1. **DEFINISI MASALAH**
2. Tugas Anda adalah membuat sebuah class yang memuat data-data pada buku alamat.

Table berikut mendefinisikan informasi yang dimiliki oleh buku alamat.

|  |  |
| --- | --- |
| Atribut | Deskripsi |
| Nama  Alamat  Nomor Telepon  Alamat E-Mail | Nama Lengkap perseorangan  Alamat Lengkap  Nomor Telepon personal  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

1. **SOURCE CODE**

|  |  |
| --- | --- |
| Alamat.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 | package Laporan4;  import java.util.Scanner;  public class Alamat {  Scanner scan = new Scanner(System.in);  private String nama;  private String alamat;  private String no\_telepon;  private String email;  public Alamat(){  this.nama = "";  this.alamat = "";  this.no\_telepon = "";  this.email = "";  }  public Alamat(String nama, String alamat, String no\_telepon, String email) {  this.nama = nama;  this.alamat = alamat;  this.no\_telepon = no\_telepon;  this.email = email;  }  public void setNama() {  System.out.print ("Nama\t\t: ");  this.nama = scan.nextLine();  }  public void setAlamat() {  System.out.print("Alamat\t\t: ");  this.alamat = scan.nextLine();  }  public void setNo\_telepon() {  System.out.print("No. Telepon\t: ");  this.no\_telepon = scan.nextLine();  }  public void setEmail() {  System.out.print("Email\t\t: ");  this.email = scan.nextLine();  }  public String getNama() {  return nama;  }  public String getAlamat() {  return alamat;  }  public String getNo\_telepon() {  return no\_telepon;  }  public String getEmail() {  return email;  }  public static void mencetakAlamat(Alamat[] alamat) {  System.out.println("--------------------------------------------------------------------------");  System.out.println(" \nNo.\t\tNama\t\t\tAlamat\t\t\tNo.Telp\t\tEmail");  System.out.println("--------------------------------------------------------------------------");  for (int x = 0; x<alamat.length; x++) {  System.out.print((x+1)+"\t");  if (alamat[x].getNama().equals(""))  System.out.print("Terhapus\n");  else{  System.out.print(alamat[x].getNama() + "\t");  System.out.print(alamat[x].getAlamat() + "\t");  System.out.print(alamat[x].getNo\_telepon() + "\t");  System.out.print(alamat[x].getEmail() + "\n");}  }  System.out.println("");  }  public static void masukanAlamat (Alamat[] alamat, int x){  System.out.println("Masukan Data Baru");  alamat[x] = new Alamat();  alamat[x].setNama();  alamat[x].setAlamat();  alamat[x].setNo\_telepon();  alamat[x].setEmail();  System.out.println("");  }  public static void menghapusAlamat (Alamat[] alamat, int y){  System.out.println("\nData Terhapus");  --y;  alamat[y] = new Alamat();  System.out.println("");  }  public static void editAlamat(Alamat[] alamat, int y){  System.out.println("\nEdit Alamat");  --y;  alamat[y].setNama();  alamat[y].setAlamat();  alamat[y].setNo\_telepon();  alamat[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 | package Laporan4;  import java.util.Arrays;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  Alamat[] alamat = new Alamat[0];  String loop = "y";  int x = 0, i = 0;  do {  cetakMenu();  switch (scan.next()) {  case "1":  alamat = Arrays.copyOf(alamat, alamat.length + 1);  Alamat.masukanAlamat(alamat, x);  x++;  break;  case "2":  Alamat.mencetakAlamat(alamat);  edit();  i = scan.nextInt();  if ((i - 1) < alamat.length) {  Alamat.editAlamat(alamat, i);  }  else dataNull();  break;  case "3":  Alamat.mencetakAlamat(alamat);  hapus();  i = scan.nextInt();  Menghapus();  if (scan.next().equals("y") && (i - 1) < alamat.length) {  Alamat.menghapusAlamat(alamat, i);  }  else dataNull();  break;  case "4":  Alamat.mencetakAlamat(alamat);  break;  case "5":  loop = "n";  break;  }  } while (loop.equals("y"));  }  public static void cetakMenu() {  System.out.println("Menu Pilih");  System.out.println("1. Tambah Daftar Alamat");  System.out.println("2. Mengubah Daftar Alamat");  System.out.println("3. Menghapus Daftar");  System.out.println("4. Daftar Tampilan Lengkap1");  System.out.println("5. Keluar");  System.out.print("Pilih : ");  }  public static void hapus() {  System.out.print("Hapus nomor? ");  }  public static void edit() {  System.out.print("Edit nomor? ");  }  public static void Menghapus() {  System.out.print("Data akan dihapus(y/n)?");  }  public static void dataNull() {  System.out.println("Data kosong\n");  }  } |

