

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	
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

```

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    }

```

```

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!
Dalam kodingan diatas bagian Student.java dan MainStudent.java terdapat kesalahan Perbaiki :
 - 1.) Pada kelas Student, konstruktor yang didefinisikan tidak sesuai dengan penamaan kelas (public student() seharusnya menjadi public Student()).
 - 2.) Nama metode konstruktor harus sama persis dengan nama kelas.
 - 3.) Pada metode getAverage(), perlu dilakukan pengecekan apakah ada pembagian dengan nol untuk menghindari error pembagian oleh nol.
 - 4.) Pada metode displayMessage(), sebaiknya tampilkan nama, alamat, umur, dan nilai matematika, bahasa Inggris, ilmu pengetahuan, serta rata-rata nilai dari setiap siswa.
 - 5.) Pada kelas MainStudent, saat mengubah informasi siswa Anna melalui konstruktor, nama konstruktor yang dipanggil tidak sesuai (new student() harus menjadi new Student()).

Berikut adalah perbaikan dan penambahan kode yang saya tambahkan :

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() { //awalan public student = seharusnya S nya besar public Student, harus sesuai nama kelas
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     }

```

```
File Edit Selection View Go Run ... modul 1
J Student.java 1 X J MainStudent.java
J Student.java > Student
1 public class Student {
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     if ((mathGrade + scienceGrade + englishGrade) == 0) {
43         return 0; // Menghindari pembagian oleh nol
44     }
45     return (mathGrade + scienceGrade + englishGrade) / 3;
46 }
47
48 public void displayMessage(){
49     System.out.println("Siswa dengan nama "+name);
50     System.out.println("beralamat di "+address);
51     System.out.println("berumur "+age);
52     System.out.println("Memiliki nilai matematika: " + mathGrade);
53     System.out.println("Memiliki nilai bahasa Inggris: " + englishGrade);
54     System.out.println("Memiliki nilai ilmu pengetahuan: " + scienceGrade);
55     System.out.println("mempunyai nilai rata rata "+getAverage());
56 }
57 }
58

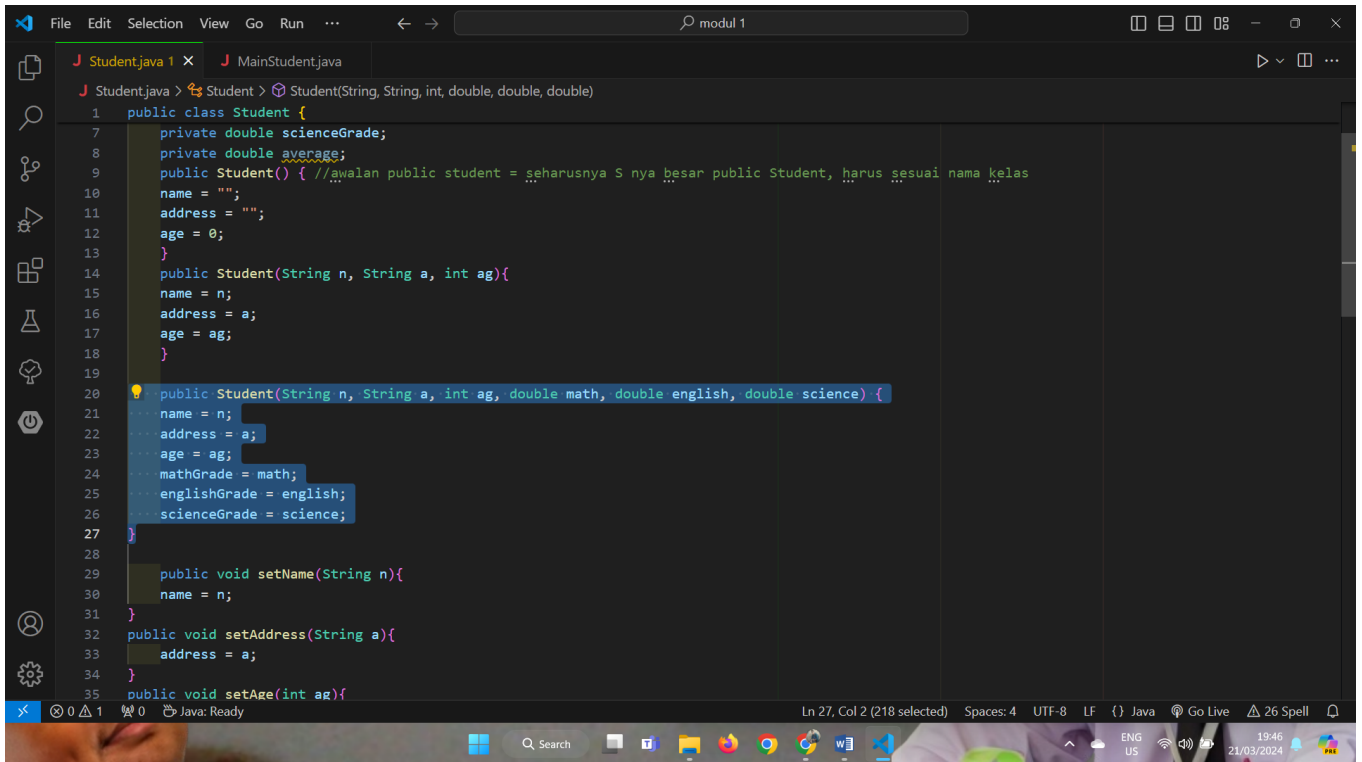
Ln 57, Col 2 Spaces: 4 UTF-8 LF {} Java Go Live 26 Spell
```

