



## JOBSHEET III ARRAY OF OBJEK

**Nama:** Rizqi Bagus Andrean

**Absen:** 25

**Kelas:** TI-1D

### 3.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Memahami dan menjelaskan fungsi array yang berisikan variabel objek.
2. Mahasiswa mampu menangkap logika tentang permasalahan array of object dalam Java
3. Mahasiswa mampu menerapkan pembuatan array of object dalam Java

### 3.2 Percobaan 1: Membuat Array dari Object, Mengisi dan Menampilkan

**Waktu:** 50 menit

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat array dari object, kemudian mengisi dan menampilkan array tersebut.

#### 3.2.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama **"ArrayObjects"**.
2. Buat class **PersegiPanjang**:

```
public class PersegiPanjang {
    public int panjang;
    public int lebar;
}
```

3. Pada fungsi main yaitu pada class **ArrayObjects**, buatlah array **PersegiPanjang** yang berisi 3 elemen:

```
public static void main(String[] args) {
    PersegiPanjang[] ppArray = new PersegiPanjang[3];
}
```

4. Kemudian isikan masing-masing atributnya:

```
ppArray[0] = new PersegiPanjang();
ppArray[0].panjang = 110;
ppArray[0].lebar = 30;

ppArray[1] = new PersegiPanjang();
ppArray[1].panjang = 80;
ppArray[1].lebar = 40;

ppArray[2] = new PersegiPanjang();
ppArray[2].panjang = 100;
ppArray[2].lebar = 20;
```

Tim Aj

Jurus

g

5. Cetak ke layar semua atribut dari objek `ppArray`:

```
System.out.println("Persegi Panjang ke-0, panjang: " + ppArray[0].panjang + ", lebar: " + ppArray[0].lebar);
System.out.println("Persegi Panjang ke-1, panjang: " + ppArray[1].panjang + ", lebar: " + ppArray[1].lebar);
System.out.println("Persegi Panjang ke-2, panjang: " + ppArray[2].panjang + ", lebar: " + ppArray[2].lebar);
```

6. Jalankan dan amati hasilnya.

### 3.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Persegi Panjang ke-0, panjang: 110, lebar: 30
Persegi Panjang ke-1, panjang: 80, lebar: 40
Persegi Panjang ke-2, panjang: 100, lebar: 20
BUILD SUCCESSFUL (total time: 0 seconds)
```

### 3.2.3 Pertanyaan

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method?Jelaskan!

Tidak, Class yang akan dibuat array of object **tidak harus** memiliki atribut dan sekaligus method, **tetapi harus memiliki minimal satu dari keduanya**.

2. Apakah class `PersegiPanjang` memiliki konstruktor?Jika tidak, kenapa dilakukan pemanggilan konstruktor pada baris program berikut :

```
ppArray[1] = new PersegiPanjang();
```

Kalau tidak ada konstruktor, maka otomatis akan ada constructor defaultnya.

3. Apa yang dimaksud dengan kode berikut ini:

```
PersegiPanjang[] ppArray = new PersegiPanjang[3];
```

Membuat array dengan tipe data persegi panjang dengan jumlah maksimal 3.

4. Apa yang dimaksud dengan kode berikut ini:

```
ppArray[1] = new PersegiPanjang();
ppArray[1].panjang = 80;
ppArray[1].lebar = 40;
```

Mengisi `ppArray` index ke 1 dengan instance dari `Persegi panjang`, lalu mengisi atributnya .

5. Mengapa class main dan juga class `PersegiPanjang` dipisahkan pada uji coba 3.2?  
Tidak apa-apa, karena class `PersegiPanjang` bersifat public jadi bisa dipanggil dari class lain.

## 3.3 Percobaan 2: Menerima Input Isian Array Menggunakan Looping

Waktu: 50 menit

Tim Ajar Algoritma dan Struktur Data 2023-2024

Jurusan Teknologi Informasi-Politeknik Negeri Malang



Pada praktikum ini kita akan mengubah hasil program dari praktikum 3.2 sehingga program dapat menerima input dan menggunakan looping untuk mengisi atribut dari semua persegi panjang yang ada di ppArray.

### 3.3.1 Langkah-langkah Percobaan

1. Import scanner pada class **ArrayObjects**.

