1. **DEFINISI MASALAH**

1. Soal 1

Tugas Anda adalah membuat sebuah class yang memuat data-data pada buku alamat.

Tabel berikut mendefinisikan informasi yang dimiliki oleh buku alamat.

|  |  |
| --- | --- |
| Attribut | Deskripsi |
| Nama | Nama Lengkap perseorangan |
| Alamat | Alamat Lengkap |
| Nomor Telepon | Nomor telepon personal |
| Alamat E-Mail | Alamat E-Mail personal |

Tabel 1: Atribut dan Deskripsi Atribut

Buat implementasi dari method sebagai berikut :

1. Menyediakan accessor dan mutator method terhadap seluruh atribut

2. Constructor

1. **SOURCE CODE**

**.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | Mohon maaf masih dalam proses penyelesaian, karena saya kira pengumpulannya seperti biasa yakni hari Senin malam. Dan saya baru sampai Malang malam ini. |

**.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | Mohon maaf masih dalam proses penyelesaian, karena saya kira pengumpulannya seperti biasa yakni hari Senin malam. Dan saya baru sampai Malang malam ini. |

1. **PEMBAHASAN**

**.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | Mohon maaf masih dalam proses penyelesaian, karena saya kira pengumpulannya seperti biasa yakni hari Senin malam. Dan saya baru sampai Malang malam ini. |

**.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | Mohon maaf masih dalam proses penyelesaian, karena saya kira pengumpulannya seperti biasa yakni hari Senin malam. Dan saya baru sampai Malang malam ini. |

1. **SCREENSHOT PROGRAM**

**.java**

Mohon maaf masih dalam proses penyelesaian, karena saya kira pengumpulannya seperti biasa yakni hari Senin malam. Dan saya baru sampai Malang malam ini.

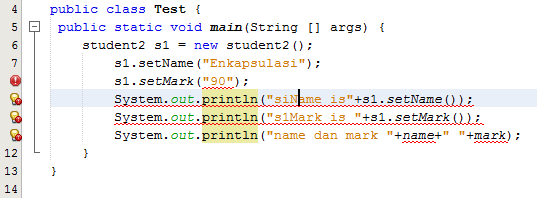
1. **PRAKTIKUM**

**A. Encapsulation 1**

Pertanyaan

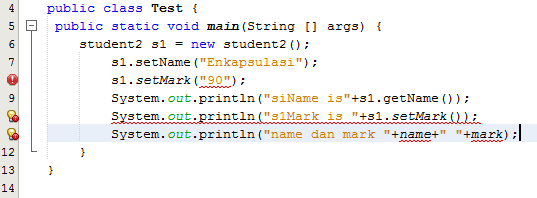
1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Jawab: Sudah terlaksana namun terdapat error pada kelas Test.



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

Jawab: Tidak ada tanda error, karena s1.setName memang bukan untuk memanggil method tersebut.



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

Jawab: Perintah perubahan baris tidak sesuai dengan sourcecode yang ada di modul.

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

Jawab:

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

Jawab: Tetap berfungsi selama access modifier yang digunakan adalah public, yang berarti memungkinkan semua kelas dapat mengakses meskipun berada pada paket yang berbeda,

**B. Encapsulation 2**

Pertanyaan

1. Method apakah yang menjadi accessor (getter) ?

Jawab:

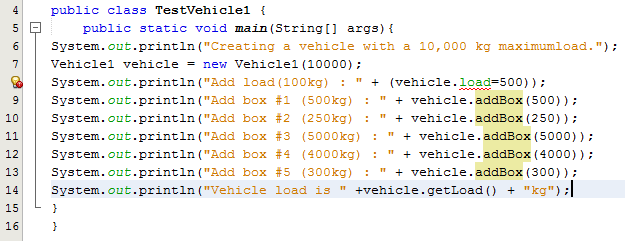
|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6. | public double getLoad(){  return this.load;  }  public double getMaxLoad(){  return this.maxLoad;  } |

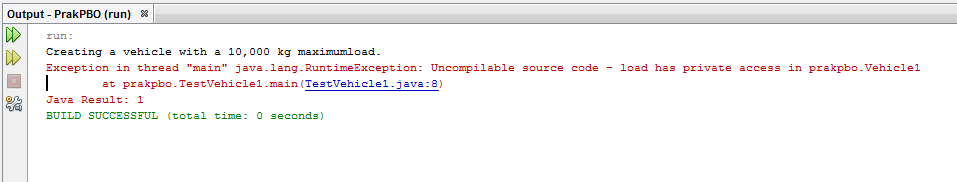
2. Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

Jawab: Tidak dapat dijalankan karena access modifier load bertipe private. 



1. **KESIMPULAN**

Encapsulation adalah suatu cara untuk menyembunyikan informasi detail dari suatu class.

Ketika sebuah program Java ingin diciptakan, dan user menginginkan beberapa data dilarang untuk diakses, pengimplementasi inilah yang sering disebut Access Modifiers. Kegunaan Access Modifiers ini untuk memberi hak akses kepada user. Tentu tidak semua data yang berada di dalam suatu kelas, atau turunannya dapat diakses, karena terdapat batasan-batasan yang berlaku.

Di dalam Java, terdapat 4 macam Access Modifiers, diantaranya:

1. Akses Default adalah tipe yang hanya class dalam package yang sama yang memiliki hak akses terhadap variabel dan methods dalam class. Contoh program sebagai berikut:

public class Car { //akses dasar terhadap variable  
 int name; //akses dasar terhadap metode  
 String getName(){  
 return name;}  
 }

2. Akses Public adalah class member untuk diakses baik dari dalam dan luar class. Sebagai contoh:

public class Car { //akses dasar terhadap variable  
 public int name; //akses dasar terhadap metode  
 public String getName(){  
 return name; }  
 }

3. Akses Private pengaksesan class hanya dapat diakses oleh class dimana tipe ini dibuat. Contoh kutipan program dari akses private:

public class Car { //akses dasar terhadap variable  
 private int name; //akses dasar terhadap metode private  
 String getName(){  
 return name; }  
 }

4. Akses Protected mengijinkan class member untuk diakses oleh method dalam class tersebut dan elemen – elemen subclass. Sebagai contoh:

public class Car { //akses dasar terhadap variable  
 private int name; //akses dasar terhadap metode  
 private String getName(){  
 return name; }  
 }