MODUL I PRAKTIKUM PEMROGRAMAN LANJUT

Disusun Oleh :
Candra Dewi, S.Kom,. M.Sc.
Adharul Muttaqin, ST., MT.
Budi Darma Setiawan, S.Kom., M.Cs

Laboratorium Komputer Dasar
Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya Malang
2015

KATA PENGANTAR

Puji syukur kehadirat Allah SWT karena atas rahmat dan hidayah-Nya, tim pelaksana dapat menyelesaikan tugas perbaikan Modul Praktikum Pemrograman Lanjut dengan sebaik-baiknya.

Modul ini sebelumnya di tulis oleh 3 dosen dari PTIIK oleh Ibu Candra Dewi, S.Kom., M.Sc., Bapak Adharul Muttaqin, ST., MT., dan Ahmad Afif Supianto, S.Si., M.Kom. Dalam pelaksanaannya, dirasa perlu dilakukan penambahan dan perubahan disana-sini sehingga modul yang ada diharapkan dapat semakin mempermudah praktikan dalam memahami materi yang ada. Oleh karena itu dibentuk suatu tim pelaksana perbaikan dengan persetujuan Ketua Laboratorium Komputer Dasar PTIIK UB untuk menunaikan kebutuhan ini.

Pada modul ini, lebih praktikan lebih diarahkan untuk mencoba materi praktikum dengan melakukan kompilasi program yang ada di modul yang kemudian menjawab pertanyaan yang ada setahap demi setahap, untuk memudahkan pemahaman terhadap materi praktikum yang disampaikan. Praktikan diharapkan juga dapat melakukan dan mencoba sendiri setiap pertanyaan dan pelaksanaan percobaan yang dilakukan sehingga praktikan dapat lebih memahami materi praktikum yang ada berdasarkan pengalaman yang telah dilakukan sendiri.

Tak ada gading yang tak retak, kami menyadari modul praktikum ini juga masih jauh dari kata sempurna. Kritik dan saran yang membangun kami harapkan dapar memperbaiki kemajuan dan kualitas modul praktikum Pemrograman Lanjut ini di masa mendatang.

Akhirnya semoga modul ini dapat memberikan manfaat bagi segenap praktikan.

Selamat belajar.

TIM PELAKSANA

Tim pelaksana perbaikan Modul Praktikum Pemrograman Lanjut

Penanggung Jawab / Pengarah	:	Ketua Laboratorium Komputer Dasar PTIIK Universitas Brawijaya
Koordinator Modul	:	Andriansyah Yusuf Rizal
Bab 1. Class dan Object	:	Andriansyah Yusuf Rizal
Bab 2. Static Method dan Overloading	:	Andriansyah Yusuf Rizal
Bab 3. Constructor dan Instance Method	:	Andriansyah Yusuf Rizal
Bab 4. Encapsulation	:	Anandhi Tristiaratri
Bab 5. Inheritance	:	Wiratama Paramasatya
Bab 6. Polymorfisme	:	Vera Rusumalawati
Bab 7. Interface	:	Sabrina Nurfadilla
Bab 8. Graphic User Interface	:	Abdullah Muhammad Farouk Hakim
Bab 9. Graphic User Interface Lanjut	:	Anandita A.Sasmito, Irnayanti Dwi Kusuma
Editor	:	Wiratama Paramasatya
Soal Latihan	:	Andriansyah Yusuf Rizal
Tata Letak / Setting	:	Andriansyah Yusuf Rizal, Anandhi Tristiaratri
Pencetakan dan penerbitan modul	:	Fransiscus Priharsono, S.Kom
Dokumentasi	:	Laboran Laboratorium Komputer Dasar PTIIK Univeristas Brawijaya

:
:

DAFTAR ISI

Sampul 1
Kata Pengantar 2
Tim Pelaksana 3
Daftar Isi 4
Bab 1. Class dan Object 5
Pengenalan OOP 5
Class 5

Bab 2. Static Method dan Overloading 9
Static Method 9
Overloading Method 9

Bab 3. Constructor dan Instance Method 14
Constructor 14
Instance Method 15

Daftar Pustaka 21

BAB 1
CLASS DAN OBJECT

Tujuan

1. Praktikan mampu memahami konsep OOP (Object Oriented Programming) dari bahasa Java
2. Praktikan mampu membuat class sebagai bentuk implementasi dari bab ini

Ringkasan Materi

A. Pengenalan OOP

OOP adalah sebuah konsep/cara pemrograman dengan menggunakan objek sebagai elemen dasar dari program. Jika kita memperhatikan dunia nyata, kita dapat menemukan beragam objek disekitar kita seperti mobil, singa, manusia dan seterusnya. Objek yang dimaksud di sini, dikarakterisasi oleh atribut dan tingkah lakunya.