```
import java.util.Scanner;
```

Note: Letakkan kode import dibawah kode package.

2. Pada praktikum 3.2 poin nomor 4, ganti kodenya dengan kode berikut ini, yaitu membuat objek **Scanner** untuk menerima input, kemudian melakukan looping untuk menerima input:

```
Scanner sc = new Scanner(System.in);

for(int i = 0; i < 3; i++)
{
    ppArray[i] = new PersegiPanjang();
    System.out.println("Persegi panjang ke-" + i);
    System.out.print("Masukkan panjang: ");
    ppArray[i].panjang = sc.nextInt();
    System.out.print("Masukkan lebar: ");
    ppArray[i].lebar = sc.nextInt();
}
```

3. Pada praktikum 3.2 poin nomor 5, ganti kodenya dengan berikut ini, yaitu melakukan looping untuk mengakses isi array **ppArray** dan menampilkannya ke layar:

```
for(int i = 0; i < 3; i++)
{
    System.out.println("Persegi Panjang ke-" + i);
    System.out.println("Panjang: " + ppArray[i].panjang + ", lebar: " + ppArray[i].lebar);
}
```

4. Jalankan dan amati hasilnya.

### 3.3.2 Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.

```
Persegi panjang ke-0
Masukkan panjang: 5
Masukkan lebar: 6
Persegi panjang ke-1
Masukkan panjang: 5
Masukkan lebar: 3
Persegi panjang ke-2
Masukkan panjang: 4
Masukkan lebar: 8
Persegi Panjang ke-0
Panjang: 5, lebar: 6
Persegi Panjang ke-1
Panjang: 5, lebar: 3
Persegi Panjang ke-2
Panjang: 4, lebar: 8
```



```
Persegi Panjang ke-2
Masukkan panjang: 4
Masukkan lebar: 8
Persegi Panjang ke-0
Panjang: 5, Lebar: 6
Persegi Panjang ke-1
Panjang: 5, Lebar: 3
Persegi Panjang ke-2
Panjang: 4, Lebar: 8
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 3>
```

### 3.3.3 Pertanyaan

1. Apakah array of object dapat diimplementasikan pada array 2 Dimensi?

Ya

2. Jika jawaban soal no satu iya, berikan contohnya! Jika tidak, jelaskan!

```
Run | Debug
public static void main(String[] args) {
    PersegiPanjang[][] pp2 = new PersegiPanjang[2][1];

    pp2[0][0] = new PersegiPanjang();
    pp2[0][0].panjang = 10;
    pp2[0][0].lebar = 5;

