

Percobaan Enkapsulasi – Access Modifier Private

1. Pendahuluan

Pemrograman berorientasi objek memodelkan dunia nyata kedalam dunia pemrograman. Dengan demikian sebuah program dapat dikorelasikan atau dipahami dengan mudah seperti halnya di dunia nyata. Enkapsulasi adalah salah satu konsep penting dalam pemodelan pemrograman berorientasi objek ini.

Enkapsulasi adalah mekanisme untuk membungkus atribut dan method menjadi satu unit/kesatuan. Didalam enkapsulasi, atribut yang ada didalam class disembunyikan dari class lain. Atribut tersebut dapat diakses dari class lain melalui method yang disediakan. Enkapsulasi disebut juga dengan data hiding.

Salah satu metode enkapsulasi adalah penggunaan access modifier private yang memungkinkan untuk suatu variabel atau method tidak bisa diakses secara langsung oleh class lain. Tujuan dari penggunaan enkapsulasi ini adalah agar class lain tidak mudah untuk memodifikasi nilai dari variabel tertentu agar tidak merusak / mengganggu program.

2. Percobaan

Untuk lebih jelasnya dalam penggunaannya di program Java, cobalah untuk membuat script sesuai dengan petunjuk berikut!

Buatlah class Siswa dengan kode program seperti di bawah ini!

```
public class Siswa {  
  
    private String nama;  
    private int nilai = 0;  
    private String kelas;  
  
    public void isiData(){  
        nama = "Steven";  
        nilai = 90;  
        kelas = "XI RPL B";  
    }  
  
    public void cetak(){  
        System.out.println("Nama      : " + nama);  
        System.out.println("Nilai   : " + nilai);  
        System.out.println("Kelas  : " + kelas);  
    }  
}
```

Lalu apa jadinya jika variabel yang sudah diberi akses modifier private kemudian diakses pada class lain?
Coba buatlah class SiswaObj dengan kode program seperti di bawah!

```
12 public class SiswaObj {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         Siswa s1 = new Siswa();
19         System.out.println(s1.nilai);
20     }
21
22 }
23
```

nilai has private access in Siswa

(Alt-Enter shows hints)

Akan muncul error seperti di atas, yang artinya nilai tidak bisa diakses secara langsung oleh class lain.

Kemudian modifikasi kode program menjadi seperti di bawah ini!

```
public class SiswaObj {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Siswa s1 = new Siswa();
        s1.isiData();
        s1.cetak();
    }

}
```

```
Output - enkapsulasi (run)

run:
Nama      : Steven
Nilai     : 90
Kelas    : XI RPL B
BUILD SUCCESSFUL (total time: 0 seconds)
```

Hasil Output akan terlihat seperti di atas.

Untuk lebih memudahkan dalam memahami enkapsulasi, silahkan kerjakan soal berikut sebagai bentuk latihan.

Soal

1. Modifikasi lah kode berikut dengan menambahkan enkapsulasi menggunakan modifier private baik di method ataupun variabel

```
public class Siswa {

    private String nama;
    private int nilai = 0;
    private String kelas;

    public void isiData(){
        nama = "Steven";
        nilai = 90;
        kelas = "XI RPL B";
    }

    public void cetak(){
        System.out.println("Nama      : " + nama);
        System.out.println("Nilai     : " + nilai);
        System.out.println("Kelas    : " + kelas);
    }

}
```

```
public class SiswaObj {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        Siswa s1 = new Siswa();  
        s1.isiData();  
        s1.cetak();  
    }  
}
```

-
2. Buatlah program untuk menghitung luas bangun datar dengan memenuhi kriteria berikut
- Terdapat deklarasi class
 - Terdapat deklarasi variabel
 - Terdapat deklarasi method
 - Penggunaan enkapsulasi (modifier private)
 - Input menggunakan Scanner atau JOptionPane

Lalu tulis laporan yang berisi screenshot dari program dan juga penjelasan dari kode program. Kumpulkan berserta dengan source code ke dalam zip/rar (atau bisa diupload di github dan link dilampirkan pada laporan)
