

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
40             return result;          +englishGrade)/3;
41     }
42     public void displayMessage(){
43         System.out.println("Siswa dengan
44         System.out.println("beralamat di  nama "+n me);
45         System.out.println("berumur "+ag  "+address );
46         System.out.println("mempunyai    e);
47         "+getAverage());          nilai      rata      rata
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();
27
28      //siswa dengan nama chris dirubah informasi alamat dan
29  umurnya melalui method
30      System.out.println("=====");
31      chris.setAddress("Surabaya");
32      chris.setAge(22);
33      chris.displayMessage();
34
35  }
```

B. Instance Method

Ketikkan program di bawah ini

```

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	
1	System.out.print("R1 : ");
1	R1.cetak();
1	System.out.print("R2 : ");
1	R2.cetak();
1	System.out.println();
8	
1	
9	R1.Sederhana();
20	R2.Sederhana();
21	
22	System.out.print("R1 : ");
23	
24	R1.cetak();
25	System.out.print("R2 : ");
26	R2.cetak();
2	System.out.println();
8	
2	
9	
3	System.out.println("Setelah dilakukan Cast ke double
0	menjadi : ");
31	System.out.println("R1 : " + R1.Cast());
32	

```

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!
terdapat kesalahan dalam merubah informasi melalui constructor, karena seharusnya dipanggil juga set nilai-nilai dari Anna supaya nilai rata-ratanya dapat dihitung dan muncul di output.

```

// siswa dengan nama anna dirubah informasi alamat dan umurnya melalui
// constructor
System.out.println(x:"=====");
anna = new Student(n:"Anna", a:"Batu", ag:18);
anna.setEnglish(english:60);
anna.setMath(math:80);
anna.setScience(science:93);
anna.displayMessage();

```

```

Siswa dengan nama Anna
beralamat di Batu
berumur 18
mempunyai nilai rata-rata 77.66666666666667
=====

```

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!

```

public Student(double mathGrade, double englishGrade, double scienceGrade) {
    this.mathGrade = mathGrade;
    this.englishGrade = englishGrade;
    this.scienceGrade = scienceGrade;
}

```

```

Student nafa = new Student(mathGrade:100, englishGrade:90, scienceGrade:95);
nafa.setName(n:"Nafa");
nafa.setAge(ag:18);
nafa.setAddress(a:"Mbetex");
nafa.displayMessage();

```

```

Siswa dengan nama Nafa
beralamat di Mbetex
berumur 18
mempunyai nilai rata-rata 95.0

```

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 adalah remidi. Nilai yang di cari adalah nilai rata rata untuk semua mapel. Kemudian nilai pada method statusAkhir tampilkan pada method displayMessage!

```
private boolean statusAkhir(){
    double average = getAverage();
    return average >= 61;
}

public void displayMessage() {
    System.out.println("Siswa dengan nama " + name);
    System.out.println("beralamat di " + address);
    System.out.println("berumur " + age);
    System.out.println("mempunyai nilai rata-rata " + getAverage());

    if (statusAkhir()) {
        System.out.println(x:"Status Akhir: Lolos");
    } else {
        System.out.println(x:"Status Akhir: Remidi");
    }
}
```

```
Siswa dengan nama Nafa
beralamat di Mbetex
berumur 18
mempunyai nilai rata-rata 95.0
Status Akhir: Lolos
```

4. Bagaimana cara memasukkan jumlah siswa sesuai dengan keinginan user? Tuliskan kodenya dengan inputan user yang interaktif! (key : menggunakan array)
Menggunakan array untuk menyimpan jumlah siswa yang nantinya dipanggil dengan membuat objek yang memanggil class Student.

