1. **DEFINISI MASALAH**
2. Soal 1

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 personal |
| Alamat | Alamat Lengkap |
| Nomor Telpon | Nomor telpon 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

1. **SOURCE CODE**

|  |  |
| --- | --- |
|  | BukuAlamat.java |
| 1 | package bab\_4; |
| 2 | public class BukuAlamat { |
| 3 | private String nama, alamat, telpon, email; |
| 4 | public BukuAlamat() { |
| 5 | } |
| 6 | public BukuAlamat(String nama, String alamat, String telpon, String email) { |
| 7 | this.nama = nama; |
| 8 | this.alamat = alamat; |
| 9 | this.telpon = telpon; |
| 10 | this.email = email; |
| 11 | } |
| 12 | public void setNama(String nama) { |
| 13 | this.nama = nama; |
| 14 | } |
| 15 | public void setAlamat(String alamat) { |
| 16 | this.alamat = alamat; |
| 17 | } |
| 18 | public void setTelpon(String telpon) { |
| 19 | this.telpon = telpon; |
| 20 | } |
| 21 | public void setEmail(String email) { |
| 22 | this.email = email; |
| 23 | } |
| 24 | public String getNama() { |
| 25 | return nama; |
| 26 | } |
| 27 | public String getAlamat() { |
| 28 | return alamat; |
| 29 | } |
| 30 | public String getTelpon() { |
| 31 | return telpon; |
| 32 | } |
| 33 | public String getEmail() { |
| 34 | return email; |
| 35 | } |
| 36 | } |

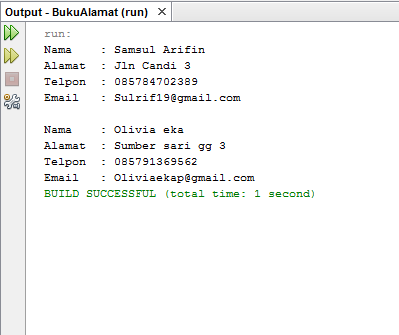
|  |  |
| --- | --- |
|  | BukuAlamat\_main.java |
| 1 | package bab\_4; |
| 2 | public class BukuAlamat\_main { |
| 3 | public static void main(String[] args) { |
| 4 | BukuAlamat a1 = new BukuAlamat("Samsul Arifin", "Jln Candi 3", "085784702389", "Sulrif19@gmail.com"); |
| 5 | BukuAlamat b1 = new BukuAlamat(); |
| 6 | b1.setNama("Olivia Eka"); |
| 7 | b1.setAlamat("Sumbersari gg3"); |
| 8 | b1.setTelpon("085712345678"); |
| 9 | b1.setEmail("Oliviaekap@gmail.com"); |
| 10 | System.out.println("Nama\t: " + a1.getNama()); |
| 11 | System.out.println("Alamat\t: " + a1.getAlamat()); |
| 12 | System.out.println("Telpon\t: " + a1.getTelpon()); |
| 13 | System.out.println("Email\t: " + a1.getEmail()); |
| 14 | System.out.println(""); |
| 15 | System.out.println("Nama\t: " + b1.getNama()); |
| 16 | System.out.println("Alamat\t: " + b1.getAlamat()); |
| 17 | System.out.println("Telpon\t: " + b1.getTelpon()); |
| 18 | System.out.println("Email\t: " + b1.getEmail()); |
| 19 | } |
| 20 | } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
|  | BukuAlamat.java |
| 1 | Memberi penamaan package class bab\_4 |
| 2 | Deklarasi kelas |
| 3 | Deklarasi atribut nama, alamat, telpon, dan email bertipe String dengan akses private |
| 4 | Deklarasi konstruktor 1 |
| 5 | Tanda untuk mengakhiri sebuah program |
| 6 | Deklarasi konstruktor 2 dengan parameter-parameter String nama, alamat, telpon, dan email |
| 7 | Men-set nama sebagai isi dari atribut nama |
| 8 | Men-set alamat sebagai isi dari atribut alamat |
| 9 | Men-set telpon sebagai isi dari atribut telpon |
| 10 | Men-set email sebagai isi dari atribut email |
| 11 | Tanda untuk mengakhiri sebuah program |
| 12 | Deklarasi method setNama dengan parameter nama sebagai mutator |
| 13 | Mengisi atribut nama dengan nama |
| 14 | Tanda untuk mengakhiri sebuah program |
| 15 | Deklarasi method setAlamat dengan parameter alamat sebagai mutator |
| 16 | Mengisi atribut alamat dengan alamat |
| 17 | Tanda untuk mengakhiri sebuah program |
| 18 | Deklarasi method setTelpon dengan parameter telpon sebagai mutator |
| 19 | Mengisi atribut telpon dengan telpon |
| 20 | Tanda untuk mengakhiri sebuah program |
| 21 | Deklarasi method setEmail dengan parameter email sebagai mutator |
| 22 | Mengisi atribut email dengan email |
| 23 | Tanda untuk mengakhiri sebuah program |
| 24 | Deklarasi method getNama sebagai accessor |
| 25 | Return atribut nama |
| 26 | Tanda untuk mengakhiri sebuah program |
| 27 | Deklarasi method getAlamat sebagai accessor |
| 28 | Return atribut alamat |
| 29 | Tanda untuk mengakhiri sebuah program |
| 30 | Deklarasi method getTelpon sebagai accessor |
| 31 | Return atribut telpon |
| 32 | Tanda untuk mengakhiri sebuah program |
| 33 | Deklarasi method getEmail sebagai accessor |
| 34 | Return atribut email |
| 35 | Tanda untuk mengakhiri sebuah program |
| 36 | Tanda untuk mengakhiri sebuah program |

