1. **DEFINISI MASALAH**
   1. 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 |

1. **SOURCE CODE**

|  |  |
| --- | --- |
| MainBuku.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 | package praktikum4;  import java.util.Scanner;  public class MainBuku {  public MainBuku() {  }  public static void main(String[] args) {  int i = 0;  String cek;  Scanner scan = new Scanner(System.in);  do {  BukuAlamat[] book = new BukuAlamat[3];  ArrayBuku(book);  System.out.println("===Selamat Datang===");  System.out.println("Data yang tersedia");  for (int a = 0; a < book.length; a++) {  System.out.println((a + 1) + "." + book[a].getNama());  }  System.out.println("====================");  cetak(book);  System.out.print("lihat data lain (y/n) :");  cek = scan.next();  } while (cek.equalsIgnoreCase("y"));  }  public static void ArrayBuku(BukuAlamat[] book) {  book[0] = new BukuAlamat();  book[0].setNama("Fakhrizal Arif P");  book[0].setAlamat("Jln.Ikan Tombro Selatan II");  book[0].setTlp("085576857695");  book[0].setEmail("fakhrizal.arif@yahoo.com");  book[1] = new BukuAlamat();  book[1].setNama("Arif Rahmadani Vinanda");  book[1].setAlamat("Jln.Ronggowuni Tumpang");  book[1].setTlp("086969696969");  book[1].setEmail("arifrahmadanivivanda@gmail.com");  book[2] = new BukuAlamat();  book[2].setNama("Tyo Rizky Bachtiar");  book[2].setAlamat("Gajayana Gg 1 ");  book[2].setTlp("082323232323");  book[2].setEmail("itachi.amatheratsu@gmail.com");  }  static void cetak(BukuAlamat[] book) {  System.out.print("pilih data :");  Scanner scan = new Scanner(System.in);  int i = scan.nextInt();  System.out.println("Nama\t:" + book[i - 1].getNama());  System.out.println("Alamat\t:" + book[i - 1].getAlamat());  System.out.println("Telp\t:" + book[i - 1].getTlp());  System.out.println("Email\t:" + book[i - 1].getEmail());  }  } |

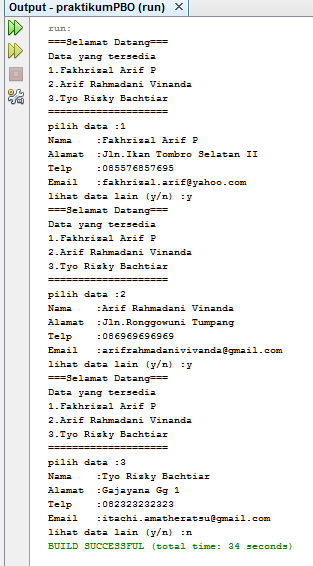
|  |  |
| --- | --- |
| BukuAlamat.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | package praktikum4;  public class BukuAlamat {  private String nama;  private String alamat;  private String tlp;  private String email;  public void setNama(String nama) {  this.nama = nama;  }  public void setAlamat(String alamat) {  this.alamat = alamat;  }  public void setTlp(String tlp) {  this.tlp = tlp;  }  public void setEmail(String email) {  this.email = email;  }  public String getNama() {  return nama;  }  public String getAlamat() {  return alamat;  }  public String getTlp() {  return tlp;  }  public String getEmail() {  return email;  }  } |