MainStudent.java

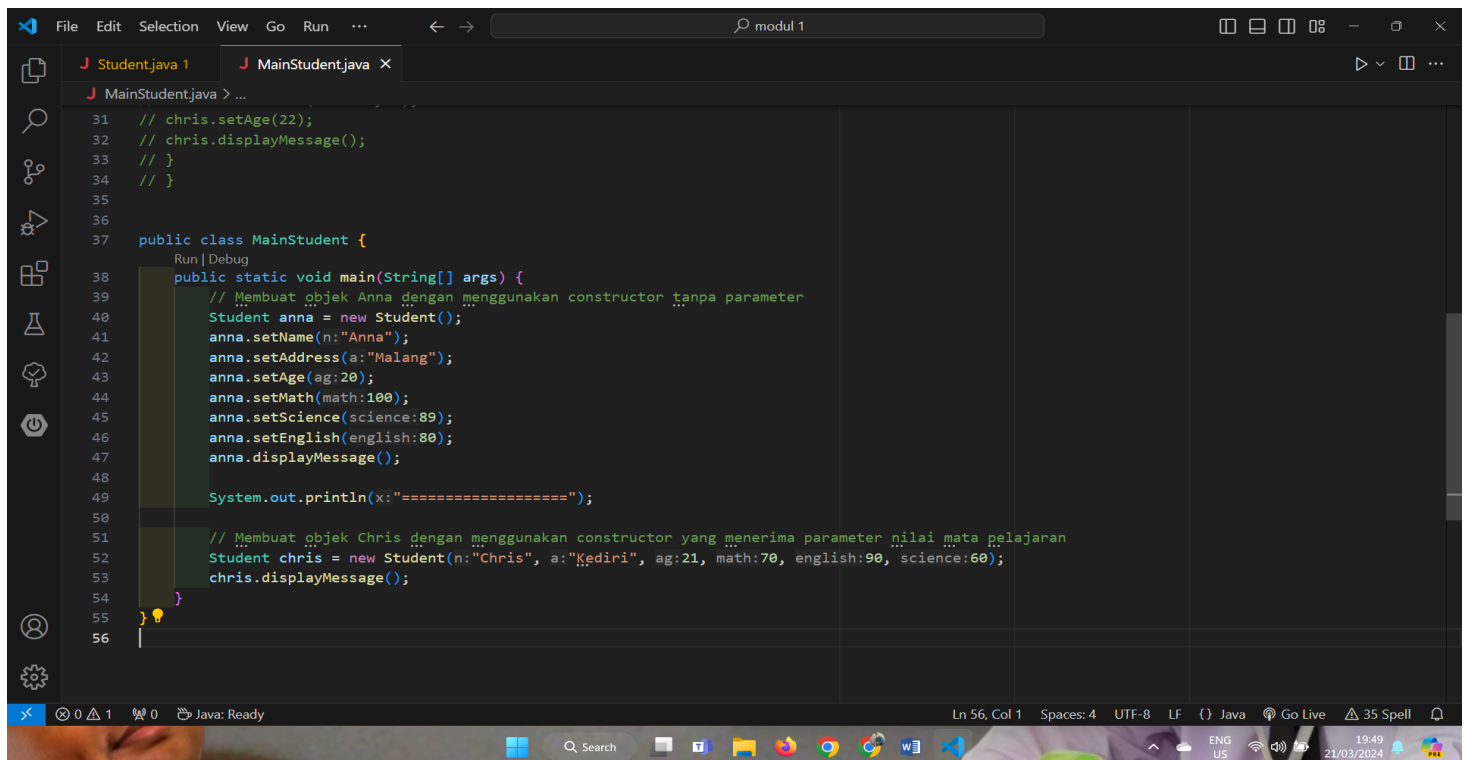
```
File Edit Selection View Go Run Terminal Help modul 1
J Student.java 1 X J MainStudent.java X
J MainStudent.java > MainStudent > main(String[])
1 public class MainStudent {
2     public static void main(String[] args) {
3         Student anna = new Student();
4         anna.setName(n:"Anna");
5         anna.setAddress(a:"Malang");
6         anna.setAge(ag:20);
7         anna.setMath(math:100);
8         anna.setScience(science:80);
9         anna.setEnglish(english:80);
10        anna.displayMessage();
11
12        //menggunakan constructor lain
13        System.out.println(x:"=====");
14        Student chris = new Student(n:"Chris", a:"Kediri", ag:21);
15        chris.setMath(math:70);
16        chris.setScience(science:60);
17        chris.setEnglish(english:90);
18        chris.displayMessage();
19
20
21
22        //siswa dengan nama anna dirubah informasi alamat dan umurnya melalui constructor
23        System.out.println(x:"=====");
24        anna = new Student(n:"anna", a:"Batu", ag:18); //nama harus sesuai kelas, memakai "s" kapital
25        anna.displayMessage();
26
27
28        //siswa dengan nama chris dirubah informasi alamat dan umurnya melalui method
29        System.out.println(x:"=====");
30        chris.setAddress(a:"Surabaya");
31        chris.setAge(ag:22);
32        chris.displayMessage();
33    }
34 }
35
36

Ln 20, Col 5 Spaces: 4 UTF-8 LF {} Java Go Live 21 Spell
```

