

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : CONSTRUCTOR DAN INSTANCE METHOD
NAMA : Najwa Afifah
NIM : 235150601111002

DATA DAN ANALISIS HASIL PERCOBAAN

A. Constructor

Pertanyaan

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

Penjelasan

Tidak ada error

Pertanyaan

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!

Syntax

Kelas MainStudent

```
1 System.out.println("-----");
2     Student b
3     agas = new Student("bagas", "bogor", 21, 50, 60, 90);
4     bagas.displayMessage();
```

Kelas Student

```
1 public Student(String n, String a, int ag, double mathGrade, double englishGrade, double scienceGrade) {
2     nama = n;
3     address = a;
4     age = ag;
5     this.mathGrade = mathGrade;
6     this.scienceGrade = scienceGrade;
7     this.englishGrade = englishGrade;
8 }
```

Penjelasan

Kode tersebut merupakan tambahan constructor pada kelas Student dengan parameter nama, alamat, umur, dan nilai ketiga pembelajaran. Pada method `displayMessage` juga disesuaikan agar menampilkan output “nilai rata rata adalah...”

Catatan : `this.(variable)` digunakan untuk merujuk pada global variabel, bukan variabel pada parameter (local variable)

Pertanyaan

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 dicari adalah nilai rata rata untuk semua mapel. Kemudian nilai pada method `statusAkhir` tampilkan pada method `displayMessage`!

Syntax

Kelas MainStudent

```
1 System.out.println("-----");
2     Student abdul = new Student("abdul", "jekardtah", 21, 50, 99, 100);
3     abdul.displayMessage();
```

Kelas Student

```
1 private boolean statusAkhir() {
2     return everage >= 61;
3 }
```

Penjelasan

Kode tersebut merupakan tambahan metode pada kelas Student dengan return type Boolean. Pada metode `displayMessage` juga disesuaikan agar menampilkan output “Apakah (nama) dinyatakan lulus? (Boolean)”

Pertanyaan

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

Syntax

Kelas MainStudent

```

1  import Modul2.Student;
2  import java.util.Scanner;
3
4  public class MainStudentt {
5  public static void main(String[] args) {
6      Scanner in = new Scanner(System.in);
7
8      System.out.println("SIAM Siswa : Input datasiswa");
9      System.out.println("Masukkan jumlah sisw yang ingin dimasukkan : ");
10     int jumlahSiswa = in.nextInt();
11     in.nextLine();
12
13     Student[] student = new Student[jumlahSiswa + 1]; // buat array utk objek
14     int id = 0;
15
16     for (id = 0; id < jumlahSiswa; id++) {
17         System.out.println("Siswa ke " + (id + 1));
18         System.out.print("\nNama siswa : ");
19         String nama = in.nextLine();
20
21         System.out.print("Lokasi : ");
22         String lokasi = in.nextLine();
23         System.out.print("Umur : ");
24         int umur = in.nextInt();
25         in.nextLine();
26
27         System.out.print("Nilai math : ");
28         int mathGrade = in.nextInt();
29         in.nextLine();
30         System.out.print("Nilai english : ");
31         int englishGrade = in.nextInt();
32         in.nextLine();
33         System.out.print("Nilai science : ");
34         int scienceGrade = in.nextInt();
35         in.nextLine();
36
37         student[id] = new Student(id, nama, lokasi, umur, mathGrade, englishGrade, scienceGrade);
38     }
39
40     for (int i = 0; i < jumlahSiswa; i++) {
41         student[i].displayMessage();
42     }
43
44 }
45 }
46

```

Kelas Student

```

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      private static int counter;
10
11  public Student(int id, String n, String a, int ag, double mg, double eg, double sg) {
12      counter++;
13      name = n;
14      address = a;
15      age = ag;
16      mathGrade = mg;
17      englishGrade = eg;
18      scienceGrade = sg;
19  }
20
21  public Student(String n, String a, int ag){
22      name = n;
23      address = a;
24      age = ag;
25  }
26
27  public Student() {
28      name = "";
29      address = "";
30      age = 0;
31  }
32
33  public void setName(String n){
34      name = n;
35  }
36
37  public void setAddress(String a){
38      address = a;
39  }
40
41  public void setAge(int ag){
42      age = ag;
43  }
44
45  public void setMath(int math){
46      mathGrade = math;
47  }
48
49  public void setEnglish(int english){
50      englishGrade = english;
51  }
52
53  public void setScience(int science){
54      scienceGrade = science;
55  }
56
57  private double getAverage(){
58      average = (mathGrade+scienceGrade+englishGrade)/3;
59      return average;
60  }
61
62  private boolean statusAkhir() {
63      return average >= 61;
64  }
65
66  public void jumlahObjek() {
67      System.out.println("Jumlah objek yang dibuat : " + counter);
68  }
69
70
71  public void displayMessage(){
72      System.out.println("\nSiswa dengan nama"+name);
73      System.out.println("beralamat di "+address);
74      System.out.println("berumur "+age);
75      System.out.println("mempunyai nilai rata rata"+ getAverage());
76      System.out.println("Apakah " + name + " dinyatakan lulus? " + statusAkhir());
77      System.out.println("");
78  }
79  }

```

Penjelasan

Kode tersebut merupakan modifikasi dari kode sebelumnya, dimana perubahan signifikan ada di kelas Main.

