PENGHITUNGAN JARI-JARI PADA OBJEK LINGKARAN DALAM BAHASA JAVA

Warno
Program studi Teknik Informatika, Fakultas Tehnik, MIPA Universitas Indraprasta PGRI
Jl. Nangka No.58C Tanjung Barat (TB Simatupang), Jagakarsa, Jakarta Selatan
warnooke@gmail.com

Abstrak

Program Orentasai Obyek(POO) adalah pemrograman prosedural, yang membagi sebuah aplikasi berdasarkan berbagai macam fungsi yang ada didalamnya. Fungsi-fungsi tersebut bisa dieksekusi dari baris manapun dalam bagian program, dimana setiap fungsi terdiri dari serangkaian langkah-langkah pengeksekusian perintah untuk menyelesaikan sebuah aktivitas.

Kata kunci: jari-jari, objek lingkaran, bahasa java

Pendahuluan

Salah satu cara untuk mengurangi kompleksitas dari perangkat lunak adalah dengan membaginya berdasarkan hirarki tertentu berdasarkan fungsi-fungsi yang ada. Salah satu pendekatan pengembangan perangkat lunak sebelum adanya pendekatan berorientasi obyek (*Object oriented Programming* atau OOP) adalah pemrograman prosedural, yang membagi sebuah aplikasi berdasarkan berbagai macam fungsi yang ada didalamnya.

Mari kita ambil sebuah contoh yaitu sebuah mobil. Mobil adalah sebuah benda yang nyata ada dalam kehidupan. Kalau kita perhatikan bahwa sebuah mobil sebenarnya terdiri dari banyak komponen-komponen yang juga merupakan sebuah obyek, sebut saja mesin, pedal gas dan rem, spion, kursi penumpang, roda, dan lain-lain. Masing-masing komponen tersebut saling berinteraksi sehingga membentuk sebuah mobil yang akhirnya bisa dikendarai dan berjalan. Tak jauh beda dengan sebuah program atau aplikasi, terdiri dari banyak obyek-obyek yang saling berinteraksi.

Hasil dan Pembahasan

Metode Pemrograman berorientasi objek (Objek Oriented Programing, OOP) menawarkan suatu teknik pembuatan dan pengembangan program aplikasi dengan cara yang mudah dan menyenangkan. Berbagai program aplikasi berbasis windows dibangun dengan menggunakan metode pemrograman ini.

Objek

Paradigma objek merupakan salah satu konsep yang sangat diperlukan dalam menguasasi pemrograman berorientasi objek karena tanpa paradigma objek kita akan mengalami kesulitan di dalam memahami dan mempelajari pemrograman berorintasi objek.

Paradigma adalah suatu cara pandang atau cara berpikir sehingga paradigma objek bisa diartikan sebagai cara pandang yang memandang segala sesuatu sebagai objek.Semua aspek dalam pemrograman Java dapat kita anggap sebagai objek, terkecuali tipe data primitive karena semua library dan objek dalam java memiliki akar awal class java.lang.Object.Oleh karena itu kita dapat mulai membiasakan dengan pemikiran ini. Sebagai langkah awal kita dapat memperhatikan benda – benda sekitar yang merupakan objek nyata yang dapat dilihat.

Classification

Classification adalah suatu proses pembuatan class. Proses pembuatan class harus dilakukan sedemikian rupa sehingga seluruh data yang diperlukan oleh objek bisa didaftarkan. Untuk class

yang kompleks hal ini tentu merupakan pekerjaan yang rumit. Konsep OOP menawarkan kemudahan dengan cara mengembangkan class yang telah kita buat. Kita bisa saja membuat suatu class yang merupakan pengembangan carai class yang telah ada sebelumnya

a. Konsep Objek dan Class

Bahasa pemrograman Java merupakan bahasa pemrograman yang berorientasi objek sehingga konsep objek dan class menjadi penting apabila kita ingin menguasai pemrograman ini. Seringkali pemrograman berorintasi objek sukar untuk dipelajari pada awalnya, terutama apabila belum pernah mempelajari dan memiliki latar belakang dari bahasa yang bersifat procedural. Jadi perbedaan Class dan Objek adalah sebagai berikut :

1. Class merupakan desan dan objek merupakan perwujudan suatu class.
2. Class bersifat abstrak dan objek bersifat konkrit

Sebuah class atau objek nantinya memiliki dua komponen :

1. Data member (anggota data) yaitu variabel – variabel yang menyatakan karakteristik suatu objek. Sebuah objek lingkaran memiliki karakteristik yang dinyatakan dengan nilai jari – jari .
2. Member function, yaitu fungsi – fungsi yang bertugas memanipulasi nilai pada data – member. Fungsi yang paling sering ada pada sebuah objek adalah fungsi untuk mengubah dan menginformasikan nilai dari data – member objek.