    System.out.println("Persegi Panjang ke-0, Panjang: " + pp2[0][0].panjang + ",
```

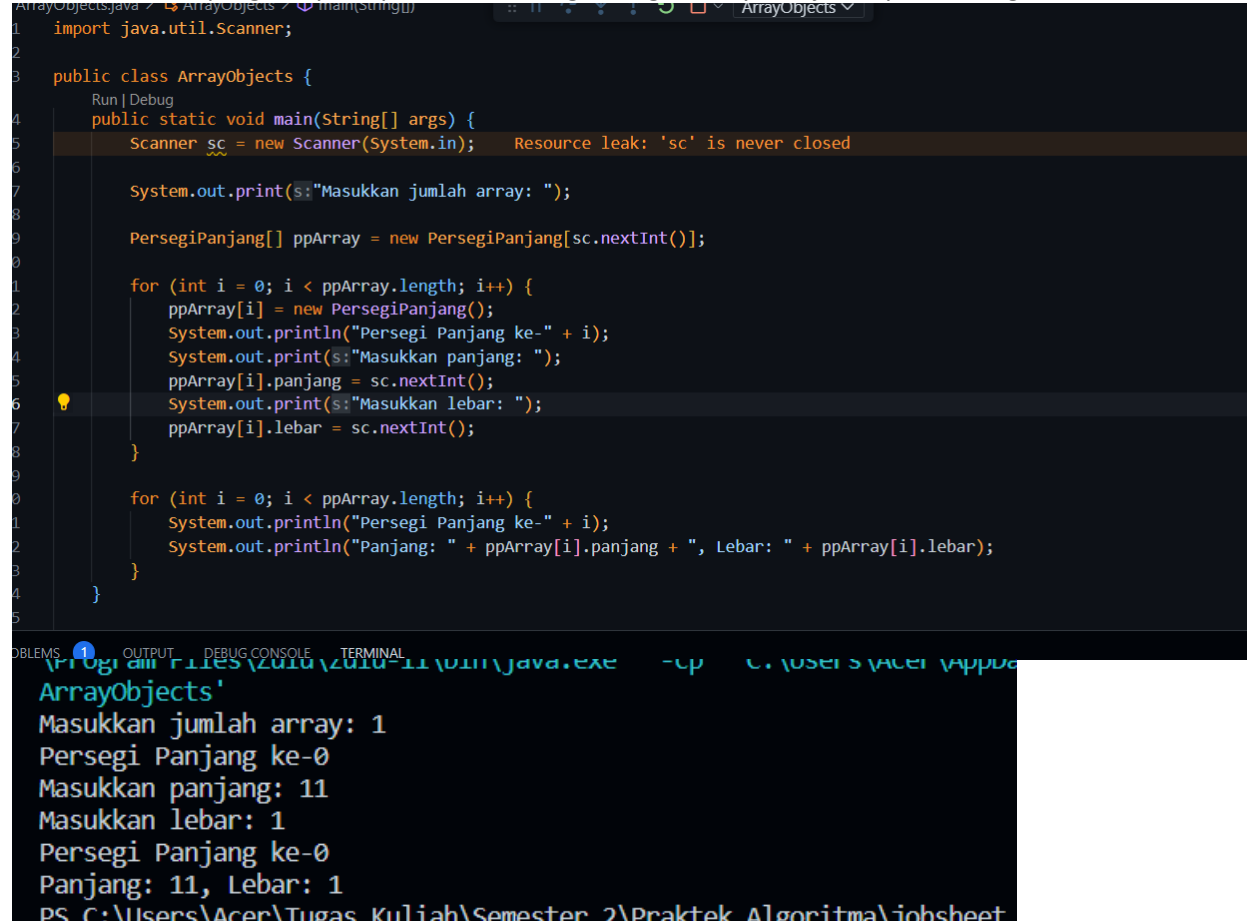
```
g:\Code\User\workspacestorage\8a7ac70f5f0a722e04ecb334981d4209\rednat.java\jul
ws\jobsheet 3_7803f0a5\bin' 'ArrayObjects'
Persegi Panjang ke-0, Panjang: 10, Lebar: 5
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 3>
```

3. Jika diketahui terdapat class **Persegi** yang memiliki atribut sisi bertipe integer, maka kode dibawah ini akan memunculkan error saat dijalankan. Mengapa?

```
Persegi[] pgArray = new Persegi[100];
pgArray[5].sisi = 20;
```

Karen ppArray pada index ke lima belum ada instance dari Persegi.

4. Modifikasi kode program pada praktikum 3.3 agar length array menjadi inputan dengan Scanner!