Kelas MainStudent:

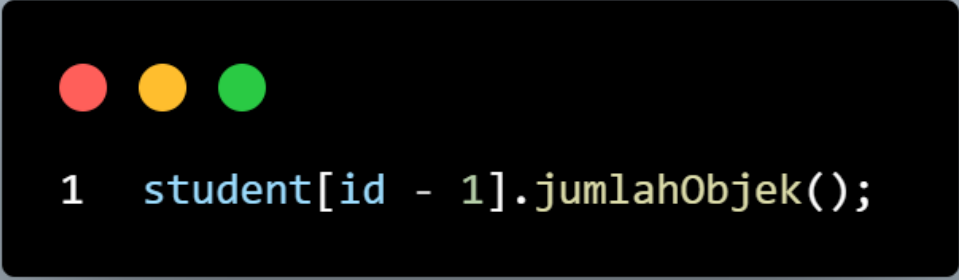
1. Menambahkan scanner
2. Menerima input jumlahSiswa
3. Menerima input data (nama, alamat, umur, dan nilai) untuk setiap siswa sesuai dengan jumlahSiswa yang diinginkan
4. Setelah menerima input dan menyimpan data, maka dibuat inisiasi objek Student dengan index array (membuat objek Student berulang kali sesuai dengan index array) dan mengisinya dengan parameter data di constructornya
5. Loop berlanjut sesuai dengan jumlahSiswa
6. Menampilkan data siswa beserta status kelulusan dengan array. Array tersebut memanggil method `displayMessage()` dengan objek Student dengan index array (`student[id]`)

Pertanyaan

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!

Syntax

Kelas MainStudent



```
1 student[id - 1].jumlahObjek();
```

Kelas Student

```

1  private static int counter;
2
3  public Student(int id, String n, String a, int ag, double mg, double eg, double sg) {
4      counter++;
5      name = n;
6      address = a;
7      age = ag;
8      mathGrade = mg;
9      englishGrade = eg;
10     scienceGrade = sg;
11 }
12
13 public void jumlahObjek() {
14     System.out.println("Jumlah objek yang dibuat : " + counter);
15 }

```

Penjelasan

Kode tersebut merupakan modifikasi dari soal sebelumnya yang menerapkan array di metode Main Kelas MainStudent:

- Memanggil metode jumlahObjek() pada instansi Student di index terakhir

Pada setiap pemanggilan constructor Student, maka ditambahkan counter. Untuk dapat mencari jumlah instansi dari kelas yang sudah dibuat, maka diambil pada instansi yang terakhir dibuat, agar dapat mengetahui nilai akhir dari counter.

Catatan: kenapa student[I - 1] bukan student[i]?

Karena pada loop, setelah instansi kelas yang terakhir dijalankan, maka nilai I bertambah. Pada nilai I terakhir, sudah tidak ada lagi instansi kelas, oleh karena itu mengambil nilai I sebelumnya yaitu pada instansi kelas yang terakhir (I-1).

Jika tetap student[i] maka error karena pada student index terakhir tidak ada instansi kelas, tidak ada nilai attribute yang dapat diambil/nilai counter-nya.

Kelas Student:

Terdapat modifikasi yaitu penambahan variable static int counter, penambahan counter pada konstruktor Student dan metode jumlahObjek();

Catatan: variabel counter bersifat static karena dianggap sebagai class variable, dimana nilainya akan terus bertambah seiring bertambahnya instance kelas dikarenakan loop Main. Berbeda dengan instance variable dimana nilai counter akan berbeda untuk setiap instance sehingga nilai counter tidak bekerja seperti yang diharapkan.

A. Instance Method

Pertanyaan

- Lakukan percobaan Instance Method diatas dan benahi jika menemukan kesalahan!

Penjelasan

Tidak ada error

Pertanyaan

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

Syntax

Kelas Rasional

```
1 public boolean lessThan (Rasional A){
2     return (pembilang * A.penyebut < penyebut * A.pembilang);
3 }
4 public boolean lessThanEqual (Rasional A){
5     return (pembilang * A.penyebut <= penyebut * A.pembilang);
6 }
7 public boolean moreThanEqual (Rasional A){
8     return (pembilang * A.penyebut >= penyebut * A.pembilang);
9 }
```

Kelas RasionalDemo

```
1 System.out.println("R1 < R2 : " + R1.lessThan(R2) + "\n");
2 System.out.println("R1 <= R2 : " + R1.lessThanEqual(R2) + "\n");
3 System.out.println("R1 >= R2 : " + R1.moreThanEqual(R2) + "\n");
```

