1. **DEFINISI MASALAH**
2. Soal 1

Tugas Anda adalah membuat sebuah class yang memuat data-data pada buku alamat. Tabel berikut mendefinisikan informasi yang dimiliki oleh buku alamat.

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

|  |  |
| --- | --- |
|  | BukuAlamat.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  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | package tugas.pkg4;  import java.util.Scanner;  public class BukuAlamat {  private String nama, alamat, telepon, email;  Scanner input = new Scanner(System.in);  public BukuAlamat (){  nama = "";  alamat = "";  telepon = "";  email = "";  }  public BukuAlamat (String n, String a, String t, String e){  nama = n;  alamat = a;  telepon= t;  email = e;  }  public String getNama() {  return nama;  }  public void setNama() {  System.out.print("Nama\t: ");  nama = input.next();  }  public String getAlamat() {  return alamat;  }  public void setAlamat() {  System.out.print("Alamat\t: ");  alamat = input.next();  }  public String getTelepon() {  return telepon;  }  public void setTelepon() {  System.out.print("Telepon\t: ");  telepon = input.next();  }  public String getEmail() {  return email;  }  public void setEmail() {  System.out.print("E-Mail\t: ");  email = input.next();  }  public static void displayBuku (BukuAlamat[] buku){  for (int i = 0; i < buku.length; i++){  System.out.println("Data Ke-"+(i+1));  if(buku[i].getNama().equals(""))  System.out.println("Deleted");  else {  System.out.println("Nama\t : "+buku[i].getNama()+"\t");  System.out.println("Alamat\t : "+buku[i].getAlamat()+"\t");  System.out.println("Telepon\t : "+buku[i].getTelepon()+"\t");  System.out.println("E-Mail\t : "+buku[i].getEmail()+"\t");  System.out.println("");  }  }  }  public static void inputBuku (BukuAlamat[] buku, int i){  buku[i] = new BukuAlamat();  buku[i].setNama();  buku[i].setAlamat();  buku[i].setTelepon();  buku[i].setEmail();  System.out.println("");  }  public static void deleteBuku (BukuAlamat[] buku, int x){  --x;  buku[x] = new BukuAlamat();  System.out.println("");  }  public static void editBuku (BukuAlamat[] buku, int x){  --x;  buku[x] = new BukuAlamat();  buku[x].setNama();  buku[x].setAlamat();  buku[x].setTelepon();  buku[x].setEmail();  System.out.println("");  }  } |

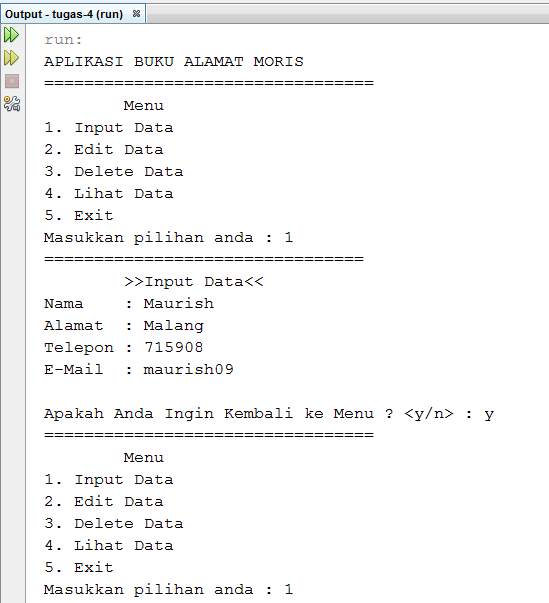
|  |  |
| --- | --- |
|  | BukuMain.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | package tugas.pkg4;  import java.util.Arrays;  import java.util.Scanner;  public class BukuMain {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  BukuAlamat[] buku = new BukuAlamat[0];  int i=0;  System.out.println("APLIKASI BUKU ALAMAT MORIS");  for (int j = 1; j > 0; j++){  System.out.println("=================================");  System.out.println("\tMenu");  System.out.println("1. Input Data");  System.out.println("2. Edit Data");  System.out.println("3. Delete Data");  System.out.println("4. Lihat Data");  System.out.println("5. Exit");  System.out.print("Masukkan pilihan anda : ");  int pil = input.nextInt();  System.out.println("================================");  switch(pil){  case 1 :  System.out.println("\t>>Input Data<<");  buku = Arrays.copyOf(buku, buku.length + 1);  BukuAlamat.inputBuku(buku, i);  i++;  break;  case 2 :  System.out.println("\t>>Edit Data<<");  BukuAlamat.displayBuku(buku);  System.out.print("Edit data nomor : ");  i = input.nextInt();  if ((i-1)<buku.length){  BukuAlamat.editBuku(buku, i);  } else  System.out.println("Data tidak tepat");;  break;  case 3 :  System.out.println("\t>>Delete Data<<");  BukuAlamat.displayBuku(buku);  System.out.print("Delete data nomor : ");  i = input.nextInt();  BukuAlamat.deleteBuku(buku, i);  break;  case 4 :  System.out.println("\t>>Lihat Data<<");  BukuAlamat.displayBuku(buku);  break;  case 5 :  System.out.println("Terimakasih telah menggunakan Program ini");  System.exit(0);  break;  default :  System.out.println("Pilihan yang anda masukkan salah");  }  System.out.print("Apakah Anda Ingin Kembali ke Menu ? <y/n> : ");  String y = input.next();  if ("y".equals(y)){  } else if ("n".equals(y)){  System.out.println("TERIMA KASIH");  System.exit(0);  }  }  }  } |

