1. **DEFINISI MASALAH**

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 implementasidari method sebagai berikut :

1. Menyediakan accessor dan mutator method terhadap seluruh atribut

2. Constructor

1. **SOURCE CODE**

|  |  |
| --- | --- |
| ClassAlamat.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 | package TugasPraktikum;  public class ClassAlamat {  private String nama;  private String alamat;  private String nmertelpn;  private String email;  public Alamat(String nama, String alamat, String nmertelpn, String email){  this.nama = nama;  this.alamat = alamat;  this.nmertelpn = nmertelpn;  this.email = email;  }  public void setNama ( String nama){  this.nama = nama;  }  public String getNama(){  return nama;  }  public void setAlamat(String alamat){  this.alamat = alamat;  }  public String getAlamat(){  return alamat;  }  public void setNomertelepon(String nmertelpn){  this.nmertelpn = nmertelpn;  }  public String getNomertelepon (){  return nmertelpn;  }  public void setEmail(String email){  this.email = email;  }  public String getEmail(){  return email;  }  } |

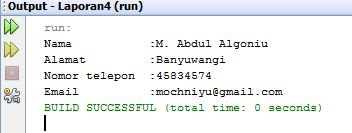
|  |  |
| --- | --- |
| MainMahasiswa.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package TugasPraktikum;  public class MainMahasiswa {  public static void main(String[] args) {  ClassAlamat a1 = new ClassAlamat("M. Abdul Algoniu", "Banyuwangi", "08955376982", "mochniyu@gmail.com");  a1.setNama("M. Abdul Algoniu");  System.out.println("Nama :" + a1.getNama());  a1.setAlamat("Banyuwangi");  System.out.println("Alamat :" + a1.getAlamat());  a1.setNomertelepon("45834574");  System.out.println("Nomor telepon :" + a1.getNomertelepon());  a1.setEmail("mochniyu@gmail.com");  System.out.println("Email :" + a1.getEmail());  }  } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| ClassAlamat.java | |
| 1  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 | Sebuah package bernama ClassAlamat  Sebuah class yang bernamaAlamat yang bersifat public  Sebuah variable bernama nama, bertipe data String dan bersifat private  Sebuah variable bernama alamat, bertipe data String dan bersifat private sebuah variable bernama nmertelepn, bertipe data String dan bersifat private  Sebuah variable bernama email, bertipe data String danbersifat private  Sebuah class bernama alamat yang bersifat constructor, yaitu String nama, String alamat, String nmertelpn, String email  Maksutdari this. Nama adalah diisi dengan nama yang ada di variable  Maksutdari this. Alamat adalah diisi dengan alamat yang ada di variable  Maksutdari this. Nmertelpn adalah diisi dengan nmertelpn yang ada di variable  Maksutdari this. Email adalah diisi dengan email yang ada di variable  Kurung penutup method alamat  Sebuah method yang bernama setNama bersifat void yang artinya tidak mengembalikan nilai dan berparameter String nama  Maksutnya adalah diisi dengan nama yang telahdiisikan  Kurung penutup method setNama  Sebuah method yang bernama getNama bersifat String  Maksut dari return nama adalah mengembalikan nilai nama  Kurung penutup method getNama  Sebuah method yang bernama setAlamat bersifat void yang artinya tidak mengembalikan nilai dan berparameter String alamat  Maksutnya adalah diisi dengan alamat yang telah diisikan  Kurung penutup method setalamat  Sebuah method yang bernama getAlamat bersifat String  Maksutdari return Alamat adalah mengembalikan nilai Alamat  Kurung penutup method getAlamat  Sebuah method yang bernama setNomertelepon bersifat void yang artinya tidak mengembalikan nilai dan berparameter String nmertelpn  Maksutnya adalah diisi dengan nmertelpn yang telah diisikan  Kurung penutup method setNomertelepon  Sebuah method yang bernama getNomertelepon bersifat String  Maksut dari return nmertelpn adalah mengembalikan nilai nmertelpn  Kurung penutup method getNomertelepon  Sebuah method yang bernama setEmail bersifat void yang artinya tidak mengembalikan nilai dan berparameter String email  Maksutnya adalah diisi dengan email yang telah diisikan  Kurung penutup method setEmail  Sebuah method yang bernama getEmail bersifat String  Maksut dari return email adalah mengembalikan nilai email  Kurung penutup method getEmail  Kurung penutup class |

