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.

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

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 | Package Bab4;  public class BukuAlamat {  private String name, address, phoneNumber, email;  public BukuAlamat(String name, String address, String phoneNumber, String email) {  this.name = name;  this.address = address;  this.phoneNumber = phoneNumber;  this.email = email;  }  public String getName() {  return name;  }  public String getAddress() {  return address;  }  public String getPhoneNumber() {  return phoneNumber;  }  public String getEmail() {  return email;  }  public void setName(String name) {  this.name = name;  }  public void setAddress(String address) {  this.address = address;  }  public void setPhoneNumber(String phoneNumber) {  this.phoneNumber = phoneNumber;  }  public void setEmail(String email) {  this.email = email;  }  } |

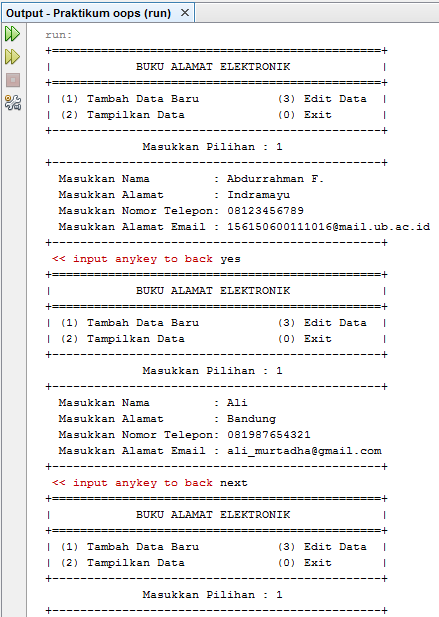
|  |  |
| --- | --- |
| **TestBukuAlamat.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84 | package Bab4;  import java.util.Scanner;  public class TestBukuAlamat {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  BukuAlamat[] data = new BukuAlamat[10];  int i = 0;  String choice;  do {  System.out.print("+=======================================+\n"  + "|\t BUKU ALAMAT ELEKTRONIK\t\t|\n"  + "+===============================================+\n"  + "| (1) Tambah Data Baru\t\t"  + " (3) Edit Data\t|\n"  + "| (2) Tampilkan Data\t\t"  + " (0) Exit\t|\n"  + "+-----------------------------------+\n"  + "\t Masukkan Pilihan : ");  choice = input.next();  switch (choice) {  case "1":  System.out.println("+------------------------------------------+");  System.out.print(" Masukkan Nama\t\t: ");  String a = inputLine();  System.out.print(" Masukkan Alamat\t: ");  String b = inputLine();  System.out.print(" Masukkan Nomor Telepon: ");  String c = inputLine();  System.out.print(" Masukkan Alamat Email\t: ");  String d = inputLine();  System.out.println("+-----------------------------------------------+");  data[i] = new BukuAlamat(a, b, c, d);  i++;  System.err.print(" << input anykey to back ");  inputLine();  break;  case "2":  if (i != 0) {  System.out.println("+-----------------------------------------------+");  System.out.println("| Attribut\t| Deskripsi\t\t\t|");  System.out.println("|-----------------------------------------------|");  for (int u = 0; u < i; u++) {  System.out.printf("| Nama\t\t| %-27s%2s |\n", data[u].getName(), (u + 1));  System.out.printf("| Alamat\t| %-30s|\n", data[u].getAddress());  System.out.printf("| Nomor Telepon\t| %-30s|\n", data[u].getPhoneNumber());  System.out.printf("| Alamat Email\t| %-30s|\n", data[u].getEmail());  System.out.println("+-----------------------------------------------+");  }  } else {  System.out.println("\t MAAF DATA MASIH KOSONG");  }  System.err.print(" << input anykey to back ");  inputLine();  break;  case "3":  if (i != 0) {  System.out.println("+-----------------------------------------------+");  System.out.print(" Masukkan No Urut data yang akan diedit: ");  int x = input.nextInt();  if (x > 0 && x <= i) {  System.out.println("+-----------------------------------------------+");  System.out.print(" Edit Nama\t\t: ");  data[x-1].setName(inputLine());  System.out.print(" Edit Alamat\t\t: ");  data[x-1].setAddress(inputLine());  System.out.print(" Edit Nomor Telepon\t: ");  data[x-1].setPhoneNumber(inputLine());  System.out.print(" Edit Alamat Email\t: ");  data[x-1].setEmail(inputLine());  System.out.println("+-----------------------------------------------+");  } else {  System.out.println("\t MAAF DATA TIDAK DITEMUKAN!");  }  } else {  System.out.println("\t MAAF DATA MASIH KOSONG");  }  System.err.print(" << input anykey to back ");  inputLine();  break;  case "0":  System.err.println(" TERIMAKASIH ANDA SUDAH KELUAR DARI PROGRAM!");  break;  default:  System.out.println("\t MAAF PILIHAN TIDAK TERSEDIA!\n");  }  } while (!"0".equals(choice));  }  public static String inputLine() {  Scanner input = new Scanner(System.in);  return input.nextLine();  }  } |