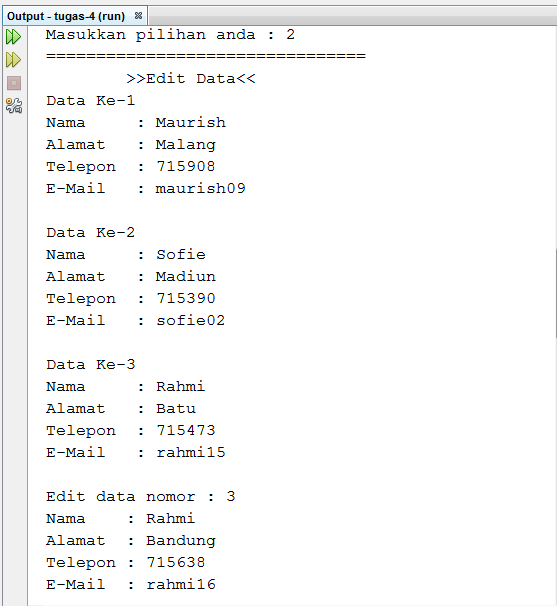
1. **PEMBAHASAN**

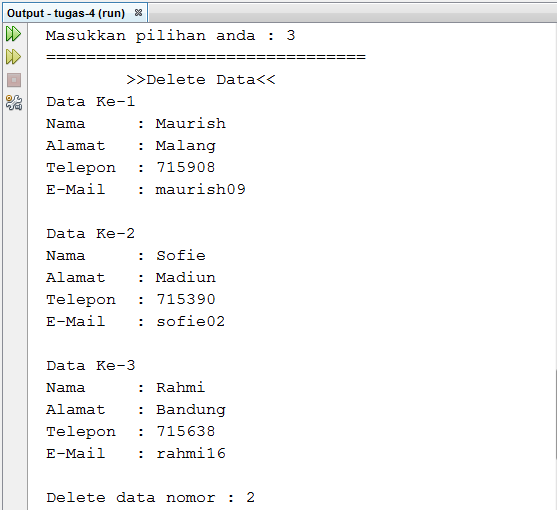
|  |  |
| --- | --- |
|  | BukuAlamat.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | Perbaris bukan paragraf memakai font times new roman 11 |

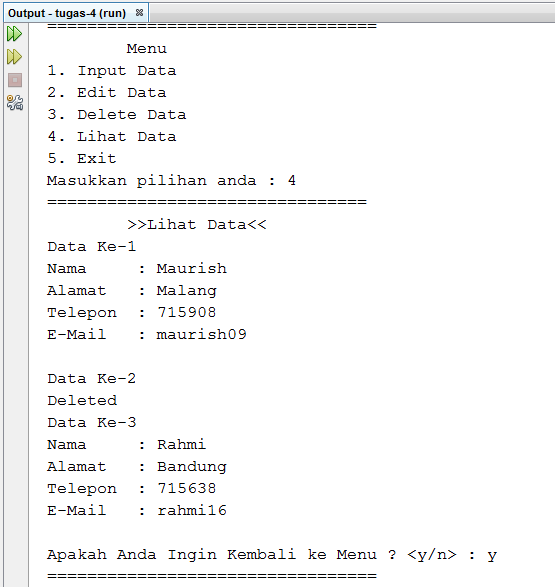
|  |  |
| --- | --- |
|  | BukuMain.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | Perbaris bukan paragraf memakai font times new roman 11 |

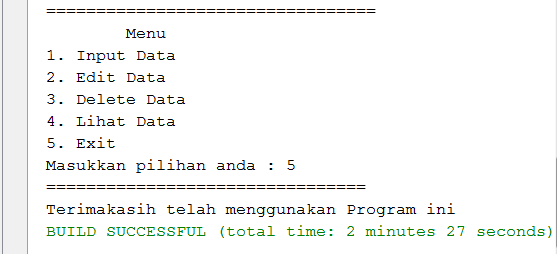
1. **SCREENSHOT PROGRAM**





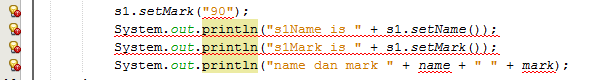






1. **PRAKTIKUM**
2. **Encapsulation 1**
3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan !