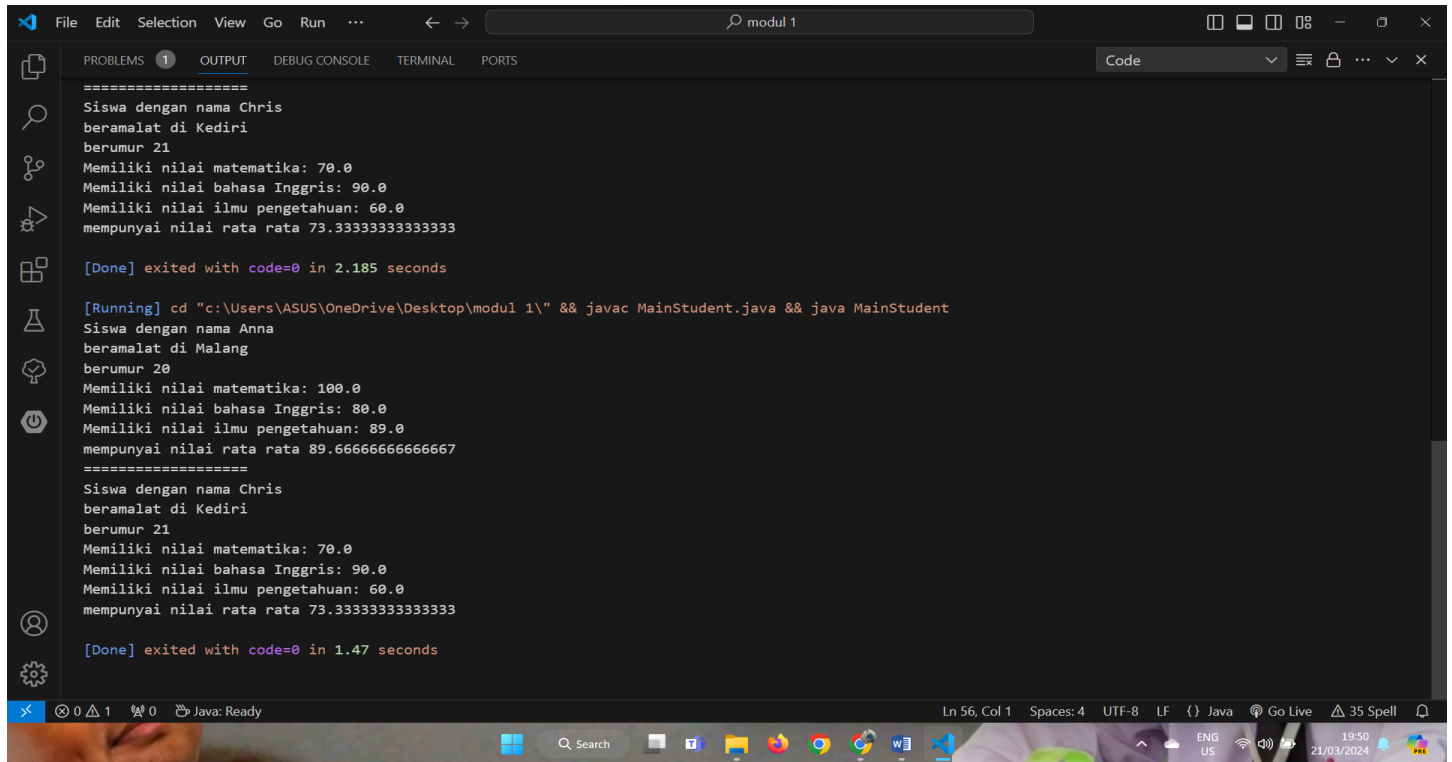
2. Tambahkan constructor pada class Student dengan parameter yang mempunyai parameter masing masing nilai dari mata pelajaran yang ada! Kemudian buat contoh objeknya pada main Class!



```
1 public class Student {
7     private double scienceGrade;
8     private double average;
9     public Student() { //awalan public student = seharusnya S nya besar public Student, harus sesuai nama kelas
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
20    public Student(String n, String a, int ag, double math, double english, double science) {
21        name = n;
22        address = a;
23        age = ag;
24        mathGrade = math;
25        englishGrade = english;
26        scienceGrade = science;
27    }
28
29    public void setName(String n){
30        name = n;
31    }
32    public void setAddress(String a){
33        address = a;
34    }
35    public void setAge(int ag){
```



```
31 // chris.setAge(22);
32 // chris.displayMessage();
33 // }
34 // }
35
36
37 public class MainStudent {
38     Run | Debug
39     public static void main(String[] args) {
40         // Membuat objek Anna dengan menggunakan constructor tanpa parameter
41         Student anna = new Student();
42         anna.setName(n:"Anna");
43         anna.setAddress(a:"Malang");
44         anna.setAge(ag:20);
45         anna.setMath(math:100);
46         anna.setScience(science:89);
47         anna.setEnglish(english:80);
48         anna.displayMessage();
49
50         System.out.println(x:"=====");
51
52         // Membuat objek Chris dengan menggunakan constructor yang menerima parameter nilai mata pelajaran
53         Student chris = new Student(n:"Chris", a:"Kediri", ag:21, math:70, english:90, science:60);
54         chris.displayMessage();
55     }
56 }
```

