

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJECT



oleh:

FARKHAN

20081010060

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR
2022

1. KOMPOSISI CLASS

Komposisi *class* pada java adalah korelasi desain antara dua kelas yang didasarkan pada asosiasi “*has-a*” dan “*whole/part*” yang disebut hubungan agregasi. Komposisi mengambil hubungan satu langkah lebih jauh dengan memastikan bahwa objek yang mengandung bertanggung jawab untuk seumur hidup objek yang dimilikinya. Jika objek B terkandung dalam objek A, maka objek A bertanggung jawab atas pembuatan dan penghancuran objek B.

1.1. SOAL dan JAWABAN

1.1.1. Buatlah program java untuk 2 class dengan ketentuan sebagai berikut.

- Class Point menyimpan sebuah titik (dalam koordinat x dan y)
- Constructor Point(x,y) menginisialisasi atribut x dan y di class Point
- Method getX() dan getY() pada class Point mengembalikan nilai variabel x dan y
- Method cetak() pada class Point mencetak nilai atribut x dan y ke layar
- Class PolyLine memiliki atribut points (bertipe Array of Object Point) yang menyimpan kumpulan titik-titik
- Constructor PolyLine(int size) menginisialisasi array points dengan sebuah array baru dan berukuran size
- Method appendPoint(Point p) menambahkan sebuah point p baru ke dalam array points
- Method cetak() pada class PolyLine melakukan looping di dalam array points untuk mencetak semua point yang tersimpan di dalamnya dalam format: (x1,y1)(x2,y2)(x3,y3) Dst

Jawaban :

Untuk menjawab soal di atas, maka dilakukan pembuatan program dengan penulisan kode beserta penjelasannya seperti di bawah ini

2. PENULISAN KODE

Di bawah ini merupakan kode dari file Point.java:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
   * default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
   * this template
4   */
5  package com.mycompany.compotition;
6
7  /**
8   *
9   * @author farhan
10  */
11 public class Point {
12     private int x;
13     private int y;
14
15     public Point(int x, int y) {
16         this.x = x;
17         this.y = y;
18     }
19
20     public int getX() {
21         return x;
22     }
23
24     public int getY(){
25         return y;
26     }
27
28     public void cetak() {
29         System.out.println ("x = " + x);
30         System.out.println ("y = " + y);
31     }
32 }
33
```

Pada baris 11, saya membuat class bernama Point. Pada baris 12 dan 13, saya membuat variabel x dan y dengan tipe data integer.

Pada class Benda ini, saya membuat constructor Point dengan parameter x dan y bertipe data integer yang digunakan untuk memasukkan nilai pada variabel x dan y yang dimiliki class. Selanjutnya saya membuat method getX() yang mengembalikan nilai dari variabel x dan method getY() yang mengembalikan nilai dari variabel y.

Selanjutnya saya membuat method cetak() bertipe void yang dapat menampilkan nilai dari variabel x dan y.

Di bawah ini merupakan kode dari file PolyLine.java.

```
1  /*
```

```

2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
   default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
   this template
4  */
5  package com.mycompany.compotition;
6
7  import java.util.Arrays;
8
9  /**
10   *
11   * @author farhan
12   */
13  public class PolyLine {
14      private Point[] points;
15      private int jumlahPoint;
16
17      public PolyLine(int size){
18          jumlahPoint = size;
19          this.points = new Point[size];
20      }
21
22      public void appendPoint(Point[] p) {
23          for (int i = 0; i < jumlahPoint; i++){
24              this.points[i] = new Point (p[i].getX(), p[i].getY());
25          }
26
27      public void cetak() {
28          for (int i=0; i < jumlahPoint; i++){
29              System.out.print("(" + points[i].getX() + "," + points[i].getY()
30              + "), ");
31          }
32
33      public static void main (String[] args) {
34          Point[] points = new Point[] {
35              new Point(4, 2),
36              new Point(8, 5),
37              new Point(1, 7)
38          };
39          points[1].cetak();
40
41          PolyLine pl1 = new PolyLine (3);
42          pl1.appendPoint(points);
43          pl1.cetak ();
44      }
45  }

```

Pada baris 13, saya membuat class dengan nama PolyLine. Clas ini memiliki atribut points bertipe array of object yang menggunakan class Point dan memiliki atribut jumlahPoint bertipe data integer yang saya tuliskan pada baris 14 dan 15.

Pada baris 17, saya membuat construct dengan parameter size bertipe data integer yang akan digunakan untuk nilai pada atribut jumlahPoint. Dengan parameter size, saya membuat atribut points memiliki panjang sejumlah size. Selanjutnya saya membuat methode

appendPoint() yang ditulis pada baris 22-25. Methode ini memerlukan satu parameter p yang bertipe array of object yang nilainya akan dimasukkan ke dalam atribut points. Di dalam methode ini, kita masukkan nilai dari p yang berisi titik x dan titik y ke dalam atribut points menggunakan perulangan. Methode selanjutnya yang saya buat ialah cetak(). Dalam methode ini, saya mencetak isi dari points yang berisi sekumpulan titik x dan y yang telah dimasukkan melalui methode appendPoint().

Selanjutnya saya buat fungsi main agar program dapat dijalankan. Pada baris 33, saya melakukan inisiasi terhadap class Point dengan nama points. Di dalam points, saya membuat beberapa objek dari class Point. Pada baris 39, saya mencoba mencetak nilai dari salah satu objek dari class Point. Pada baris 41, saya menginisiasi class PolyLine dengan nama pl1 dan memberikan argumen 3. Nilai argumen yang diberikan harus sama dengan jumlah objek yang dibuat dari class Point. Dari objek pl1, saya menggunakan methode appendPoint() dengan memberikan argumen points, yaitu objek yang dibuat dari class Point sebelumnya. Terakhir, saya cetak nilai dari PolyLine menggunakan methode cetak().

3. SCREENSHOT HASIL PERCOBAAN

Berikut ini merupakan tangkapan layar dari hasil dari program yang telah dibuat.

```
-----[ jar ]-----  
  
--- exec-maven-plugin:3.0.0:exec (default-cli) @ compotition ---  
x = 8  
y = 5  
(4,2), (8,5), (1,7),  
-----  
  
BUILD SUCCESS
```

Gambar 3.1. Luaran Program

Pada gambar 3.1. menunjukkan hasil dari program yang menunjukkan nilai x dan y dari salah satu objek dari class Point. Kemudian pada baris selanjutnya menunjukkan sekumpulan pasangan nilai x dan y yang terkumpul pada objek dari class PolyLine.