1. **DEFINISI MASALAH**
2. 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
3. **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  83  84  85  86  87  88  89  90  91  92  93  94 | import java.util.\*;  public class BukuAlamat {  Scanner scan = new Scanner(System.in);  private String Nama;  private String Alamat;  private String NomorTelepon;  private String Email;  public BukuAlamat() {  this.Nama = "";  this.Alamat = "";  this.NomorTelepon = "";  this.Email = "";  }  public BukuAlamat(String nama, String alamat, String  nomorTelepon, String email) {  this.Nama = nama;  this.Alamat = alamat;  this.NomorTelepon = nomorTelepon;  this.Email = email;  }  public void setNama() {  System.out.print("Nama\t: ");  this.Nama = scan.nextLine();  }  public void setAlamat() {  System.out.print("Alamat\t: ");  this.Alamat = scan.nextLine();  }  public void setNomorTelepon() {  System.out.print("Telepon\t: ");  this.NomorTelepon = scan.nextLine();  }  public void setEmail() {  System.out.print("Email\t: ");  this.Email = scan.nextLine();  }  public String getNama() {  return Nama;  }  public String getAlamat() {  return Alamat;  }  public String getNomorTelepon() {  return NomorTelepon;  }  public String getEmail() {  return Email;  }  public static void cetakBukuAlamat(BukuAlamat[] bukualamat) {  System.out.println("\nNO\tNAMA\t\t\tALAMAT  LENGKAP\t\t\tNOMOR TELEPON\t\tEMAIL");  System.out.println("------------------------------------");  for (int x = 0; x < bukualamat.length; x++) {  System.out.print((x+1)+"\t");  if(bukualamat[x].getNama().equals(""))  System.out.print("Dihapus\n");  else{  System.out.print(bukualamat[x].getNama() + "\t");  System.out.print(bukualamat[x].getAlamat() + "\t");  System.out.print(bukualamat[x].getNomorTelepon() +  "\t");  System.out.print(bukualamat[x].getEmail() + "\n");}  }  System.out.println("");  }  public static void inputBukuAlamat(BukuAlamat[] bukualamat,  int x){  System.out.println("\nINPUT DATA BARU");  bukualamat[x] = new BukuAlamat();  bukualamat[x].setNama();  bukualamat[x].setAlamat();  bukualamat[x].setNomorTelepon();  bukualamat[x].setEmail();  System.out.println("");  }  public static void deleteBukuAlamat(BukuAlamat[] bukualamat,  int y){  System.out.println("\nDATA BERHASIL DIDELETE");  --y;  bukualamat[y] = new BukuAlamat();  System.out.println("");  }  public static void editBukuAlamat(BukuAlamat[] bukualamat,  int y){  System.out.println("\nEDIT DATA BUKU");  --y;  bukualamat[y] = new BukuAlamat();  bukualamat[y].setNama();  bukualamat[y].setAlamat();  bukualamat[y].setNomorTelepon();  bukualamat[y].setEmail();  System.out.println("");  }  } |

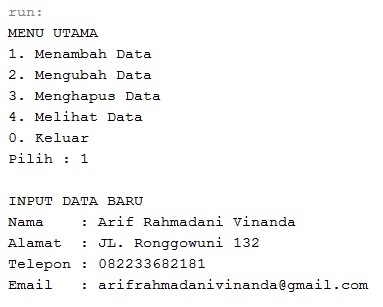
|  |  |
| --- | --- |
| **Main.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | import java.util.\*;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  BukuAlamat[] bukualamat = new BukuAlamat[0];  String loop = "y";  int x = 0, i = 0;  do {  cetakMenu();  switch (scan.next()) {  case "1":  bukualamat = Arrays.copyOf(bukualamat,  bukualamat.length + 1);  BukuAlamat.inputBukuAlamat(bukualamat, x);  x++;  break;  case "2":  BukuAlamat.cetakBukuAlamat(bukualamat);  edit();  i = scan.nextInt();  if((i-1)<bukualamat.length){  BukuAlamat.editBukuAlamat(bukualamat,  i);}  else dataNull();  break;  case "3":  BukuAlamat.cetakBukuAlamat(bukualamat);  hapus();  i = scan.nextInt();  yakinHapus();  if (scan.next().equals("y")&&(i-1)<bukualamat.length)  {  BukuAlamat.deleteBukuAlamat(bukualamat, i);  }  else dataNull();  break;  case "4":  BukuAlamat.cetakBukuAlamat(bukualamat);  break;  case "0":  loop = "n";  break;  }  } while (loop.equals("y"));  }  public static void cetakMenu() {  System.out.println("MENU UTAMA");  System.out.println("1. Menambah Data");  System.out.println("2. Mengubah Data");  System.out.println("3. Menghapus Data");  System.out.println("4. Melihat Data");  System.out.println("0. Keluar");  System.out.print("Pilih : ");  }  public static void hapus() {  System.out.print("Hapus nomor ke? ");  }  public static void edit() {  System.out.print("Edit nomor ke? ");  }  public static void yakinHapus() {  System.out.print("Yakin data dihapus(y/n)?");  }  public static void dataNull(){  System.out.println("Data tidak ada\n");  }  } |