**PEMBAHASAN**

|  |  |
| --- | --- |
| MainBuku.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 | Kelas berada di package praktikum4  import Scanner untuk inputan  Deklarasi class dengan nama MainBuku  Konstruktor  Deklarasi suatu main method  Variabel i bertipe integer bernilai 0  Variabel cek bertipe String  Instansiasi scanner dengan nama scan  Fungsi perulangan do  Instansiasi BukuAlamat dengan nama book  Memanggil method ArrayBuku berparameter book  Mencetak ===Selamat Datang===  Mencetak Data yang tersedia  Fungsi perulangan for merubah nilai variabel a sebanyak panjang indeks array  Mencetak menu nama-nama yang ada dengan mencetak array yang berisi nama  Mencetak =========================  Memamnggil method cetak berparameter book  Mencetak lihat data lain (y/n) :  Proses input variabel cek  Fungsi perulangan while dengan statment mengecek variabel cek = y  Deklarasi method ArrayBuku  Instansisasi array berindeks 0  Mengisi nama indeks 0 dengan memanggil mutator setNama  Mengisi alamat indeks 0 dengan memanggil mutator setAlamat  Mengisi no telp indeks 0 dengan memanggil mutator setTlp  Mengisi Email indeks 0 dengan memanggil mutator setEmail  Instansisasi array berindeks 1  Mengisi nama indeks 1 dengan memanggil mutator setNama  Mengisi alamat indeks 1 dengan memanggil mutator setAlamat  Mengisi no telp indeks 1 dengan memanggil mutator setTlp  Mengisi Email indeks 1 dengan memanggil mutator setEmail  Instansisasi array berindeks 2  Mengisi nama indeks 2 dengan memanggil mutator setNama  Mengisi alamat indeks 2 dengan memanggil mutator setAlamat  Mengisi no telp indeks 2 dengan memanggil mutator setTlp  Mengisi Email indeks 2 dengan memanggil mutator setEmail  Deklarasi method cetak  Mencetak pilih data :  Instansiasi scanner dengan nama scan  Proses input variabel i bertipe int  Mencetak nama dan memanggil nilai dari getNama  Mencetak alamat dan memanggil nilai dari getAlamat  Mencetak no telp dan memanggil nilai dari getTelp  Mencetak email dan memanggil nilai dari getEmail |

|  |  |
| --- | --- |
| BukuAlamat.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Kelas berada di package praktikum4  Deklarasi class dengan nama BukuAlamat  Deklarasi variable nama bertipe String yang bersifat private  Deklarasi variable alamat bertipe String yang bersifat private  Deklarasi variable tlp bertipe String yang bersifat private  Deklarasi variable email bertipe String yang bersifat private  Deklarasi mutator method void dengan nama setNama  Mengisi variabel angka pada method setNama  Deklarasi mutator method void dengan nama setAlamat  Mengisi variabel angka pada method setAlamat  Deklarasi mutator method void dengan nama setTlp  Mengisi variabel angka pada method setTlp  Deklarasi mutator method void dengan nama setEmail  Mengisi variabel angka pada method setEmail  Deklarasi method getNama  Mengambil nilai dari variabel nama  Deklarasi method getAlamat  Mengambil nilai dari variabel alamat  Deklarasi method getTlp  Mengambil nilai dari variabel Tlp  Deklarasi method getEmail  Mengambil nilai dari variabel email |

1. **SCREENSHOT PROGRAM**



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

Jawab : Pada baris ke 5 class main parametertidek sesuai tipe data yang trlah di inisialisasikan tanda petik dua di hapus agar sesuai ,baris ke 6 dan 7 class main pemanggilanmethod salah harusnya jika di cetak yang di panggil adalah accsesor ,baris ke 8 terjadi eror karena variabel yang di panggil bertipe private maka pada class Student di baris 2dan 3 di ganti public dan pada class main baris ke 8 variabel di tambah instansiasi objek

|  |  |
| --- | --- |
| Main.java | |
| 1  2  3  4  5  6  7  8  9  10 | 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.name + " " + s1.mark);  }  } |

|  |  |
| --- | --- |
| Student.java | |
| 1  2  3  4  5  6  7  8  9  10  11 | public class Student {  public String name;  public int mark;  public void setName(String n) {  name = n;  }  public String getName() {  return name;  }  public void setMark(int m) {  mark = m;  }  public int getMark() {  return mark;  }  } |

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

Jawab : Maka tidak terjadi error karenan pemanggilan method sudah tepat

* 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 : Maka tidak terjadi eror karena parameter telah sesuai dengan tipe data

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

Jawab : setNama tidak dapat di akses di main karena bertippe private

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

Jawab : iya selama dalam satu package enkapsulasi tetap berfungsi

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

Jawab : Method yang mempunyai pengembalian nilai return

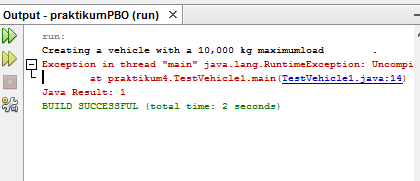
* 1. Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

