1. **DEFINISI MASALAH**

Membuat sebuah class yang memuat data pada buku alamat. Dengan atribut Nama, Alamat, Nomor Telepon, Alamat E-Mail,dan disertai deskripsi Nama Lengkap perseorangan, Alamat Lengkap, Nomor Telepon personal, Alamat E-Mail personal. Menggunakan method dengan implementasi, menyediakan accessor dan mutator method terhadap seluruh atribut, dan constructor.

1. **SOURCE CODE**

**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 | import java.util.\*;  public class Main {  public static void main(String[] args) {  Bukualamat[] buku = new Bukualamat[0];  int i = 0, u = 0, w = 0;  Scanner input = new Scanner(System.in);  do {  System.out.println("1. Masukkan Data");  System.out.println("2. Lihat Data");  System.out.println("3. Hapus Data");  System.out.println("4. Ubah Data");  System.out.println("0. Keluar");  System.out.print("> ");  int pil = input.nextInt();  switch (pil) {  case 1:  buku = Arrays.copyOf(buku, buku.length + 1);  Bukualamat.input(buku, i);  i++;  break;  case 2:  Bukualamat.display(buku);  break;  case 3:  Bukualamat.display(buku);  System.out.print("Hapus Data Ke- ? ");  w = input.nextInt();  System.out.print("y/n untuk menghapus ");  if (input.next().equals("y") && (w - 1) < buku.length) {  Bukualamat.delete(buku, w);  }  break;  case 4:  Bukualamat.display(buku);  System.out.print("Edit Data Ke- ? ");  w = input.nextInt();  if ((w - 1) < buku.length) {  Bukualamat.edit(buku, w);  }  break;  case 0:  u++;  break;  }  } while (u == 0);  }  } |

**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 | import java.util.Scanner;  public class Bukualamat {  Scanner in = new Scanner(System.in);  private String nama;  private String alamat;  private String nope;  private String email;  public Bukualamat() {  this.nama = "";  this.alamat = "";  this.nope = "";  this.email = "";  }  public Bukualamat(String nama, String alamat, String nope, String email) {  this.alamat = alamat;  this.email = email;  this.nama = nama;  this.nope = nope;  }  public static void display(Bukualamat[] buku) {  int i;  for (i = 0; i < buku.length; i++) {    if (buku[i].getNama().equals("")) {  System.out.println("DATA KE-"+(i + 1)+" DATA TERHAPUS");  } else {  System.out.println("DATA KE-"+(i + 1));  System.out.println("NAMA : " + buku[i].getNama());  System.out.println("ALAMAT : " + buku[i].getAlamat());  System.out.println("NOMER TELEPON : " + buku[i].getNope());  System.out.println("EMAIL : " + buku[i].getEmail());  }  }  }  public static void input(Bukualamat[] buku, int i) {  buku[i] = new Bukualamat();  System.out.print("NAMA : ");  buku[i].setNama();  System.out.print("ALAMAT : ");  buku[i].setAlamat();  System.out.print("NOMER TELEPON : ");  buku[i].setNope();  System.out.print("EMAIL : ");  buku[i].setEmail();  }  public static void edit(Bukualamat[] buku, int a) {  --a;  buku[a] = new Bukualamat();  System.out.print("NAMA : ");  buku[a].setNama();  System.out.print("ALAMAT : ");  buku[a].setAlamat();  System.out.print("NOMER TELEPON : ");  buku[a].setNope();  System.out.print("EMAIL : ");  buku[a].setEmail();  }  public static void delete(Bukualamat[] buku, int a) {  --a;  buku[a] = new Bukualamat();  }  public String getNama() {  return nama;  }  public void setNama() {  this.nama = in.nextLine();  }  public String getAlamat() {  return alamat;  }  public void setAlamat() {  this.alamat = in.nextLine();  }  public String getNope() {  return nope;  }  public void setNope() {  this.nope = in.nextLine();  }  public String getEmail() {  return email;  }  public void setEmail() {  this.email = in.nextLine();  }  } |