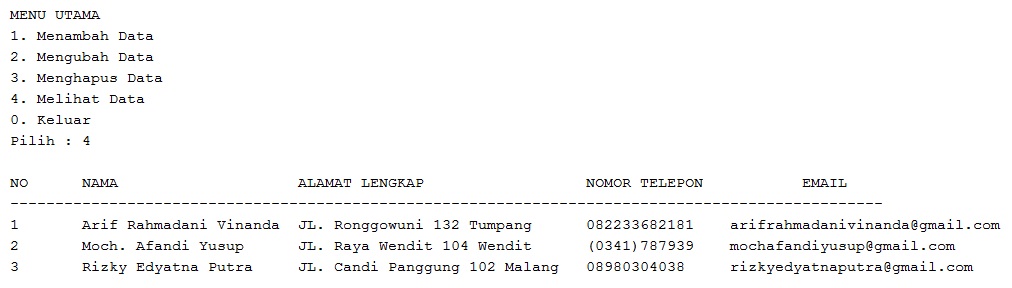
1. **PEMBAHASAN**

|  |  |
| --- | --- |
| **BukuAlamat.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  14  16  17  18  19  21  22  23  25  26  27  29  30  31  33  34  35  37  38  40  41  43  44  46  47  48  49  50  52  53  54  55  56  57  58  59  60  62  64  66  68  69  70  71  72  73  74  76  78  79  80  81  83  85  86  87  88  89  90  91  92 | Proses memanggil semua class dari package java.util.  Deklarasi class dengan nama BukuAlamat.  Pendeklarasian objek scan dari class Scanner.  Inisialisasi variable Nama sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable Alamat sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable NomorTelepon sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable Email sebagai atribut bertipe data String dengan access modifier private.  Pendeklarasian constructor BukuAlamat tanpa parameter.  Proses pengisian variable Nama dengan “ ”.  Proses pengisian variable Alamat dengan “ ”.  Proses pengisian variable NomorTelepon dengan “ ”.  Proses pengisian variable Email dengan “ ”.  Pendeklarasian constructor BukuAlamat dengan parameter nama, alamat, nomorTelepon, email bertipe data String.  Proses pengisian variable Nama dengan parameter nama.  Proses pengisian variable Alamat dengan parameter alamat.  Proses pengisian variable NomorTelepon dengan parameter nomorTelepon.  Proses pengisian variable Email dengan parameter email.  Pendeklarasian method setNama sebagai mutator tanpa pengembalian nilai (void) beraccess modifier public.  Proses mencetak “Nama : ”.  Proses pengisian variable Nama dengan inputan dari user bertipe data String.  Pendeklarasian method setAlamat sebagai mutator tanpa pengembalian nilai (void) beraccess modifier public.  Proses mencetak “Alamat : ”.  Proses pengisian variable Alamat dengan inputan dari user bertipe data String.  Pendeklarasian method setNomorTelepon sebagai mutator tanpa pengembalian nilai (void) beraccess modifier public.  Proses mencetak “Telepon : ”.  Proses pengisian variable NomorTelepon dengan inputan dari user bertipe data String.  Pendeklarasian method setEmail sebagai mutator tanpa pengembalian nilai (void) beraccess modifier public.  Proses mencetak “Email : ”.  Proses pengisian variable Email dengan inputan dari user bertipe data String.  Pendeklarasian method getNama sebagai accessor dengan pengembalian nilai bertipe data String beraccess modifier public.  Proses pengembalian nilai ke method getNama dari variable Nama.  Pendeklarasian method getAlamat sebagai accessor dengan pengembalian nilai bertipe data String beraccess modifier public.  Proses pengembalian nilai ke method getAlamat dari variable Alamat.  Pendeklarasian method getNomorTelepon sebagai accessor dengan pengembalian nilai bertipe data String beraccess modifier public.  