```
import java.util.Scanner;

public class ArrayObjects {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan jumlah array: ");

        PersegiPanjang[] ppArray = new PersegiPanjang[sc.nextInt()];

        for (int i = 0; i < ppArray.length; i++) {
            ppArray[i] = new PersegiPanjang();
            System.out.println("Persegi Panjang ke-" + i);
            System.out.print("Masukkan panjang: ");
            ppArray[i].panjang = sc.nextInt();
            System.out.print("Masukkan lebar: ");
            ppArray[i].lebar = sc.nextInt();
        }

        for (int i = 0; i < ppArray.length; i++) {
            System.out.println("Persegi Panjang ke-" + i);
            System.out.println("Panjang: " + ppArray[i].panjang + ", Lebar: " + ppArray[i].lebar);
        }
    }
}
```

Program Files (x86)\Java\jre-7\bin\java.exe -cp C:\Users\Acer\AppData\Local\Program Files (x86)\Java\jre-7\bin\java.exe ArrayObjects'

Masukkan jumlah array: 1

Persegi Panjang ke-0

Masukkan panjang: 11

Masukkan lebar: 1

Persegi Panjang ke-0

Panjang: 11, Lebar: 1

PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\iobsheet

5. Apakah boleh Jika terjadi duplikasi instansiasi array of objek, misalkan saja instansiasi dilakukan pada **ppArray[i]** sekaligus **ppArray[0]**?Jelaskan !

Bisa, instance yang ada di ppArray index yang lama akan diganti yang baru..

### 3.4 Percobaan 3: Penambahan Operasi Matematika di Dalam Method

Waktu: 50 menit

Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

#### 3.4.1 Langkah-langkah Percobaan

1. Buat package baru "**ArrayBalok**".
2. Buat class **Balok**:



```
public class Balok {
    public int panjang;
    public int lebar;
    public int tinggi;

    public Balok(int p, int l, int t)
    {
        panjang = p;
        lebar = l;
        tinggi = t;
    }

    public int hitungVolume()
    {
        return panjang * lebar * tinggi;
    }
}
```

3. Pada fungsi **main** yaitu pada class **ArrayBalok**, buat array **Balok** yang berisi 3 elemen:

```
public static void main(String[] args) {
    Balok[] blArray = new Balok[3];
}
```

4. Kemudian tambahkan kode berikut ini untuk mengisi array **blArray** menggunakan konstruktor dari class **Balok**:

```
blArray[0] = new Balok(100, 30, 12);
blArray[1] = new Balok(120, 40, 15);
blArray[2] = new Balok(210, 50, 25);
```

5. Tampilkan semua volume balok tersebut dengan cara memanggil method `hitungVolume()` di dalam looping seperti berikut ini:

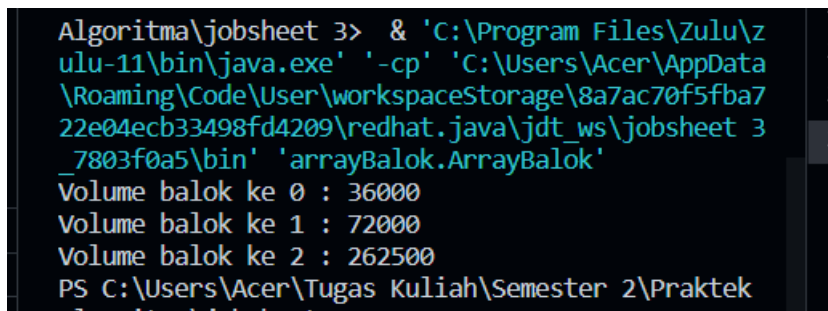
```
for(int i = 0; i < 3; i++)
{
    System.out.println("Volume balok ke " + i + ": " + blArray[i].hitungVolume());
}
```

6. Jalankan dan amati hasilnya.

### 3.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Volume balok ke 0: 36000
Volume balok ke 1: 72000
Volume balok ke 2: 262500
```