|  |  |
| --- | --- |
| MainMahasiswa.java | |
| 1  3  4  5  6  7  8  9  10  11  12  13  14  15 | Sebuah package bernama TugasPraktikum  Sebuah class yang bernama MainMahasiswa yang bersifat public  Public static void main(String[] args) adalah fungsi main dimana public pada bagian ini berarti bahwa metode-metode dapat dipanggil didalam dan juga diluar kelas. Static sendiri berarti sama untuk seluruh instant dari kelas dan void sendiri berarti bahwa metode tidak akan mengirim apapun setelah selesainya. Fungsi main ini berisi argument yang diambil dari ruang eksekusi.  Maksutnya adalah penginstansiasi class melalui objek bernama a1, karena constructor maka langsung ditulis isinya berupa M. Abdul Algoniu, Banyuwangi, 08955376982, mochniyu@gmail.com  Maksutnya objek dari a1 memanggil method setNama dan diisi dengan M. Abdul Algoniu  Artinya menampilkan hasil output yaitu Nama + diisi dengan pengembalian nilainya dengan getNama  Maksutnya objek dari a1 memanggil method setAlamat dan diisi dengan Banyuwangi  Artinya menampilkan hasil output yaitu Alamat + diisi dengan pengembalian nilainya dengan getAlamat  Maksutnya objek dari a1 memanggil method setNomertelepon dan diisi dengan 08955376982  Artinya menampilkan hasil output yaitunomertelepon + diisi dengan pengembalian nilainya dengan getNomertelepon  Maksutnya objek dari a1 memanggil method setEmail dan diisi dengan mochniyu@gmail.com  Artinya menampilkan hasil output yaitu Email + diisi dengan pengembalian nilainya dengan getEmail  Kurung penutup class  Kurung penutup main |

1. **SCREENSHOT PROGRAM**

****

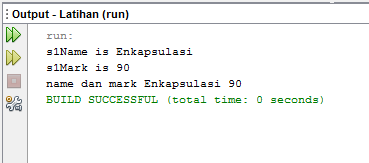
1. **PRAKTIKUM**

**Encapsulation 1**

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Terjadi kesalahan pada sintax berikut

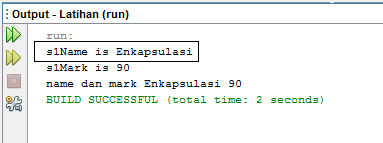
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | s1.setName("Enkapsulasi");  s1.setMark("90");**//seharusnya tidak mengunakan tanda petik ganda karna setMark bertipe int.**  **S1.setMark(90);//seperti ini yang benar.**  System.out.println("s1Name is "+s1.setName());  //s1 bukan memanggil setNama, tapi seharusnya memanggil getName karna disini akan diprint.  **System.out.println(“s1Name is ”+s1.getName());**//seperti ini yang benar.  System.out.println("s1Mark is "+s1.setMark());  //s1 bukan memanggil setMark, tapi seharusnya memanggil getMark karna disini akan diprint.  **System.out.println(“s1Mark is”+s1.getMark());**//seperti ini yang benar.  System.out.println("name dan mark "+name+" "+mark);  //disini seharunya memanggil getName dan getMark.  **System.out.println(“name dan mark ”+getName+)””+getMark());**  //seperti ini yang benar |



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

|  |  |
| --- | --- |
| 1  2  3  4 | System.out.println("s1Name is "+s1.setName());  //Di rubah menjadi  System.out.println("s1Name is "+s1.getName**());** |

Maka programnya akan benar karna pada baris 6 s1.setName adalah salah, yang benar s1.getName, dan akan negeprint nama yaitu “Enkapsulasi”.