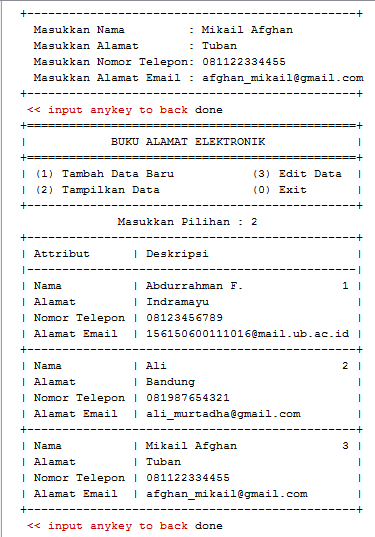
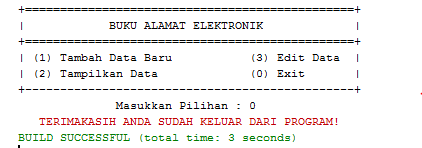
1. **PEMBAHASAN**

|  |  |
| --- | --- |
| **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 | Nama package dari file project.  Nama class dari project yang bernama BukuAlamat.  Penginisialisasian tipe data dari setiap field.  Konstruktor yang memiliki parameter.  Pemindahan data dari variabel name terhadap field name.  Pemindahan data dari variabel address terhadap field address.  Pemindahan data dari variabel phoneNumber terhadap field phoneNumber.  Pemindahan data dari variabel email terhadap field email.  Penutup sintaks.  Method yang bertipe String dan bernama getName. Method tersebut tergolong method yang mengembalikan nilai/data.  Pengembalian data dari field name.  Penutup sintaks.  Method yang bertipe String dan bernama getAddress. Method tersebut tergolong method yang mengembalikan nilai/data.  Pengembalian data dari field address.  Penutup sintaks.  Method yang bertipe String dan bernama getPhoneNumber. Method tersebut tergolong method yang mengembalikan nilai/data.  Pengembalian data dari field phoneNumber.  Penutup sintaks.  Method yang bertipe String dan bernama getEmail. Method tersebut tergolong method yang mengembalikan nilai/data.  Pengembalian data dari field email.  Penutup sintaks.  Method yang bertipe void dan bernama setName. Method tersebut memiliki parameter yang bertipe String dan tidak mengembalikan nilai/data.  Pemindahan data dari variabel name terhadap field name.  Penutup sintaks.  Method yang bertipe void dan bernama setAddress. Method tersebut memiliki parameter yang bertipe String dan tidak mengembalikan nilai/data.  Pemindahan data dari variabel address terhadap field address.  Penutup sintaks.  Method yang bertipe void dan bernama setPhoneNumber. Method tersebut memiliki parameter yang bertipe String dan tidak mengembalikan nilai/data.  Pemindahan data dari variabel phoneNumber terhadap field phoneNumber.  Penutup sintaks.  Method yang bertipe void dan bernama setEmail. Method tersebut memiliki parameter yang bertipe String dan tidak mengembalikan nilai/data.  Penutup sintaks.  Penutup sintaks. |