Abstraction

Abstraction adalah suatu proses dimana kita melakukan desain class dan menentukan data dan method yang akan dimiliki oleh sebuah class. Sebuah bangun geometri lingkaran dideskripsikan dengan bentuk seperti dibawah ini :

Gambar 1
Lingkaran

Lingkaran memiliki jari – jari (radius atau r) untuk menyatakan ukurannya. Semakin besar nilai jari – jarinya semakin besar ukuran lingkaran tersebut dan sebaliknya. Jadi jari – jari adalah data yang dimiliki oleh sembarang lingkaran dan inilah yang menjadi karakteristik bangun geometri lingkaran.

Class lingkaran ini memiliki karakteristik sebagai berikut :

1. Memiliki jari – jari
2. Memiliki beberapa method
3. Mengubah nilai jari – jari
4. Memperoleh informasi jari – jari
5. Menghitung luas lingkaran.
6. Menghitung keliling lingkarann karakteristik tersebut di atas.

Langkah berikutnya adalah membuat format class lingkaran berdasarka

Algoritma dalam membentuk lingkaran.

```
Class Lingkaran
{
        data : jari – jari
        method : update jari – jari
              informasi jari – jari
                hitung luas
                hitung keliling
}
```

Sampai disini tahap perancangan class bisa dianggap selesai. Pada bagian selanjutnya kita akan melakukan implementasi class ini secara mendetail.

Encapsulation

Data yang ada pada suatu objek tidak boleh diubah secara langsung dari luar objek tersebut. Perubahan secara langsung dari luar objek dapat berakibat data – data didalam objek mengalami"bad value" yang pada akhirnya berimbas pada tidak berfungsinya sebuah objek seperti yang dikehendaki. Berikut ini diberikan kode program untuk class lingkaran secara lengkap. Perhatikan bagaimana kita melakukan "Pengamanan" terhadap data objek kata kunci privat digunakan untuk mengunci data atau method agar tidak terlihat dari luar objek sebaliknya kata kunci public digunakan untuk mempublish data atau method agar dikenal dari luar objek

Program -01

```
public class Lingkaran
{
private double jarijari;
public void setjarijari (double r)
        {if (r>0){jarijari=r;
                        }}
public double getjarijari()
        {return(jarijari);}
public double hitungluas()
        {return(Math.PI * jarijari * jarijari);
        } public double hitungkeliling()
        {return(Math.PI * 2 * jarijari);          }
}
```

Penamaan method mengikuti ciri – ciri dari kebanyakan perintah standar Java. Ciri – ciri tersebut adalah :

1. Method yang bertugas mengubah nilai kedalam objek diberi awalan **set**, misalnya setJariJari(). Nama Method setelah kata set sama dengan nama variabel data objek yang akan diubah nilainya.
2. Method yang betugas mengambil nilai dari dalam objek diberi awalan **get**, misalnya getLuas(). Nama method setelah kata get sama dengan nama variabel data objek yang akan diketahui nilainya.
3. Method dengan tugas selain point 1 dan 2 akan diberi nama sesuai dengan tugas yang dikerjakan oleh method tersebut.

Secara bagan kita bisa menggambarkan class lingkaran ini dengan bentuk sebagai berikut:

Lingkaran

1. Double jarijari
2. void setJariJari(double)
3. double getJarijari()
4. double hitungLuas()
5. double hitungKeliling()

Constructor

Metode OOP hadir dengan adanya sebuah method khusus yang disebut constructor yang akan dipanggil secara otomatis pada saat sebuah objek diciptakan. Kita tidak usah melakukan pemanggilan terhadap constructor ini, javalah yang akan melakukannya. Keunikan ini bisa kita manfaatkan untuk melakukan proses inisialisasi dilakukan melalui constructor. Isi constructor itu sendiri bisa berupa pemanggilan method internal atau instruksi lain.

Sebagaimana fungsi atau method pada umumnya, constructor juga bisa memiliki variasi parameter. Kita bisa menerapkan konsep function – overloading pada constructor. Secara umum ada tiga variasi constructor.

Singgle Constuctor

Program berikut ini merupakan modifikasi dari program – 01 dimana kita sudah menambahkan constructor pada kelas lingkaran perhatikan bagaimana constructor ini bekerja.