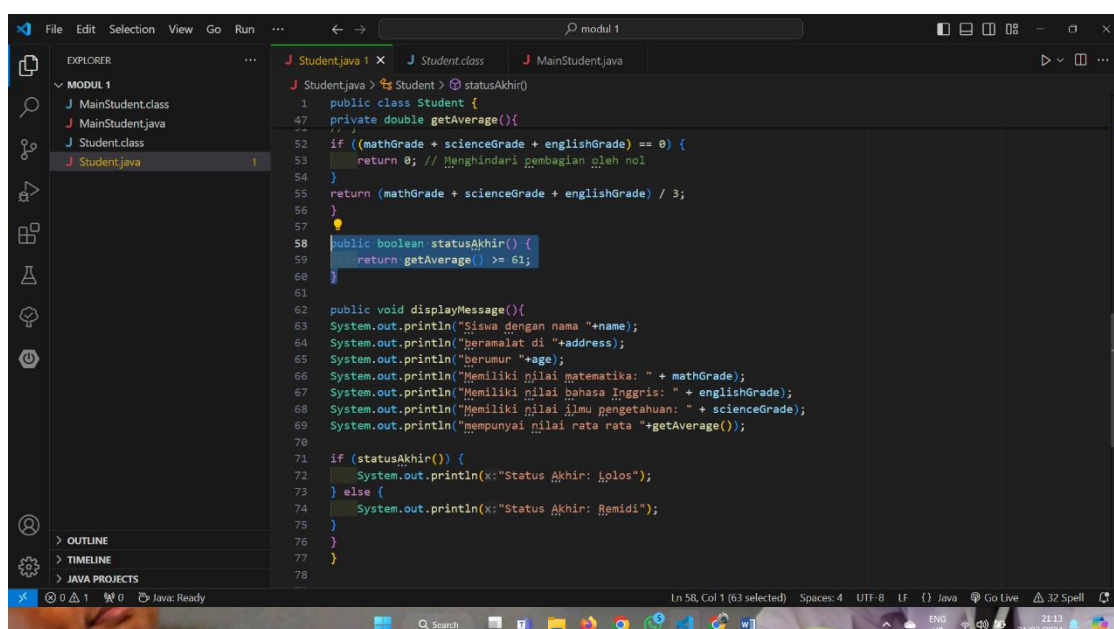



```
=====
Siswa dengan nama Chris
beralamat di Kediri
berumur 21
Memiliki nilai matematika: 70.0
Memiliki nilai bahasa Inggris: 90.0
Memiliki nilai ilmu pengetahuan: 60.0
mempunyai nilai rata rata 73.33333333333333
[Done] exited with code=0 in 2.185 seconds

[Running] cd "c:\Users\ASUS\OneDrive\Desktop\modul 1\" && javac MainStudent.java && java MainStudent
Siswa dengan nama Anna
beralamat di Malang
berumur 20
Memiliki nilai matematika: 100.0
Memiliki nilai bahasa Inggris: 80.0
Memiliki nilai ilmu pengetahuan: 89.0
mempunyai nilai rata rata 89.66666666666667
=====
Siswa dengan nama Chris
beralamat di Kediri
berumur 21
Memiliki nilai matematika: 70.0
Memiliki nilai bahasa Inggris: 90.0
Memiliki nilai ilmu pengetahuan: 60.0
mempunyai nilai rata rata 73.33333333333333
[Done] exited with code=0 in 1.47 seconds
```

Perbedaan antara kodingan sebelumnya dan kodingan yang telah diperbarui terletak pada penambahan constructor baru pada class `Student` yang menerima parameter nilai dari mata pelajaran. Dalam kodingan sebelumnya, hanya terdapat dua jenis constructor, yakni yang tidak memiliki parameter dan yang memiliki parameter untuk nama, alamat, dan umur. Namun, dalam kodingan yang diperbarui, telah ditambahkan constructor baru yang memungkinkan inisialisasi nilai mata pelajaran (matematika, bahasa Inggris, dan ilmu pengetahuan) sejak pembuatan objek. Sehingga, penggunaannya mempermudah dan lebih efisien dalam pembuatan objek `Student` dengan nilai mata pelajaran yang telah ditentukan.

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!



```
J Student.java > Student > statusAkhir()
1 public class Student {
2     private double getAverage(){
3         if ((mathGrade + scienceGrade + englishGrade) == 0) {
4             return 0; // Menghindari pembagian oleh nol
5         }
6         return (mathGrade + scienceGrade + englishGrade) / 3;
7     }
8     public boolean statusAkhir(){
9         return getAverage() >= 61;
10    }
11
12    public void displayMessage(){
13        System.out.println("Siswa dengan nama "+name);
14        System.out.println("beralamat di "+address);
15        System.out.println("berumur "+age);
16        System.out.println("Memiliki nilai matematika: " + mathGrade);
17        System.out.println("Memiliki nilai bahasa Inggris: " + englishGrade);
18        System.out.println("Memiliki nilai ilmu pengetahuan: " + scienceGrade);
19        System.out.println("mempunyai nilai rata rata "+getAverage());
20    }
21
22    if (statusAkhir()) {
23        System.out.println(x:"Status Akhir: Lolos");
24    } else {
25        System.out.println(x:"Status Akhir: Remidi");
26    }
27 }
28 }
```

```

[Running] cd "c:\Users\ASUS\OneDrive\Desktop\modul 1\" && javac Student.java && java Student
Error: Main method not found in class Student, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application

[Done] exited with code=1 in 2.926 seconds

[Running] cd "c:\Users\ASUS\OneDrive\Desktop\modul 1\" && javac MainStudent.java && java MainStudent
Siswa dengan nama Anna
beralamat di Malang
berumur 20
Memiliki nilai matematika: 100.0
Memiliki nilai bahasa Inggris: 80.0
Memiliki nilai ilmu pengetahuan: 89.0
mempunyai nilai rata rata 89.66666666666667
Status Akhir: Lolos
=====
Siswa dengan nama Chris
beralamat di Kediri
berumur 21
Memiliki nilai matematika: 70.0
Memiliki nilai bahasa Inggris: 90.0
Memiliki nilai ilmu pengetahuan: 60.0
mempunyai nilai rata rata 73.33333333333333
Status Akhir: Lolos

[Done] exited with code=0 in 1.284 seconds

