



## JOBSHEET III ARRAY OF OBJEK

### 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;
```



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

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

3. Apa yang dimaksud dengan kode berikut ini:

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

4. Apa yang dimaksud dengan kode berikut ini:

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

5. Mengapa class main dan juga class `PersegiPanjang` dipisahkan pada uji coba 3.2?

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

**Waktu: 50 menit**

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
```

### 3.3.3 Pertanyaan

1. Apakah array of object dapat diimplementasikan pada array 2 Dimensi?
2. Jika jawaban soal no satu iya, berikan contohnya! Jika tidak, jelaskan!



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;
```

4. Modifikasi kode program pada praktikum 3.3 agar length array menjadi inputan dengan Scanner!
5. Apakah boleh Jika terjadi duplikasi instansiasi array of objek, misalkan saja instansiasi dilakukan pada **ppArray[i]** sekaligus **ppArray[0]**?Jelaskan !

### 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!
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.

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



### 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***
2. 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
```

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)