|  |  |
| --- | --- |
| **TestBukuAlamat.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 | Nama package dari file project.  Proses memanggil fungsi dari method Scanner.  Nama class dari project yang bernama TestBukuAlamat.  Method static void mainyang berparameter String.  Pembuatan objek baru dari fungsi Scanner.  Pembuatan objek baru dari class BukuAlamat yang bertipe array.  Penginisialisasian variabel i terhadap tipe data integer.  Penginisialisasian variabel choice terhadap tipe data String.  Perulangan yang menggunakan sintaks perulangan do-while.  Proses mencetak untuk tampilan menu.  Proses menerima masukan dari pengguna dan memindahkan data masukan tersebut terhadap variabel choice.  Penyeleksian menggunakan sintaks switch-case yang berparameter variabel choice.  Penyeleksian untuk sintaks case 1.  Proses mencetak tampilan garis batas atas.  Proses mencetak kalimat “Masukkan Nama”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Proses mencetak kalimat “Masukkan Alamat”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Proses mencetak kalimat “Masukkan Nomor Telepon”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Proses mencetak kalimat “Masukkan Alamat Email”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Proses mencetak tampilan garis batas bawah.  Pembuatan objek baru yang bertipe array dari konstruktor yang bernama BukuAlamat.  Proses penambahan nilai secara otomatis dari variabel i (post increament).  Proses mencetak kalimat “input anykey to back”.  Proses menerima masukan dari pengguna tanpa menyimpan data/nilai masukan tersebut terhadap sebuah variabel.  Proses keluar dari perulangan.  Penyeleksian untuk sintaks case 2.  Penyeleksian menggunakan sintaks if.  Proses mencetak tampilan garis batas atas.  Proses mencetak tampilan menu.  Proses mencetak tampilan garis batas bawah.  Perulangan menggunakan sintaks for.  Proses mencetak keluaran dari data nama yang telah dimasukkan sebelumnya.  Proses mencetak keluaran dari data alamat yang telah dimasukkan sebelumnya.  Proses mencetak keluaran dari data nomer telepon yang telah dimasukkan sebelumnya.  Proses mencetak keluaran dari data email yang telah dimasukkan sebelumnya.  Proses mencetak tampilan garis batas bawah.  Penutup sintaks.  Penyeleksian menggunakan sintaks else dari variabel yang ada.  Proses mencetak kalimat “MAAF DATA MASIH KOSONG”.  Penutup sintaks.  Proses mencetak kalimat “input anykey to back”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Proses keluar dari perulangan.  Penyeleksian untuk sintaks case 2.  Penyeleksian menggunakan sintaks if.  Proses mencetak tampilan garis batas atas.  Proses mencetak kalimat “Masukkan No Urut data yang akan diedit:”.  Proses menerima masukan dari pengguna untuk menentukan indeks data sesuai urutan yang telah diminta oleh pengguna.  Penyeleksian menggunakan sintaks if.  Proses mencetak tampilan garis batas atas.  Proses mencetak kalimat “Edit Nama :”.  Proses menerima masukan dari pengguna dan mengganti data nama pengguna tersebut terhadap indeks data yang telah ditentukan pengguna sebelumnya.  Proses mencetak kalimat ‘Edit Alamat :”.  Proses menerima masukan dari pengguna dan mengganti data alamat pengguna tersebut terhadap indeks data yang telah ditentukan pengguna sebelumnya.  Proses mencetak kalimat “Edit Nomor Telepon :”.  Proses menerima masukan dari pengguna dan mengganti data nomor telepon pengguna tersebut terhadap indeks data yang telah ditentukan pengguna sebelumnya.  Proses mencetak kalimat “Edit Alamat Email :”.  Proses menerima masukan dari pengguna dan mengganti data alamat email pengguna tersebut terhadap indeks data yang telah ditentukan pengguna sebelumnya.  Proses mencetak tampilan garis batas bawah.  Penyeleksian menggunakan sintaks else dari variabel yang ada.  Proses mencetak kalimat “MAAF DATA TIDAK DITEMUKAN!”.  Penutup sintaks.  Penyeleksian menggunakan sintaks else dari variabel yang ada.  Proses mencetak kalimat “MAAF DATA MASIH KOSONG”.  Penutup sintaks.  Proses mencetak kalimat “input anykey to back”.  Proses menerima masukan dari pengguna tanpa menyimpan data/nilai masukan tersebut terhadap sebuah variabel.  Proses keluar dari perulangan.  Penyeleksian untuk sintaks case 0.  Proses mencetak kalimat “TERIMA KASIH ANDA SUDAH KELUAR DARI PROGRAM!”.  Proses keluar dari perulangan.  Sintaks default dari switch –case.  Proses mencetak kalimat “MAAF PILIHAN ANDA TIDAK TERSEDIA!”.  Penutup sintaks.  Perulangan menggunakan sintaks do-while.  Penutup sintaks.  Method static yang bertipe String dan bernama inputLine. Method tersebut digunakan agar setiap masukan dari pengguna tidak terlewati (ter-skip).  Pembuatan objek baru dari fungsi Scanner.  Pengembalian data dari masukan pengguna.  Penutup sintaks.  Penutup sintaks. |

1. **SCREENSHOT PROGRAM**





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

Jawab : Ditemukan kesalahan pada baris ke 5 sampai 8

|  |
| --- |
| s1.setMark(**"90"**);  System.out.println("s1Name is "+s1.**setName**());  System.out.println("s1Mark is "+s1.**setMark**());  System.out.println("name dan mark "+ **name** +" "+ **mark**); |

Seharusnya

|  |
| --- |
| 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!