```

Dengan adanya method statusAkhir, kita dapat mengevaluasi apakah seorang siswa perlu mengikuti remedi atau tidak berdasarkan rata-rata nilai dari semua mata pelajaran yang diambil. Hasil evaluasi ini kemudian akan disertakan dalam informasi lengkap yang ditampilkan saat memanggil method `displayMessage`.

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

```

jumlahSiswa.java > jumlahSiswa > main(String[])
1 import java.util.Scanner;
2
3 public class jumlahSiswa {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         System.out.print(s:"Masukkan jumlah siswa : ");
8         int jumlahSiswa = scanner.nextInt();
9
10        Student[] siswa = new Student[jumlahSiswa];
11
12        for (int i = 0; i < jumlahSiswa; i++) {
13            System.out.println("Masukkan data untuk siswa ke-" + (i+1) + ":");
14            System.out.print(s:"Nama : ");
15            String nama = scanner.next();
16            System.out.print(s:"Alamat : ");
17            String alamat = scanner.next();
18            System.out.print(s:"Umur : ");
19            int umur = scanner.nextInt();
20            System.out.print(s:"Nilai Matematika: ");
21            double nilaiMatematika = scanner.nextDouble();
22            System.out.print(s:"Nilai Bahasa Inggris: ");
23            double nilaiBahasaInggris = scanner.nextDouble();
24            System.out.print(s:"Nilai Ilmu Pengetahuan: ");
25            double nilaiIlmuPengetahuan = scanner.nextDouble();
26
27            siswa[i] = new Student(nama, alamat, umur, nilaiMatematika, nilaiBahasaInggris, nilaiIlmuPengetahuan);
28        }
29
30        System.out.println(x:"\nData Siswa:");
31        for (int i = 0; i < jumlahSiswa; i++) {
32            System.out.println("Siswa ke-" + (i+1) + ":");
33            siswa[i].displayMessage();
34            System.out.println(); // pemisahannya
35        }
36
37        scanner.close();
38    }
39 }
40

```

Data Siswa:

Siswa ke-1:
 Siswa dengan nama Eka
 beralamat di Tulungagung
 berumur 23
 Memiliki nilai matematika: 80.0
 Memiliki nilai bahasa Inggris: 23.0
 Memiliki nilai ilmu pengetahuan: 44.0
 mempunyai nilai rata rata 49.0
 Status Akhir: Remidi

Siswa ke-2:
 Siswa dengan nama Ngunyen
 beralamat di Semarang
 berumur 67
 Memiliki nilai matematika: 99.0
 Memiliki nilai bahasa Inggris: 99.0
 Memiliki nilai ilmu pengetahuan: 99.0
 mempunyai nilai rata rata 99.0
 Status Akhir: Lolos

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!

```
85 public class Student {
96 }
97
98
99 public Student(String name, String address, int age, double mathGrade, double englishGrade, double scienceGrade) {
100     this.name = name;
101     this.address = address;
102     this.age = age;
103     this.mathGrade = mathGrade;
104     this.englishGrade = englishGrade;
105     this.scienceGrade = scienceGrade;
106     jumlahObjek++;
107 }
108
109 public void setName(String name) {
110     this.name = name;
111 }
112
113 public void setAddress(String address) {
114     this.address = address;
115 }
116
117 public void setAge(int age) {
118     this.age = age;
119 }
120
121 public void setMath(double mathGrade) {
122     this.mathGrade = mathGrade;
123 }
124
125 public void setEnglish(double englishGrade) {
126     this.englishGrade = englishGrade;
127 }
128
129 public void setScience(double scienceGrade) {
130     this.scienceGrade = scienceGrade;
131 }
132
133 public void displayMessage() {
134     System.out.println("Siswa dengan nama " + name);
135     System.out.println("beralamat di " + address);
136     System.out.println("berumur " + age);
137     System.out.println("Memiliki nilai matematika: " + mathGrade);
138     System.out.println("Memiliki nilai bahasa Inggris: " + englishGrade);
139     System.out.println("Memiliki nilai ilmu pengetahuan: " + scienceGrade);
140     System.out.println("Memiliki nilai rata-rata " + getAverage());
141     System.out.println("Status Akhir: " + statusAkhir());
142 }
143
144 private double getAverage() {
145     return (mathGrade + scienceGrade + englishGrade) / 3;
146 }
147
148 private String statusAkhir() {
149     return getAverage() >= 61 ? "Lolos" : "Remidi";
150 }
151
152 public static int getJumlahObjek() {
153     return jumlahObjek;
154 }
155 }
156
157
```