1. **PEMBAHASAN**

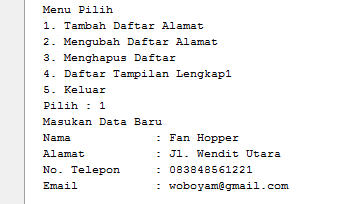
|  |  |
| --- | --- |
| Alamat.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | Pendeklarasian pada package  Scanner scan sebagai inputan  Penginisialisasian class Alamat  Scanner scan sebagai inputan  Inputan private nama dengan string  Inputan private alamat dengan string  Inputan private no\_telepon dengan string  Inputan private email dengan string  Method constructor dengan nama Alamat  Mengisi variable nama dengan “”  Mengisi variable alamat dengan ""  Mengisi variable no\_telepon dengan ""  Mengisi variable email dengan ""  Method constructor dengan nama Alamat dengan parameter String nama, String alamat, String no\_telepon, String email  Mengisi variable nama = nama  Mengisi variable alamat = alamat  Mengisi variable no\_telepon = no\_telepon  Mengisi variable email = email;  Pendeklarasian mutator dengan nama public void setNama  Proses mencetak Nama  Pengisian nama dengan inputan scan.nextLine  Pendeklarasian mutator dengan nama public void setAlamat  Proses mencetak Alamat  Pengisian alamat dengan inputan scan.nextLine  Pendeklarasian mutator dengan nama public void setNo\_telepon  Proses mencetak No. Telepon  Pengisian no\_telepon dengan inputan scan.nextLine  Method constructor dengan nama public void setEmail  Proses mencetak Email  Pengisian email dengan inputan scan.nextLine  Method constructor dengan nama public berparameter String getNama  Proses pemanggilan nilai balik nama  Method constructor dengan nama public berparameter String getAlamat  Proses pemanggilan nilai balik alamat  Method constructor dengan nama public berparameter String getNo\_telepon  Proses pemanggilan nilai balik no\_telepon  Method constructor dengan nama public berparameter String getEmail  Proses pemanggilan nilai balik email  Method public static dengan mengambil nilai void mencetakAlamat berparameter Alamat[] alamat  Proses mencetak "--------------------------------------------------------------------------"  Proses mencetak " \nNo.\t\tNama\t\t\tAlamat\t\t\tNo.Telp\t\tEmail"  Proses mencetak "--------------------------------------------------------------------------"  Perulangan for berparameter int x = 0; x<alamat.length; x++  Proses mencetak (x+1)+"\t"  Percabangan if berparameter alamat[x].getNama().equals("")  Proses mencetak "Terhapus\n"  Seleksi kondisi dengan else  Proses mencetak alamat[x].getNama() + "\t"  Proses mencetak alamat[x].getAlamat() + "\t"  Proses mencetak alamat[x].getNo\_telepon() + "\t"  Proses mencetak alamat[x].getEmail() + "\n"  Proses mencetak ("") untuk jarak  Method public static dengan mengambil nilai void masukanAlamat berparameter Alamat[] alamat, int x  Proses mencetak "Masukan Data Baru"  Penginisialisasian objek alamat  Pemanggilan method mutator setNama()  Pemanggilan method mutator setAlamat()  Pemanggilan method mutator setNo\_telepon()  Pemanggilan method mutator setEmail()  Proses mencetak ("") untuk jarak  Method public static dengan mengambil nilai void menghapusAlamat berparameter Alamat[] alamat, int y  Proses mencetak "\nData Terhapus"  Proses pengurangan variable dengan --y  Penginisialisasian objek dengan new Alamat()  Proses mencetak ("")  Method public static dengan mengambil nilai void editAlamat berparameter Alamat[] alamat, int y  Proses mencetak "\nEdit Alamat"  Proses pengurangan variable dengan --y  Pemanggilan method mutator setNama()  Pemanggilan method mutator setAlamat()  Pemanggilan method mutator setNo\_telepon()  Pemanggilan method mutator setEmail()  Proses mencetak ("") untuk jarak  Deklarasi penutup program |