Penjelasan

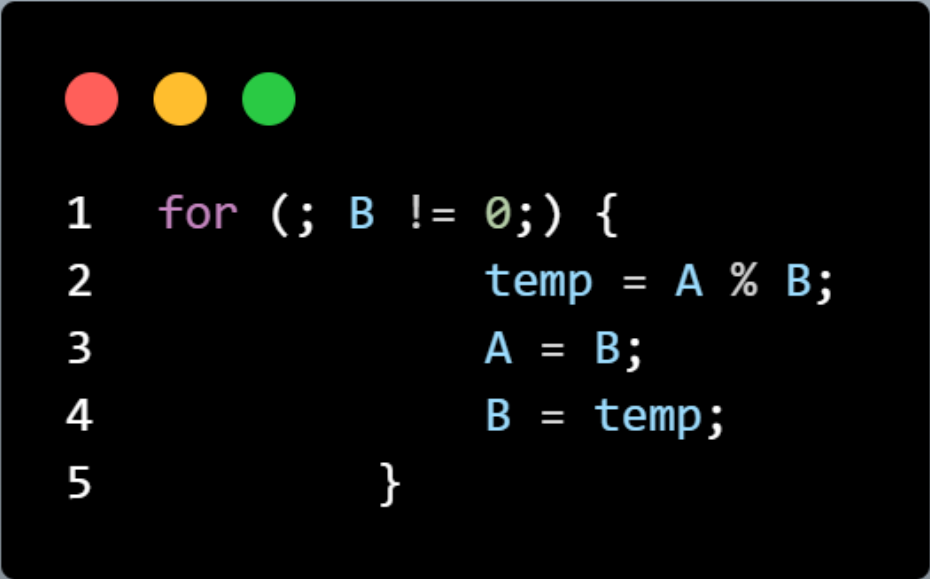
Kode tersebut merupakan tambahan metode untuk perbandingan <, <=, >=, metode yang ditambahkan mirip dengan metode moreThan(), dimana disini hanya tinggal mengganti tanda nya saja. Pemanggilan metode di main juga sama

Pertanyaan

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

Syntax

Kelas Rasional



```
1  for (; B != 0;) {  
2      temp = A % B;  
3      A = B;  
4      B = temp;  
5  }
```

Penjelasan

Hasilnya tidak berubah. Untuk mengubah dari while ke for, hanya perlu penambahan sintaks dimana di dalam for loop tidak ada nilai awal (initial condition), dan nilai increment. Di dalam for loop hanya ada syarat loop berjalan, yaitu Ketika nilai B !=0. Ketika nilai B belum mencapai 0, maka loop terus berjalan.

Pertanyaan

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

Syntax

Kelas Rasional

```

1  public void kurangkan(Rasional B) {
2      this.pembilang = this.pembilang * B.penyebut - B.pembilang * this.penyebut;
3      this.penyebut = this.penyebut * B.penyebut;
4  }
5
6  public void kalikan(Rasional C) {
7      this.pembilang = this.pembilang * C.pembilang;
8      this.penyebut = this.penyebut * C.penyebut;
9  }
10
11 public void bagi(Rasional D) {
12     Rasional rasionalBagi = new Rasional(D.penyebut, D.pembilang);
13     this.kalikan(RasionalBagi);
14 }

```

Kelas RasionalDemo

```

1  System.out.println("Nilai dari 'R1 - R2' ");
2      R1.kurangkan(R2);
3      R1.cetak();
4
5      System.out.println("Nilai dari 'R1 * R2' ");
6      R1.kalikan(R2);
7      R1.cetak();
8
9      System.out.println("Nilai dari 'R1/R2' ");
10     R1.bagi(R2);
11     R1.cetak();

```

Penjelasan

Kode ini merupakan penambahan method pada kelas Rasional dan RasionalDemo.

Pada kelas Rasional, 3 method baru yaitu method untuk kurang(-), kali(*), dan bagi(/)

- Method kurangkan membuat instance kelas Rasional baru dengan nama B, dengan nilai parameter di kelas RasionalDemo. Dalam kasus ini, nilai R2 menjadi parameter Rasional B.

This.pwmbilang dan this.penyebut adalah pembilang dan penyebut pada R1, B.pembilang dan B.penyebut adalah pembilang dan penyebut pada R2 (dari parameter di instance Rasional B)

- b. Method kalikan membuat instance kelas Rasional baru dengan nama C, dengan nilai parameter di kelas RasionalDemo. Dalam kasus ini, nilai R2 menjadi parameter Rasional C.

This.pembilang dan this.penyebut adalah pembilang dan penyebut pada R1, B.pembilang dan B.penyebut adalah pembilang dan penyebut pada R2 (dari parameter di instance Rasional C)

- c. Method bagi membuat instance kelas Rasional baru dengan nama D, dengan nilai parameter di kelas Rasionaldemo. Dalam kasus ini, nilai R2 menjadi parameter Rasional D.

- 1) Membuat instance baru dikelas Rasional Bernama rasionalBagi dengan parameter Rasional D(nilai pembilang dan penyebut) yang dibalik. Nilai pembilang menjadi penyebut, nilai penyebut menjadi pembilang

Catatan : yang dibalik nilainya hanya nilai R2 atau parameter Rasional D

- 2) Mengalikan kedua nilai pecahan dengan rumus berikut:

$R1 * R2$ yang dibalik