

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET PERTEMUAN KE-3**



**RIO TRI PRAYOGO**

**TI 1A**

**26**

**2341720236**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Praktikum 1 : Membuat Array dari Object, Mengisi dan Menampilkan

### Percobaan :

#### Persegi Panjang

```
package minggu3;

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

#### Object

```
package minggu3;

public class ArrayObjects26 {
    public static void main(String[] args) {
        PersegiPanjang[] ppArray = new PersegiPanjang[3];

        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;

        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);
    }
}
```

```
}
```

#### Verifikasi :

```
Persegi Panjang ke-0, panjang: 110, lebar: 30
Persegi Panjang ke-1, panjang: 80, lebar: 40
Persegi Panjang ke-2, panjang: 100, lebar: 20
```

#### Pertanyaan :

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method?Jelaskan!  
= Tidak, array of object tidak harus memiliki atribut dan method sekaligus. Array of object bisa hanya memiliki atribut atau object saja, tidak harus keduanya.
2. Apakah class PersegiPanjang memiliki konstruktor?Jika tidak, kenapa dilakukan pemanggilan konstruktor pada baris program berikut :  

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

  
= Tidak, class PersegiPanjang tidak memiliki konstruktor. Baris kode diatas dilakukan untuk menginstansiasi object pada array index ke '1' dari array of object PersegiPanjang yang telah dipanggil pada line: 

```
PersegiPanjang[] ppArray = new PersegiPanjang[3];
```
3. Apa yang dimaksud dengan kode berikut ini:  

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

  
= Baris kode diatas melakukan instansiasi atau pemanggilan dari array of object PersegiPanjang dengan panjang array sebanyak '3'.
4. Apa yang dimaksud dengan kode berikut ini:  

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

  
= Baris kode pertama melakukan instansiasi object pada array index ke '1' dari array of object PersegiPanjang. Baris kode kedua dan ketiga melakukan instansiasi dari atribut '*panjang*' dan '*lebar*' dari object PersegiPanjang didalam index '1' seperti yang telah dipanggil sebelumnya.
5. Mengapa class main dan juga class PersegiPanjang dipisahkan pada uji coba 3.2?  
= Karena class PersegiPanjang merupakan public class dan bukan merupakan sebuah konstruktor. Sehingga class PersegiPanjang digabung didalam class 'ArrayObjects26'. Baik didalam class maupun diluar.

#### Praktikum 2 : Menerima Input Isian Array Menggunakan Looping

#### Percobaan :

```
package minggu3;

import java.util.Scanner;

public class ArrayObjects26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        PersegiPanjang[] ppArray = new PersegiPanjang[3];
```

```

        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 = scan.nextInt();
            System.out.print("Masukkan lebar: ");
            ppArray[i].lebar = scan.nextInt();
        }

        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);
        }
    }
}

```

#### Verifikasi :

```

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

```

#### Pertanyaan :

1. Apakah array of object dapat diimplementasikan pada array 2 Dimensi?  
= Bisa, array of object tidak hanya bisa diimplementasikan pada array biasa tetapi juga dapat diimplementasikan pada array 2 dimensi.

2. Jika jawaban soal no satu iya, berikan contohnya! Jika tidak, jelaskan!  
= Berikut merupakan contoh implementasi dari array 2 dimensi di dalam array of object:

```
package minggu3;

public class DataOrang26 {
    public static void main(String[] args) {
        Orang[][] dataOrang = new Orang[2][2];

        dataOrang[0][0] = new Orang();
        dataOrang[0][0].nama = "Rio";
        dataOrang[0][0].umur = 20;
        dataOrang[0][1] = new Orang();
        dataOrang[0][1].nama = "Ren";
        dataOrang[0][1].umur = 25;

        System.out.println(dataOrang[0][0].nama + " berumur " +
            dataOrang[0][0].umur + " tahun");

        System.out.println(dataOrang[0][1].nama + " berumur " +
            dataOrang[0][1].umur + " tahun");
    }
}
```

```
package minggu3;

public class Orang {
    public String nama;
    public int umur;
}
```

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;
```

= Kode diatas memunculkan error saat dijalankan karena tidak ada instansiasi object Persegi pada array index ke '5'. Seharusnya diinstansiasi terlebih dahulu seperti kode ini:

```
pgArray[5] = new Persegi();
```