Proses pengembalian nilai ke method getNomorTelepon dari variable NomorTelepon.  Pendeklarasian method getEmail sebagai accessor dengan pengembalian nilai bertipe data String beraccess modifier public.  Proses pengembalian nilai ke method getEmail dari variable Email.  Pendeklarasian method cetakBukuAlamat tanpa pengembalian nilai (void) bertipe static dengan parameter array bukualamat bertipedata arrayofobject BukuAlamat.  Proses mencetak “NO NAMA ALAMAT LENGKAP NOMOR TELEPON EMAIL”.  Proses mencetak “----------------------------------------------------------------------------”.  Inisialisasi perulangan for dengan variable x bertipe data integer bernilai 0. Perulangan akan terus berulang hingga x kurang dari panjang array bukualamat dan nilai x akan bertambah 1 setiap perulangan.  Proses mencetak nilai x ditambah dengan 1.  Pengecekan apakah nilai dari method getNama berisi “ ”.  Proses mencetak “Dihapus”.  Fungsi yang dijalankan apabila fungsi diatas tidak terpenuhi.  Proses mencetak nilai dari method accessor getNama.  Proses mencetak nilai dari method accessor getAlamat.  Proses mencetak nilai dari method accessor getNomorTelepon.  Proses mencetak nilai dari method accessor getEmail.  Proses mencetak “ ” (enter).  Pendeklarasian method inputBukuAlamat tanpa pengembalian nilai (void) bertipe static dengan parameter array bukualamat bertipedata arrayofobject BukuAlamat dan variable x bertipe data integer.  Proses mencetak “INPUT DATA BARU”.  Inisialiasi objek bukualamat indeks ke x dari class BukuAlamat.  Proses pemanggilan method mutator setNama.  Proses pemanggilan method mutator setAlamat.  Proses pemanggilan method mutator setNomorTelepon.  Proses pemanggilan method mutator setEmail.  Proses mencetak “ ” (enter).  Pendeklarasian method deleteBukuAlamat tanpa pengembalian nilai (void) bertipe static dengan parameter array bukualamat bertipedata arrayofobject BukuAlamat dan variable y bertipe data integer.  Proses mencetak “DATA BERHASIL DIDELETE”.  Proses pengurangan nilai variable y dengan angka 1.  Inisialisasi objek bukualamat indeks ke y dari class BukuAlamat.  Proses mencetak “ ” (enter).  Pendeklarasian method editBukuAlamat tanpa pengembalian nilai (void) bertipe static dengan parameter array bukualamat bertipedata arrayofobject BukuAlamat dan variable y bertipe data integer.  Proses mencetak “EDIT DATA BUKU”.  Proses pengurangan nilai variable y dengan angka 1.  Inisialisasi objek bukualamat indeks ke y dari class BukuAlamat.  Proses pemanggilan method mutator setNama.  Proses pemanggilan method mutator setAlamat.  Proses pemanggilan method mutator setNomorTelepon.  Proses pemanggilan method mutator setEmail.  Proses mencetak “ ” (enter). |