|  |  |
| --- | --- |
|  | BukuAlamat\_main.java |
| 1 | Memberi penamaan package class bab\_4 |
| 2 | Deklarasi Kelas |
| 3 | Deklarasi kelas main |
| 4 | Inisialisasi object a1 dengan pengisian data pada konstruktor |
| 5 | Inisialisasi object b1 |
| 6 | Penggunaan method setNama oleh b1 |
| 7 | Penggunaan method setAlamat oleh b1 |
| 8 | Penggunaan method setTelpon oleh b1 |
| 9 | Penggunaan method setEmail oleh b1 |
| 10 | Menampilkan nama dengan penggunaan method getNama oleh a1 |
| 11 | Menampilkan alamat dengan penggunaan method getAlamat oleh a1 |
| 12 | Menampilkan telpon dengan penggunaan method getTelpon oleh a1 |
| 13 | Menampilkan email dengan penggunaan method getEmail oleh a1 |
| 14 | Menampilkan baris kosong |
| 15 | Menampilkan nama dengan penggunaan method getNama oleh b1 |
| 16 | Menampilkan alamat dengan penggunaan method getAlamat oleh b1 |
| 17 | Menampilkan telpon dengan penggunaan method getTelpon oleh b1 |
| 18 | Menampilkan email dengan penggunaan method getEmail oleh b1 |
| 19 | Tanda untuk mengakhiri sebuah program |
| 20 | Tanda untuk mengakhiri sebuah program |

1. **SCREENSHOT PROGRAM**



1. **PRAKTIKUM**
2. Encapsulation 1

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

**Jawab:**

pembenahan dilakukan di baris 5 sampai 8

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:**

Berubah dari eror menjadi tidak eror. Eror terjadi karena setName bertipe void, dan akan berubah menjadi tidak eror jika diganti getName yang memiliki return type.

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:**

Berubah dari eror menjadi tidak eror. Pada baris 5, s1.setMark membutuhkan parameter int, jadi baris 5 tidak terjadi eror hanya jika parameter bertipe int.Pada baris ke 7, jika s1.setMark() diganti menjadi s1.getMark() maka akan beubah menjadi tidak eror karena memlilki return type.

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

**Jawab:**

Terjadi eror karena method tersebut tidak bisa digunakan di class main.

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

**Jawab:**

Ya, karena menggunakan akses private (yang hanya dapat diakses di class tersebut) untuk atribut dan public (yang dapat diakses oleh semua class dalam network) untuk method.

1. Encapsulation 2

Pertanyaan

1. Method apakah yang menjadi accessor (getter) ?

**Jawab:**

getLoad() dan getMaxLoad()

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:**

Eror, dikarenakan adanya atribut load yang tidak dapat diakses.

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.

**Jawab:**

Creating a vehicle with a 10,000 kg maximum load.

Add box #1 (500kg) : true

Add load(100kg) : 500.0

Add box #2 (250kg) : true

Add box #3 (5000kg) : true

Add box #4 (4000kg) : true

Add box #5 (300kg) : false

Vehicle load is 9750.0kg

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.

**Jawab:**

Creating a vehicle with a 10,000 kg maximum load.

Add box #1 (500kg) : true

Add box #2 (250kg) : true

Add box #3 (5000kg) : true

Add box #4 (4000kg) : true

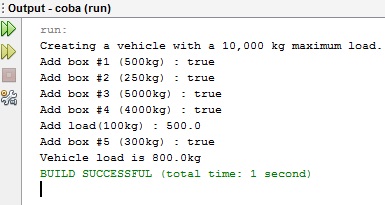
Add load(100kg) : 500.0

Add box #5 (300kg) : true

Vehicle load is 800.0kg

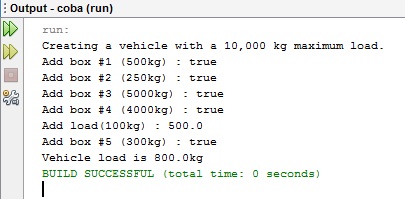
1. Ulangi instruksi pada nomer 3 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

**Jawab:**



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

**Jawab:**



1. **KESIMPULAN**

Enkapsulasi adalah pembungkus, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain. Konsep enkapsulasi sangat penting dilakukan untuk menjaga kebutuhan program agar dapat diakses sewaktu-waktu, sekaligus menjaga program tersebut.

Acces modifier adalah ketika sebuah program Java ingin diciptakan, dan user menginginkan beberapa data dilarang untuk diakses, pengimplementasi inilah yang sering disebut Access Modifiers. Kegunaan Access Modifiers ini untuk memberi hak akses kepada user. Tentu tidak semua data yang berada di dalam suatu kelas, atau turunannya dapat diakses, karena terdapat batasan-batasan yang berlaku.

4 tipe acces modifier antara lain adalah :

-Default : tipe akses ini hanya dalam satu class itu sendiri yang dapat mengaksesnya tergantung dari blok kodenya.

-Public : Tipe ini mengijinkan seluruh class dari luar atau dari dalam package bisa mengaksesnya.

-Protected : Tipe ini di gunakan untuk pewarisan dalam java, jadi klo suatu super class mempunyai sebuah member, ini dapat diakses oleh subclassnya.

-Private : Tipe ini hanya dapat diakses dimana class tersebut di buat.