Contohnya, objek sebuah mobil mempunyai atribut tipe transimisi, warna dan manufaktur. Objek Mobil juga mempunyai tingkah laku berbelok, mengerem dan berakselerasi. Dengan cara yag sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari objek singa. Coba perhatikan tabel dibawah ini sebagai contoh perbandingan :

Objek	Atribut	Tingkah Laku
Mobil	Setir/kemudi Rem Pegas	Berbelok Mengerem Mempercepat
Singa	mata kaki Taring	Tidur Berlari Memangsa

Dengan deskripsi ini, objek pada dunia nyata dapat secara mudah diasumsikan sebagai objek perangkat lunak menggunakan atribut sebagai data dan tingkah laku sebagai method. Data dan method dapat digunakan dalam pemrograman game atau perangkat lunak interaktif untuk membuat simulasi objek pada dunia nyata. Contohnya adalah perangkat lunak objek mobil dalam permainan balap mobil ataupun singa dalam perangkat lunak pendidikan interaktif untuk anak-anak.

B. Class

Class adalah struktur dasar dari OOP. Class inilah yang nantinya digunakan sebagai *template* atau cetakan dari sebuah objek. Pembentukan objek dilakukan dengan menggunakan class. Class terdiri dari 2 dua komponen yang disebut dengan fied (menggambarkan attribut/properti) dan method (menggambarkan tingkah laku). Field merupakan tipe data yang didefinisikan oleh class, sementara method merupakan operasi. Sedangkan objek adalah sebuah instance dari class. Untuk dapat membedakan class dan objek, dapat dilihat contoh pada tabel dibawah ini :

Class Mobil		Objek Mobil A	Objek Mobil B
Variabel instance	Nomor plat	N 7221 WZ	N 2010 KT
	Warna	Biru	Merah
	Manufaktur	Mitsubishi	Toyota
	Kecepatan	50 km/h	100 km/h

Ketika diinisialisasi, setiap objek mendapat satu set variabel yang baru. Bagaimanapun, implementasi dari method dibagi diantara objek pada class yang sama. Class menyediakan keuntungan dari *reusability* artinya programmer perangkat lunak dapat menggunakan sebuah class beberapa kali untuk membuat objek (blueprint).

Untuk mendefinisikan class dapat dituliskan sebagai berikut

```
<modifier> class <name> {
   <attribut declaration>
   <constructor declaration>
  <method declaration>
}
```

Contoh :

```
public class mobil{
    int a;
    public mobil(int nilai){
      a = nilai;
    }
    Publig int getNilai(){
      return a;
    }
}
```

Instansiasi

Instansiasi adalah proses untuk membuat objek dari sebuah class. Membuat instan Objek dari sebuah class dilakukan dengan menggunakan kata kunci ***new***. Contohnya pada suatu kasus kita memiliki Class bernama mobil dan kita ingin menginstan objek dari class Mobil pada class mainMobil dan kita beri nama mobil_A.

```
Mobil.java
public class Mobil{
}

mainMobil.java
public class mainMobil{
  public static void main(String[] args){
    Mobil mobil_A = new Mobil();
  }
}
```

Pelaksanaan Percobaan

A. Class

Ketikkan program di bawah ini

Mobil.java

```
1  public class Mobil {
2      private String noPlat;
3      private String warna;
4      private String manufaktur;
5      private int kecepatan;

6      public void setNoPlat(String s){
7          noPlat = s;
8      }
9      public void setWarna(String s){
10         warna = s;
11     }
12     public void setManufaktur(String s){
13         manufaktur = s;
14     }
15     public void setKecepatan(int i){
```

```
16          kecepatan = i;
17      }
18      public void displayMessage(){
19          System.out.println("Mobil anda adalah bermerek
    "+manufaktur);
20          System.out.println("mempunyai nomor plat "+noPlat);
21          System.out.println("serta memililki warna "+warna);
22          System.out.println("dan mampu menempuh kecepatan
    "+kecepatan);
23      }
24  }
```