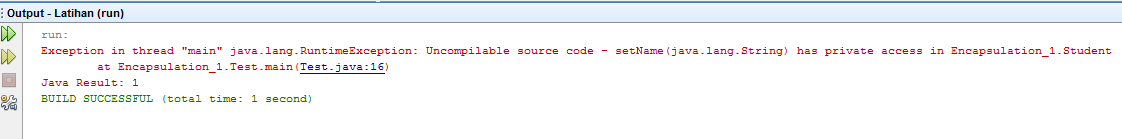


1. Lakukan perubahan pada baris 5 dengan menghilangkan String di tanda dalam kurung lalu ubah baris 7 seperti no. 2 apa yang terjadi? jelaskan!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | public class Test {  public static void main(String [] args) {  Student s1=new Student();  s1.setName("Enkapsulasi");  s1.setMark("90");// akan dihapus tanda petik gandanya, menjadi  **s1.setMark(90);//maka program akan berjalan karna Mark bertipe data int.**  System.out.println("s1Name is "+s1.setName()); System.out.println("s1Mark is "+s1.setMark());//dirubah seperti nomer 2.  **System.out.println(“s1Mark is “+s1.getMark());//maka program akan berjalan karna pada source code sebelum dirubah seperti ini adalah salah.**  System.out.println("name dan mark "+name+" "+mark);  }  } |

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public void setName(String n){  name=n;  }  **//Di rubah menjadi**  **Private void setName(String n){**  **name=n;**  **}** |

Maka program tidak akan berjalan karna pada class main setName tidak dapat dipanggil sehingga program error. Access modifier private tidak dapat dipanggil di class yang lain.

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

Ya, karena Enkapsulasi bertujuan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan oleh program lain.

**Ecapsulation 2**

1. Method apakah yang menjadi accessor (getter) ?

Method yang mempunyai nilai return. Yaitu sperti method :

public double getLoad(){

return this.load;

}

public double getMaxLoad(){

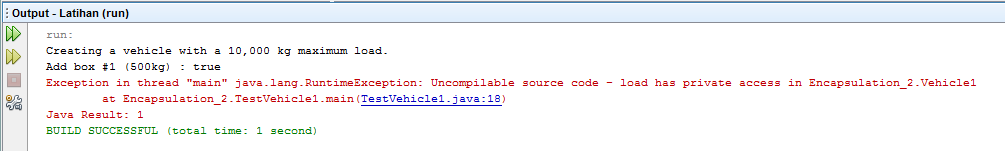
return this.maxLoad;

}

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.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public class TestVehicle1 {  public static void main(String[] args) {  System.out.println("Creating a vehicle with a 10,000 kg maximum load.");  Vehicle1 vehicle = new Vehicle1(10000);  System.out.println("Add box #1 (500kg) : " + vehicle.addBox(500));  System.out.println("Add load(100kg) : " + (vehicle.load=500)); |

Program akan error karna pada class “Vehiclel” atribut “load” access modifiernya private sehingga tidak bisa di panggil di kelas yang lain.



1. Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi public. Jalankan program, apakah output dari program tersebut?

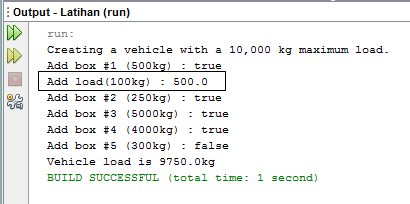
a. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1. System.out.println("Add load(100kg) : " + (vehicle.load=500)); Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.

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.

