1. **DEFINISI MASALAH**

Tugas anda adalah membuat sebuah class yang memuat data – data pada buku alamat.

Table berikut mendefinisikan informasi yang di miliki oleh buku alamat.

|  |  |
| --- | --- |
| Atribut | Deskripsi |
| Nama | Nama Lengkap Perseorangan |
| Alamat | Alamat Lengkap |
| Nomor Telpeon | Nomor Telepon Personal |
| Alamat E-Mail | Alamat E-Mail Personal |

Table 1: Atribut dan Deskripsi Atribut

Buat implementasi dari method sebagai berikut:

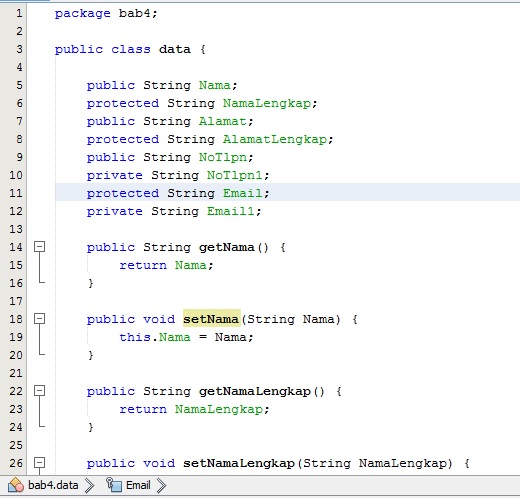
1. Menyediakan accessor dan mutator method terhadap seluruh atribut
2. Constructor
3. **SOURCE CODE**

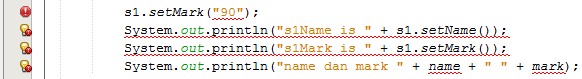
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  60  71  72  73  74  75  76  77  78  79 | package bab4;  public class data {  public String Nama;  protected String NamaLengkap;  public String Alamat;  protected String AlamatLengkap;  public String NoTlpn;  private String NoTlpn1;  protected String Email;  private String Email1;  public String getNama() {  return Nama;  }  public void setNama(String Nama) {  this.Nama = Nama;  }  public String getNamaLengkap() {  return NamaLengkap;  }  public void setNamaLengkap(String NamaLengkap) {  this.NamaLengkap = NamaLengkap;  }  public String getAlamat() {  return Alamat;  }  public void setAlamat(String Alamat) {  this.Alamat = Alamat;  }  public String getAlamatLengkap() {  return AlamatLengkap;  }  public void setAlamatLengkap(String AlamatLengkap) {  this.AlamatLengkap = AlamatLengkap;  }  public String getNoTlpn() {  return NoTlpn;  }  public void setNoTlpn(String NoTlpn) {  this.NoTlpn = NoTlpn;  }  public String getNoTlpn1() {  return NoTlpn1;  }  public void setNoTlpn1(String NoTlpn1) {  this.NoTlpn1 = NoTlpn1;  }  public String getEmail() {  return Email;  }  public void setEmail(String Email) {  this.Email = Email;  }  public String getEmail1() {  return Email1;  }  public void setEmail1(String Email1) {  this.Email1 = Email1;  }      } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| 1  3  5  6  7  8  9  10  11  12  13  14  15  18  19  20  23  24  27  28  31  32  33  36  37  40  41  44  45  46  49  50  53  54  57  58  61  62  65  66  69  60  73  74  77  78 | Deklarasikan pacage bab4  Deklarasikan class data  Dekalrasikan variable Nama yang bertipedata String dan memiliki accessor public  Dekalrasikan variable NamaLengkap yang bertipedata String dan memiliki accessor protected  Dekalrasikan variable alamat yang bertipedata String dan memiliki accessor public  Dekalrasikan variable Alamat Lengkap yang bertipedata String dan memiliki accessor protected  Dekalrasikan variable No Tlpn yang bertipedata String dan memiliki accessor public  Dekalrasikan variable No Tlpn1 yang bertipedata String dan memiliki accessor private  Dekalrasikan variable email yang bertipedata String dan memiliki accessor protected  Dekalrasikan variable email1 yang bertipedata String dan memiliki accessor private  Deklarasikan method getNama yang bertipe data String  Kembalikan nilai Nama;  Deklarasikan mehod set Nama yang bertipedatakan String dengan accessor public  Kembalikan nilai nama  Deklarasikan method getNamaLengkap yang bertipe data String  Kembalikan nilai NamaLengkap;  Deklarasikan mehod set Namalengkap yang bertipedatakan String dengan accessor public  Kembalikan nilai NamaLengkap  Deklarasikan method getAlamatyang bertipe data String  Kembalikan nilai Alamat  Deklarasikan mehod set Alamat yang bertipedatakan String dengan accessor public  Kembalikan nilai Alamat  Deklarasikan method getAlamatLengkap yang bertipe data String  Kembalikan nilai AlamatLengkap;  Deklarasikan mehod set Alamat lengkap yang bertipedatakan String dengan accessor public  Kembalikan nilai AlamatLengkap  Deklarasikan method getNoTlpn yang bertipe data String  Kembalikan nilai NoTlpn  Deklarasikan mehod set NoTlp yang bertipedatakan String dengan accessor public  Kembalikan nilai NoTlpn  Deklarasikan method getNoTlpn1 yang bertipe data String  Kembalikan nilai NoTlpn1  Deklarasikan mehod set NoTlpn1 yang bertipedatakan String dengan accessor public  Kembalikan nilai NoTlpn1  Deklarasikan method getEmail yang bertipe data String  Kembalikan nilai Email  Deklarasikan mehod set Email yang bertipedatakan String dengan accessor public  Kembalikan nilai Email  Deklarasikan method getEmail1 yang bertipe data String  Kembalikan nilai Email1  Deklarasikan mehod set Email1 yang bertipedatakan String dengan accessor public  Kembalikan nilai Email1 |