Program – 02

```
public class Lingkaran
{
        private double jarijari;
        public Lingkaran()
    {setJariJari(1);}
    public void setJariJari (double r)
    {if (r>0){jarijari=r;}
        }
        public double getjarijari()
    {return(jarijari);          }
        public double hitungluas()
    {return(Math.PI * jarijari * jarijari);}
        public double hitungkeliling()
    {return(Math.PI * 2 * jarijari);}
}
```

Karakteristik *constructor*

1. Didefinisikann public adalah sebagai awal dalam membuat program atau code.
2. Memiliki nama sama dengan nama class.

Kedua program diatas bisa kita compile tanpa kesalahan tapi tidak bisa kita jalankan karena kode–kode tersebut hanya merupakan implementasi class. Dengan kata lain kita baru saja membuat sebuah modul bernama lingkaran. Jika kita mencoba menjalankan kedua program ini java akan menampilkan pesan kesalahan yang menyatakan bahwa fungsi main (0 tidak ada dalam file).

Program berikut ini adalah program lengkap yang bisa decompile dan dijalankan. Program ini kita beri nama CobaLingkaran_1 sesuai dengan nama class utamanya.

Program – 03

```
class Lingkaran
{
        private double jarijari;
        public Lingkaran()
        {
                setJariJari(1);
        }
        public void setjarijari (double r)
        {
                if (r>0)
                {
                        jarijari=r;
                }
        }
        public double getjarijari()
        {
                return(jarijari);
        }
        public double hitungluas()
        {
                return(Math.PI * jarijari * jarijari);
        }
        public double hitungkeliling()
        {
                return(Math.PI * 2 * jarijari);
        }
}
public class CobaLingkaran_1
{
        public static void main (String [] argas)
        {
                Lingkaran abc = new Lingkaran();
                Lingkaran pqr = new Lingkaran();
                Lingkaran xyz = new Lingkaran();
                System.out.println (" Data default :");
                System.out.println (" _____ :");
                System.out.println (" Jari-jari ABC :" + abc.getJariJari());
                System.out.println ("Jari-jari PQR :" + pqr.getJariJari());
                System.out.println ("Jari-jari XYZ :" + xyz.getJariJari());
                System.out.println ();
                abc.setJariJari (5);
                pqr.setJariJari (3.2);
                xyz.setJariJari (1.8);
                System.out.println (" Data sekarang :");
                System.out.println (" _____ :");
                System.out.println (" Jari-jari ABC :" + abc.getJariJari());
                System.out.println ("Jari-jari PQR :" + pqr.getJariJari());
                System.out.println ("Jari-jari XYZ :" + xyz.getJariJari());
                System.out.println ();  }}
```

3. Constructor Dengan Parameter

Kekurangan dari pemakaian single constructor seperti pada program di atas adalah kita tidak bisa menentukan secara langsung nilai default jari – jari lingkaran. Setidaknya kita bisa memerlukan dua langkah untuk melakukan setting data jari – jari sesuai yang kita kehendaki :

- Memanggil objek
- Memanggil method setJariJari()

Program 03 akan kita modifikasi menjadi program 04, program ini kita beri nama Cobalingkaran _2

Program – 3

```
class Lingkaran
{
        private double jarijari;
        public Lingkaran(double r)
        {
                if ( r > 0 )
                {
                        JariJari=r;
                }
                else
                {
                        JariJari=1;
                }
        }
        public void setjarijari (double r)
        {
                if (r>0)
                {
                        jarijari=r;
                }
        }
        public double getjarijari()
        {
                Return(jarijari);
        }
        public double hitungluas()
        {
                return(Math.PI * jarijari * jarijari);
        }
        public double hitungkeliling()
        {
                return(Math.PI * 2 * jarijari);
        }
}
public class CobaLingkaran_2
{
        public static void main (String [] argas)
        {
```

```
        Lingkaran abc = new Lingkaran(6);
        Lingkaran pqr = new Lingkaran(8.9);
        Lingkaran xyz = new Lingkaran(4.8);
        System.out.println (" Data default :");
        System.out.println (" _____ :");
        System.out.println (" Jari-jari ABC :" +
        abc.getJariJari());
        System.out.println ("Jari-jari PQR :" +
        pqr.getJariJari());
        System.out.println ("Jari-jari XYZ :" +
        xyz.getJariJari());
        System.out.println ();
        }
    }
```

4. Multiple Construction

Seakan harus memenuhi semua keinginan. Kedua kondisi constructor di atas dirasakan belum bisa memenuhi keinginan user, ada user yang menginginkan constructor jenis pertama ada yang lebih suka pada model constructor jenis kedua. Oleh karena kondisi tersebutlah yang menyebabkan kita perlu memasukkan beberapa model constructor kedalam class, dengan kata lain kita akan membuat beberapa constructor yang bisa dipilih oleh user, semakin komplek class yang kita buat semakin banyak variasi constructor yang akan terjadi.