|  |  |
| --- | --- |
| 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 | Pendeklarasian pada package  Scan inputan sebagai Scanner dan Arrays  Deklarasi class dengan nama Main  Main method java  Pendeklarasian scan Scanner.  Inisialisasi ArrayOfObject alamat dari class Alamat dengan panjang 0.  Deklarasi variable loop bertipe data String “y”.  Deklarasi variable x, i bertipe int dengan nilai 0.  Perulangan do-while  Proses memanggil method cetakMenu pilih  Proses pengecekan input user bertipe String dengan switch  Fungsi yang dijalankan apabila nilai inputan user bernilai “1”.  Pengcopyan array alamat ke array baru alamat dengan panjang array ditambah 1.  Proses pemanggilan method static Alamat dengan array alamat dan variable x dari class Alamat.  Penambahan nilai variable x dengan 1.  Break untuk menghentikan case “1”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “2”.  Proses pemanggilan method static mencetakAlamat dengan array alamat dari class Alamat.  Proses pemanggilan method edit.  Pengisian variable i inputan bertipe integer.  Pengecekan nilai variable i dikurangi angka 1 kurang dari panjang array alamat.  Proses pemanggilan method static editAlamat dengan parameter array alamat dan inputan bertipe integer class Alamat.  Fungsi yang dijalankan apabila fungsi if tidak terpenuhi untuk memanggil method dataNull.  Break untuk menghentikan case “2”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “3”.  Proses pemanggilan method static mencetakAlamat dengan array alamat dari class Alamat.  Proses memanggil method hapus.  Pengisian variable i inputan bertipe integer.  Proses memanggil method Hapus.  Pengecekan nilai inputan bertipe String = “y” dan nilai i-1 kurang dari panjang array alamat.  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 memanggil method static mencetakAlamat dengan array alamat dari class Alamat.  Break untuk menghentikan case “4”.  Fungsi yang dijalankan apabila nilai inputan user bernilai “5”.  Pengisian variable loop dengan “n”.  Perulangan do-while yang berulang ketika nilai variable loop bernilai “y”.  Pendeklarasian method static mencetakMenu tanpa pengembalian nilai.  Proses mencetak “Menu Pilih”.  Proses mencetak “1. Tambah Daftar Alamat”.  Proses mencetak “2. Mengubah Daftar Alamat”.  Proses mencetak “3. Menghapus Daftar”.  Proses mencetak “4. Daftar Tampilan Lengkap”.  Proses mencetak “5. Keluar”.  Proses mencetak “Pilih : “.  Pendeklarasian method static hapus.  Proses mencetak “Hapus nomor? “.  Pendeklarasian method static edit.  Proses mencetak “Edit nomor? “.  Pendeklarasian method static menghapus.  Proses mencetak “Data Akan Dihapus(y/n)? “.  Pendeklarasian method static dataNull.  Proses mencetak “Data Kosong“.  Deklarasi penutup program. |

