

Nama: Isri Amirul Haq  
NIM: 235150600111011

## JAWABAN BAB 4 ENCAPSULATION

Data dan Analisis hasil percobaan

### A. Encapsulation 1

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

**Jawab:**

- Saat memanggil setMark() di main(), nilai yang diberikan adalah string "90", sedangkan parameter untuk setMark() adalah integer.
- Saat mencoba mencetak nilai name dan mark di main(), variabel tersebut tidak dapat diakses karena memiliki hak akses private. Berikut ini adalah koreksi dari kedua kesalahan tersebut:

```
// (before) menggunakan string padahal yang diinputkan adalah integer
// s1.setMark("90");

s1.setMark(m:90); // Menggunakan integer

// System.out.println("s1Name is " + s1.setName());
// System.out.println("s1Mark is " + s1.setMark());

System.out.println("s1Name is " + s1.getName()); // Menggunakan getName()
System.out.println("s1Mark is " + s1.getMark()); // Menggunakan getMark()

// System.out.println("name dan mark " + name + " " + mark);
System.out.println("name dan mark " + s1.getName() + " " + s1.getMark());
```

2. Jika pada baris 6 s1.setName diubah menjadi s1.getName apa yang terjadi? Jelaskan!

**Jawab:**

Pengaruh Mengubah s1.setName Menjadi s1.getName: Jika kita mengubah s1.setName() menjadi s1.getName(), itu berarti kita mencoba untuk memanggil method getName() tanpa mengirimkan nilai apapun (tanpa argumen). Namun, method getName() tidak memiliki argumen, sehingga pemanggilan ini akan berhasil dan mengembalikan nilai dari atribut name dari objek s1.

3. Setelah diperbaiki, ubahlah hak akses pada baris 4 (pada class Student) menjadi private apa yang terjadi jika class Test dijalankan? Jelaskan!

**Jawab:**

Hak Akses Private pada Kelas Student: Jika kita mengubah hak akses dari atribut name dan mark menjadi private di kelas Student, maka mereka tidak akan dapat diakses secara langsung dari luar kelas Student. Oleh karena itu, ketika mencoba mengakses name dan mark dari kelas Test, akan terjadi error karena tidak ada akses ke atribut tersebut.

4. Jika kedua kelas diatas terdapat dalam package yang sama apakah konsep enkapsulasi

tetap berfungsi? Jelaskan!

**Jawab:**

Konsep Enkapsulasi dalam Satu Package: Ya, konsep enkapsulasi tetap berfungsi dalam satu package. Meskipun atribut name dan mark memiliki hak akses private, mereka masih dapat diakses oleh kelas yang berada dalam package yang sama (Test dalam hal ini) melalui penggunaan metode getName() dan getMark() yang telah disediakan. Ini adalah salah satu keuntungan dari penggunaan package dalam pemrograman Java, yaitu kelas-kelas dalam package yang sama dapat saling berinteraksi dengan atribut dan metode yang bersifat private.

## B. Encapsulation 2

Pertanyaan

1. Method apakah yang menjadi accessor (getter) ?

**Jawab:**

- a. getLoad() untuk mendapatkan nilai dari atribut load.
- b. getMaxLoad() untuk mendapatkan nilai dari atribut maxLoad.

Mereka digunakan untuk mengambil nilai dari atribut load dan maxLoad secara publik.

2. Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

**Jawab:**

Apabila program dijalankan, maka akan terjadi error karena load bersifat private.

3. Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi public.

Jalankan program, apakah output dari program tersebut?

- a. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

**Jawab:**

Apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"

- b. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

```
System.out.println("Add load(100kg) : " + (vehicle.load=500));
```

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

**Jawab:**

apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"

4. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload

pada class Vehicle1 menjadi protected.

**Jawab:**

- a. apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"
- b. apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"

5. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi default.

**Jawab:**

- a. apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"
- b. apabila program dijalankan, maka akan menampilkan "Add load(100kg) : 500.0"