

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
PERTEMUAN KE-1
OBJECT ORIENTATION

Class merupakan template dari objek yang memiliki atribut dan perilaku serupa. Objek merupakan instansiasi dari sebuah class. Objek diciptakan menggunakan konstruktor dari class-nya.

TUJUAN PRAKTIKUM

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. menerapkan konsep dasar kelas dan objek ke dalam program,
2. menerapkan konstruktor dan membuat overloading-nya.

TOOLS

Tools yang diperlukan untuk melakukan praktikum ini adalah Java Development Kit (JDK) untuk mengkompilasi dan menjalankan program Java serta code editor atau IDE untuk menulis program Java.

LANGKAH PRAKTIKUM

Bagian 1 - Membuat Class dan Objek Titik

1. Tuliskan nama file, deskripsi, pembuat, dan tanggal pada bagian awal file Anda sebagai komentar.
2. Buatlah sebuah class Titik yang memiliki atribut berupa absis dan ordinat serta memiliki beberapa method seperti contoh berikut:

```
7      /* Nama File      : Titik.java
8      * Deskripsi       : berisi atribut dan method dalam class Titik
9      * Pembuat        :
10     * Tanggal         :
11     */
12
13     public class Titik {
14     □      /******ATRIBUT******/
15         double absis;
16         double ordinat;
17
18     □      /******METHOD******/
19         //konstruktor untuk membuat titik (0,0)
20         Titik(){
21     □         absis = 0;
22             ordinat = 0;
23         }
24
25         //mengembalikan nilai absis
26     □         double getAbsis(){
27             return absis;
28         }
29     }
```

```

30      //mengembalikan nilai ordinat
31      double getOrdinat() {
32          return ordinat;
33      }
34
35      //mengeset absis titik dengan nilai baru x
36      void setAbsis(double x) {
37          absis = x;
38      }
39
40      //mengeset ordinat titik dengan nilai baru y
41      void setOrdinat(double y) {
42          ordinat = y;
43      }
44
45      //menggeser nilai absis dan ordinat titik masing-masing sejauh x dan y
46      void geser(double x, double y) {
47          absis = absis + x;
48          ordinat = ordinat + x;
49      }
50
51      //mencetak koordinat titik
52      void printTitik() {
53          System.out.println("Titik (" + absis + "," + ordinat + ")");
54      }
55  } //end class Titik

```

3. Buatlah sebuah main class bernama MTitik yang menggunakan class Titik untuk membuat objek Titik dan memanggil method yang telah dibuat dalam class Titik, seperti contoh berikut:

```

7      /* Nama File      : Titik.java
8      * Deskripsi       : berisi atribut dan method dalam class Titik
9      * Pembuat        :
10     * Tanggal         :
11     */
12     public class MTitik {
13     public static void main(String[] args) {
14         Titik T1 = new Titik(); //Membuat objek titik T1 (0,0)
15         T1.setAbsis(3); //mengubah absis T1 dengan nilai 3
16         T1.setOrdinat(4); //mengubah ordinat T1 dengan nilai 4
17         T1.printTitik(); //mencetak koordinat T1 ke layar
18         T1.geser(3,4); //menggeser T1 sejauh (3,4)
19         T1.printTitik(); //menampilkan koordinat T1 setelah digeser
20     }
21 }

```

Jalankan class MTitik dan amati hasilnya.

Bagian 2 – Object Reference

4. Tambahkan kode berikut dalam method main di dalam class MTitik sebagai berikut:

```

21     Titik T2 = T1;
22     T2.printTitik();
23     T1.setAbsis(10);
24     T1.setOrdinat(10);
25     T2.printTitik();

```

Jalankan program tersebut dan amati hasilnya. Apakah koordinat titik T2 ikut berubah saat nilai absis dan ordinat titik T1 diubah? Mengapa?

Bagian 3 – Overloading Constructor

5. Pada contoh nomor 2 konstruktor class Titik selalu menghasilkan titik (0,0). Dengan prinsip *overloading*, tambahkanlah sebuah konstruktor pada class Titik yang menerima masukan nilai absis dan ordinat untuk menghasilkan titik baru sesuai dengan nilai absis dan ordinat tersebut.

Bagian 4 – Static Attribute & Method

6. Tambahkan sebuah atribut static counterTitik untuk menyimpan jumlah objek titik yang telah diinstantiasi dari class Titik. Lalu, modifikasi konstruktor Titik dan tambahkan method static getCounterTitik() untuk mengembalikan nilai counterTitik, seperti berikut:

```
13 public class Titik {
14     /*****ATRIBUT*****/
15     double absis;
16     double ordinat;
17     static int counterTitik = 0;
18
19     /*****METHOD*****/
20     //konstruktor untuk membuat titik (0,0)
21     Titik(){
22         absis = 0;
23         ordinat = 0;
24         counterTitik++;
25     }
26
27     //mengembalikan nilai counterTitik
28     static int getCounterTitik(){
29         return counterTitik;
30     }
```

7. Panggilah method getCounterTitik di dalam method main dari class MTitik, kemudian amati hasilnya.

```
14 Titik T1 = new Titik(); //Membuat objek titik T1 (0,0)
15 Titik T2 = new Titik(3,5); //Membuat objek titik T2 (3,5)
16
17 System.out.println("Jumlah Objek Titik = " + Titik.getCounterTitik());
18 System.out.println("Jumlah Objek Titik = " + T2.getCounterTitik());
```

Bagaimana perbedaan antara atribut class yang dideklarasikan menggunakan keyword static dengan atribut class biasa (tanpa keyword static)?