Selanjutnya kita akan membuat main class dengan MainMobil dan menginstan objek baru dari class tersebut. Perhatikan pada baris 4 dan 12 terdapat deklarasi ***new*** yang artinya perintah untuk menginstan objek baru dari class mobil

MainMobil.java

```
1   public class MainMobil {
2       public static void main(String[] args) {
3           //instan objek bernama m1
4           Mobil m1 = new Mobil();
5           m1.setKecepatan(50);
6           m1.setManufaktur("Toyota");
7           m1.setNoPlat("AB 1231 UA");
8           m1.setWarna("Merah");
9           m1.displayMessage();
10          System.out.println("================");
11          //instan objek baru bernama m2
12          Mobil m2 = new Mobil();
13          m2.setKecepatan(100);
14          m2.setManufaktur("Mitsubishi");
15          m2.setNoPlat("N 1134 AG");
16          m2.setWarna("Pink");
17          m2.displayMessage();
18          System.out.println("================");
19          //merubah warna dari objek m1
20          System.out.println("mobil pada objek m1 di rubah menjadi
    warna hijau");
21          m1.setWarna("Hijau");
22          //menampilkan hasil perubahan
23          m1.displayMessage();
24      }
25  }
```

Data dan Analisis hasil percobaan

A. Class

Pertanyaan

1. Apakah yang disebut dengan variabel instance dan lokal variabel? Jelaskan perbedaanya!
   ……..
   ……..

2. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!
   ……..

……..

3. Rubah kode pada mainMobil diatas menjadi proses meminta masukan dari user dan buat menjadi interaktif!
   ……..
   ……..

4. Tambahkan method pada class mobil bernama setWaktu yang berparameter double, yang kemudian disimpan pada variabel waktu! (Ketetuannya adalah user harus menginputkan dalam satuan jam)
   ……..
   ……..

5. Tambahkan method bernama rubahSekon mempunyai parameter bertipe double dan hanya dapat dipanggil pada class mobil. Method ini memiliki fungsi untuk merubah masukan user yaitu jam menjadi sekon. Method tersebut di panggil pada method setWaktu dengan nilai parameter adalah nilai dari variabel parameter method setWaktu!
   ……..
   ……..

6. Tambahkan method pada class mobil dan hanya dapat dipanggil pada class mobil bernama rubahKecepatan yang mempunyai fungsi untuk merubah format kecepatan yang awalnya km/h menjadi m/s. Dipanggil di method setKecepatan!
   ……..
   ……..
7. Tambahkan method pada class mobil bernama hitungJarak yang mempunyai aksi untuk menghitung jarak yang dapat di tempuh oleh mobil dengan rumus jarak = kecepatan * waktu!
   ……..
   ……..

8. Tambahkan informasi jarak yang dapat ditempuh pada method displayMessage kemudian rubah satuannya yang awalnya m (meter) menjadi km (kilometer)!
   ……..
   ……..

9. Mahasiswa A ingin menulis pada sebuah buku tulis yang ingin dia miliki, isi lembar buku tersebut adalah 50 lembar. Setiap harinya ia menulis sebanyak 100 kata perhari yang cukup untuk 1/2 halaman buku. Buatlah rumus untuk menghitung berapa lama ia menghabiskan 1 buku tersebut serta identifikasilah objek, dan karakteristiknya kemudian implementasikan dalam bentuk class.
   ……..
   ……..

Tugas Praktikum

Sebelum mengerjakan soal di bawah ini tentukan dahulu objek, attribut, behaviour dan class.