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

```
package minggu3;
```

```

import java.util.Scanner;

public class ArrayObjects26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Masukkan banyak persegi panjang yang dimiliki:
");
        int n = scan.nextInt();
        PersegiPanjang[] ppArray = new PersegiPanjang[n];

        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 = scan.nextInt();
            System.out.print("Masukkan lebar: ");
            ppArray[i].lebar = scan.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);
        }
    }
}

```

Output :

```

Masukkan banyak persegi panjang yang dimiliki: 2
Persegi panjang ke-0
Masukkan panjang: 10
Masukkan lebar: 10
Persegi panjang ke-1
Masukkan panjang: 5
Masukkan lebar: 5
Persegi panjang ke-0
Panjang: 10, lebar: 10
Persegi panjang ke-1
Panjang: 5, lebar: 5

```

5. Apakah boleh Jika terjadi duplikasi instansiasi array of objek, misalkan saja instansiasi dilakukan pada `ppArray[i]` sekaligus `ppArray[0]`?Jelaskan !  
= Duplikasi instansiasi array of object bisa dilakukan, akan tetapi hasil instansiasi yang paling baru akan menumpuk hasil instansiasi sebelumnya pada index yang sama. Sehingga index pada array of object tersebut hanya akan mengambil isi dari instansiasi array of object yang paling baru dan menghapus isi dari instansiasi sebelumnya.

### Praktikum 3 : Penambahan Operasi Matematika di Dalam Method

#### Percobaan :

##### Balok

```
package minggu3;

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;
    }
}
```

##### Object

```
package minggu3;

public class ArrayBalok26 {
    public static void main(String[] args) {
        Balok[] bilArray = new Balok[3];

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

```

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

```

#### Verifikasi :

```

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

```

#### Pertanyaan :

1. Dapatkah konstruktor berjumlah lebih dalam satu kelas? Jelaskan dengan contoh!  
= Iya dapat, satu buah class bisa memiliki lebih dari satu konstruktor. Hal ini dapat dilihat pada percobaan Praktikum 3 diatas atau lebih tepatnya pada baris:
  - Konstruktor :

```

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

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

```

- Instansiasi :

```

...
bilArray[2] = new Balok(210, 50, 25);

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

```



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.

```
package minggu3;  
  
public class Segitiga {  
    public int alas;  
    public int tinggi;  
  
    public Segitiga(int a, int t) {  
        alas = a;  
        tinggi = t;  
    }  
}
```

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)

```
...  
    public double hitungLuas() {  
        return 0.5 * alas * tinggi;  
    }  
  
    public double hitungKeliling() {  
        double sisiMiring = Math.sqrt(Math.pow(alas, 2) +  
Math.pow(tinggi, 2));  
        return alas + tinggi + sisiMiring;  
    }  
}
```

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

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

```

...
public static void main(String[] args) {
    Segitiga[] sgArray = new Segitiga[4];
    sgArray[0] = new Segitiga(10, 4);
    sgArray[1] = new Segitiga(20, 10);
    sgArray[2] = new Segitiga(15, 6);
    sgArray[3] = new Segitiga(25, 10);

    for (int i = 0; i < sgArray.length; i++) {
        System.out.println("sgArray ke-" + i + "\ntalas: " +
            sgArray[i].alas + ", tinggi: " + sgArray[i].tinggi);
    }
}

```

```

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

```

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

```

...
for (int i = 0; i < sgArray.length; i++) {
    System.out.println("Luas segitiga ke-" + i + ": " +
        sgArray[i].hitungLuas());
    System.out.println("Keliling segitiga ke-" + i + ": " +
        sgArray[i].hitungKeliling());
}
...

```

```

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
Luas segitiga ke-0: 20.0
Keliling segitiga ke-0: 24.77032961426901
Luas segitiga ke-1: 100.0
Keliling segitiga ke-1: 52.3606797749979
Luas segitiga ke-2: 45.0
Keliling segitiga ke-2: 37.15549442140351
Luas segitiga ke-3: 125.0
Keliling segitiga ke-3: 61.92582403567252

```

### Latihan Praktikum :

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***

**Jawab:**

a) Main

```
package minggu3.BangunRuang;

import java.util.Scanner;

public class BangunMain26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        boolean run = true;
        do {
            System.out.println("Pilih Bangun: ");
            System.out.println("1 = Kerucut");
            System.out.println("2 = Limas Segi Empat");
            System.out.println("3 = Bola");
            System.out.print("(1/2/3): ");
            int input = scan.nextInt();
            switch (input) {
                case 1:
                    System.out.println("Masukkan jumlah bangun kerucut yang dimiliki: ");
                    int n = scan.nextInt();
                    Kerucut[] krArray = new Kerucut[n];
                    for (int i = 0; i < krArray.length; i++) {
                        System.out.println("Bangun Ruang Kerucut ke-" + (i + 1));
                        krArray[i] = new Kerucut();
                    }
                }
            }
        } while (run);
    }
}
```

