

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET PERTEMUAN KE-3



NAMA : ALVINO VALERIAN D.R

KELAS : 1A

NO. ABSEN : 05

NIM : 2341720027

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

3.2.3 Pertanyaan 1

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 :
3. Apa yang dimaksud dengan kode berikut ini:
4. Apa yang dimaksud dengan kode berikut ini:
5. Mengapa class main dan juga class PersegiPanjang dipisahkan pada uji coba 3.2?

Jawaban:

1. Sebuah class dapat dibuat array of object meskipun tidak memiliki method. Keberadaan atribut dan method pada class tergantung pada kebutuhan dan desain program
2. class persegi panjang memiliki konstruktor, Pada baris kode ppArray[1]=new persegiPanjang();, terdapat pemanggilan konstruktor persegiPanjang().Konstruktor ini digunakan untuk membuat object baru dari class persegiPanjang.
3. deklarasi array persegiPanjang[] : Tipe data array yang dapat menampung object dari class persegiPanjang. ppArray : Nama variabel array.
4. pparray = new persegi panjang adalah menamahkan object baru kedalam array, untuk array.panjang dan lebar untuk menginisialisasi nilai atribut
5. Memisahkan class membantu menjaga kesederhanaan code, terutama pada class Main. Class Main hanya perlu fokus pada alur program utama, tanpa terbebani dengan detail implementasi class PersegiPanjang.

3.3.3 Pertanyaan 2

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?
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 !

jawaban:

1. array of object dapat diimplementasikan pada array 2 dimensi.

2.

```
for (int i = 0; i < ppArray2D.length; i++) {
    for (int j = 0; j < ppArray2D[i].length; j++) {
        ppArray2D[i][j] = new persegiPanjang();

        System.out.println("Persegi Panjang ke-" + i + "," + j);
        System.out.print("Masukkan Panjang: ");
        ppArray2D[i][j].panjang = sc.nextInt();
        System.out.print("Masukkan Lebar: ");
        ppArray2D[i][j].lebar = sc.nextInt();
    }
}
```

3. Penggunaan pgArray[5].sisi tidak valid:

Kode mencoba mengakses elemen ke-5 dari array pgArray. Namun, array pgArray belum diinisialisasi dengan nilai. Baris pgArray[5].sisi akan menghasilkan NullPointerException karena elemen ke-5 array pgArray masih bernilai null.

4.

```
System.out.print("Masukkan Baris Array: ");
int baris = sc.nextInt();
persegiPanjang[] ppArray = new persegiPanjang[baris];

for (int i=0;i<ppArray.length;i++) {
```

5. tidak dianjurkan menduplikasi instansiasi Saat Anda menduplikasi instansiasi seperti ini, kedua elemen array (ppArray[i] dan ppArray[0]) akan merujuk ke objek yang sama di memori. Mengubah nilai properti pada salah satu elemen akan memengaruhi elemen lainnya.

3.4.3 Pertanyaan

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

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

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()**.

Jawaban

1. Ya, konstruktor dapat berjumlah lebih dari satu dalam satu kelas.

```
public balok(int p,int l,int t){

    panjang = p;
    lebar = l;
    tinggi = t;
}

public balok(int p,int l){

    this.panjang=p;
    this.lebar=l;

}
```

2.

```
public class segitiga {
    public int alas;
    public int tinggi;

    public segitiga(int alas,int tinggi){
        this.alas=alas;
        this.tinggi=tinggi;
    }
}
```

3.

```
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.

```
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);
```

5.

```
for(int i=0;i<4;i++){
    System.out.println("Hitung Luas Segitiga Ke- :"+i+",
Luas:"+String.format("%.2f",sgArray[i].hitungLuas()));
    System.out.println("Hitung Kell Segitiga Ke- :"+i+",
Kell:"+String.format("%.2f",sgArray[i].hitungKeliling()));
}
```

PRAKTIKUM

```
class kerucut {

    private double jariJari;
    private double sisiMiring;
    private double luasPermukaan;
    private double Volume;

    public kerucut(double jariJari, double sisiMiring, double tinggi){
        this.jariJari=jariJari;
        this.sisiMiring=sisiMiring;
        hitungluasPermukaan();
        hitungVolume();
    }
    private void hitungluasPermukaan(){
        luasPermukaan = Math.PI*jariJari*(jariJari+sisiMiring);
    }
    private void hitungVolume(){
        double tinggiKerucut;
        tinggiKerucut= Math.pow(sisiMiring,2) - Math.pow(jariJari,
2);

        tinggiKerucut= Math.sqrt(2);
        Volume = (1.0/3.0)*Math.PI*jariJari*jariJari*tinggiKerucut;
    }
    public double hasilLp(){
        return luasPermukaan;
    }
    public double hasilV(){
        return Volume;
    }
}
```

```
public class limasSegiEmpatSamaSisi {
    private double sisiAlas;
    private double tinggiLimas;
    private double luasPermukaan;
    private double volume;

    public limasSegiEmpatSamaSisi(double sisiAlas, double
tinggiLimas){
        this.sisiAlas=sisiAlas;
        this.tinggiLimas=tinggiLimas;
        hitungLuasPermukaan();
        hitungVolume();
    }
    private void hitungLuasPermukaan(){
        luasPermukaan =
2*sisiAlas*sisiAlas+4*(sisiAlas*tinggiLimas/2);
    }
    private void hitungVolume(){
        volume= (1.0/3.0)*sisiAlas*sisiAlas*tinggiLimas;
    }
    public double hasilLp(){
        return luasPermukaan;
    }
    public double hasilV(){
        return volume;
    }
}
```