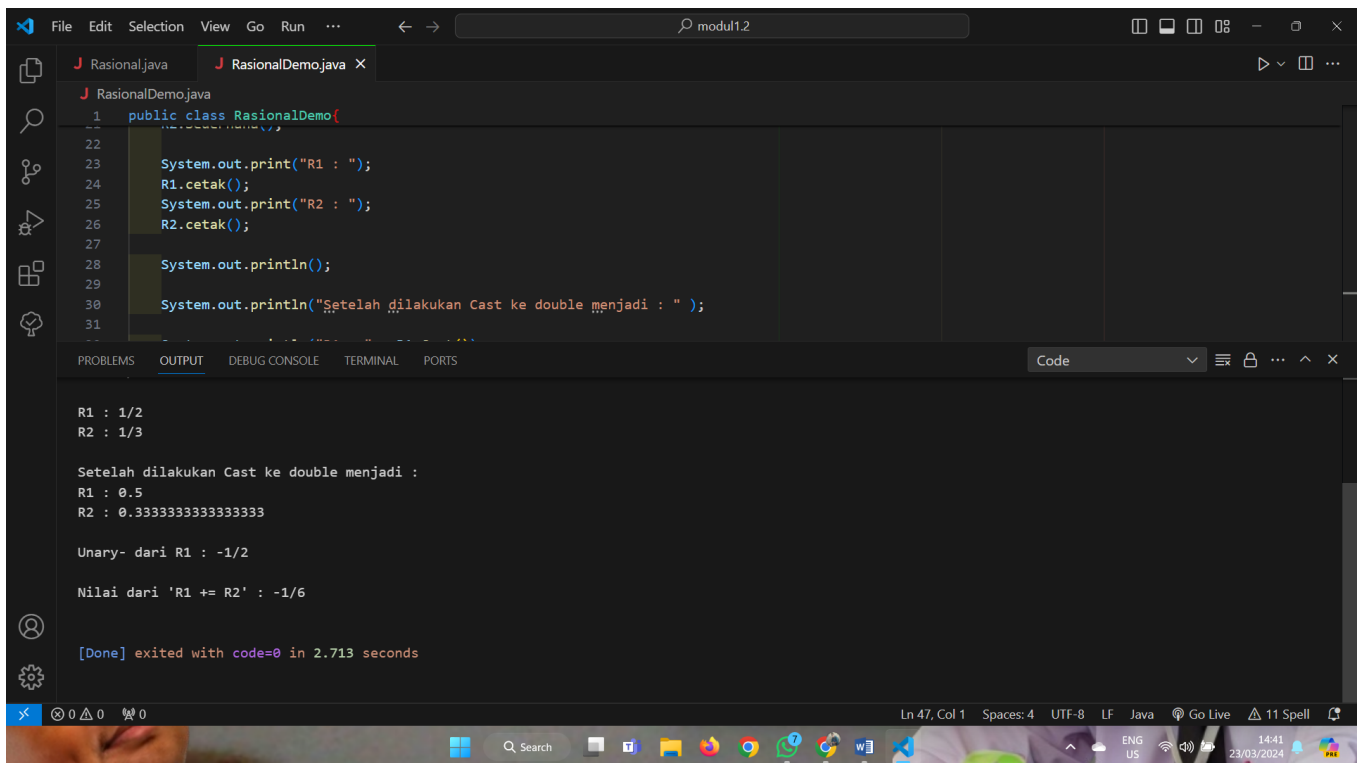
Siswa dengan nama Anna
beralamat di Malang
berumur 20
Memiliki nilai matematika: 100.0
Memiliki nilai bahasa Inggris: 80.0
Memiliki nilai ilmu pengetahuan: 90.0
Memiliki nilai rata-rata 90.0
Status Akhir: Lolos
=====
Siswa dengan nama Chris
beralamat di Surabaya
berumur 21
Memiliki nilai matematika: 70.0
Memiliki nilai bahasa Inggris: 60.0
Memiliki nilai ilmu pengetahuan: 75.0
Memiliki nilai rata-rata 68.33333333333333
Status Akhir: Lolos
Jumlah objek Student yang dibuat: 2

Dalam contoh program ini, kita membuat kelas Student yang merepresentasikan data siswa, dan kelas MainStudent untuk menjalankan program. Saat objek Student dibuat, jumlah objek secara otomatis terhitung menggunakan variabel statis jumlahObjek.

B. Instance Method

Pertanyaan

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



```
1 public class RationalDemo{
2     //constructor
3     RationalDemo(){
4         //initialization
5         //...
6     }
7     //method
8     //...
9     System.out.print("R1 : ");
10    R1.cetak();
11    System.out.print("R2 : ");
12    R2.cetak();
13
14    System.out.println();
15
16    System.out.println("Setelah dilakukan Cast ke double menjadi : ");
17
18    //...
19 }
20
21 //operator unary += \
22 public void unaryPlus(Rasional A){
23     pembilang = pembilang * A.penyebut + penyebut * A.pembilang;
24     penyebut *=A.penyebut;
25 }
26 public void cetak(){
27     System.out.println(pembilang + "/" + penyebut);
28 }
29 //operator <
30 public boolean lessThan(Rasional A){
31     return (pembilang * A.penyebut < penyebut * A.pembilang);
32 }
33 //operator <=
34 public boolean lessThanOrEqualTo(Rasional A){
35     return (pembilang * A.penyebut <= penyebut * A.pembilang);
36 }
37 //operator >=
38 public boolean greaterThanOrEqualTo(Rasional A){
39     return (pembilang * A.penyebut >= penyebut * A.pembilang);
40 }
41 }
```

