**Modul 2**

**Pengenalan Pemrograman Berorientasi Objek**

* 1. **Class dan Object**

Class adalah cetak biru dari suatu objek riil di dunia. Penerapan objek riil direpresentasikan dalam bentuk class. Sebuah class mempunyai anggota yang terdiri dari atribut dan method. Atribut adalah semua field identitas yang kita berikan pada suatu class.

Contoh :

**Class segiempat 🡪 object**

 **Data : panjang, lebar**

Object adalah gambaran dari entity, baik dunia nyata atau konsep dengan batasan-batasan dan pengertian yang tepat. Objek-objek ini kemudian juga dapat berupa gabungan dari beberapa objek yang lebih kecil.

Sebagai contoh, objek kubus mempunyai ciri-ciri khusus yang dimiliki oleh kubus. Misalnya mempunyai panjang, lebar, dan tinggi. Ciri-ciri khusus kubus tersebut harus menjadi identitas utama sebagai objek kubus.

Keyword “new” digunakan untuk menambahkan instansiasi/membuat sebuah object baru.

Contoh :

**kubus hit**

**hit = new kubus()**

* 1. **Metode**

Metode biasa kita kenal sebagai *function* dan *procedure*. Dikatakan fungsi bila method tersebut melakukan suatu proses dan mengembalikan suatu nilai (*return value*), dan dikatakan prosedur bila method tersebut hanya melakukan suatu proses dan tidak mengembalikan nilai (*void*). Dalam OOP, method digunakan untuk memodularisasi program melalui pemisahan tugas dalam suatu class. Pemanggilan metode menspesifikasikan nama metode dan menyediakan informasi (parameter) yang diperlukan untuk melaksanakan tugasnya.

1. Metode tanpa nilai balik

Metode tanpa nilai balik harus dideklarasikan menggunakan kata kunci void, yang artinya tidak memiliki tepe kembalian.

Bentuk umum :

TipeAkses void NamaMetode (parameter)

{

 //BadanMetode

}

1. Metode dengan nilai balik

Bentuk umum :

TipeAkses TipeKembalian NamaMetode (parameter)

{

 //BadanMetode

 return nilai;

}

Access Modifier ada 3 jenis yang umum:

1. private, akses terbatas pada class itu sendiri
2. protected, akses terbatas pada class itu sendiri atau dapat diturunkan ke class

turunannya dan berganti akses ke private.

1. public, akses tidak terbatas pada dan dari luar bebas mengakses ke class tersebut.

**Contoh 1**

public class persegipanjang

{

 public double panjang; public double lebar;

 public double luas;

 public void setdata(double x, double y)

 {

 panjang = x, lebar = y; }

 public void luaspersegipanjang(double x, double y)

 {

 luas = x \* y;

 System.out.println("luas segiempat 2 = " +luas); }

 public double luaspersegi (double x, double y)

 {

 luas = x \* y; return(luas) }

}

class demopersegipanjang

{

 public static void main (String [] arg)

 {

 double x, y, luas;

 //instansiasi kelas segiempat

 persegipanjang se1 = new persegipanjang();

 se1.panjang = 8;

 se1.lebar = 6;

 luas = se1.panjang \* se1.lebar;

 System.out.println("luas segiempat 1 = " +luas);

 persegipanjang se2 = new persegipanjang();

 x = 5;

 y = 6;

 se2.setdata(x, y);

 luas = x \* y;

 System.out.println("luas segiempat 2 = " +luas);

 persegipanjang se3 = new persegipanjang();

 x = 5;

 y = 4;

 se3.luaspersegipanjang(x, y);

 persegipanjang se4 = new persegipanjang();

 x = 5;

 y = 5;

 luas = se4.luaspersegi(x, y);

 System.out.println("luas segitiga 4 : " +luas);

 }

}

* 1. **Class Diagram**

Pada pembuatan perangkat lunak berbasis Object Oriented, terdapat mekanisme untuk merancang perangkat lunak tersebut menggunakan UML (Unified Modeling Language). Salah satu bentuk perancangan dari UML adalah **Class Diagram**. Contoh class diagram diperlihatkan pada gambar di bawah ini.