1. Buatlah sebuah sistem sederhana yang menyerupai Sistem Informasi Akademik Mahasiswa (SIAM), dengan ketentuan user menginputkan Nama, Nim, IP serta jurusan. Selain itu mahasiswa juga dapat memasukkan kode Mata kuliah, Nama Mata kuliah dan jumlah sks mata kuliah tersebut. Jumlah sks yang di ambil harus sesuai dengan IP yang di dapat pada semester lalu. Buat skenario dengan banyak mahasiswa minimal 3 orang.

BAB 2
STATIC METHOD DAN OVERLOADING

Tujuan

1. Praktikan mampu memahami konsep static method yang ada di java
2. Mampu membedakan perbedaan method yang menggunakan kata kunci static atau tidak
3. Mampu memahami dan mengimplementasikan method overloading

Ringkasan Materi

A. Static Method

Seperti diketahui bahwa setiap class pada java mengimplementasikan method untuk melakukan aksi. Sebagai contoh adalah proses memasukkan sebuah data dari keyboard, dimana dapat dilakukan dengan memanggil method dari object *scanner* yang di inisialisasikan dari constructor untuk meng-input-kan dari input stream *(system.in)*. Proses ini dapat dilakukan dengan meng-instance obyek dari Class. Kemudian dari instan objek tersebut dapat kita panggil method yang sesuai. Kadangkala kita tidak perlu melakukan instansiasi class terlebih dahulu untuk memanggil method yang ada pada class di library java, namun cukup memanggil nama method. Inilah yang disebut dengan method *static*. Untuk mendeklarasikan method static, dapat dilakukan dengan meletakkan keyword ***static*** sebelum tipe method yang digunakan (return atau void). Method static dapat dipanggill dengan menyebutkan nama kelas dan diikuti dengan dot (.) seperti sintaks berikut:

ClassName.methodName(arguments)

Contoh dari implementasi *static method* adalah pemanggilan method **pow()** dari class Math, dimana cukup dipanggil dengan perintah : *Math.pow().*

Namun sebaliknya jika method yang kita panggil adalah method yang bukan static, maka kita perlu menginstan obyek dari class tersebut dan kemudian memanggil method tersebut. Sebagai contoh adalah pemanggilan method *getAlignmentY()* pada kelas *JPanel* berikut:

Jpanel j = new Jpanel();
j.getAlignmentY();

B. Overloading Method

Penamaan method pada OOP (Object Oriented Programming) menjadi sangat penting terutama pada pemrograman menggunakan bahasa java. Dalam penamaan method, terkadang tanpa sadar kita memberi nama yang sama pada method yang berbeda sehingga dapat mengakibatkan kesalahan pada saat program dijalankan. Untuk mengatasi hal ini, Java memperkenalkan istilah *overloading*, Overloading adalah teknik penamaan method dengan nama yang sama namun memiliki tipe dan jumlah argumen atau parameter yang berbeda. Sebagai contoh adalah method Hitung pada class Lingkaran, dimana pada class ini terdapat method bernama Hitung dengan parameter a dengan tipe integer.

```
public class Lingkaran{
   public static void Hitung(int a){
   //kode program
   }
}
```

Kemudian pada class tersebut dibuat method baru bernama Hitung namun parameternya bertipe double dengan nama value

```
public static void Hitung(double value){
   //kode program
}
```

Kedua method ini disebut overloading method karena memiliki nama yang sama tetapi tipe dari argumennya berbeda.

Pelaksanaan Percobaan

A. Static Method

Ketikkan program di bawah ini

Aritmatika.java

```
1  public class Aritmatika {
2      public void hitungPenjumlahan(int a,int b){
3          int nilai = a+b;
4          System.out.println("nilai penjumlahan adalah : "+nilai);
5      }
6      public static void hitungPerkalian(int a, int b){
7          int nilai = a*b;
8          System.out.println("nilai perkalian adalah : "+nilai);
9      }
10     public static void hitungPengurangan(int a, int b){
11         int nilai = a-b;
12         System.out.println("nilai pengurangan adalah : "+nilai);
13     }
14
15 }
16
17
```

Selanjutya adalah membuat class main dari class Aritmatika bernama MainAritmatika dan di beri instan objek untuk memanggil method static dan non static.