Program 04 berikut ini menyajikan bagaimana ide multiple - constructor diterapkan pada class lingkaran. Program ini kita beri nama CobaLingkaran_3

Program -04

```
class Lingkaran
{
    private double jarijari;
    public Lingkaran()
    {
        setJariJari(1);
    }
    public Lingkaran (double r)
  {
        if (r>0)
        {
            jarijari=r;
        }
        Else
        {
            jariJari=1;
        }
    }
    public void setJariJari(double r)
    {
        if (r>0)
        {
            jariJari=r;
          }
```

```
        }
        public double getjarijari()
    {
            return(jarijari);
    }
        public double hitungluas()
    {
        return(Math.PI * jarijari * jarijari);
    }
        public double hitungkeliling()
    {
        return(Math.PI * 2 * jarijari);
    }
}
public class CobaLingkaran_1
{
    public static void main (String [] argas)
    {
        Lingkaran abc = new Lingkaran();
        Lingkaran pqr = new Lingkaran(8.9);
            Lingkaran xyz = new Lingkaran(4.8);
        System.out.println (" Data default2 :");
            System.out.println (" _____ :");
            System.out.println (" Jari-jari ABC :" +
            abc.getJariJari());
            System.out.println ("Jari-jari PQR :" +
            pqr.getJariJari());
            System.out.println ("Jari-jari XYZ :" +
            xyz.getJariJari());
            System.out.println ();
            abc.setJariJari (5);
            pqr.setJariJari (3.2);
            xyz.setJariJari (1.8);
            System.out.println (" Data sekarang :");
            System.out.println (" _____ :");
            System.out.println (" Jari-jari ABC :" +
            abc.getJariJari());
            System.out.println ("Jari-jari PQR :" +
            pqr.getJariJari());
            System.out.println ("Jari-jari XYZ :" +
            xy z.getJariJari());
            System.out.println ();
    }
}
```

1. INHERITANCE

Inheritance bisa diterjemahkan secara bebas sebagai pewarisan. Yang dimaksud dengan Inheritance adalah konsep dimana kita bisa membuat sebuah class baru dengan mengembangkan class yang pernah dibuat sebelumnya, method OOP memungkinkan kita

memperoleh seluruh data dari class induk untuk diberikan pada class tanpa harus melakukan copy and paste seluruh kode induk.
Istilah dalam inheritance yang diperhatikan :

- Extends
- Keyword ini harus kita tambahkan pada definisi class yang menjadi sub class.
- Super class digunakan untuk menunjukkan hirarki class yang berarti class dasar sub class / class anak.
- Sub class adalah class anak atau turunan secara hirarki dari superclass.
- Keyword ini digunakan untuk memanggil konstruktor dari super class atau menjadi variabel yang mengacu pada super class.
- Method overriding
- Pendefinisian ulang method yang sama pada sub class.

Program 05 berikut ini merupakan implementasi java untuk class silinder. Kata kunci extends menyatakan bahwa suatu class merupakan turunan dari class lain pada baris 1 kita mendefinisikan class silinder sebagai turunan sebagai class lingkaran.

Program - 05

```
class Silinder extends Lingkaran
{
        private double tinggi;
        public Silinder ()
        {
                setJariJari (1);
                setTinggi (1);
        }
        public void setTinggi (double t)
        {
                if (t > 0 )
                {
                        Tinggi=t
                }
        }
        public double getTinggi()
        {
                Return(tinggi);
        }
        public double hitungluas()
        {
        Return(super.hitungLuas() * 2 + super.hitungKeliling() *
        tinggi);
        {
                Return(Math.PI * 2.0 * jarijari);
        }
}
```

Polymorphism

Secara mudah polymorphism bisa disamakan dengan method overloading, dimana didalam sebuah class terdapat beberapa method dengan nama sama. Kata Polymorphism memiliki arti kemampuan untuk memiliki bentuk atau wujud berbeda. Dalam istilah pemrograman, kata ini

memiliki arti kemampuan aksi berbeda bila method yang sama dipanggil, dimana aksi method tergantung dari tipe objeknya.

Kita juga perlu mengerti kondisi yang harus dipenuhi supaya polymorphism dapat diimplementasikan :

1. Method yang dipanggil harus melalui variable dari basis class atau super class.
2. Method yang dipanggil harus juga menjadi method dari basis class.
3. Signature method harus sama baik pada super class maupun sub class
4. Method acces attribute pada subclass tidak boleh lebih terbatas dari basis class.

