



unindra
universitas indraprasta PGRI

Pembungkusan

Encapsulation

PERTEMUAN 10

Encapsulation

- *Encapsulation* merupakan salah satu dari konsep OOP pada bahasa pemrograman java, konsep ini memiliki mekanisme untuk membuangkus sebuah data/variable, data tersebut tidak bisa diakses oleh class lain, dengan menggunakan konsep Encapsulation, semua data harus kita set menjadi private, agar class lain tidak dapat mengaksesnya.
- Kata kunci *private* adalah sebuah modifier untuk membatasi hak akses pada variabel, class dan method, jika data-data di set hak aksesnya menjadi private, maka data tersebut hanya bisa digunakan oleh class itu sendiri dan tidak bisa digunakan oleh class luar
- **Tujuan encapsulation** adalah untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interfensi atau penyalahgunaan dari luar system dan sekaligus menyederhanakan system itu sendiri

Dalam kehidupan sehari-hari, penyembunyian data sudah sering kita alami.

- Contohnya adalah objek Handphone; dimana setiap orang yang menggunakannya tidak mengetahui secara pasti apa yang ada di dalam Handphone tersebut. Tidak mengetahui bagaimana cara kerja HP tersebut. Yang penting adalah menggunakannya melalui interface yang disediakan.
- Kita mengirim pesan WA ke teman kita, tapi kita tidak tahu proses system bagaimana cara nya pengiriman pesan tersebut dari media Handphone kita
- Kita membuat Rekening Bank dengan menyerahkan data-data seperti KTP, data pribadi dll, tapi pihak bank tidak mengetahui kode PIN kita

Getter dan Setter

- Dalam mengakses data tersebut dalam encapsulation bisa dengan menggunakan method public *Getter* dan *Setter*
- Dinamakan accessor (get) karena metode ini hanya mengambil tanpa mengubah state dari member tersebut, dan dinamakan mutator (set) karena metode ini juga mengubah state dari member tersebut. Method-method ini dideklarasikan sebagai public, sedangkan variable dideklarasikan sebagai private.
- Getter dan setter digunakan untuk secara efektif melindungi data, terutama saat membuat kelas. Untuk setiap variabel, metode get mengembalikan nilai, sedangkan metode set, meng set kan nilai.
- Pada dasarnya metode get digunakan untuk mengembalikan nilai pada dari variable dengan menggunakan fungsi *return* dan set merupakan method void untuk mensetting atau memberikan nilai pada variable.

- Getter dan setter memungkinkan kita untuk memiliki kontrol atas nilai-nilai. misalnya, memvalidasi nilai yang diberikan dalam setter sebelum benar-benar menetapkan nilai.
- Getter mulai dengan get, lalu diikuti dengan nama variabel, dengan huruf pertama dari nama variabel dikapitalisasi. Setter mulai dengan set, diikuti dengan nama variabel, dengan huruf pertama dari nama variabel dikapitalisasi.
- Metode getter mengembalikan nilai dari atribut. Metode setter mengambil parameter dan memberikan ke atribut.
- Method setter tidak memiliki nilai kembalian void (kosong). Karena tugasnya hanya untuk mengisi data ke dalam atribut.
- Sedangkan method getter memiliki nilai kembalian sesuai dengan tipe data yang akan diambil.

Contoh Program encapsulation

```
1 class charlie
2   { private int a = 5;
3     protected int b = 7;
4     public void info()
5     {
6       System.out.println("Di panggil pada =" + this.getClass().getName());
7       System.out.println("a=" + a);
8       System.out.println("b=" + b);
9     }
10  }
```

```
1 class delta extends charlie
2   { private int c = 9;
3     public void info()
4     {
5       System.out.println("Di panggil pada =" + this.getClass().getName());
6       System.out.println("a=" + a);
7       System.out.println("b=" + b);
8       System.out.println("c=" + c);
9     }
10    public static void main(String[] args)
11    {
12      charlie c = new charlie();
13      c.info();
14      delta d = new delta ();
15      d.info ();
16    }
17  }
```

a has private access in charlie

Atribut a tidak bisa di akses oleh class turunan, karena atribut a di kontrol aksesnya private di kelas parents/induk
Jadi analisis terlebih dahulu untuk atribut mana yang bisa di akses atau tidak oleh turunan

Contoh Program encapsulation

```
public class encap_nilai2 extends encap_nilai {
    private int nilaiku;

    void setNilai (int nilai){
        nilaiku=nilai;
    }

    int getNilai(){
        return nilaiku;
    }
}

public class encap_nilaimain {
    public static void main(String[] args) {
        encap_nilai Biodata = new encap_nilai();
        encap_nilai2 Biodata2 = new encap_nilai2();
        Biodata>Nama = "Adelya";
        Biodata.NIM = 201743501;
        Biodata2.setNilai(90);

        Biodata.data ();
        System.out.println("Nilai = "+ Biodata2.getNilai());
    }
}

public class encap_nilai {
    public String>Nama;
    public int>NIM;

    void data ( ){
        System.out.println("Nama = "+ Nama);
        System.out.println("NIM = "+ NIM);
    }
}

Options
Nama = Adelya
NIM = 201743501
Nilai = 90
```

Konsep yg di gunakan pada program tersebut adalah mengprivatekan atribut nilaiku agar tidak di akses class lainnya. Untuk atribut di kelas induk bisa di turunkan lagi ke class turunan yang lainnya. Contoh : class mata pelajaran, dll