MainAritmatika.java

```
1  import java.util.Scanner;
2  public class MainAritmatika {
3      public static void main(String[] args) {
4          Scanner in = new Scanner(System.in);
5          System.out.print("masukkan nilai 1 : ");
6          int nil1 = in.nextInt();
7          System.out.print("masukkan nilai 2 : ");
8          int nil2 = in.nextInt();
9          //memanggil method static
10         Aritmatika.hitungPengurangan(nil1, nil2);
11         System.out.print("masukkan nilai 1 : ");
12         nil1  = in.nextInt();
13         System.out.print("masukkan nilai 2 : ");
14         nil2 = in.nextInt();
15         //memanggil method static
16         Aritmatika.hitungPerkalian(nil1, nil2);
17         System.out.print("masukkan nilai 1 : ");
18         int value1  = in.nextInt();
19         System.out.print("masukkan nilai 2 : ");
```

```
20          int value2 = in.nextInt();
21          //memanggil method NONstatic harus melalui objek
22          Aritmatika a = new Aritmatika();
23          a.hitungPenjumlahan(value1, value2);
24      }
25  }
```

B. Overloading Method

Ketikkan program di bawah ini

```
1   import java.util.Scanner;
2   public class Overloading {
3      public static void HitungLuas(int a,int b){
4         int nilai = a*b;
5         System.out.println("maka hasil luas : "+nilai);
6      }
7      public static double HitungLuas(double value, double value2){
8         double nilai = value* value2;
9         return nilai;
10     }
11     public static void main(String[] args) {
12        Scanner in = new Scanner(System.in);
13        System.out.print("masukkan nilai integer 1 : ");
14        int integer1 = in.nextInt();
15        System.out.print("masukkan nilai integer 2 : ");
16        int integer2 = in.nextInt();
17        HitungLuas(integer1, integer2);
18        System.out.print("masukkan nilai double 1 : ");
19        double double1 = in.nextDouble();
20        System.out.print("masukkan nilai double 2 : ");
21        double double2 = in.nextDouble();
22        HitungLuas(integer1, integer2);
23        System.out.println("Maka        hasil        luas        :
24        "+HitungLuas(double1, double2));
25     }
26  }
```

Data dan Analisis hasil percobaan

A. Static Method

Pertanyaan

1. Apakah yang disebut dengan static variabel? Dan apa fungsi dari static variabel serta kapan kita dapat menggunakan static variabel?

   ……..
   ……..

2. Mengapa pada main method harus dituliskan static? Jelaskan jawaban anda beserta dengan alasan!

   ……..
   ……..

3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

   ……..
   ……..

4. Jika pada tubuh method hitungPenjumlahan ditambahkan syntax *hitungPerkalian(a,b)* apa yang terjadi? Jelaskan?
   ……..
   ……..

5. Jika pada tubuh method hitungPerkalian ditambahkan syntax *hitungPenjumlahan(a,b)* apa yang terjadi? Jelaskan?
   ……..
   ……..

6. Tambahkan method non static dengan nilai balikan double untuk menghitung pembagian dengan parameter String nil dan String nil2, dan panggil method tersebut pada method main!
   ……..
   ……..

B. Overloading Method

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!
   ……..
   ……..

2. Jika pada baris 7, pada parameter double value dan double value2 di hapus dan di ganti menjadi int a dan int b apa yang terjadi? Jelaskan!
   ……..
   ……..

3. Rubah method pada baris ketujuh menjadi method bertipe void, dan lakukan juga perubahan main method.
   ……..
   ……..

4. Tambahkan method non static bertipe void bernama HitungLuas yang mempunyai parameter String bernama value1 dan String bernama value2! Kemudian panggil method tersebut pada main method!
   ……..
   ……..

5. Tambahkan method non static bernama cekKondisi yang mempunyai parameter bertipe double bernama value. Method ini di gunakan untuk mengecek nilai value yang merupakan nilai dari luas sebuah bidang. Jika nilai value bernilai lebih dari sama dengan 60 dan kurang dari sama dengan 100 maka akan mencetak “Nilai luas bidang termasuk kategori Besar”, jika nilai value bernilai lebih dari sama dengan 101 maka akan mencatak “Nilai luas bidang termasuk kategori sangat besar”, dan jika selain kondisi di atas akan mencetak “Nilai luas bidang termasuk kategori kecil”. Kemudian panggil method ini di dalam method main!
   ……..
   ……..