1. **PEMBAHASAN**

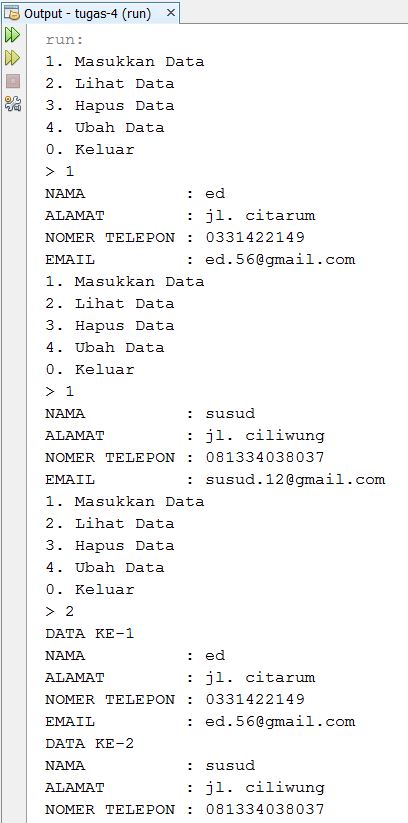
**Main.java**

|  |  |
| --- | --- |
| 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  31  33  34  35  36  37  38  39  41  42  43  44  46 | Memanggil java util pada class  Membuat class bernama Main  Main method  Membuat variable array bernama buku bertipe data class Bukualamat  Membuat variable i dengan deklarasi 0, variable u dengan deklarasi 0, variable w dengan deklarasi 0  Memanggil fungsi scanner bernama input  Melakukan perulangan do while  Mencetak, 1. Masukkan Data  Mencetak, 2. Lihat Data  Mencetak, 3. Hapus Data  Mencetak, 4. Ubah Data  Mencetak, 0. Keluar  Mencetak, >  Deklarasi variable pil bertipe data interger, dengan menggunakan inputan user  Membuat switch case dengan masukkan variable pil  Case 1  Mengcopy array buku ke array baru bernama sama dengan panjang array yg berbeda, yaitu ditambah 1  Memanggil method input dengan patameter array buku dan variable 1  i + 1  Proses break pada case 1  Case 2  Memanggil method display dengan parameter array buku  Proses break pada case 2  Case 3  Memanggil method display dengan parameter array buku  Mencetak, Hapus Data ke- ?  Inisialisasi variable w, dengan menggunakan inputan user  Mencetak, y/n untuk menghapus  Melakukan penyeleksian, apabila inputan user “y” dan (w-1) kurang dari panjang array buku  Memanggil method delete dengan parameter array buku dan variable w  Proses break pada case 3  Case 4  Memanggil method display dengan parameter array buku  Mencetak, Edit Data Ke- ?  Inisialisasi variable w, dengan menggunakan inputan user  Melakukan penyeleksian, apabila (w-1) kurang dari panjang array buku  Mamanggil method edit dengan parameter array buku dan variable w  Proses break pada case 4  Case 0  u+1  Proses break pada case 0  Melakukan perulangan terus menerus apabila nilai dari variable u sama dengan 0 |

**Bukualamat.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  14  16  17  18  19  21  22  23  25  26  28  29  30  32  34  36  41  42  43  44  45  46  47  48  49  50  52  53  54  55  56  57  58  59  60  61  62  64  65  66  68  71  72  74  76  77  78  80  83  84  86  90 | Memanggil fungsi java util scanner  Membuat class bernama Bukualamat  Memanggil fungsi scanner dengan nama in  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 nope sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable email sebagai atribut bertipe data String dengan access modifier private.  Membuat constructor Bukualamat tidak disertai dengan parameter  Deklarasi variable nama dengan teks kosong  Deklarasi variable alamat dengan teks kosong  Deklarasi variable nope dengan teks kosong  Deklarasi variable email dengan teks kosong  Pengisisan parameter contstructor Bukualamat dengan parameter String nama, String alamat, String nope, String email  Deklarasi variable alamat dengan variable alamat  Deklarasi variable email dengan variable email  Deklarasi variable nama dengan variable nama  Deklarasi variable nope dengan variable nope  Membuat method static void bernama display dengan parameter variable array buku bertipe data Bukualamat  Inisialisasi variable i  Melakukan perulangan sampai panjang array buku  Melakukan penyeleksian apabila isi array ke-i get nama berisi data kosong  Mencetak, DATA KE- (i+1) DATA TERHAPUS  Pengecualian  Mencetak, Mencetak, DATA KE- (i+1)  Mencetak, NAMA : , disertai dengan pengembalian method nama  Mencetak, ALAMAT : , disertai dengan pengembalian method alamat  Mencetak, NOMER TELEPON : , disertai dengan pengembalian method nope  Mencetak, EMAIL : , disertai dengan pengembalian method email  Membuat method static void bernama input dengan parameter variable array buku bertipe data Bukualamat dan variable i bertipe data interger  Deklarasi array ke i pada array buku  Mencetak, NAMA :  Deklarasi method setNama pada array buku indeks ke i  Mencetak, ALAMAT :  Deklarasi method setAlamat pada array buku indeks ke i  Mencetak, NOMER TELEPON :  Deklarasi method setNope pada array buku indeks ke i  Mencetak, EMAIL :  Deklarasi method setEmail pada array buku indeks ke i  Membuat method static void bernama edit dengan parameter variable array buku bertipe data Bukualamat dan variable a bertipe data interger  a-1  Deklarasi array ke a pada array buku  Mencetak, NAMA :  Deklarasi method setNama pada array buku indeks ke a  Mencetak, ALAMAT :  Deklarasi method setAlamat pada array buku indeks ke a  Mencetak, NOMER TELEPON :  Deklarasi method setNope pada array buku indeks ke a  Mencetak, EMAIL :  Deklarasi method setEmail pada array buku indeks ke a  Membuat method static void bernama delete dengan parameter variable array buku bertipe data Bukualamat dan variable a bertipe data interger  a-1  Deklarasi array ke a pada array buku  Membuat method String bernama getNama dengan pengenbalian data  Membuat method void bernama setNama  Pendeklarasian varibale nama dengan inputan user  Membuat method String bernama getAlamat dengan pengenbalian data  Membuat method void bernama setAlamat  Pendeklarasian varibale alamat dengan inputan user  Membuat method String bernama getNope dengan pengenbalian data  Membuat method void bernama setNope  Pendeklarasian varibale nope dengan inputan user  Membuat method String bernama getEmail dengan pengenbalian data  Membuat method void bernama setEmail  Pendeklarasian varibale email dengan inputan user |