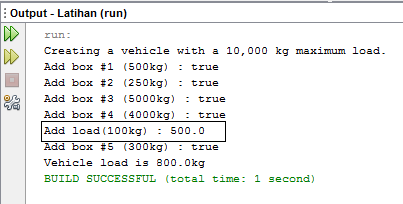
|  |  |
| --- | --- |
| 1  2  3  4 | private double load, maxLoad;  **//menjadi**  **public double load, maxLoad;** |

Progam akan berjalan karna atribut load dan maxload akan bisa di access oleh class lain.

a.



b.

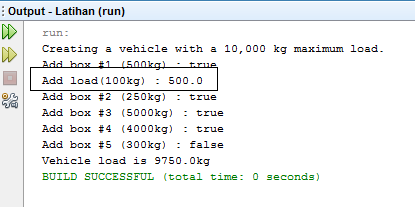


1. Ulangi instruksi pada nomer 3 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

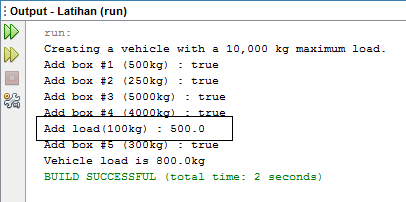
|  |  |
| --- | --- |
| 1  2  3  4 | private double load, maxLoad;  **//di rubah menjadi**  **protected double load, maxLoad;** |

Jika atribut load dan maxload access modifiernya dirubah menjadi protected program akan berjalan karna protected bisa di access di kelas yang lain.

a.



b.

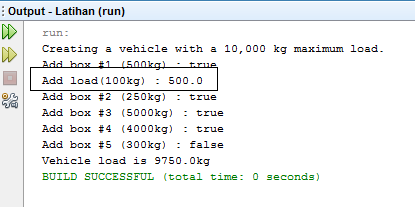


1. Ulangi instruksi pada nomer 3 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi default**.**

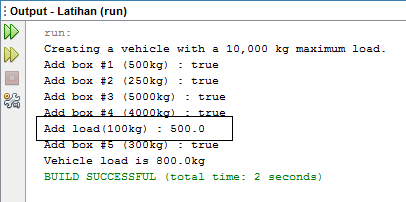
|  |  |
| --- | --- |
| 1  2  3  4 | private double load, maxLoad;  **//di rubah menjadi**  **double load, maxLoad;** |

Jika atribut load dan maxload access modifiernya dirubah menjadi default, program akan tetep berjalan, karna default juga bisa di access di kelas lain.

a.



b.



1. **KESIMPULAN**

* Enkapsulasi adalah suatu cara untuk menyembunyikan informasi detail dari suatu class. Dalam enkapsulasi terdapat hak akses public, protected, dan private. Hak akses public memungkinkan semua kelas dapat mengakses meskipun berada pada paket yang berbeda, hak akses protected hanya diberikan kepada kelasnya sendiri dan turunannya, serta kelas kelas dalam satu paket. Sedangkan private hanya boleh diakses oleh kelasnya sendiri. Enkapsulasi bertujuan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain.Enkapsulasi memiliki manfaat sebagai Modularitas, dan Information Hiding.
* Cara penggunaan encapsulations harus mempunyai / memakai variable pembungkus
* Ketika kita menggunakan pembungkus dan untuk mengakses menggunakan setter dan getter
* Kelebihan Encapsulation

1. Bersifat independen. Suatu modul yang terenkapsulasi dengan baik akan bersifat independen dari yang lain. Sehingga dapat digunakan pada bagian manapun dari program.
2. Bersifat transparan. Jika Anda melakukan modifikasi pada suatu modul, maka perubahan tersebut akan dirasakan oleh semua bagian yang menggunakan modul tersebut.
3. Menghindari dari efek yang diluar perencanaan. Modul yang terenkapsulasi dengan baik hanya akan berinteraksi dengan bagian program melalui variable input dan output yang telah didefinisikan sebelumnya. Sehingga dapat mengurangi kemungkinan bug.