### 3.4.3 Pertanyaan

1. Dapatkah konstruktor berjumlah lebih dalam satu kelas? Jelaskan dengan contoh!  
Bisa.



```

arrayBalok > Balok.java > B ... ArrayOb
1  package arrayBalok;
2
3  public class Balok {
4      public int panjang;
5      public int lebar;
6      public int tinggi;
7
8      public Balok(int p, int l, int t) {
9          panjang = p;
10         lebar = l;
11         tinggi = t;
12     }
13
14     public Balok() {
15         panjang = 0;
16         lebar = 0;
17         tinggi = 0;
18     }
19
20     public int hitungVolume() {
21         return panjang * lebar * tinggi;
22     }
23
24 }
25

```

2. Jika diketahui terdapat class `Segitiga` seperti berikut ini:

```

public class Segitiga {
    public int alas;
    public int tinggi;
}

```

Tambahkan konstruktor pada class `Segitiga` tersebut yang berisi parameter `int a`, `int t` yang masing-masing digunakan untuk mengisi atribut alas dan tinggi.

```

arrayBalok > Segitiga.java >
1  package arrayBalok;
2
3  public class Segitiga {
4      public int alas;
5      public int tinggi;
6
7      public Segitiga(int a, int t) {
8          alas = a;
9          tinggi = t;
10     }
11 }
12

```

3. Tambahkan method `hitungLuas()` dan `hitungKeliling()` pada class `Segitiga` tersebut. **Asumsi segitiga adalah segitiga siku-siku.** (Hint: Anda dapat menggunakan bantuan library `Math` pada Java untuk mengkalkulasi sisi miring)

```

arrayBalok > Segitiga.java >
1  package arrayBalok;
2
3  public class Segitiga {
4      public int alas;
5      public int tinggi;
6
7      public Segitiga(int a, int t) {
8          alas = a;
9          tinggi = t;
10     }
11
12     public int hitungLuas() {
13         return (alas * tinggi) / 2;
14     }
15
16     public int hitungKeliling() {
17         return alas * 3;
18     }
19 }
20

```

4. Pada fungsi `main`, buat array `Segitiga sgArray` yang berisi 4 elemen, isikan masing-masing atributnya sebagai berikut:



sgArray ke-0	alas: 10, tinggi: 4
sgArray ke-1	alas: 20, tinggi: 10
sgArray ke-2	alas: 15, tinggi: 6
sgArray ke-3	alas: 25, tinggi: 10

```

12 public int hitungLuas() {
13     return (alas * tinggi) / 2;
14 }
15
16 public int hitungKeliling() {
17     return alas * 3;
18 }
19
20 public static void main(String[] args) {
21     Segitiga[] sgArray = new Segitiga[3];
22
23     sgArray[0] = new Segitiga(a:10, t:4);
24     sgArray[1] = new Segitiga(a:20, t:10);
25     sgArray[2] = new Segitiga(a:15, t:6);
26
27     for (int i = 0; i < sgArray.length; i++) {
28         System.out.println("Luas segitiga ke " + i + " : " + sgArray[i].hitungLuas());
29         System.out.println("Keliling segitiga ke " + i + " : " + sgArray[i].hitungKeliling());
30         System.out.println(x:"=====");
31     }
32 }
33

```

- Kemudian menggunakan looping, cetak luas dan keliling dengan cara memanggil method `hitungLuas()` dan `hitungKeliling()`.

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Keliling segitiga ke 0 : 30
=====
Luas segitiga ke 1 : 100
Keliling segitiga ke 1 : 60
=====
Luas segitiga ke 2 : 45
Keliling segitiga ke 2 : 45
=====
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algor

```

### 3.5 Latihan Praktikum

Waktu: 150 menit

1. Buatlah program yang dapat **menghitung luas permukaan dan volume** bangun ruang **kerucut, limas segi empat sama sisi, dan bola**. Buatlah 3 (tiga) class sesuai dengan jumlah jenis bangun ruang. Buatlah satu main class untuk membuat *array of objects* yang menginputkan atribut-atribut yang ada menggunakan konstruktor semua bangun ruang tersebut. Dengan ketentuan,
  - a. **Buat looping** untuk menginputkan masing-masing atributnya, kemudian **tampilkan luas permukaan dan volume** dari tiap jenis bangun ruang tersebut.
  - b. *Pada kerucut, inputan untuk atribut hanya jari-jari dan sisi miring*
  - c. *Pada limas segi empat sama sisi, inputan untuk atribut hanya panjang sisi alas dan tinggi limas*
  - d. *Pada bola, inputan untuk atribut hanya jari-jari*

#### LimasSegiEmpat

