



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 Membuat Array dari Object, Mengisi dan Menampilkan

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**”. Buat package dengan nama minggu3.
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

- Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method?Jelaskan! atribut dan method tidak harus selalu ada dalam class yang akan dibuat array of object, tetapi akan tergantung pada kebutuhan desain class yang lebih besar yang ingin kita buat.
- Apakah class `PersegiPanjang` memiliki konstruktor?Jika tidak, kenapa dilakukan pemanggilan konstruktor pada baris program berikut : `class PersegiPanjang` tidak memiliki konstruktor.

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

Pemanggilan konstruktor pada baris program tersebut merupakan contoh dari pembuatan objek tanpa menentukan nilai atributnya.

3. Apa yang dimaksud dengan kode berikut ini: `Kode di atas adalah kode untuk membuat array PersegiPanjang dg ukuran 3.`

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

4. Apa yang dimaksud dengan kode berikut ini:

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

menambahkan objek baru dari kelas `PersegiPanjang` ke dalam array `ppArray` pada indeks ke-1.
`ppArray[1].panjang = 80;` mengatur nilai dari properti panjang objek `PersegiPanjang` yang terletak pada indeks ke-1 di array `ppArray`
`ppArray[1].lebar = 40;` mengatur nilai dari properti lebar objek `PersegiPanjang` yang terletak pada indeks ke-1 di array `ppArray`

5. Mengapa class main dan juga class `PersegiPanjang` dipisahkan pada uji coba 3.2?
 agar kode program menjadi lebih rapi, mudah dibaca, dan didebug

3.3 Menerima Input Isian Array Menggunakan Looping

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? Ya, array of object dpt diimplementasikan pada array 2 dimensi.
2. Jika jawaban soal no satu iya, berikan contohnya! Jika tidak, jelaskan!

```
PersegiPanjang[][] pp2DArray = new PersegiPanjang[5][5];
pp2DArray[0][0] = new PersegiPanjang(5, 5);
pp2DArray[0][1] = new PersegiPanjang(6, 6);
```
3. Jika diketahui terdapat class **Persegi** yang memiliki atribut sisi bertipe integer, maka kode dibawah ini akan memunculkan error saat dijalankan. Mengapa?

elemen dalam array belum diinisialisasi dengan objek Persegi sebelum mengakses atribut sisi

```
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 ! Boleh, karena hal ini tidak menyebabkan error pada kode program. ketika kita membuat objek baru untuk variabel tertentu, kita tidak akan mengubah data dari objek yang sudah ada sebelumnya.

3.4 Operasi Matematika Atribut Object Array

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 Tidak ada batasan formal yang menghendaki konstruktor berjumlah tunggal dalam satu kelas. Tetapi, tergantung pada konvensi dan praktek yang umum, sebuah kelas dapat memiliki beberapa konstruktor yang memiliki argumen yang berbeda

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.
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

1. Buatlah program yang dapat menghitung luas permukaan dan volume beberapa bangun ruang (minimal 3, jenis bangun luas bebas). Buatlah 3 (tiga) class sesuai dengan jumlah jenis bangun ruang. Dan buat satu main class untuk membuat array of objek yang menginputkan atribut-atribut yang ada menggunakan konstruktor semua bangun ruang tersebut.
Keterangan : Buat looping untuk menginputkan masing-masing atributnya, kemudian tampilkan luas permukaan dan volume dari tiap jenis bangun ruang tersebut.
2. Sebuah perusahaan jual beli tanah membutuhkan sebuah program yang dapat memberitahu luas tanah. Program tersebut dapat menerima input jumlah tanah yang akan dihitung dan atribut



panjang dan lebar masing-masing dari tanah yang diinputkan. Program dapat menampilkan luas tanah masing-masing tanah yang diinputkan tadi dan juga tanah terluas dari ketiga tanah yang ada. Contoh output program:

```
Jumlah tanah: 3

Tanah 1
Panjang: 100
Lebar: 40
Tanah 2
Panjang: 250
Lebar: 100
Tanah 3
Panjang: 120
Lebar: 100

Luas Tanah 1: 4000
Luas Tanah 2: 25000
Luas Tanah 3: 12000

Tanah terluas: Tanah 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
```