Dari penjelasan saja mungkin kita akan sulit memahami dengan jelas mengenai konsep polymorphism. Oleh karena itu kita memperhatikan contoh penggunaan polymorphism berikut ini.

Program – 07

```
//-----------------------------------------
//Definisi class Tampildata
//-----------------------------------------
class Tampildata
{
        public void tulisData (String data)
        {
                System.out.println(data);
        }
        public void tulisData(int data);
        {
                System.out.println(data);
        }
        public void tulisData (double data)
        {
                System.out.println(data);
        }
        public void tulisData (char data)
        {
                System.out.println(data);
        }
}
//-----------------------------------------
//Definisi class Utama
//-----------------------------------------
public class Polymorphism
{
        public static void main (String{} argas)
        {
        TampilData xyz = new TampilData();
        System.out.println("Cetak data String");
        xyz.tulisdata("Universitas Indraprasta, Jakarta");
        System.out.println("Cetak data Integer");
        xyz.tulisdata(530);
        System.out.println("Cetak data Double");
        xyz.tulisdata(22.0/7.0);
        System.out.println("Cetak data CHAR");
        xyz.tulisdata("U");
        }}
```

Package

Package adalah sebuah direktori yang berisi kumpulan file.class dengan kegunaan spesifik berbeda dengan bahasa lain java mengorganisasi package dalam bentuk direktori dimana dalam direktori ini bisa terdapat satu atau lebih file java yang telah decompile atau dalam bentuk class di dalam direktori tersebut bisa juga terdapat satu atau lebih sub – direktori yang memiliki file.class atau sub-direktori lain. Instruksi untuk meng-compile class java menjadi package adalah :

Javac –d <directory_target><nama_file_**java**>

Diatas kita telah membuat dua class : lingkaran dan silinder sekarang kita akan membuat kedua class tersebut menjadi package agar program lain yang menggunakannya tidak menyertakan kode kedua class ini didalamnya.

Program – 08

```
package modul;
public class Lingkaran
{
    private double jarijari;
        public Lingkaran()
        {
                setJariJari (1);
        }
        public void setJariJari (double r)
        {
                if (r>0)
                {
                JariJari=r;
                }
        }
        public double getJariJari()
        {
            return(jariJari);
        }
        public double hitungLuas()
        {
                return(Math.PI * JariJari * JariJari);
        }
        public double hitungKeliling()
        {return(Math.PI * 2 * JariJari);}}
```

Kita teruskan dengan melakukan modifikasi class Silinder beberapa hal harus dipahami terlebih dahulu :

1. Class Silinder merupakan turunan dari class Lingkaran.
2. Class lingkaran harus sudah decompile terlebih dahulu.
3. Di dalam Class silinder harus ada instruksi yang mengarahkan java agar
4. mengambil modul Lingkaran dari directori modul.
5. Class silinder itu sendiri harus menyertakan package modul agar hasil compilenya disimpan di directory modul.

Kesimpulan

Setelah mencoba memikirkannya, kita tentu mengambil kesimpulan untuk mengambil foto mereka lalu menscannya sehingga dapat dimengerti computer. Komputer dapat memahami bahasa mesin di mana bahasa mesin ini sangat sulit untuk kita pahami sebagai manusia. Oleh karena iu diperlukan bahasa pemrograman yang menjadi jembatan dengan bahasa mesin tersebut. Bahasa pemrograman Java merupakan salah satu sarana bagi kita untuk mengolah data dan memberikan instruksi untuk diproses dan dijalankan oleh *computer*. Dengan adanya mencoba belajar bahasa java maka kita harus rajin mencoba dan mencoba berlatih dalam membuat salah satu aplikasi yang sederhana sehingga dapat menyelesaikan suatu masalah dari terkecil sampai dengan masalah yang besar untuk itu jika anda ingin menjadi seorang program maka seringlah berlatih.

Daftar Pustaka

Setiyo cahyono, Februari 2006 penerbit teknik informatika bandung.

Luwis @2011, penerbit PT elex Media Komputindo, Jakarta.

Deitel, H.M. Java How To Program, sixth Edition. 2004, Prentice Hall.

Exercise 10, Introduction to Computer Science Januari 2003, TU Darmstadt

H.M Deitel, Java How to Program, Sixth Edition. Prentice Hall, August 2004

Rickyanto, Isak, *Dasar Pemrograman Berorientasi Objek*, Yogyakarta : Andy , 2002

Purnama, Rangsang, *Tuntutan Pemrograman Java*, Jakarta : Prestasi Pustaka Publisher.