```

1 package bangunruang;
2
3 public class LimasSegiEmpatSamaSisi {
4
5     private double panjangSisiAlas;
6     private double tinggilimas;
7
8     public LimasSegiEmpatSamaSisi(double panjangSisiAlas, double tinggilimas) {
9         this.panjangSisiAlas = panjangSisiAlas;
10        this.tinggilimas = tinggilimas;
11    }
12
13    public double getLuasPermukaan() {
14        double luasAlas = panjangSisiAlas * panjangSisiAlas;
15        double luasSelimut = 4 * (1.0 / 2) * panjangSisiAlas * tinggilimas;
16        return luasAlas + luasSelimut;
17    }
18
19    public double getVolume() {
20        return (1.0 / 3) * getLuasAlas() * tinggilimas;
21    }
22
23    private double getLuasAlas() {
24        return panjangSisiAlas * panjangSisiAlas;
25    }
26 }

```

```

gunruang > Kerucut.java > Main
package bangunruang;

public class Kerucut {

    private double jariJari;
    private double sisiMiring;

    public Kerucut(double jariJari, double sisiMiring) {
        this.jariJari = jariJari;
        this.sisiMiring = sisiMiring;
    }

    public double getLuasPermukaan() {
        double luasAlas = Math.PI * jariJari * jariJari;
        double luasSelimut = Math.PI * jariJari * sisiMiring;
        return luasAlas + luasSelimut;
    }

    public double getVolume() {
        return (1.0 / 3) * Math.PI * jariJari * jariJari * sisiMiring;
    }
}

```

## Bola

```

gunruang > Bola.java > Main
package bangunruang;

public class Bola {

    private double jariJari;

    public Bola(double jariJari) {
        this.jariJari = jariJari;
    }

    public double getLuasPermukaan() {
        return 4 * Math.PI * jariJari * jariJari;
    }

    public double getVolume() {
        return (4.0 / 3) * Math.PI * jariJari * jariJari * jariJari;
    }
}

```

## BangunRuang

```

1  package bangunruang;
2
3  public class BangunRuang {
4      Bola bola;
5      Kerucut kerucut;
6      LimasSegiEmpatSamaSisi limasSegiEmpatSamaSisi;
7
8      BangunRuang(Bola bola) {
9          this.bola = bola;
10     }
11
12     BangunRuang(Kerucut kerucut) {
13         this.kerucut = kerucut;
14     }
15
16     BangunRuang(LimasSegiEmpatSamaSisi limasSegiEmpatSamaSisi) {
17         this.limasSegiEmpatSamaSisi = limasSegiEmpatSamaSisi;
18     }
19
20     public double getLuasPermukaan() {
21         if (bola != null) {
22             return bola.getLuasPermukaan();
23         } else if (kerucut != null) {
24             return kerucut.getLuasPermukaan();
25         } else {
26             return limasSegiEmpatSamaSisi.getLuasPermukaan();
27         }
28     }
29
30     public double getVolume() {
31         if (bola != null) {
32             return bola.getVolume();
33         } else if (kerucut != null) {
34             return kerucut.getVolume();
35         } else {
36             return limasSegiEmpatSamaSisi.getVolume();
37         }
38     }
39 }

```

Main