Berdasarkan gambar di atas, terlihat bahwa class diagram merupakan sebuah bujur sangkar dengan 3 tingkatan, dengan konten di masing-masing tingkat sebagai berikut:

1. Tingkat pertama: nama Class
2. Tingkat kedua: nama atribut beserta tipe datanya
3. Tingkat ketiga: nama method beserta keterangannya
4. **Penulisan Atribut**

Atribut dituliskan dengan aturan berikut:

*visibility name: type-expression= initial-value{ property-string}*

Keterangan:

1. Visibility: private (-), protected(#), dan public (+)
2. Name: nama atribut
3. Type-expression: tipe data dari atribut
4. Initial-value: nilai awal atribut (jika ada)
5. Property-string: aturan perancangan lainnya (jika ada)

Jika sebuah atribut bersifat final (merupakan konstanta), maka pada class diagram ditulis dengan huruf kapital. Jika atribut bersifat static, berikan *underline*.

1. **Penulisan Method**

Method dituliskan dengan aturan berikut:

*visibility name( parameter-list) : return-type-expression{ property-string}*

Keterangan:

1. Visibility: private (-), protected(#), dan public (+)
2. Name: nama method
3. Parameter-list: penulisan parameter yang dilewatkan oleh method. Terdapat aturan lain untuk ini.
4. Return -ype-expression: tipe data yang dikembalikan oleh method
5. Property-string: aturan perancangan lainnya (jika ada)

Aturan penulisan parameter-list:

 *kind name: type-expression= default-value*

Keterangan:

1. Kind: jenis parameter dari method. Terdapat 3 jenis parameter: in, out, inout. Khusus bahasa pemrograman J ava, hany a ada ti pe “i n”/
2. Name: nama parameter yang dilewatkan
3. Type-expression: tipe data parameter yang dilewatkan
4. Default-value: nilai awal parameter (jika ada).

Class diagram “Mahasiswa” di atas memiliki 2 konstruktor. Dengan kata lain, overloading constructor terjadi pada class “Mahasiswa”. Sebuah class dapat memiliki lebih dari 1konstruktor. Bahkan, sebuah class dapat tidak dituliskan konstruktornya. Secara default konstruktor kosong akan terbentuk jika tidak dituliskan konstruktor secara eksplisit pada sebuah class. Bentuk dari konstruktor kosong:

**Contoh 2 :**

Mengimplementasikanclass diagram ke dalam program :



public class Tabungan

{

 public int saldo;

 /\*public Tabungan (int initsaldo)

 {

 saldo = initsaldo;

 }\*/

 public void settabungan(int initsaldo)

 {

 saldo = initsaldo;

 }

 public void ambiluang (int jumlah)

 {

 saldo = saldo - jumlah;

 }

}

class tesTabungan

{

 public static void main(String args[])

 {

 int a;

 System.out.println("objek1 : ");

 Tabungan t=new Tabungan();

 t.settabungan(50000);

 System.out.println("Saldo awal Tabungan Anda : "+t.saldo);

 t.ambiluang(1500);

 System.out.println("Jumlah uang yang diambil : 1500");

 System.out.println("Saldo Tabungan Anda sekarang adalah: " + t.saldo);

 }

}

* 1. **Tugas Praktikum**
1. Conan merupakan anak SD penggemar novel detektif. Ia ingin membuat sebuah program untuk mendata setiap novel detektif yang ia miliki. Setiap novel memiliki judul, nama pengarang dan tahun terbit. Tapi Conan ingin mengetahui isi dari setiap novel, sehingga ia tahu deskripsi novel tersebut. Conan juga menginginkan informasi harga beli tercantum di program. Dikarenakan sewaktu-waktu ia ingin menjual kembali novelnya, terdapat mekanisme untuk menghitung harga jual novel. Rumus yang ia gunakan adalah “harga jual = harga beli – 20% \* harga beli”.
2. Tentukanlah atribut dan method terlibat, gambarkan class diagramnya.
3. Buat programnya dengan membuat 3 objek berdasarkan class tersebut (data bebas), dan tampilkan informasi dari setiap buku tersebut. Lalu, tampilkan harga total beli buku serta harga total buku jika dijual.

|  |  |
| --- | --- |
| **Diperiksa Tanggal :** | **Paraf Dosen :** |
| **Catatan :** |