Tugas Praktikum

1. Soal 1

   Susun program dengan menggunakan overloading function dengan ketentuan :
   Terdapat method bernama overloadingMeth berparameter String dan integer, dimana method tersebut mempunyai fungsi untuk merubah input teks menjadi bilangan dan input bilangan menjadi teks
   Misal :
   Input : overloadingMeth(71)
   Output : tujuh puluh satu
   Input : overloadingMeth(tiga puluh lima)
   Output : 35
   Range untuk input parameter adalah 0-100

2. Soal 2

   Buatlah program dengan menggunakan class untuk menghitung penjumalahan, pengurangan, perkalian dan pembagian. Method penjumlahan dan pengurangan menggunakan static method sedangkan sisanya menggunakan method non static. Tambahkan method bertipe non static bernama Sederhana untuk menyederhanakan sebuah pecahan.

BAB 3
Constructor dan Instance Method

Tujuan

1. Praktikan dapat mendeklarasikan konstruktor, membuat default konstruktor dan overloading konstruktor dari class yang sudah mereka buat
2. Praktikan mampu membuat Instance Method pada class yang telah di buat

Ringkasan Materi

A. Constructor

Constructor sangatlah penting pada pembentukan sebuah object. Constructor adalah method dimana seluruh inisialisasi object ditempatkan. Saat kita menginstan sebuah object pada main class atau class lain, kita sebenarnya telah memanggil sebuah konstruktor pada sebuah class yang kita instan objeknya.

Berikut ini adalah property dari construstor :

1. Constructor memiliki nama yang sama dengan class
2. Constructor tidak memilki return value, meskipun void
3. Constructor tidak dapat dipanggil secara langsung, namun harus dipanggil dengan menggunakan operator ***new*** pada saat menginstan objek dari class

Untuk mendeklarasikan sebuah constructor dapat kita tuliskan dengan sintaks berikut :

```
<modifier> <classname> (parameter){
     <statement>
}
```

Contoh : misalnya dibuat constructor pada class mahasiswa :

```
public class mahasiswa{
    public mahasiswa(){
       //statement
   }
}
```

A.1 Default Constructor

Setiap class memiliki default constructor. Sebuah default constructor adalah constructor yang tidak memiliki parameter apapun. Jika didalam class tidak didefinisikan constructor apapun, maka sebuah default constructor akan dibentuk secara implisit oleh Java.

Sebagai contoh, pada class mahasiswa, bentuk default constructor akan terlihat dibawah ini :

```
public mahasiswa(){
  //area inisialisasi kode
}
```

A.2 Overloading Constructor

Tidak hanya method saja yang memiliki sifat overloading, constructor juga dapat dibuat overloading. Sama dengan halnya overloading method, overloading constructor adalah constructor dengan nama yang sama namun memliki jumlah atau tipe parameter yang berbeda. Contoh dari overloading method adalah sebagai berikut :

```
public Mahasiswa(){
   //area inisialisasi kode
}
```

```
    Public Mahasiswa(String temp){
        this.name = temp;
    }
    Public Mahasiswa(String name, String address){
        this.name = name;
        this.address = address
    }
    public Mahasiswa(String mGrade, double eGrade, double
sGrade){
        mathGrade = mGrade;
        englishGrade = eGrade;
        scienceGrade = sGrade;
    }
```

A.3 Menggunakan Constructor

Untuk menggunakan constructor kita dapat menggunakan kode-kode sebagai berikut :

```
    public static void main(String[]{
        //membuat 3 objek
        Mahasiswa m1 = new Mahasiswa("Anna");
        Mahasiswa m2 = new Mahasiswa("Chris","Malang");
        Mahasiswa m3 = new Mahasiswa(80,90,100);

    }
```

B. Instance Method

Sebuah class juga memiliki method yang dikaitkan dengan instan tertentu. Sewaktu method instan dipanggil, dia akan mengakses data yang terdapat pada instan yang dikaitkannya. Untuk lebih jelasnya mari kita melihat pada pelaksanaan percobaan bagian instance method.