1. **SCREENSHOT PROGRAM**

****

****

****

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

Terjadi kesalahan, berikut pembenahannya.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public class Test {  public static void main(String [] args) {  Student s1=new Student();  s1.setName("Enkapsulasi");  s1.setMark("90");  System.out.println("s1Name is "+s1.getName()); System.out.println("s1Mark is "+s1.getMark()); System.out.println("name dan mark "+s1.getName()+" "+s1.getMark());  }  } |

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

Maka programnya akan benar, karena baris 6 s1.setName masih terjadi error, yang benar s1.getName, dan akan memunculkan output nama yaitu “Enkapsilasi”.

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public class Test {  public static void main(String [] args) {  Student s1=new Student();  s1.setName("Enkapsulasi");  s1.setMark("90");  System.out.println("s1Name is "+s1.getName()); System.out.println("s1Mark is "+s1.getMark()); System.out.println("name dan mark "+s1.getName()+" "+s1.getMark());  }  } |

Akan menjadi tambah benar karena disitu pada baris ke 5 parameter bertipe String yang harus dirubah ke bentuk interger, sedangkan pada no 7 sama seperti no 2. Karena pada baris 7 s1.setMark masih terjadi error, yang benar s1.getMark, dan akanmemunculkan output nama yaitu “90”.

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

Maka program akan error karena pada class main setName tidak dapat dipanggil. Acces 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 membutuhkan minimal dua kelas dalam satu package yang sama.

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

Method yang mempunyai nilai pengembalian (return).

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 !

Program akan error karena pada class Vehicle1 atribut load access modifiernya private, sehingga tidak bisa dipanggil di kelas lain.

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. System.out.println("Add load(100kg) : " + (vehicle.load=500)); Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula !

Program akan menghasilkan output “Add load (100kg) : 500.0” dibawah add box #1.

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 !

Program akan menghasilkan output “Add load (100kg) : 500.0” dibawah add box #4.

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

Jika atribut load dan maxload access modifiernya dirubah menjadi protected, maka program akan berjalan karena protected dapat di access di kelas lainnya.

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

Jika atribut load dan maxload access modifiernya dirubah menjadi default, maka program akan tetap berjalan karena default juga dapat di access di kelas lainnya.

1. **KESIMPULAN**

Enkapsulasi adalah suatu cara untuk menyembunyikan informasi detail dari suatu class. Dalam enkapsulasi terdapat hak akses public, protected, dan private.

Tipe akses public memungkinkan semua class dapat mengakses meskipun berada pada package yang berbeda. Tipe akses protected hanya diberikan kepada classnya sendiri dan turunannya, serta class – class dalam satu package. Sedangkan tipe akses private hanya boleh diakses oleh classnya sendiri. Yang terakhir tipe data default dapat diakses pada classnya sendiri dan juga class lain yang berada pada satu package.

Package adalah suatu field atau dinamakan seperti folder dimana terdapat berbagai kelas yang biasanya saling berhubungan untuk menjalankan program yang lebih kompleks dengan cara saling akses data pada masing – masing class dengan yang memiliki tipe akses default, protected, dan public.

Subclass adalah suatu class yang dapat warisan atau hasil turunan dari class yang pertama kali dibuat atau sepuerclass, memiliki beberapa atau keseluruhan kesamaan sifat atau dengan kata lain subclass memiliki dara yang terdiri dari apapun yang dipunyai oleh class utamanya atau superclass.