1. **DEFINISI MASALAH**

Tasasasas

1. **SOURCE CODE**

|  |  |
| --- | --- |
| 1 | Courier New 10 |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| 1 | Perbaris bukan paragraf memakai font times new roman 11 |

1. **SCREENSHOT PROGRAM**
2. **PRAKTIKUM**

**A. Encapsulation 1**

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Jawab:

* Pada baris ke-6 tanda petik dihilangkan atau dari String diganti dengan integer.
* Pada baris ke-7 dan ke-8 seharusnya memanggil method getter bukan setter.
* Di baris ke-9 seharusnya variabel name dan mark diganti menjadi s1.getName () dan s1.getMark()

1. Jika pada baris 6 *s1.setName* diubah menjadi *s1.getName* apa yang terjadi? jelaskan!

Jawab:

Benar karena memang seharusnya yang dipanggil adalah accesornya bukan mutatornya.

1. Lakukan perubahan pada baris 5 dengan menghilangkan String di tanda dalam kurung lalu ubah baris 7 seperti no. 3 apa yang terjadi? jelaskan!

Jawab:

Eror, karena method setName memiliki parameter String

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

Jawab:

Eror, karena modifier tidak bias diakses pada kelas yang berbeda sedangkan pada kelas Test ada instansi objek untuk memanggil method setName

1. Jika kedua kelas diatas terdapat dalam package yang sama apakah konsep enkapsulasi tetap berfungsi? jelaskan!

Jawab:

Tetap berfungsi, karena salah satu kelas menyembunyikan informasi sehingga tidak bisa diakses oleh kelas. Ini berarti menerapkan salah satu fungsi dari enkapsulasi, yaitu informasi hiding.

**B. Encapsulation 2**

Pertanyaan

1. Method apakah yang menjadi accessor (getter) ?

Jawab:

getLoad() dan detMaxLoad()

1. 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:

Eror, karena variabel load pada kelas vehicle memiliki modifier private sehingga tidak dapat diakses pada kelas lainnya.

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

Jalankan program, apakah output dari program tersebut?

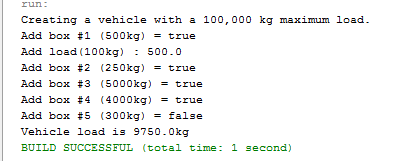
a. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

Jawab:



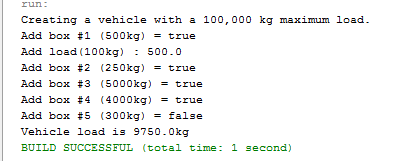
b. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

Jawab:



1. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi **protected.**

Jawab:

Program berjalan seperti soal nomor 3, karena modifier protected ini dapat diakses dikelas yang sama dan package yang sama pula. Asal masih dalam satu package

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

1. **KESIMPULAN**

Berisi kesimpulan program sesuai dengan bab yang di ajarkan. Font Times New Roman 11