Jawab :

|  |
| --- |
| System.out.println("s1Name is "+**s1.getName**()); |

Maka program yang sebelumnya error akan benar karena sebelumnya terjadi kesalahan. Di baris tersebut diminta untuk menampilkan jadi harus menggunakan method get (accessor).

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

Jawab :

|  |
| --- |
| s1.setMark(**90**); |

jika tanda String (“ ”) dihilangkan maka program akan benar karena setMark() merupakan method yang menerima parameter bertipe integer.

|  |
| --- |
| System.out.println("s1Mark is "+s1.**getMark**()); |

Jika pada baris ketujuh diganti seperi soal sebelumnya maka akan benar karena pada baris tersebut diminta untuk menampilkan bukan untuk mengeset, jadi menggunakan method get (accessor).

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

Jawab :

|  |
| --- |
| private void setName(String n){ |

Maka akan terjadi error, class Test tidak dapat mengaksesnya karena method tersebut bersifat private tidak dapat diakses dari class yang berbeda.

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

Jawab : Ya tetap berfungsi karena ada method setter dan getter yang bisa mengakses enkapsulasi pada program tersebut. Namun jika method set dan getnya di private maka class pada package yang berbeda tidak dapat mengaksesnya.

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

Pada source code Encapsulation 2, method yang umumnya mempunyai return value adalah sebagai accessor, diantaranya:

getLoad()

getMaxLoad()

addBox()

Jawab :

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.

Maka akan terjadi Error karena field **load** bersifat private jadi tidak bisa langsung dipanggil namun harus dibuatkan mutator/method set terlebih dahulu.

***Kembalikan program seperti semula.***

Jawab:

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.

Tidak akan mengubah nilai hasil dari load, karena pada sebelumnya load juga bernilai 500. Jadi tidak berpengaruh apa-apa, hanya saja menambah cetakan pada hasil outputnya.

***Kembalikan program seperti semula.***

Jawab :

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.

Maka nilai dari vehicle load akan berubah menjadi 800kg dan bernilai true semuanya karena proses sebelumnya dihilangkan dan diganti dengan nilai 500kg ditambah 300kg.

***Kembalikan program seperti semula.***

Jawab :

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

Pada saat modifier dari variabel diubah menjadi protected, maka variabel hanya bisa diakses pada class, package, dan subclass yang sama.

Karena program ini memenuhi semua syarat maka output yang dihasilkan sama dengan output yang dihasilkan pada intruksi nomer 3.

Jawab :

1. **KESIMPULAN**

**Enkapsulasi** data merupakan cara melindungi atribut/fungsi tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi di suatu bagian program sehingga validitas data bisa terjaga. Akses ke internal sistem diatur sedemikian rupa melalui seperangkat interface atau yang biasa disebut dengan setter (Mutator) dan getter (Accessor).

**Access modifier** adalah batasan dan cara mengakses suatu properti dan atau fungsi dari sebuah kelas. Dengan access modifier inilah salah satu tujuan belajar OOP dapat dicapai yaitu enkapsulasi data.

**4 macam akses modifier :**

1. ***Default***  
   Access Modifier ini hanya menspesifikasikan kelas-kelas (classes) di paket yang sama yang bisa mengakses variabel dan method suatu kelas. Anggota kelas dengan default access bisa melihat kelas lain dipaket yang sama. Dalam pendeklarasiannya tidak ada keyword khusus yang digunakan untuk default access.
2. ***Public***  
   Akses modifier ini bisa digunakan oleh siapa saja, baik oleh kelas yang sama,  kelas lain maupun lingkungan luar kelas. Sehingga anggota kelas yang dideklarasikan sebagai public akan bisa diakses oleh sembarang object lain.
3. ***Protected***  
   Access modifier protected menspesifikasikan atau menentukan anggota kelas yang hanya bisa diakses oleh method-methodyang ada di dalam kelas yang sama dan juga dapat diakses oleh subclass (kelas turunan) dari kelas tersebut.
4. ***Private***  
   Access modifier private adalah yang paling terbatas, menspesifikasikan anggota kelas(variable dan method) hanya dapat diakses oleh kelas dimana anggota itu didefinisikan. Access modifier private mengakibatkan bahwa tidak ada kelas lain yang bisa mengakses anggota kelas private, sekaligus termasuk subclass – subkelas (subclassess)-nya kecuali dengan cara menambahkan Accessor atau Mutator agar bisa mengakses variable/method yang di private tersebut.