Jawab : akan terjadi eror karena load tidak memiliki hak akses karena beripe perivate



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

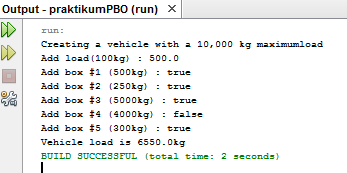
Jalankan program, apakah output dari program tersebut?

* + 1. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1.

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

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

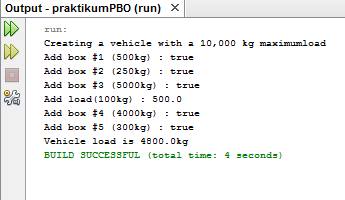


* + 1. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1.

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

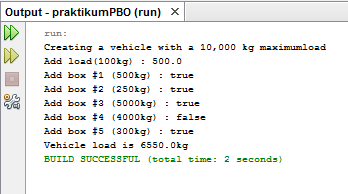
Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.



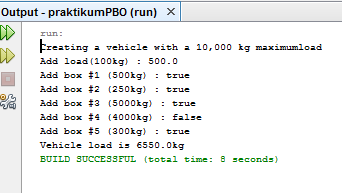
4. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

Jawab : load tetap dapat diakses karena masih dalam satu package



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

Jawab : load tetap dapat diakses karena masih dalam satu package



1. **KESIMPULAN**

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

Access modifier merupakan keyword yang digunakan untuk menentukan spesifikasi tingkat akses suatu vaiable atau method (anggota kelas). Peng-aksesan yang dimaksud bisa berupa pengaksesan dalam kelas yang sama, turunan maupun dari luar kelas di mana variable dan method dideklarasikan.

1. Default

Access Modifier ini hanya menspesifikasikan kelas-kelas (classes) di paket yang sama yang bisa mengakses variable dan method suatu kelas. Anggota kelas dengan default access bisa melihat kelas lain dipaket yang sama. Tidak ada keyword khusus untuk mendeklarasikan modifier default access. Sehingga apabila tidak ada access modifier pada pendeklarasian tersebut, berarti yang dimaksud adalah default access.

Contoh :

class Mahasiswa {

String nama;

String kata() {

return “I LOVE JAVA”;

}

}

1. Public

Suatu anggota kelas (class member) dideklarasikan sebagai public, maka anggota tersebut akan bisa digunakan oleh siapa saja, yaitu baik oleh kelas yang sama, kelas lain maupun lingkungan luar kelas. Sehingga anggota kelas (class member) yang dideklarasikan sebagai public akan bisa diakses oleh sembarang object lain.

Contoh :

class Mahasiswa {

public String nama;

public String kata() {

return “I LOVE JAVA”;

}

}

1. Protected

Access modifier protected menspesifikasikan atau menentukan anggota kelas (class member) yang hanya bisa diakses oleh method-methodyang ada di dalam kelas yang sama dan juga dapat diakses oleh subclass (kelas turunan) dari kelas tersebut.

Contoh :

class Mahasiswa {

protected String nama;

protected String kata() {

return “I LOVE JAVA”;

}

}

1. Private

Access modifier private merupakan tingkatan akses yang sangat terbatas. Kemudian seperti modifier-modifier lain, modifier private menspesifikasikan anggota kelas (class member) yang hanya bisa diakses oleh kelas dimana anggota-anggota kelas tersebut dideklarasikan. Keadaan ini mengakibatkan bahwa tidak ada kelas lain yang bisa mengakses anggota kelas private, sekaligus termasuk subkelas-subkelas (subclassess)-nya

Access modifier private adalah yang paling terbatas, menspesifikasikan anggota kelas(variable dan method) hanya dapat diakses oleh kelas dimana anggota itu didefinisikan.

Contoh :

class Mahasiswa {

private String nama;

private String kata() {

return “I LOVE JAVA”;

}

}