Mofidier (lanjutan)

- final
  - Hanya satu kali diinisiasi, nilai tidak dapat diubah
  - Sering digunakan untuk mendefinisikan konstanta
  - Field static final harus diinisialisasi ketika class diinisialisasi
  - Field non-static final harus diinisialisasi ketika object dari class dikonstruksi

Inisialisasi Fields

Inisialisasi Field

Tidak selalu menjadi konstanta, selama tipenya benar

Jika tidak ada inisialisasi, maka nilai inisial default diisi tergantung pada jenis tipenya

Type	Initial Value
boolean	false
char	‘\u0000’
byte, short, int, long	0
float	+0.0f
double	+0.0
object reference	null

Objek

- Objek adalah realisasi dari kelas
- Semua yang kita lihat di sekitar kita adalah objek, seperti: Orang, Mobil, Binatang, dan lain-lain adalah objek
- Hubungannya dengan kelas adalah:

contoh

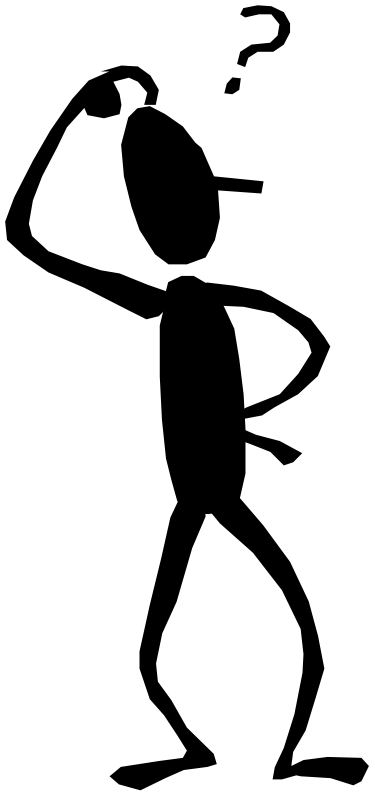
Objek Orang. Orang adalah objek, dan bentuk dasar dari objek adalah Kelas. Bentuk dasar dari Orang adalah: nama memiliki nama, alamat, usia, dan lain-lain, inilah yang dimaksud dengan kelas.

Contoh penulisan kelas dalam program java

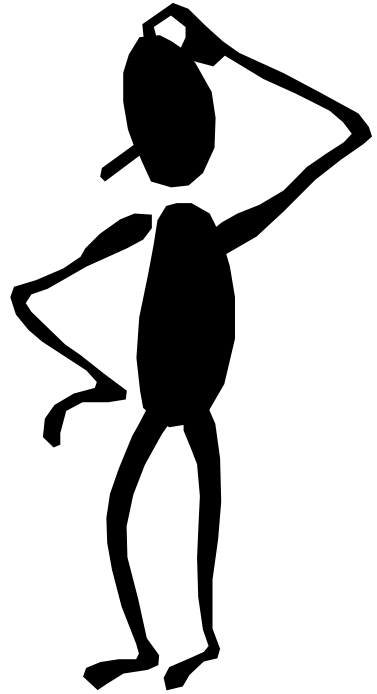
```
public class Mobil{}
```

Contoh program membuat obyek mb dari kelas Mobil

```
public static void main(String[] args){
        Mobil mb = new Mobil();
}
```



Pertanyaan?



Terima Kasih

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI INDRAMAYU
Jalan Raya Lohbener Lama Nomor 8 Lohbener – Indramayu 45252
Telepon/Faximile: (0234) 5746464
Laman : http://www.polindra.ac.id e-mail: info@polindra.ac.id