Output:

```
R1 : 1/2
R2 : 1/3

Setelah dilakukan Cast ke double menjadi :
R1 : 0.5
R2 : 0.3333333333333333

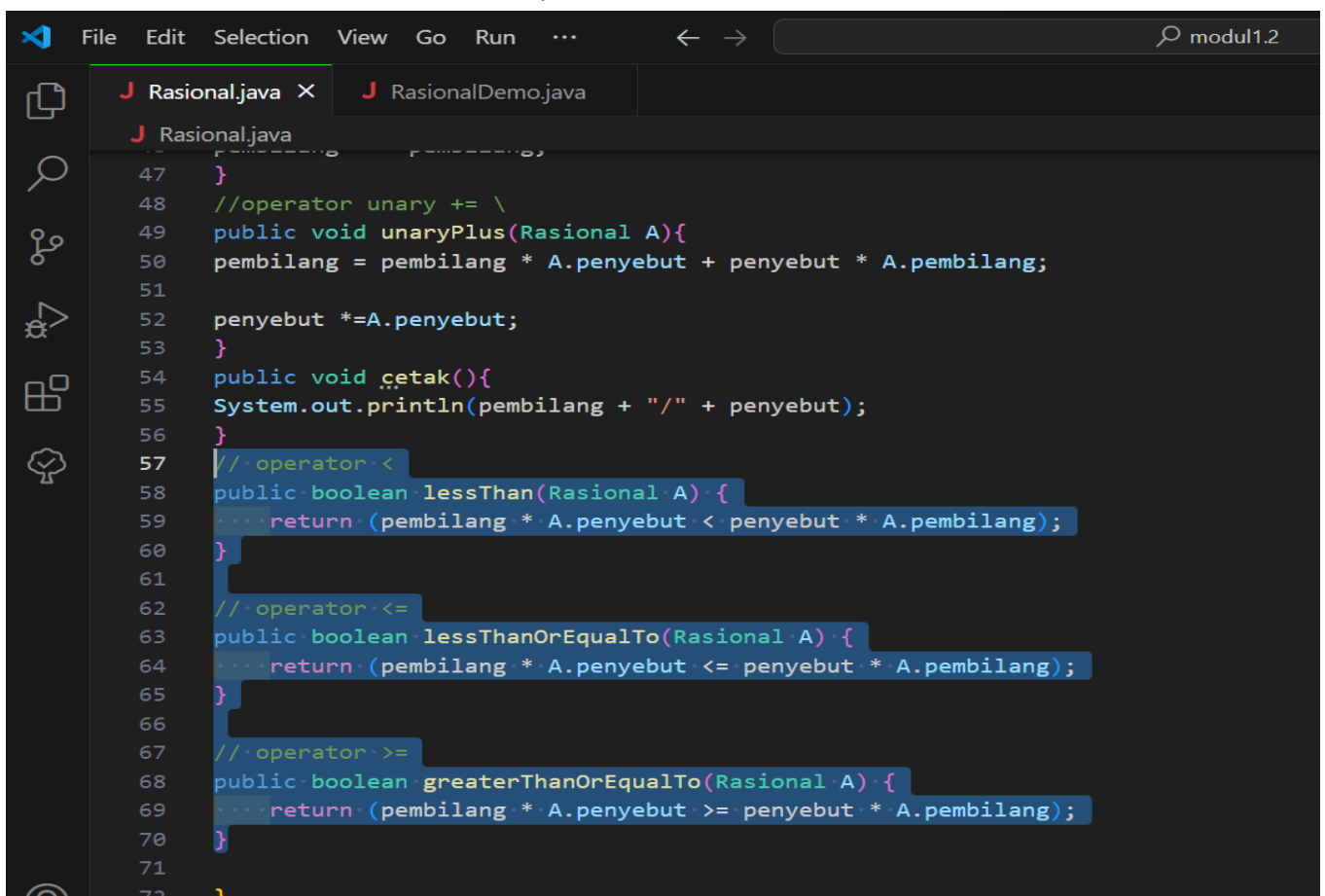
Unary- dari R1 : -1/2

Nilai dari 'R1 += R2' : -1/6

[Done] exited with code=0 in 2.713 seconds
```

Tidak menemukan kesalahan, dapat berjalan.

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



```
47 }
48 //operator unary += \
49 public void unaryPlus(Rasional A){
50     pembilang = pembilang * A.penyebut + penyebut * A.pembilang;
51     penyebut *=A.penyebut;
52 }
53 public void cetak(){
54     System.out.println(pembilang + "/" + penyebut);
55 }
56 //operator <
57 public boolean lessThan(Rasional A){
58     return (pembilang * A.penyebut < penyebut * A.pembilang);
59 }
60 //operator <=
61 public boolean lessThanOrEqualTo(Rasional A){
62     return (pembilang * A.penyebut <= penyebut * A.pembilang);
63 }
64 //operator >=
65 public boolean greaterThanOrEqualTo(Rasional A){
66     return (pembilang * A.penyebut >= penyebut * A.pembilang);
67 }
68 }
```

