1. **DEFINISI MASALAH**

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 |

1. **SOURCE CODE**

|  |  |
| --- | --- |
|  | Encapsulation.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 | public class encapsulation {  private String nama;  private String alamat;  private String notlp;  private String email;  public encapsulation(){  nama = "";  alamat = "";  notlp = "";  email = "";  }  public void setNama(String n){  nama=n;  }  public String getNama(){  return nama;  }  public void setAlamat(String a){  alamat=a;  }  public String getAlamat(){  return alamat;  }  public void setNoTlp(String t){  notlp=t;  }  public String getNoTlp(){  return notlp;  }  public void setEmail(String e){  email=e;  }  public String getEmail(){  return email;  }  } |

|  |  |
| --- | --- |
| 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 | package Encapsulation;  import java.util.Scanner;  public class EtestE {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  encapsulation s[] = new encapsulation[5];  System.out.println("------- Selamat datang -------");  System.out.println("Silahkan masukan identitas lengkap anda : ");  for (int i = 0; i < 3; i++) {  s[i] = new encapsulation();  System.out.print("Nama Lengkap : ");  String nam = in.nextLine();  s[i].setNama(nam);  System.out.print("Alamat : ");  String al = in.nextLine();  s[i].setAlamat(al);  System.out.print("Nomor Telpon : ");  String no = in.nextLine();  s[i].setNoTlp(no);  System.out.print("E-mail : ");  String em = in.nextLine();  s[i].setEmail(em);  System.out.println("-------------------------");  }  }  } |

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?

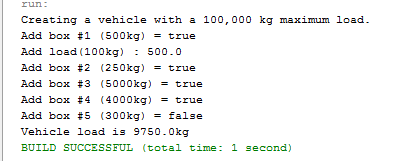
a. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

Jawab:



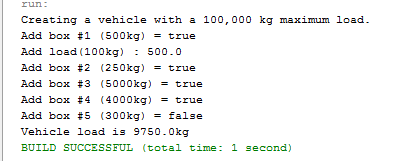
b. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

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

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

Jawab:

Program berjalan seperti pada soal nomor 3, karena modifier default ini dapat diakses pada package yang sama.

1. **KESIMPULAN**

Encapsulation adalah pembungkus variabel dan method dalam sebuah objek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut. Atau encapsulation untuk memastika pengguna sebuah objek tidak dapat mengganti keadaan dalam/dari sebuah objek dengan cara yang tidak layak, hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya.

Access modifier adalah batasan dan cara mengakses suatu properti dan/atau fungsi dari sebuah kelas. Dengan access modifier inilah salah satu tujuan belajar OOP dapat dicapai yaitu enkapsulasi data.

Macam-macam acces modifier yaitu,

* Public adalah kode akses yang bersifat umum. dengan kata lain, data maupun method dalam suatu kodingan tersebut dapat diakses oleh semua bagian di dalam program.
* Protected adalah kode akses yang membuat suatu data atau method yang didefinisikan dengan tingkatan akses ini dapat diakses oleh kelas yang memilikinya saja dan juga kelas-kelas yang memasih memiliki keturunan.
* Private adalah atribut atau fungsi yang diberi akses private hanya bisa diakses didalam kelas itu sendiri. Umumnya atribut yang menggambarkan sifat objek diberi akses private.
* Default digunakan dalam sebuah pernyataan switch untuk label blok pernyataan yang akan dieksekusi jika tidak ada case cocok dengan nilai yang ditentukan
* Package adalah tempat atau sarana mengelompokan dan pengorganisasian kelas-kelas menjadi suatu grup dalam library, sedangkan subclass adalah class yang diwarisi sifat-sifat dari superclassnya.