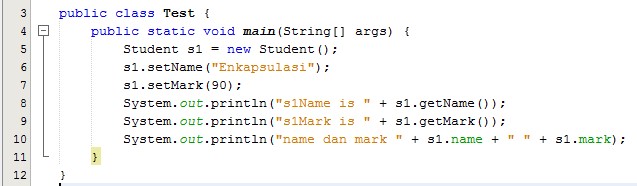
1. **SCREENSHOT PROGRAM**

****

1. **PRAKTIKUM**
2. Encapsulation
3. Laukan percobaan di atas dan benahi jika menemukan kesalahan!

Jawab: terjadi kesalahan pada Test.java

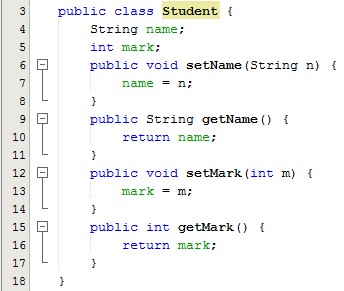
Pembetulanya:

Test.java

Pada Test.java terdapat beberapa pembetulan yaitu pada s1.setMark(“90”); di ubah menjadi s1.setMark(90) karena nilai yang di minta bertupe integer sedangkan tanda petik merupakan ciri dari tie string. Kemudian pada proses mencetak method yang di pangil kurang tepat seharusnya method yang di pangil getName() dan getMark() bukan set, karena pada method get terdapat pengembalian nilai.

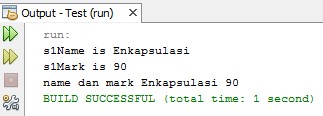
Pada name dan mark tidak dapat memangil dalam name dan mark karena tidak erdapat pendeklarasian objek yang di tuju.

Studend.java



Pada Student.java terdapat pembetulan pada pendeklarasian variabel private string nama dan private int mark menjadi string nama dan int mark karena jika menggunakan privat nilai dalam variabel nama dan mark tidak dapat diakses oleh class main.

Run



1. Jika pada baris 6 S1.setName di ubah menjadi s1.getName apa yg terjadi? Jelaskan!

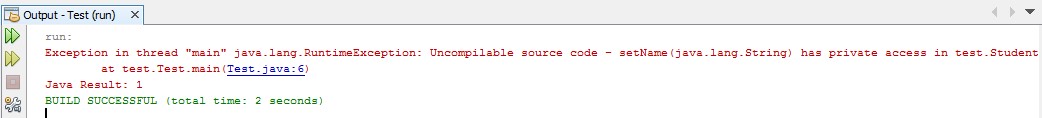
Jawab: Tidak terjadi kesalahan karena pada baris 6 bertujuan untuk menampilkan pernyataan yang berada di method getName sehingga memerlukan pengembalian nilai.

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

Jawab: Tidak terjadi kesalahan karena s1.setMark(“90”); pada baris 5 yang diubah menjadi s1.setMark(90) karena nilai yang diminta bertipe integer sedangkan tanda petik merupakan ciri dari tipe string.sedangkan pada baris 7 bertujuan untuk menampilkan pernyataan yang berada di method getName sehingga memerlukan pengembalian nilai.

1. Setelah di perbaiki, ubalah hak akses pada baris 4 (pada class Student) mejadi private apa yang terjadi jika class Test di jalankan? Jelaskan!

Jawab:

Ketika di run

Test.java



Karena private hanya dapat di akses pada class tersebut sedangkan pada class memberikan iputan pada method baris ke 4 pada class Student.

