

## BAB 2

# 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 constructor :

1. Constructor memiliki nama yang sama dengan class
2. Constructor tidak memiliki 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 memiliki 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("beralamat 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	
	public class MainStudent {
	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 dengan 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(){
17        int temp, A, B;
18        if (penyebut ==0){
19            return;
20        }
21        A = (pembilang<penyebut) ? penyebut:pembilang;
22        B = (pembilang<penyebut) ? pembilang:penyebut;
23
24        while (B != 0){
25            temp= A % B;
26            A = B;
27            B = temp;
28        }
29        pembilang /=A;
30        penyebut /=A;
31    }
32    public double Cast(){
33        return (penyebut==0.0) ? 0.0 : (double)pembilang /
34        (double)penyebut;
35    }
36
37

```

```

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();
17
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();
27
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!  
**Pada class Student dan mainStudent ada beberapa konstruktor yang typo, student yang seharusnya Student (S besar), lalu pada displayMessage juga terdapat kesalahan pada**

```
System.out.println("mempunyai nilai rata-rata " +getAverage);
```

**Yang seharusnya getAvarage adalah average**

```
System.out.println("mempunyai nilai rata-rata " +average);
```

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!

**Student.java**

```

public Student(String n, String a, int ag, double math, double english, double
science) {
    name = n;
    address = a;
    age = ag;
    mathGrade = math;
    englishGrade = english;
    scienceGrade = science;
    average = getAverage();
}

```

**MainStudent.java**

```

Student david = new Student("Dapit", "Cikarang", 19, 85, 70, 80);
david.displayMessage();

```

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!

**Boolean statusAkhir:**

```

public boolean statusAkhir() {
    return average >= 61;
}

```

**displayMessage:**

```
System.out.println("Status: " + (statusAkhir() ? "Lolos" : "Remidi"));
```

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

```
Student[] students = new Student[jumlahSiswa];

for (int i = 0; i < jumlahSiswa; i++) {
    System.out.println("Siswa ke-" + (i + 1));

    System.out.print("Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Alamat: ");
    String alamat = scanner.nextLine();

    System.out.print("Umur: ");
    int umur = scanner.nextInt();
    scanner.nextLine();

    System.out.print("Nilai Matematika: ");
    double math = scanner.nextDouble();

    System.out.print("Nilai Bahasa Inggris: ");
    double english = scanner.nextDouble();

    System.out.print("Nilai IPA: ");
    double ipa = scanner.nextDouble();
    scanner.nextLine();

    students[i] = new Student(nama, alamat, umur, math, english, ipa);
}
```

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!

**Student.java**

```
public class Student {
    private static int count = 0;

    // Konstruktor
    public Student() {
        count++;
    }

    // Method untuk mendapatkan jumlah objek yang dibuat
    public static void jumlahObjek() {
        System.out.println("Jumlah objek Student yang dibuat: " + count);
    }

    // Method lainnya
}
```

**MainStudent.java**

```

public class MainStudent {
    public static void main(String[] args) {
        // Membuat beberapa objek Student
        Student anna = new Student();
        Student chris = new Student();
        Student david = new Student();

        Student.jumlahObjek
        // Kode lainnya ...
    }
}

```

**B. Instance Method**

Pertanyaan

1. Lakukan percobaan Instance Method diatas dan benahi jika menemukan kesalahan!

**Kesalahan hanya saya temukan pada**

```

System.out.println("R1.isRasional: " +R1.isRasional());
System.out.println("R2.isRasional: " + R1.isRasional());

```

**Yang seharusnya adalah**

```

System.out.println("R1.isRasional: " +R1.isRasional());
System.out.println("R2.isRasional: " + R2.isRasional());

```

2. Tambahkan method untuk operator <, <=, >= !

```

public boolean lessThan(Rasional A) {
    return (pembilang * A.penyebut < penyebut * A.pembilang);
}

public boolean lessThanOrEqual(Rasional A) {
    return (pembilang * A.penyebut <= penyebut * A.pembilang);
}

public boolean greaterThan(Rasional A) {
    return (pembilang * A.penyebut > penyebut * A.pembilang);
}

public boolean greaterThanOrEqual(Rasional A) {
    return (pembilang * A.penyebut >= penyebut * A.pembilang);
}

```



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

```
public void Sederhana() {
    int temp, A, B;
    if (penyebut == 0) {
        return;
    }
    A = (pembilang < penyebut) ? penyebut : pembilang;
    B = (pembilang < penyebut) ? pembilang : penyebut;

    for (; B != 0;) {
        temp = A % B;
        A = B;
        B = temp;
    }

    pembilang /= A;
    penyebut /= A;
}
```

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

```
public Rasional subtract(Rasional A) {
    int newPembilang = (pembilang * A.penyebut) - (penyebut * A.pembilang);
    int newPenyebut = penyebut * A.penyebut;
    return new Rasional(newPembilang, newPenyebut);
}

public Rasional multiply(Rasional A) {
    int newPembilang = pembilang * A.pembilang;
    int newPenyebut = penyebut * A.penyebut;
    return new Rasional(newPembilang, newPenyebut);
}

public Rasional divide(Rasional A) {
    int newPembilang = pembilang * A.penyebut;
    int newPenyebut = penyebut * A.pembilang;
    return new Rasional(newPembilang, newPenyebut);
}
```

### Tugas Praktikum

Dari jawaban bab 1, implementasikan hal-hal berikut di dalam class :

1. Tambahkan attribute 'sinopsis' (minimal 10 kata),
2. Constructor pada semua attribute yang ada,
3. Method hitung jumlah kata pada atribut synopsis,

4. Method cek tingkat kesamaan, dengan parameter object lain dan nilai kembalian 0-100% sesuai kesamaan attribute kedua object,
5. Method copy yang mengembalikan object yang nilainya sama.