

Nama: Isri Amirul Haq
NIM: 235150600111011

Modul 1 Bab 2 Constructor dan Instance Method

Data dan Analisis hasil percobaan

A. Constructor

Pertanyaan

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

Jawab: Pada class Student, terdapat kesalahan penulisan constructor. Seharusnya nama constructor harus sama dengan nama class dan huruf awalnya kapital. Jadi, ubah public student() menjadi public Student().

```
public Student(){    // no 1, sebelumnya "public student()"
    name = "";
    address = "";
    age = 0;
}
```

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!

Jawab:

```
// no 2, menambahkan constructor pada class Student dengan parameter untuk masing-masing nilai dari mata pelajaran yang ada
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;
}
```

```
// no 2, penggunaan constructor baru pada main class:
Student isri = new Student(n:"isri", a:"Lombok", ag:19, math:95, english:90, science:90);
isri.displayMessage();
System.out.println(x:"=====");
```

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!

Jawab:

```
// no 3, method yang mengembalikan nilai boolean untuk menentukan apakah siswa tersebut remidi atau tidak
public boolean statusAkhir() {
    double average = getAverage();
    return average >= 61;
}
```

```
// Display sebelumnya
// 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());
// }
```

```
// Display baru
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());
    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)

Jawab:

```

// no 4, dapat memasukkan jumlah siswa sesuai dengan keinginan pengguna dengan
// inputan yang interaktif dan menggunakan array
Scanner scanner = new Scanner(System.in);

System.out.print(s:"Masukkan jumlah siswa: ");
int jumlahSiswa = scanner.nextInt();
scanner.nextLine(); // membersihkan newline character
Student[] students = new Student[jumlahSiswa];

for (int i = 0; i < jumlahSiswa; i++) {
    System.out.println("Data siswa ke-" + (i + 1) + ":");
    System.out.print(s:"Nama: ");
    String name = scanner.nextLine();
    System.out.print(s:"Alamat: ");
    String address = scanner.nextLine();
    System.out.print(s:"Umur: ");
    int age = scanner.nextInt();
    System.out.print(s:"Nilai Matematika: ");
    double math = scanner.nextDouble();
    System.out.print(s:"Nilai Bahasa Inggris: ");
    double english = scanner.nextDouble();
    System.out.print(s:"Nilai IPA: ");
    double science = scanner.nextDouble();

    students[i] = new Student(name, address, age, math, english, science);
    scanner.nextLine();
}

// Menampilkan data siswa
System.out.println(x:"\nData Siswa:");
for (int i = 0; i < jumlahSiswa; i++) {
    students[i].displayMessage();
    System.out.println();
}
scanner.close();

```

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!

```

// no 5
private static int banyakObjek = 0;

// no 1, sebelumnya "public student()"
public Student(){
    name = "";
    address = "";
    age = 0;
}

// no 2, menambahkan constructor pada class Student dengan parameter untuk masing-masing nilai dari mata pelajaran yang ada
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;

    // no 5
    banyakObjek++;
}

// no 3, method yang mengembalikan nilai boolean untuk menentukan apakah siswa tersebut remidi atau tidak
public boolean statusAkhir() {
    double average = getAverage();
    return average >= 61;
}

// no 5
public static void jumlahObjek() {
    System.out.println("Jumlah objek Student yang telah dibuat: " + banyakObjek);
}

```

```

// no 5, memanggil method jumlahObjek()
Student.jumlahObjek();

```

B. Instance Method

Pertanyaan

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

Jawab:

Percobaan Instance Method aman saja apabila di jalankan. Namun, ada kesalahan kecil, di mana R1.isRasional() seharusnya menggunakan R2.isRasional() untuk mengecek keberhasilan pembuatan objek R2. tetapi walau begitu program tetap berjalan dengan baik.

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

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

Modul Praktikum – Pemrograman Lanjut

Jawab:

```
// no 2  
// operator <  
public boolean kurangDari(Rasional A) {  
    return (pembilang * A.penyebut < penyebut * A.pembilang);  
}  
  
// operator <=  
public boolean kurangDariAtauSamaDengan(Rasional A) {  
    return (pembilang * A.penyebut <= penyebut * A.pembilang);  
}  
  
// operator >=  
public boolean lebihBesarDariAtauSamaDengan(Rasional A) {  
    return (pembilang * A.penyebut >= penyebut * A.pembilang);  
}
```

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

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

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

    // no 3
    for (; B != 0;) {
        temp = A % B;
        A = B;
        B = temp;
    }

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

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

```
// no 4
// operator -
public Rasional subtract(Rasional A) {
    int pbl = pembilang * A.penyebut - penyebut * A.pembilang;
    int pyb = penyebut * A.penyebut;
    return new Rasional(pbl, pyb);
}

// operator *
public Rasional multiply(Rasional A) {
    int pbl = pembilang * A.pembilang;
    int pyb = penyebut * A.penyebut;
    return new Rasional(pbl, pyb);
}

// operator /
public Rasional divide(Rasional A) {
    int pbl = pembilang * A.penyebut;
    int pyb = penyebut * A.pembilang;
    return new Rasional(pbl, pyb);
}
```