Pelaksanaan Percobaan

A. Constructor

Student.java

```
1  public class Student {
2      private String name;
3      private String address;
4      private int age;
5      private double mathGrade;
6      private double englishGrade;
7      private double scienceGrade;
8      private double average;
9      public student(){
10         name = "";
11         address = "";
12         age = 0;
13     }
14     public Student(String n, String a, int ag){
15         name = n;
16         address = a;
17         age = ag;
18     }
19     public void setName(String n){
```

```
20          name = n;
21      }
22      public void setAddress(String a){
23          address = a;
24      }
25      public void setAge(int ag){
26          age = ag;
27      }
28      public void setMath(int math){
29          mathGrade = math;
30      }
31      public void setEnglish(int english){
32          englishGrade = english;
33      }
34      public void setScience(int science){
35          scienceGrade = science;
36      }
37      private double getAverage(){
38          double result = 0;
39          result = (mathGrade+scienceGrade+englishGrade)/3;
40          return result;
41      }
42      public void displayMessage(){
43          System.out.println("Siswa dengan nama "+name);
44          System.out.println("beramalat di "+address);
45          System.out.println("berumur "+age);
46          System.out.println("mempunyai    nilai    rata    rata
47  "+getAverage());
48      }
49  }
```

Ketikkan program di bawah ini

MainStudent.java

```
1   public class MainStudent {
2       public static void main(String[] args) {
3           Student anna = new Student();
4           anna.setName("Anna");
5           anna.setAddress("Malang");
6           anna.setAge(20);
7           anna.setMath(100);
8           anna.setScience(89);
9           anna.setEnglish(80);
10          anna.displayMessage();
11
12          //menggunakan constructor lain
13          System.out.println("====================");
14          Student chris = new Student("Chris", "Kediri", 21);
15          chris.setMath(70);
16          chris.setScience(60);
18          chris.setEnglish(90);
19          chris.displayMessage();
20
21
```

```
22          //siswa dengan nama anna dirubah informasi alamat dan
23  umurnya melalui constructor
24          System.out.println("====================");
25          anna = new student("anna", "Batu", 18);
26          anna.displayMessage();
28
29          //siswa denagan nama chris dirubah informasi alamat dan
30  umurnya melalui method
31          System.out.println("====================");
32          chris.setAddress("Surabaya");
33          chris.setAge(22);
34          chris.displayMessage();
35      }
    }
```

B. Instance Method

Ketikkan program di bawah ini

Rasional.java

```
1   public class Rasional{
2       private int pembilang, penyebut;
3       public Rasional(){
4         pembilang=0;
5         penyebut=0;
6       }
7       public Rasional(int pbl, int pyb){
8         pembilang=pbl;
9         penyebut=pyb;
10      }
11        //mengecek suatu bilangan adalah rasional atau bukan
12      public boolean isRasional(){
13          return (penyebut!= 0);
14      }
15        //menyederhanakan bilangan rasional
16      public void Sederhana(){
18          int temp, A, B;
19        if (penyebut ==0){
20              return;
21        }
22        A = (pembilang<penyebut) ? penyebut:pembilang;
23        B = (pembilang<penyebut) ? pembilang:penyebut;
24
25        while (B != 0){
26              temp= A % B;
28              A = B;
29              B = temp;
30        }
31
32        pembilang /=A;
33        penyebut /=A;
34      }
35      public double Cast(){
36          return  (penyebut==0.0)  ?  0.0  :  (double)pembilang  /
37  (double)penyebut;
```

```
38      }
39        //oprator >
40      public boolean moreThan (Rasional A){
41          return (pembilang * A.penyebut > penyebut * A.pembilang
42  );
43      }
44        //operator Unary- ---> A = -A
45      public void negasi(){
46          pembilang = - pembilang;
47      }
48        //operator unary += \
49      public void unaryPlus(Rasional A){
50          pembilang = pembilang * A.penyebut + penyebut *
51  A.pembilang;
52        penyebut *=A.penyebut;
53      }
54      public void cetak(){
55          System.out.println(pembilang + "/" + penyebut);
56      }
57  }
```