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

```

12 public boolean isRasional(){
13     return (penyebut != 0);
14 }
15 //menyederhanakan bilangan rasional
16 public void sederhana(){
17
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     for (; B != 0; ) {
26         temp = A % B;
27         A = B;
28         B = temp;
29     }
30
31     pembilang /= A;
32     penyebut /= A;
33 }
34 public double Cast(){
35     return (penyebut == 0) ? 0.0 : (double)pembilang /
36         (double)penyebut;
37 }
38 //operator >
39 public boolean moreThan (Rasional A){
40     return (pembilang * A.penyebut > penyebut * A.pembilang

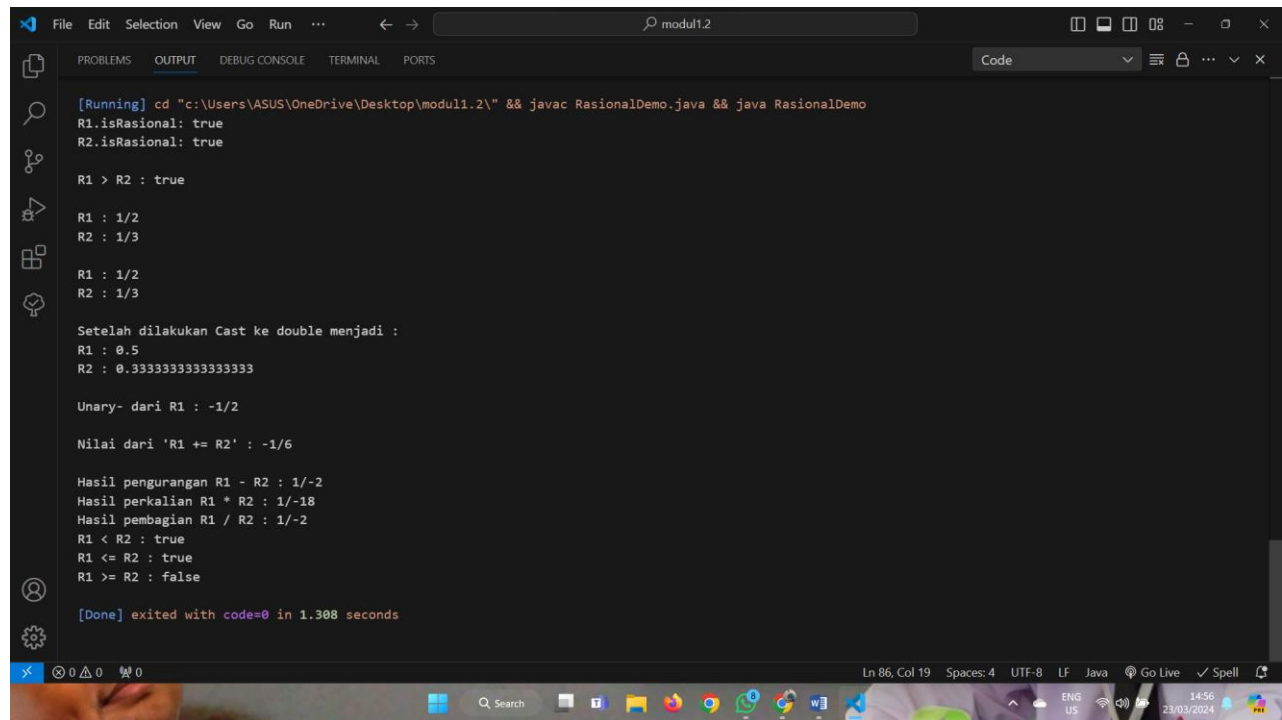
```

- Tambahkan method untuk operasi -, *, / !

```

66 // operator >=
67 public boolean greaterThanOrEqualTo(Rasional A) {
68     return (pembilang * A.penyebut >= penyebut * A.pembilang);
69 }
70
71 // Operator pengurangan (-)
72 public Rational subtract(Rasional A) {
73     int newPembilang = pembilang * A.penyebut - penyebut * A.pembilang;
74     int newPenyebut = penyebut * A.penyebut;
75     Rational result = new Rational(newPembilang, newPenyebut);
76     result.sederhana();
77     return result;
78 }
79
80 // Operator perkalian (*)
81 public Rational multiply(Rasional A) {
82     int newPembilang = pembilang * A.pembilang;
83     int newPenyebut = penyebut * A.penyebut;
84     Rational result = new Rational(newPembilang, newPenyebut);
85     result.sederhana();
86     return result;
87 }
88
89 // operator pembagian (/)
90 public Rational divide(Rasional A) {
91     if (A.pembilang == 0) {
92         System.out.println("Tidak bisa melakukan pembagian oleh nol.");
93         return null;
94     }
95     int newPembilang = pembilang * A.penyebut;
96     int newPenyebut = penyebut * A.pembilang;
97     Rational result = new Rational(newPembilang, newPenyebut);
98     result.sederhana();
99     return result;
100 }
101
102

```



```
File Edit Selection View Go Run ... modult1.2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

[Running] cd "c:\Users\ASUS\OneDrive\Desktop\modul1.2\" && javac RasionalDemo.java && java RasionalDemo
R1.isRasional: true
R2.isRasional: true

R1 > R2 : true

R1 : 1/2
R2 : 1/3

R1 : 1/2
R2 : 1/3

Setelah dilakukan Cast ke double menjadi :
R1 : 0.5
R2 : 0.3333333333333333

Unary- dari R1 : -1/2

Nilai dari 'R1 += R2' : -1/6

Hasil pengurangan R1 - R2 : 1/-2
Hasil perkalian R1 * R2 : 1/-18
Hasil pembagian R1 / R2 : 1/-2
R1 < R2 : true
R1 <= R2 : true
R1 >= R2 : false

[Done] exited with code=0 in 1.308 seconds

Ln 86, Col 19 Spaces: 4 UTF-8 LF Java Go Live Spell
```

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.