```

1 package bangunruang;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // Jumlah bangun ruang
7         int jumlahBangunRuang = 3;
8
9         // Array of objects
10        BangunRuang[] bangunRuangs = new BangunRuang[jumlahBangunRuang];
11
12        // Looping untuk input data dan menghitung luas permukaan dan volume
13        for (int i = 0; i < jumlahBangunRuang; i++) {
14            System.out.println("\nBangun Ruang " + (i + 1) + ":");
15            System.out.print("Jenis bangun ruang (kerucut, limas, bola): ");
16            String jenisBangunRuang = System.console().readLine().toLowerCase();
17
18            switch (jenisBangunRuang) {
19                case "kerucut":
20                    System.out.print("Jari-jari: ");
21                    double jariJariKerucut = Double.parseDouble(System.console().readLine());
22                    System.out.print("Sisi miring: ");
23                    double sisiMiringKerucut = Double.parseDouble(System.console().readLine());
24                    bangunRuangs[i] = new BangunRuang(new Kerucut(jariJariKerucut, sisiMiringKerucut));
25                    System.out.println("Luas permukaan: " + bangunRuangs[i].getLuasPermukaan());
26                    System.out.println("Volume: " + bangunRuangs[i].getVolume());
27                    break;
28                case "limas":
29                    System.out.print("Panjang sisi alas: ");
30                    double panjangSisiAlasLimas = Double.parseDouble(System.console().readLine());
31                    System.out.print("Tinggi limas: ");
32                    double tinggilimas = Double.parseDouble(System.console().readLine());
33                    bangunRuangs[i] = new BangunRuang(new LimasSegiEmpatSamaSisi(panjangSisiAlasLimas, tinggilimas));
34                    System.out.println("Luas permukaan: " + bangunRuangs[i].getLuasPermukaan());
35                    System.out.println("Volume: " + bangunRuangs[i].getVolume());
36                    break;
37                case "bola":
38                    System.out.print("Jari-jari: ");
39                    double jariJariBola = Double.parseDouble(System.console().readLine());
40                    bangunRuangs[i] = new BangunRuang(new Bola(jariJariBola));
41                    System.out.println("Luas permukaan: " + bangunRuangs[i].getLuasPermukaan());
42                    System.out.println("Volume: " + bangunRuangs[i].getVolume());
43                    break;
44                default:
45                    System.out.println("Jenis bangun ruang tidak valid!");
46                    i--;
47                    continue;
48            }
49        }
50
51        // Tampilkan hasil
52        for (BangunRuang bangunRuang : bangunRuangs) {
53            System.out.println("\n" + bangunRuang.getClass().getSimpleName() + ":");
54            System.out.println("Luas permukaan: " + bangunRuang.getLuasPermukaan());
55            System.out.println("Volume: " + bangunRuang.getVolume());
56        }
57    }
58 }
59

```



```

\Code\User\workspaceStorage\8a7ac70f5fba722e04ecb33498fd4209\redhat.java\jdt_ws\jobsheet 3

Bangun Ruang 1:
Jenis bangun ruang (kerucut, limas, bola): kerucut
Jari-jari: 7
Sisi miring: 10
Luas permukaan: 373.84952577718536
Volume: 513.1268000863329

Bangun Ruang 2:
Jenis bangun ruang (kerucut, limas, bola): bola
Jari-jari: 7
Luas permukaan: 615.7521601035994
Volume: 1436.755040241732

Bangun Ruang 3:
Jenis bangun ruang (kerucut, limas, bola): limas
Panjang sisi alas: 10
Tinggi limas: 10
Luas permukaan: 300.0
Volume: 333.3333333333326
    
```

- Sebuah kampus membutuhkan program untuk menampilkan informasi mahasiswa berupa nama, nim, jenis kelamin dan juga IPK mahasiswa. Program dapat menerima input semua informasi tersebut, kemudian menampilkanya kembali ke user. Implementasikan program tersebut jika dimisalkan terdapat 3 data mahasiswa yang tersedia. Contoh output program:

```

Masukkan data mahasiswa ke- 1
Masukkan nama :Rina
Masukkan nim :1234567
Masukkan jenis kelamin :P
Masukkan IPK :3.5
Masukkan data mahasiswa ke- 2
Masukkan nama :Rio
Masukkan nim :7654321
Masukkan jenis kelamin :L
Masukkan IPK :4.0
Masukkan data mahasiswa ke- 3
Masukkan nama :Reza
Masukkan nim :8765398
Masukkan jenis kelamin :L
Masukkan IPK :3.8
    
```

```

Data Mahasiswa ke-1
nama : Rina
nim : 1234567
Jenis kelamin : P
Nilai IPK : 3.5
Data Mahasiswa ke-2
nama : Rio
nim : 7654321
Jenis kelamin : L
Nilai IPK : 4.0
Data Mahasiswa ke-3
nama : Reza
nim : 8765398
Jenis kelamin : L
Nilai IPK : 3.8
    
```

- Modify the resulting program at no.2, so that it could be used to calculate Average GPA, as well as to display student information that has biggest GPA! (use method to implement each process)



```

1  package mahasiswa;
2
3  import java.util.Scanner;
4
5  public class Mahasiswa {
6      String nim, nama;
7      char jenisKelamin;
8      double ipk;
9
10     public Mahasiswa(String nim, String nama, char jenisKelamin, double ipk) {
11         this.nim = nim;
12         this.nama = nama;
13         this.jenisKelamin = jenisKelamin;
14         this.ipk = ipk;
15     }
16
17
18     public static void main(String[] args) {
19         Scanner sc = new Scanner(System.in);
20
21         double sumIpk = 0;
22         double topIpk = 0;
23
24         Mahasiswa[] mhsArray = new Mahasiswa[3];
25
26         for(int i = 0; i < mhsArray.length; i++) {
27             System.out.println("Mahasiswa ke-" + i);
28             System.out.print("NIM: ");
29             String nim = sc.next();
30             System.out.print("Nama: ");
31             String nama = sc.next();
32             System.out.print("Jenis Kelamin: ");
33             char jenisKelamin = sc.next().charAt(0);
34             System.out.print("IPK: ");
35             double ipk = sc.nextDouble();
36
37             sumIpk += ipk;
38             if (ipk > topIpk) {
39                 topIpk = ipk;
40             }
41
42             mhsArray[i] = new Mahasiswa(nim, nama, jenisKelamin, ipk);
43         }
44
45         for (Mahasiswa m: mhsArray) {
46
47             System.out.println("NIM: " + m.nim);
48             System.out.println("Nama: " + m.nama);
49             System.out.println("Jenis Kelamin: " + m.jenisKelamin);
50             System.out.println("IPK: " + m.ipk);
51
52             if (m.ipk == topIpk) {
53                 System.out.println("Mahasiswa ini memiliki IPK tertinggi");
54             }
55
56             System.out.println("-----");
57         }
58
59         System.out.println("Rata-rata IPK kelas: " + sumIpk / mhsArray.length);
60
61         sc.close();
62     }
63 }
64
65
66
67
68

```



```
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 3> & C:\Program
\Code\User\workspaceStorage\8a7ac70f5fba722e04ecb33498fd4209\redhat.java\jdt_ws\jobs
Mahasiswa ke-0
NIM: 123
Nama: Agus
Jenis Kelamin: L
IPK: 3.5
Mahasiswa ke-1
NIM: 1234
Nama: Anto
Jenis Kelamin: L
IPK: 3.2
Mahasiswa ke-2
NIM: 12345
Nama: Rani
Jenis Kelamin: P
IPK: 3.95
NIM: 123
Nama: Agus
Jenis Kelamin: L
IPK: 3.5
-----
NIM: 1234
Nama: Anto
Jenis Kelamin: L
IPK: 3.2
-----
NIM: 12345
Nama: Rani
Jenis Kelamin: P
IPK: 3.95
Mahasiswa ini memiliki IPK tertinggi
-----
Rata-rata IPK kelas: 3.5500000000000003
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 3> |
```