***Jawab :*** Terdapat eror karena terjadi ketidak tepatan tipe data pada class yang dipanggil



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

***Jawab :*** Tidak akan terjadi error, karena dengan menggunakan acessor (method get) akan dapat mengambil nilai dari kelas Student.

|  |  |
| --- | --- |
|  | Test.java |
| 6 | System.out.println("s1Name is " + s1.getName()); |

1. Lakukan perubahan pada baris 5 dengan menghilangkan String di tanda dalam kurung lalu ubah baris 7 seperti nomor 3 apa yang terjadi? Jelaskan!

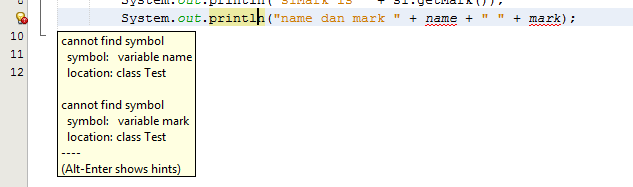
***Jawab :*** Pada baris 5 tidak akan terjadi error karena method setMark di class Student berparameter integer jadi harus menerima inputan bertipe integer. Pada baris 7 tidak akan error karena dengan menggunakan accessor (method get) akan dapat mengambil nilai dari kelas Student

|  |  |
| --- | --- |
|  | Test.java |
| 5  6  7 | s1.setMark(90);  System.out.println("s1Name is " + s1.getName());  System.out.println("s1Mark is " + s1.getMark()); |

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

***Jawab :*** Terjadi error di class Test, karena baris 4 di kelas Student bertipe private maka tidak bisa akan bisa dipanggil di kelas lain melainkan hanya bisa dipangil di class Sudent

|  |  |
| --- | --- |
|  | Student.java |
| 4  5  6 | private void setName(String n) {  name = n;  } |



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

***Jawab :*** Tetap berfungsi jika hak akses bertipe public, karena hak akses public dapat diakses semua kelas meskipun berbeda package dan juga hak akses protected dapat diakses di kelasnya sendiri, turunannya dan juga class yang berada dalam satu package, kemudian jika hak aksesnya default, dapat diakses oleh classnya dan juga class lain yang dalam satu package.Namun jika hak akses bertipe private, berarti hanya classnya sendiri yang dapat mengaksesnya.

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

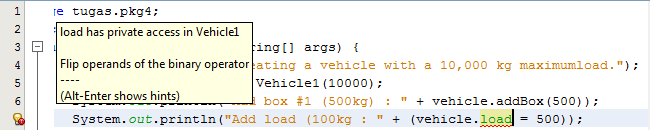
***Jawab :*** Method getLoad() , getMaxLoad() , addBox() , yaitu method yang dapat mengambil atau merubah nilai dari suatu variable.

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 :*** Program eror, karena hak akses variabel load merupakan hak akses private



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

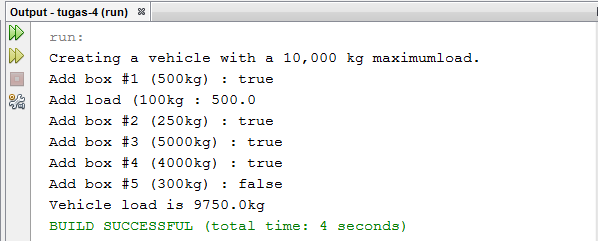
a. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.

***Jawab :***

|  |  |
| --- | --- |
|  | TestVehicle1.java |
| 7 | System.out.println("Add load(100kg) : " + (vehicle.load=500)); |



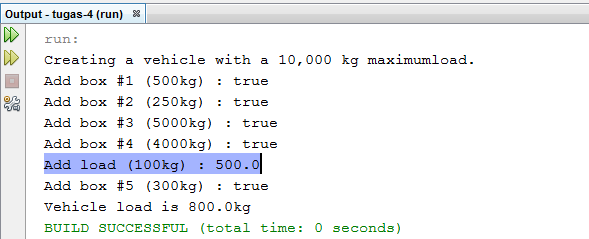
b. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.

***Jawab :***

|  |  |
| --- | --- |
|  | TestVehicle1.java |
| 13 | System.out.println("Add load(100kg) : " + (vehicle.load=500)); |



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

***Jawab :*** Program dapat dijalankan, output yang didapatkan sama seperti nomor 3

|  |  |
| --- | --- |
|  | Vehicle1.java |
| 2  3 | public class Vehicle1 {  protected double load, maxLoad; |

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

***Jawab :*** Program dapat dijalankan, output yang didapatkan sama seperti nomor 3

|  |  |
| --- | --- |
|  | Vehicle1.java |
| 2  3 | public class Vehicle1 {  double load, maxLoad; |

1. **KESIMPULAN**

Enkapsulasi adalah pembungkus data, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain. Konsep enkapsulasi sangat penting dilakukan untuk menjaga kebutuhan program agar dapat diakses sewaktu-waktu, sekaligus menjaga program tersebut.

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 variabel

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 variabel

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 variabel

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 variabel

private int name; //akses dasar terhadap metode

private String getName(){

return name; }

}