1. Jika kedua kelas di atas terdapat dalam package yang sama apakah konsep enkapsulasi tetap berfungsi? Jelaskan!

Jawab:

Masih bias karena masing masing menggunacan accessor yang bias di akses di pakage mereka masing masing

1. Encapsulation 2
2. Method apakah yang menjadi accessor (getter)?

Jawab:

* Method getLoad
* Method getMaxLoad
* Method addBox

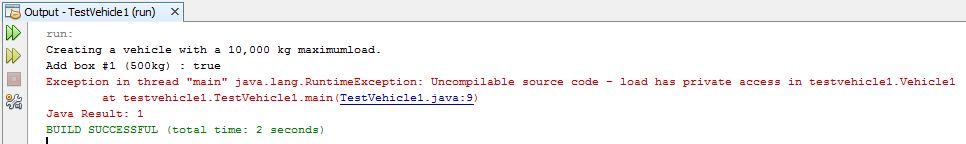
1. Tambahkan source code berikut di bawah baris ke 6 pada classTestVehicle1.

System.out.println(“add load(100kg): “ + (vhivle.load=500));

Jalankan program, apakah output dari program tersebut?

Kembaliak program seperti semula

Jawab:



Karena load pada class Vehicle1 memiliki hak akses private senhingga tidak dapat diakses dari class TestVehicle1.

1. Ubah tipe data pada atribut load dan maxload pada class Vihecle1menjadi public.

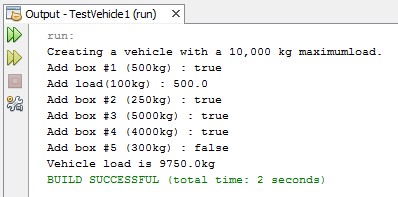
Jalankan program, apakah output dari program tersebut?

1. Tambahkan source kode bekut di bawah baris ke 6 pada class TestVehicle1.

System.out.println(“add load(100kg): “ + (vhivle.load=500))

Jalankan program, apakah output dari program tersebut?

Kembaliak program seperti semula

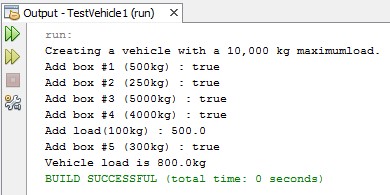


1. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.



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

Jawab:



1. Ulangi instuksi pada nomer 4 dengan mengubah tipe data pada atrubut load dan maxload pada class Vehicle1 menjadi default

Jawab:



1. **KESIMPULAN**

Encaptulation adalah konsep darsar dalam OOP yang bertujuan (mangkapsul) data dan metode dalam suatu class sehinga data dan metode yang tersembunyi dalam suatu class sehingga data dan method yang tersembunyi tersebut dapat di akses di luar class. Encaptulation juga direalisir dengan modifier : private dan data yang dapat di akses dari luar di beri modifier dengan public

Acces modifier adalah keyword yang di gunakan untuk menentukan spesifikasi tingkaat akses sebuah variable atau method (anggota kelas). Peng-aksesan yang di maksut berupa pengaksesan dalam kelas yang sama, turunan maupun dari kelas di mana variable dan method di deklarasikan. Dan ada beberapa macam access modifier yaitu :

Default : Access Modifier ini hanya menspesifikasikan kelas-kelas (classes) di paket yang sama yang bisa mengakses variable dan method suatu kelas. Anggota kelas dengan default access bisa melihat kelas lain dipaket yang sama. Tidak ada keyword khusus untuk mendeklarasikan modifier default access. Sehingga apabila tidak ada access modifier pada pendeklarasian tersebut, berarti yang dimaksud adalah default access.

Public : Suatu anggota kelas (class member) dideklarasikan sebagai public, maka anggota tersebut akan bisa digunakan oleh siapa saja, yaitu baik oleh kelas yang sama, kelas lain maupun lingkungan luar kelas. Sehingga anggota kelas (class member) yang dideklarasikan sebagai public akan bisa diakses oleh sembarang object lain.

Protected : Access modifier protected menspesifikasikan atau menentukan anggota kelas (class member) yang hanya bisa diakses oleh method-methodyang ada di dalam kelas yang sama dan juga dapat diakses oleh subclass (kelas turunan) dari kelas tersebut.

Ptivate : Access modifier private merupakan tingkatan akses yang sangat terbatas. Kemudian seperti modifier-modifier lain, modifier private menspesifikasikan anggota kelas (class member) yang hanya bisa diakses oleh kelas dimana anggota-anggota kelas tersebut dideklarasikan. Keadaan ini mengakibatkan bahwa tidak ada kelas lain yang bisa mengakses anggota kelas private, sekaligus termasuk subkelas-subkelas (subclassess)-nya

Access modifier private adalah yang paling terbatas, menspesifikasikan anggota kelas(variable dan method) hanya dapat diakses oleh kelas dimana anggota itu didefinisikan.