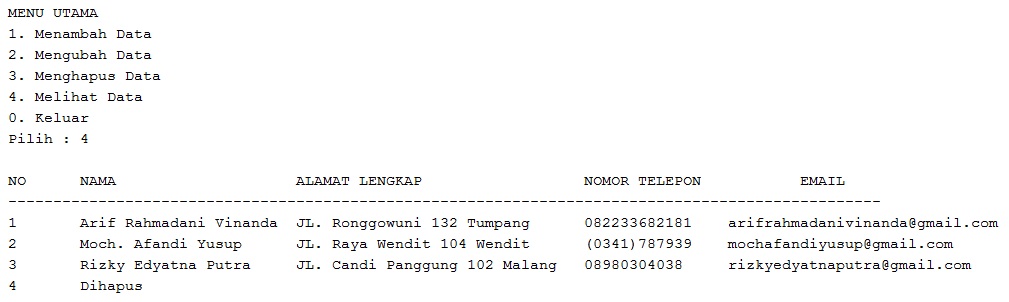
|  |  |
| --- | --- |
| **Main.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  14  15  16  17  18  19  20  21  22  24  25  26  27  28  29  30  31  33  35  36  37  38  39  40  41  44  46  47  48  49  50  51  52  53  55  56  58  59  61  62  64 | Proses memanggil semua class dari package java.util.  Deklarasi class dengan nama Main.  Main method java.  Pendeklarasian objek scan dari class Scanner.  Inisialisasi arrayofobject bukualamat dari class BukuAlamat dengan panjang array 0.  Deklarasi variable loop bertipe data String dengan nilai “y”.  Deklarasi variable x, i bertipe data integer dengan nilai 0.  Inisialisasi perulangan do-while.  Proses pemanggilan method cetakMenu.  Proses pengecekan inputan user bertipe data String dengan fungsi switch.  Fungsi yang dijalankan apabila nilai inputan user bernilai “1”.  Fungsi untuk pengcopyan array bukualamat ke array baru bukualamat dengan panjang array sebelumnya ditambah 1.  Proses pemanggilan method static inputBukuAlamat dengan argumen array bukualamat dan variable x dari class BukuAlamat.  Proses penambahan nilai variable x dengan 1.  Fungsi untuk menghentikan case “1”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “2”.  Proses pemanggilan method static cetakBukuAlamat dengan argumen array bukualamat dari class BukuAlamat.  Proses pemanggilan method edit.  Pengisian variable i dari inputan user bertipe data integer.  Pengecekan apakah nilai dari variable i dikurangi angka 1 kurang dari panjang array bukualamat.  Proses pemanggilan method static editBukuAlamat dengan parameter array bukualamat dan inputan user bertipe data ingeger dari class BukuAlamat.  Fungsi yang dijalankan apabila fungsi if diatas tidak terpenuhi serta memanggil method dataNull.  Fungsi untuk menghentikan case “2”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “3”.  Proses pemanggilan method static cetakBukuAlamat dengan argumen array bukualamat dari class BukuAlamat.  Proses pemanggilan method hapus.  Pengisian variable i dari inputan user bertipe data integer.  Proses pemanggilan method yakinHapus.  Pengecekan pakah nilai dari inputan user bertipe data String sama dengan “y” dan nilai i dikurangi angka 1 kurang dari panjang array bukualamat.  Proses pemanggilan method static deleteBukuAlamat dengan argumen array bukualamat dan variable i.  Fungsi yang dijalankan apabila fungsi if diatas tidak terpenuhi serta memanggil method dataNull.  Fungsi untuk menghentikan case “3”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “4”.  Proses pemanggilan method static cetakBukuAlamat dengan argumen array bukualamat dari class BukuAlamat.  Fugsi untuk menghentikan case “4”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “0”.  Proses pengisian variable loop dengan “n”.  Eksrepsi perulangan do-while yang berulang ketika nilai variable loop bernilai “y”.  Pendeklarasian method static cetakMenu tanpa pengembalian nilai (void).  Proses mencetak “MENU UTAMA”.  Proses mencetak “1. Menambah Data”.  Proses mencetak “2. Mengubah Data”.  Proses mencetak “3. Menghapus Data”.  Proses mencetak “4. Melihat Data”.  Proses mencetak “0. Keluar”.  Proses mencetak “Pilih : “.  Pendeklarasian method static hapus tanpa pengembalian nilai (void).  Proses mencetak “Hapus nomor ke? “.  Pendeklarasian method static edit tanpa pengembalian nilai (void).  Proses mencetak “Edit nomor ke? “.  Pendeklarasian method static yakin tanpa pengembalian nilai (void).  Proses mencetak “Yakin data dihapus(y/n)? “.  Pendeklarasian method static dataNull tanpa pengembalian nilai (void).  Proses mencetak “Data tidak ada“. |