```

        System.out.print("Masukkan jari-jari: ");
        krArray[i].jariJari = scan.nextDouble();
        System.out.print("Masukkan sisi miring : ");
        krArray[i].sisiMiring = scan.nextDouble();
        System.out.println("Luas Permukaan Bangun
Kerucut ke-" + (i + 1)+": " + krArray[i].lpKerucut());
        System.out.println("Volume Bangun Kerucut ke-" +
(i + 1)+": " + krArray[i].volKerucut());
    }
    break;
    case 2:
        System.out.println("Masukkan jumlah bangun limas
segi empat yang dimiliki: ");
        n = scan.nextInt();
        Limas[] lmArray = new Limas[n];
        for (int i = 0; i < lmArray.length; i++) {
            System.out.println("Bangun Ruang Limas Segi
Empat ke-" + (i + 1);
            lmArray[i] = new Limas();
            System.out.print("Masukkan alas: ");
            lmArray[i].sisi = scan.nextDouble();
            System.out.print("Masukkan tinggi: ");
            lmArray[i].tinggi = scan.nextDouble();
            System.out.println("Luas Permukaan Bangun
Kerucut ke-" + (i + 1)+": " + lmArray[i].lpLimas());
            System.out.println("Volume Bangun Kerucut ke-" +
(i + 1)+": " + lmArray[i].volLimas());
        }
        break;
    case 3:
        System.out.println("Masukkan jumlah bangun bola yang
dimiliki: ");
        n = scan.nextInt();
        Bola[] blArray = new Bola[n];
        for (int i = 0; i < blArray.length; i++) {
            System.out.println("Bangun Ruang Bola ke-" + (i
+ 1);
            blArray[i] = new Bola();
            System.out.print("Masukkan jari-jari: ");

```

```

        blArray[i].jariJari = scan.nextDouble();

        System.out.println("Luas Permukaan Bangun
Kerucut ke-" + (i + 1) + ": " + blArray[i].lpBola());

        System.out.println("Volume Bangun Kerucut ke-" +
(i + 1) + ": " + blArray[i].volBola());

    }

    break;

default:

    System.out.println("Pilihan tidak valid");

    break;

}

} while (run);

}

}

```

## b) Kerucut

```

package minggu3.BangunRuang;

public class Kerucut {

    public double jariJari;

    public double sisiMiring;

    public double lpKerucut() {

        double selimut = Math.PI * jariJari * sisiMiring;

        return Math.PI * jariJari * (selimut + jariJari);

    }

    public double volKerucut() {

        double tinggi = Math.sqrt(Math.pow(sisiMiring, 2) -
Math.pow(jariJari, 2));

        return (1.0 / 3) * Math.PI * jariJari * jariJari * tinggi;

    }

}

```

```

Pilih Bangun:
1 = Kerucut
2 = Limas Segi Empat
3 = Bola
(1/2/3): 1
Masukkan jumlah bangun kerucut yang dimiliki:
2
Bangun Ruang Kerucut ke-1
Masukkan jari-jari: 5
Masukkan sisi miring : 6
Luas Permukaan Bangun Kerucut ke-1: 1558.9804765031483
Volume Bangun Kerucut ke-1: 86.82903396745259
Bangun Ruang Kerucut ke-2
Masukkan jari-jari: 4
Masukkan sisi miring : 5
Luas Permukaan Bangun Kerucut ke-2: 839.8338345445853
Volume Bangun Kerucut ke-2: 50.26548245743669

```

### c) Limas Segi Empat

```
package minggu3.BangunRuang;

public class Limas {
    public double sisi;
    public double tinggi;

    public double lpLimas() {
        double sisiTegak = 0.5 * sisi * tinggi;
        return (sisi * sisi) + (4 * sisiTegak);
    }

    public double volLimas() {
        return (1.0 / 3) * (sisi * sisi) * tinggi;
    }
}
```

Pilih Bangun:

1 = Kerucut

2 = Limas Segi Empat

3 = Bola

(1/2/3): 2

Masukkan jumlah bangun limas segi empat yang dimiliki:

1

Bangun Ruang Limas Segi Empat ke-1

Masukkan alas: 5

Masukkan tinggi: 5

Luas Permukaan Bangun Kerucut ke-1: 75.0

Volume Bangun Kerucut ke-1: 41.666666666666666

### d) Bola

