



Nama

Raudhil Firdaus Naufal

NIM

2341720164

Kelas

1E

Jurusan

Teknologi Informasi

Prodi

Teknik Informatika

Percobaan 1: Membuat Array dari Object, Mengisi dan Menampilkan.

Hasil Percobaan:

```
Persegi Panjang ke-0, panjang 110 lebar: 30  
Persegi Panjang ke-0, panjang 80 lebar: 40  
Persegi Panjang ke-0, 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 selalu, karena pada percobaan 3.2 hanya diisi dengan atribut, jadi tanpa method masih bisa dijalankan. Tetapi untuk kasus sebenarnya, lebih baik untuk memakai atribut dan juga method.

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

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

Tidak, karena dilakukan untuk membuat objek saja. Pemanggilan konstruktor memang terjadi tetapi akan otomatis menggunakan konstruktor default untuk membuat objek dari java.

3. Apa yang dimaksud dengan kode berikut ini:

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

Maksud dari kode ini adalah, merupakan deklarasi dan instansiasi untuk membuat array dengan nama variabel ppArray dengan 3 elemen yang akan disimpan di class persegiPanjang.

4. Apa yang dimaksud dengan kode berikut ini:

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

Pada kode berikut kita membuat objek baru dari persegiPanjang pada ppArray[1], lalu diisi dengan panjang dan lebarnya, sehingga pada ppArray[1] memiliki nilai panjang sebesar 80 dan lebar sebesar 40.

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

Karena masing masing dari class tersebut memiliki tugas tersendiri, seperti class persegiPanjang yang digunakan untuk mendeklarasikan atribut sedangkan class main digunakan untuk menggunakan atribut yang ada pada class persegiPanjang.

Percobaan 2: Menerima Input Isian Array Menggunakan Looping.

Hasil Percobaan:

```
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?

Ya, array of object dapat diimplementasikan pada array 2 Dimensi.

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

Contohnya seperti ini:

```
persegiPanjang[][] ppArrays = new persegiPanjang[2][3];
```

Disini kita membuat array 2 dimensi dengan 2 baris dan 3 kolom yang bisa diisi dengan objek.

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;
```

Karena kita belum menginisialisasi objek pada array yang ke 5 dan keseluruhannya, kita hanya membuat array dengan 100 memori, tetapi objeknya belum dibuat, untuk membuatnya bekerja, kita harus menambahkan ini:

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

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

```
System.out.print(s:"Masukkan jumlah Persegi Panjang: ");  
persPanjang = sc.nextInt();  
persegiPanjang [] ppArray = new persegiPanjang[persPanjang];
```

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

Ya boleh, karena maksud dari instansiasi ini adalah agar value ppArray[i] sama dengan value ppArray[0], namun masalahnya hanya jika satu nilai berubah, maka nilai lainnya juga akan ikut berubah.

Percobaan 3: Penambahan Operasi Matematika di Dalam Method.

Hasil Percobaan:

```
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!

Ya, konstruktor dapat berjumlah lebih dari satu dalam satu kelas. Contohnya seperti ini:

```
public buku22(){  
}  
  
public buku22(String jud, String pg, int hal, int stok, int har){  
    judul = jud;  
    pengarang = pg;  
    halaman = hal;  
    this.stok = stok;  
    harga = har;  
}
```

Pada kode ini konstruktor berjumlah 2, yang pertama adalah konstruktor tanpa parameter, dan yang kedua adalah konstruktor dengan parameter.

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.

```
public class segitiga {  
    public int alas;  
    public int tinggi;  
  
    public segitiga(int a,int t){  
        this.alas = a;  
        this.tinggi = t;  
  
        System.out.println(a);  
        System.out.println(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)

```
void hitungLuas(){  
    luas = alas * tinggi / 2;  
    System.out.println("Luasnya adalah: " + luas);  
}  
  
void hitungKeliling(){  
    sisiMiring = Math.pow((1/2 * alas),2) + Math.pow(tinggi, 2);  
    keliling = sisiMiring * 2 + alas;  
    System.out.println("Kelilingnya adalah: " + keliling);  
}
```

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

```
public static void main(String[] args) {  
    segitiga [] sgArray = new segitiga[4];  
  
    sgArray[0] = new segitiga(a:10, t:4);  
    sgArray[1] = new segitiga(a:20, t:10);  
    sgArray[2] = new segitiga(a:15, t:6);  
    sgArray[3] = new segitiga(a:25, t:10);  
}
```

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

```
for(int i = 0; i < 3; i++){  
    sgArray[i].hitungLuas();  
    sgArray[i].hitungKeliling();  
}
```

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

Hasil:

```
Bangun Ruang Kerucut  
  
Masukkan Jari - jari: 14  
Masukkan Sisi Miring: 10  
Luas Permukaannya adalah: 1055.04  
Volumenya adalah: 2052.5072003453315  
  
Bangun Ruang Limas Segi Empat Sama Sisi  
  
Masukkan Panjang Sisi Alas: 10  
Masukkan Tinggi Limas: 12  
Luas Permukaannya adalah: 340.0  
Volumenya adalah: 399.99999999999994  
  
Bangun Ruang Bola  
  
Masukkan Jari - jari: 10  
Luas Permukaannya adalah: 1256.6370614359173  
Volumenya adalah: 4188.790204786391
```

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 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
```

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)

```
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
Rata-rata IPK adalah: 3.766666666666667
IPK terbesar adalah: 4.0
```