Bagian 5 – Keyword This

8. Keyword this dalam Java memiliki beberapa kegunaan antara lain: sebagai referensi ke dirinya sendiri untuk menunjuk ke atribut, method, ataupun konstruktor. Perhatikan contoh berikut, lalu praktikkan dan amati hasilnya.

```

13 public class Titik {
14     /*****ATRIBUT*****/
15     double absis;
16     double ordinat;
17     static int counterTitik = 0;
18
19     /*****METHOD*****/
20     //konstruktor untuk membuat dengan nilai absis dan ordinat tertentu
21     Titik(double absis, double ordinat){
22         this.absis = absis;
23         this.ordinat = ordinat;
24         counterTitik++;
25     }
26
27     //konstruktor untuk membuat titik (0,0)
28     Titik(){
29         this(0,0);
30     }

```

9. Bagaimana jika this digunakan untuk mengakses atribut counterTitik seperti berikut?

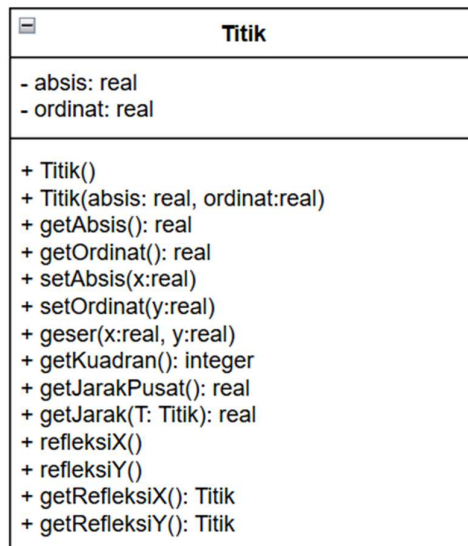
```

void printCounterTitik(){
    System.out.println(this.counterTitik);
}

```

Bagian 6 – Finalisasi Method Lainnya

10. Lengkapilah method-method lain yang berkaitan dengan class Titik memiliki semua method seperti class diagram berikut:



Catatan: prosedur refleksiX() langsung mengubah absis dan ordinat titik setelah direfleksikan terhadap sumbu X, sedangkan fungsi getRefleksiX() menghasilkan sebuah titik baru yang merupakan hasil refleksi sumbu X dari sebuah objek titik.

TUGAS/LATIHAN

Buatlah class Garis dalam program Java, lalu buatlah sebuah main class MGaris yang menggunakan class Garis tersebut.

1. Atribut Garis terdiri atas titik awal dan titik akhir yang masing-masing bertipe Titik. Tambahkan pula atribut static counterGaris untuk menyimpan banyaknya objek garis yang pernah dibuat.
2. Method Garis terdiri atas:
 - a. Konstruktor tanpa parameter yang menginisialisasi titik awal dengan (0,0) dan titik akhir dengan (1,1).
 - b. Konstruktor dengan parameter masukan titik awal dan titik akhir.
 - c. Selektor (getter) untuk setiap atribut beserta mutatornya (setter). Tambahkan pula selektor untuk mendapatkan atribut static counterGaris.
 - d. Method untuk mendapatkan panjang sebuah garis.
 - e. Method untuk mendapatkan gradien dari sebuah garis.
 - f. Method untuk mendapatkan titik tengah dari sebuah garis.
 - g. Method untuk mengecek apakah garis tersebut sejajar dengan sebuah garis lainnya, mengembalikan true jika sejajar dan false jika sebaliknya.
 - h. Method untuk mengecek apakah garis tersebut tegak lurus dengan sebuah garis lainnya, mengembalikan true jika sejajar dan false jika sebaliknya.
 - i. Method untuk menampilkan ke layar titik awal dan titik akhir garis.
 - j. Method untuk menampilkan persamaan garis dalam bentuk string $y = mx + c$.
Contoh: garis yang dibuat dari titik (-2,0) dan (0,4) mempunyai persamaan garis $y = 2x + 4$.

PELAPORAN

Pada akhir sesi praktikum, laporkan hasil praktikum pada link <http://tiny.cc/pbo25>.

Lengkapi semua file program yang harus dikerjakan dalam modul ini dan kumpulkan hasil akhirnya di kulon.

*****Selamat Mengerjakan dan Berlatih *****