1. **SCREENSHOT PROGRAM**

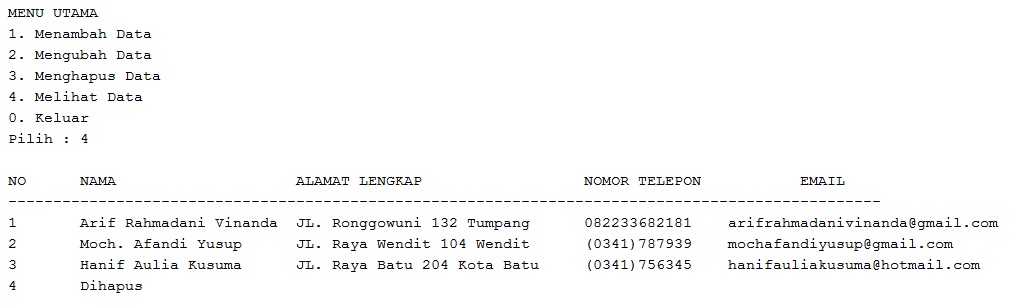
****

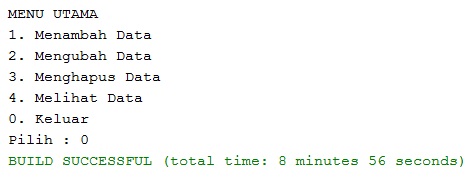
****

****

****

****

****

****

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

|  |  |
| --- | --- |
| Jawaban : | Terjadi kesalahan Test.java baris ke-5 yakni pada pemanggilan method setMark dengan argumen bertipe String dengan nilai “90” yang seharusnya berparameter integer 90. Lalu pada baris 6 dan 7 waktu mencetak nilai dari objek s1 seharusnya menggunakan accessor getName dan getMark bukan mutator setName dan setMark. Yang terakhir pada baris ke-8 pemanggilan variable dengan access modifier private harus menggunakan accessor tidak langsung memanggil nama variablenya. |

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

|  |  |
| --- | --- |
| Jawaban : | Error akan hilang karena setName adalah mutator yang harusnya pada proses pencetakan atau pengambilan nilai dari objek s1 menggunakan accessor getName. |

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

|  |  |
| --- | --- |
| Jawaban : | Pada baris 5 error akan hilang karena parameter yang dibutuhkan sudah memenuhi syarat yakni bertipe data integer. Sebaliknya, baris 7 akan error karena ada dua kali penginisialisasian objek s1. |

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

|  |  |
| --- | --- |
| Jawaban : | Pada class Student.java tidak terjadi error, namun pada class Test.java terjadi error pada baris ke-4 karena baris tersebut memanggil method yang beraccess modifier private dimana access modifier tersebut hanya meperbolehkan method diakses pada class itu saja. |

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

|  |  |
| --- | --- |
| Jawaban : | Tetap, karena attribut pada class student beraccess modifier private yang hanya bisa diakses pada class tersebut. Metode pengaksesan dua attribut tersebut harus menggunakan accessor dan mutator sehingga konsep enkapsulasi tetap berfungsi. |

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

|  |  |
| --- | --- |
| Jawaban : | Method getLoad dan getMaxLoad. |

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

|  |  |
| --- | --- |
| 1  2 | System.out.println("Add load(100kg) : " + (vehicle.load=500)); |

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

|  |  |
| --- | --- |
| Jawaban : | Akan terjadi error pada baris tersebut karena objek memanggil attribut langsung yang beraccess modifier private tanpa melewati accessor yang ada. |

1. Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi **public**. Jalankan program, apakah output dari program tersebut?
2. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

|  |  |
| --- | --- |
| 1  2 | System.out.println("Add load(100kg) : " + (vehicle.load=500)); |