```
public class MainStudent {
    public static void main(String[] args) {

        System.out.print(s:"Masukkan jumlah siswa: ");
        int jumlahSiswa = input.nextInt();

        Student[] siswa = new Student[jumlahSiswa];

        for (int i = 0; i < jumlahSiswa; i++){
            System.out.println("Data siswa ke-" + (i+1));
            input.nextLine();
            System.out.print(s:"Nama: ");
            String nama = input.nextLine();
            System.out.print(s:"Alamat: ");
            String alamat = input.nextLine();
            System.out.print(s:"Umur: ");
            int umur = input.nextInt();
            System.out.print(s:"Nilai Matematika: ");
            int math = input.nextInt();
            System.out.print(s:"Nilai IPA: ");
            int science = input.nextInt();
            System.out.print(s:"Nilai Bahasa Inggris: ");
            int bing = input.nextInt();

            siswa[i] = new Student(nama, alamat, umur);
            siswa[i].setMath(math);
            siswa[i].setScience(science);
            siswa[i].setEnglish(bing);
            siswa[i].displayMessage();
            System.out.println(x:"");
        }
    }
}
```

```

Data siswa ke-1
Nama: Nafa
Alamat: Mbetek
Umur: 18
Nilai Matematika: 100
Nilai IPA: 98
Nilai Bahasa Inggris: 95
=====
Siswa dengan nama Nafa
beralamat di Mbetek
berumur 18
mempunyai nilai rata-rata 97.66666666666667
Status Akhir: Lolos

Data siswa ke-2
Nama: Nopal
Alamat: Araya
Umur: 17
Nilai Matematika: 90
Nilai IPA: 80
Nilai Bahasa Inggris: 85
=====
Siswa dengan nama Nopal
beralamat di Araya
berumur 17
mempunyai nilai rata-rata 85.0
Status Akhir: Lolos

```

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

Membuat variabel static untuk menyimpan jumlah objek, yang nantinya akan ditambahkan di tiap konstruktor, sehingga jika kita membuat objek di kelas main yang pastinya juga akan memanggil konstruktor, jumlah objek akan dapat dihitung.

```
private static int jumlahObject = 0;
```

```

public Student(double mathGrade, double englishGrade, double scienceGrade) {
    this.mathGrade = mathGrade;
    this.englishGrade = englishGrade;
    this.scienceGrade = scienceGrade;
    jumlahObject++;
}

public Student(){
    name = "";
    address = "";
    age = 0;
    jumlahObject++;
}

public Student(String n, String a, int ag) {
    name = n;
    address = a;
    age = ag;
    jumlahObject++;
}

public static void jumlahObject() {
    System.out.println("Jumlah objek Student yang telah dibuat: " + jumlahObject);
}

```

```
Jumlah objek Student yang telah dibuat: 5
```

B. Instance Method

Pertanyaan

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


```
// operator >
public boolean moreThan(Rasional A) {
    return (pembilang * A.penyebut > penyebut * A.pembilang);
}

// operator <
public boolean lessThan(Rasional B){
    return (pembilang * B.penyebut < penyebut * B.pembilang);
}

// operator <=
public boolean lessThanEquals(Rasional C){
    return (pembilang * C.penyebut <= penyebut * C.pembilang);
}

// operator >=
public boolean moreThanEquals(Rasional D){
    return (pembilang * D.penyebut >= penyebut * D.pembilang);
}
```

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

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

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

```
// Operasi pengurangan (-)
public Rasional kurang(Rasional E) {
    int newPembilang = pembilang * E.penyebut - penyebut * E.pembilang;
    int newPenyebut = penyebut * E.penyebut;
    Rasional result = new Rasional(newPembilang, newPenyebut);
    result.Sederhana();
    return result;
}

// Operasi perkalian (*)
public Rasional multiply(Rasional F) {
    int newPembilang = pembilang * F.pembilang;
    int newPenyebut = penyebut * F.penyebut;
    Rasional result = new Rasional(newPembilang, newPenyebut);
    result.Sederhana();
    return result;
}

// Operasi pembagian (/)
public Rasional divide(Rasional G) {
    int newPembilang = pembilang * G.penyebut;
    int newPenyebut = penyebut * G.pembilang;
    Rasional result = new Rasional(newPembilang, newPenyebut);
    result.Sederhana();
    return result;
}
```

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.