Ketikkan program di bawah ini yang bertindak sebagai main program

RasionalDemo.java

```
1   public class RasionalDemo{
2       public static void main(String[] args){
3           Rasional R1 = new Rasional(1,2);
4         Rasional R2 = new Rasional(1,3);
5
6           System.out.println("R1.isRasional: " + R1.isRasional());
7           System.out.println("R2.isRasional: " + R1.isRasional());
8         System.out.println();
9
10          System.out.println("R1 > R2 : " + R1.moreThan(R2));
11        System.out.println();
12
13        System.out.print("R1 : ");
14        R1.cetak();
15        System.out.print("R2 : ");
16        R2.cetak();
18        System.out.println();
19
20        R1.Sederhana();
21        R2.Sederhana();
22
23        System.out.print("R1 : ");
24        R1.cetak();
25        System.out.print("R2 : ");
26        R2.cetak();
28        System.out.println();
29
30          System.out.println("Setelah dilakukan Cast ke double
31  menjadi : " );
32        System.out.println("R1 : " + R1.Cast());
```

```
33          System.out.println("R2 : " + R2.Cast());
34          System.out.println();
35
36          R1.negasi();
37          System.out.print("Unary- dari R1 : ");
38          R1.cetak();
39          System.out.println();
40
41          R1.unaryPlus(R2);
42          System.out.print("Nilai dari 'R1 += R2' : ");
43          R1.cetak();
44          System.out.println();
45      }
46  }
```

Data dan Analisis hasil percobaan

A. Constructor

Pertanyaan

1. Lakukan percobaan constructor diatas dan benahi jika menemukan kesalahan!
   ……..
   ……..

2. Tambahkan constructor pada class Student dengan parameter yang mempunyai parameter masing masing nilai dari mata pelajaran yang ada! Kemudian buat contoh objeknya pada main Class!
   ……..
   ……..

3. Tambahkan method dengan nilai balikan berupa boolean pada class student bernama statusAkhir untuk menentukan apakah siswa tersebut remidi atau tidak. Ketentuannya adalah jika nilai lebih dari atau sama dengan 61 adalah lolos sedangkan nilai kurang dari atau sama dengan 60 adakah remidi. Nilai yang di cari adalah nilai rata rata untuk semua mapel. Kemudian nilai pada method statusAkhir tampilkan pada method displayMessage!
   ……..
   ……..

4. Bagaimana cara memasukkan jumlah siswa sesuai dengan keinginan user? Tuliskan kodenya dengan inputan user yang interaktif! (key : menggunakan array)
   ……..
   ……..

5. Bagaimana cara menghitung banyaknya objek yang kita buat dari sebuah menginstance objek dari mein class? Tuliskan kodenya kemudian tampilkan informasinya dengan memanggil method jumlahObjek() bertipe void!
   ……..
   ……..

B. Instance Method

Pertanyaan

1. Lakukan percobaan Instance Method diatas dan benahi jika menemukan kesalahan!
   ……..
   ……..
2. Tambahkan method untuk operator <, <=, >= !

……..
……..

3. Ubah method sederhana pada baris 25 – 30 yang awalnya adalah menggunakan while menjadi for!
   ……..
   ……..

4. Tambahkan method untuk operasi -, * , / !
   ……..
   ……..

Tugas Praktikum

1. Buatlah implementasi sebuah mesin ATM. Dalam mesinATM, password dapat dimasukkan oleh user. Selain itu, sistem memilki pilihan menu melihat saldo, menarik uang, dan mentransfer ke rekening lain!

DAFTAR PUSTAKA

- Horstmann, Cay. 2010. *Big Java 4 Edition*. Hoboken : John Willey & Sons Inc
- Horton, Ivor. 2002. *Beginning Java 2, SDK 1.4 Edition*. Birmingham : Wrox Press Ltd.
- Deitel, Paul., dan Deitel, Harvey. 2012. *Java How To Program Ninth Edition*. Boston : Prentice Hall
- Avestro, Joyce. 2007. *JENI (Java Education Network Indonesia)*. Jakarta : Jardiknas
- Dewi, Candra., Muttaqin, Adharul dan Supianto, Afif. 2012. *Modul Pemrograman Lanjut*. Malang : Laboratorium Komputer Dasar PTIIK UB