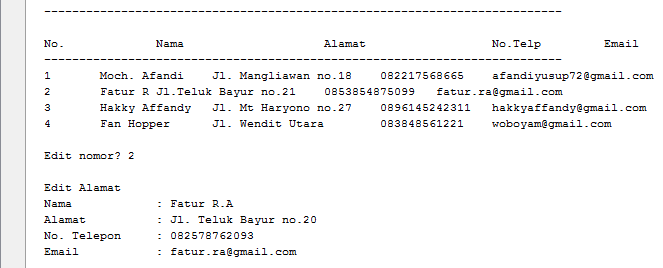
1. **SCREENSHOT PROGRAM**

* **Menambah Dafttar**

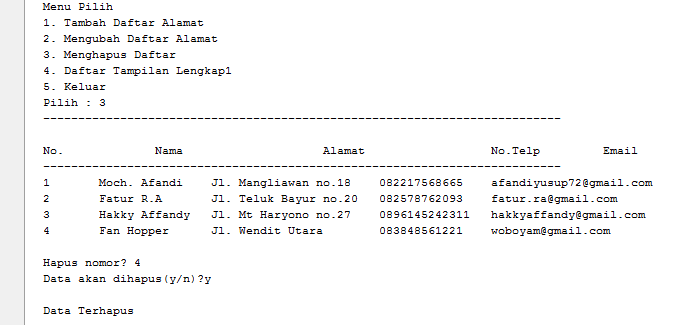




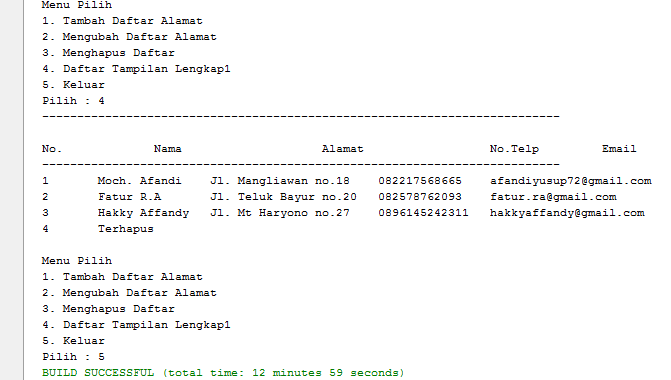
* **Mengubah Daftar**



* **Menghapus Daftar**



* **Melihat Tampilan Lengkap**



1. **PRAKTIKUM**
2. **ENCAPSULATION 1**
3. Lakukan percobaan di atas dan benahi jika menemukan kesalahan!

* Terjadi kesalahan pada baris 7 yang seharusnya inputan bertipe data integer dan pada baris 8 dan 9 yang seharusnya s1 di ganti dengan setMark.

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

* Program akan terjadi eror karena get adalah variable untuk mengambil nilai sedangkan tipe data yang di pakai seharusnya setter.

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

* Eror pada baris ke 5 akan hilang karema telah memenuhi syarat dan sebaliknya jika pada baris ke 7 akan eror karena melakukan penginisialisasian s1 dua kali.

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

* Pada class Test.java terjadi error pada baris ke-4 karena baris tersebut memanggil method yang beraccess modifier private.

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

* Tetap berjalan karena attribut pada class student beraccess modifier private yang hanya bisa diakses pada class tersebut.

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

* Yaitu Accessor Method dan Method getLoad, contoh

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class StudentRecord {  private String name;  public String getName(){  return name;  }  } |

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

1. Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi public.

Jalankan program, apakah output dari program tersebut?

1. 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.

* 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.

* 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 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

* Tetap bisa berjalan tanpa error dan menghasilkan output yang sama seperti diatas karena access modifier protected dapat diakses juga dari class yang satu package.

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

* 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.

1. **KESIMPULAN**
2. **Encapsulation**

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.

1. **Access Modifier**

Access Modifier batasan dan cara mengakses suatu property dana tau fungsi dari sebuah class, dengan access modifier inilah salah satu tujuan belajar OOP dapat dicapai yaitu enkapsulasi data. Enkapsulasi data merupakan cara melindungi atribut/fugsi tertentu dari sebuah kelas agar tidak sembarangan di akses dan di modifikasi di suatu bagian program sehingga validitas data bias terjaga.

1. **Macam-Macam Access Modifier**

Access Modifier terdiri dari 3 macam, dan di antaranya yaitu;

1. Public

Yaitu atribut atau fungsi yang di beri access modifier ini bebas di akses dari manapun di program yang umumnya sebuah fugsi constructor dan destructor pasti public.

1. Private

Atribut atau fungsi diberi akses private hanya bisa diakses di dalam kelas itu sendiri.

1. Protected

Merupakan access modifier yang hanya akan muncul jika ada lebih dari 1 kelas dalam suatu suatu masalah dan ada kelas yang mendapat pewaris sifat inherintance dari kelas lain.