```
package minggu3.BangunRuang;

public class Bola {
    public double jariJari;

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

```

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

```

```

Pilih Bangun:
1 = Kerucut
2 = Limas Segi Empat
3 = Bola
(1/2/3): 3
Masukkan jumlah bangun bola yang dimiliki:
1
Bangun Ruang Bola ke-1
Masukkan jari-jari: 7
Luas Permukaan Bangun Kerucut ke-1: 615.7521601035994
Volume Bangun Kerucut ke-1: 1436.755040241732

```

- 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

```

Object :

```

package minggu3;

public class Mahasiswa {
    public String nama;
    public int nim;
    public char jk;
    public float ipk;
}

```

Main :

```
package minggu3;

import java.util.Scanner;

public class MahasiswaMain26 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Mahasiswa[] mhsArray = new Mahasiswa[3];

        for (int i = 0; i < mhsArray.length; i++) {
            mhsArray[i] = new Mahasiswa();
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1));
            System.out.print("Masukkan nama: ");
            mhsArray[i].nama = scan.nextLine();
            System.out.print("Masukkan NIM: ");
            mhsArray[i].nim = scan.nextInt();
            System.out.print("Masukkan jenis kelamin: ");
            mhsArray[i].jk = scan.next().charAt(0);
            System.out.print("Masukkan IPK: ");
            mhsArray[i].ipk = scan.nextFloat();
            scan.nextLine();
        }
        System.out.println();
        for (int i = 0; i < mhsArray.length; i++) {
            System.out.println("Data mahasiswa ke-" + (i + 1));
            System.out.println("Nama: " + mhsArray[i].nama);
            System.out.println("NIM: " + mhsArray[i].nim);
            System.out.println("Jenis kelamin: " + mhsArray[i].jk);
            System.out.println("Nilai IPK " + mhsArray[i].ipk);
        }
    }
}
```



Output :

```
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
```

3. Modifikasi program Latihan no.2 di atas, sehingga bisa digunakan untuk menghitung rata-rata IPK, serta menampilkan data mahasiswa dengan IPK terbesar! (gunakan method untuk masing-masing proses tersebut)

Object

```
...

    public float hitungRataRataIPK(Mahasiswa[] mahasiswa) {
        float totalIPK = 0;
        for (Mahasiswa mhs : mahasiswa) {
            totalIPK += mhs.ipk;
        }
        return totalIPK / mahasiswa.length;
    }

    public Mahasiswa ipTertinggi(Mahasiswa[] mahasiswa) {
        Mahasiswa mahasiswaTerbaik = mahasiswa[0];
        for (Mahasiswa mhs : mahasiswa) {
            if (mhs.ipk > mahasiswaTerbaik.ipk) {
                mahasiswaTerbaik = mhs;
            }
        }
        return mahasiswaTerbaik;
    }
}
```

## Main

```
...

    float rataRataIPK = mhsArray[0].hitungRataRataIPK(mhsArray);

    System.out.println("\nRata-rata IPK: " + rataRataIPK);

    Mahasiswa mahasiswaTerbaik = mhsArray[0].ipTertinggi(mhsArray);

    System.out.println("\nData Mahasiswa dengan IPK Terbesar:");

    mahasiswaTerbaik.tampilData();

}

...
```

## Output

```
Masukkan data mahasiswa ke-1
Masukkan nama: Rio
Masukkan NIM: 123
Masukkan jenis kelamin: L
Masukkan IPK: 3.5
Masukkan data mahasiswa ke-2
Masukkan nama: Yui
Masukkan NIM: 456
Masukkan jenis kelamin: P
Masukkan IPK: 4.0
Masukkan data mahasiswa ke-3
Masukkan nama: Ken
Masukkan NIM: 789
Masukkan jenis kelamin: L
Masukkan IPK: 3.0

Data mahasiswa ke-1
Nama: Rio
NIM: 123
Jenis kelamin: L
IPK: 3.5
Data mahasiswa ke-2
Nama: Yui
NIM: 456
Jenis kelamin: P
IPK: 4.0
Data mahasiswa ke-3
Nama: Ken
NIM: 789
Jenis kelamin: L
IPK: 3.0

Rata-rata IPK: 3.5

Data Mahasiswa dengan IPK Terbesar:
Nama: Yui
NIM: 456
Jenis kelamin: P
IPK: 4.0
```