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

|  |  |
| --- | --- |
| Jawaban : | Output :  Add load(100kg) : 500.0  Program tersebut bisa karena objek vehicle mengakses attribut load yang sekarang beraccess modifier public sehingga dapat langsung diakses tanpa melalui accessor maupun mutator |

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

|  |  |
| --- | --- |
| 1  2 | System.out.println("Add load(100kg) : " + (vehicle.load=500)); |

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

|  |  |
| --- | --- |
| Jawaban : | Output :  Add load(100kg) : 500.0  Program tersebut bisa (seperti point a) karena objek vehicle mengakses attribut load yang sekarang beraccess modifier public sehingga dapat langsung diakses tanpa melalui accessor maupun mutator |

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

|  |  |
| --- | --- |
| Jawaban : | Tetap bisa berjalan tanpa error dan menghasilkan output yang sama seperti diatas karena access modifier protected dapat diakses juga dari class yang satu package dengan class tersebut sehingga attribut yang beraccess modifier protected class Vehicle1.java juga dapat diakses dari TestVehicle1.java yang berada dalam satu package. |

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

|  |  |
| --- | --- |
| Jawaban : | Sama seperti nomor 4, tetap bisa berjalan tanpa error dan menghasilkan output yang sama seperti diatas karena access modifier default dapat diakses juga dari class yang satu package dengan class tersebut sehingga attribut yang beraccess modifier default class Vehicle1.java juga dapat diakses dari TestVehicle1.java yang berada dalam satu package. |

1. **KESIMPULAN**

Enkapsulasi jika diartikan secara harfiah adalah pembungkus. Maksud dari pembungkus adalah suatu cara untuk meyembunyikan implementasi detil dari class untuk mencegah akses yang ilegal dari class lain. Ada dua hal yang  mendasar dari enkapsulasi yaitu *information hiding* dan *interface to access data. Information hiding* adalah menyembunyikan informasi suatu class agar tidak bisa diakses dari luar class, caranya hanya dengan memberikan access modifier *private* pada varibel yang ingin kita *hiding*. Selanjutnya mengenai *interface to acces data,* ialah cara kita untuk mengubah nilai pada suatu variabel yang telah dilakukan *information hiding.* Cara yang biasa dilakukan adalah melalui *method,* lewat *method* ini kita bisa merubah nilai suatu variabel yang telah mengalami *information hiding.* Untuk merubah suatu nilai yang telah dienkapsulasi menggunakan method yang bernama mutator (set) untuk merubah nilai dari attribut yang beraccess modifier *private.* Sedangkan untuk mengambil nilainya, membutuhkan method yang bernama accessor (get) sehingga class lain harus melalui dua method tersebut untuk menggunakan attribut yang ada.

Selanjutnya ialah access modifier. Telah dibahas diatas bahwa enkapsulasi memerlukan access modifier *private*. Access modifier merupakan keyword yang digunakan untuk menentukan spesifikasi tingkat akses suatu vaiable atau method (anggota kelas). Pengaksesan yang dimaksud bisa berupa pengaksesan dalam kelas yang sama, turunan, satu package maupun dari luar kelas di mana variable dan method dideklarasikan. Terdapat 4 access modifier yang dipakai dalam java :

1. Private

Access modifier ini memberikan proteksi yang cukup untuk variable atau method dalam satu class. Setelah diberi access modifier tersebut, variable atau method tidak bisa diakses secara langsung oleh class lain meskipun class tersebut berada dalam satu package yang sama. Cara untuk mengaksesnya yakni menggunakan method mutator dan accessor (telah dijelaskan diatas).

1. Default

Access modifier ini hampir sama dengan access modifier private namun variable atau method dengan access modifier ini masih bisa diakses secara langsung apabila class yang mengakses masih satu class dengan class tersebut.

1. Protected

Access modifier ini sudah hampir bertipe terbuka yakni memberikan access hingga subclass atau class turunannya. Namun class ini masih belum bisa diakses secara langsung dari class lain yang berbeda package.

1. Public

Access modifier ini dapat diakses dari class lain meskipun class tersebut berbeda package. Access modifier ini bisanya hanya diberikan kepada variable atau method yang tidak bersifat harus dilindungi atau diproteksi sehingga pengakses bisa mengakses secara langsung.