```
public class bola {  
    private double jariJari;  
    private double luasPermukaan;  
    private double volume;  
  
    public bola(double jariJari){  
        this.jariJari=jariJari;  
        hitungLuasPermukaan();  
        hitungVolume();  
    }  
    private void hitungLuasPermukaan(){  
        luasPermukaan= 4*Math.PI*jariJari*jariJari;  
    }  
    private void hitungVolume(){  
        volume= (4.0/3.0)*Math.PI*jariJari*jariJari*jariJari;  
    }  
    public double hasilLp(){  
        return luasPermukaan;  
    }  
    public double hasilV(){  
        return volume;  
    }  
}
```



```
import java.util.Scanner;

public class mainbaru {

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

        // Input jumlah perhitungan untuk setiap bangun ruang
        System.out.print("Berapa Kali Menghitung Kerucut: ");
        int jumlahKerucut = scanner.nextInt();

        System.out.print("Berapa Kali Menghitung Limas: ");
        int jumlahLimas = scanner.nextInt();

        System.out.print("Berapa Kali Menghitung Bola: ");
        int jumlahBola = scanner.nextInt();

        // Deklarasi array of objects untuk setiap bangun ruang
        kerucut[] kerucut = new kerucut[jumlahKerucut];
        limasSegiEmpatSamaSisi[] limas = new
limasSegiEmpatSamaSisi[jumlahLimas];
        bola[] bolla = new bola[jumlahBola];

        // Input data dan hitung untuk setiap kerucut
        for (int i = 0; i < kerucut.length; i++) {
            System.out.println("\n-KERUCUT-" + (i + 1));
            System.out.print("Masukkan Jari-Jari: ");
            double jariJariKerucut = scanner.nextDouble();

            System.out.print("Masukkan Sisi Miring: ");
            double sisiMiringKerucut = scanner.nextDouble();

            kerucut[i] = new kerucut(jariJariKerucut,
sisiMiringKerucut, i);
        }

        // Input data dan hitung untuk setiap limas
```

```

        for (int i = 0; i < limas.length; i++) {
            System.out.println("\n-LIMAS-" + (i + 1));
            System.out.print("Masukkan Sisi Alas: ");
            double sisiAlasLimas = scanner.nextDouble();

            System.out.print("Masukkan Tinggi Limas: ");
            double tinggiLimas = scanner.nextDouble();

            limas[i] = new limasSegiEmpatSamaSisi(sisiAlasLimas,
tinggiLimas);
        }
        for (int i = 0; i < bolla.length; i++) {
            System.out.println("\n-BOLA-" + (i + 1));
            System.out.print("Masukkan Jari-jari: ");
            double jariJari = scanner.nextDouble();

            bolla[i] = new bola(jariJari);
        }
        // tampil
        kerucut kerucut2 = new kerucut(jumlahKerucut, jumlahLimas,
jumlahBola);
        System.out.println("KERUCUT");
        System.out.println("Luas Permukaan: " + kerucut2.hasilLp());
        System.out.println("Volume: " + kerucut2.hasilV());

        limasSegiEmpatSamaSisi limas2 = new
limasSegiEmpatSamaSisi(jumlahLimas, jumlahBola);
        System.out.println("LIMAS");
        System.out.println("Luas Permukaan: " + limas2.hasilLp() );
        System.out.println("Volume: " + limas2.hasilV() );

        bola bol = new bola(jumlahBola);
        System.out.println("BOLA");
        System.out.println("Luas Permukaan: " + bol.hasilLp() );
        System.out.println("Volume: " + bol.hasilV() );
        scanner.close();
    }

```

```

PS D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet3> & '
ceptionMessages' '-cp' 'C:\Users\lenovo\AppData\Roaming\Code\User\wor
5\bin' 'mainbaru'
Berapa Kali Menghitung Kerucut: 1
Berapa Kali Menghitung Limas: 1
Berapa Kali Menghitung Bola: 1

-KERUCUT-1
Masukkan Jari-Jari: 6
Masukkan Sisi Miring: 7

-LIMAS-1
Masukkan Sisi Alas: 9
Masukkan Tinggi Limas: 3

-BOLA-1
Masukkan Jari-jari: 6
KERUCUT
Luas Permukaan: 6.283185307179586
Volume: 1.480960979386122
LIMAS
Luas Permukaan: 4.0
Volume: 0.3333333333333333
BOLA
Luas Permukaan: 12.566370614359172
Volume: 4.1887902047863905
PS D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet3>

```

2.

```

package praktiklatihan2;

public class mahasiswa {
    private String nama;
    private String jenisKelamin;
    private int nim ;
    private double ipk;

    public mahasiswa(String nama,String jenisKelamin,int nim,double
ipk){
        this.nama=nama;
        this.jenisKelamin=jenisKelamin;
        this.nim=nim;
        this.ipk=ipk;
    }

    public String hasilNama(){
        return nama;
    }
    public String hasilJenisKelamin(){
        return jenisKelamin;
    }
}

```

```

    public int hasilNim() {
        return nim;
    }

    public double hasilIpk() {
        return ipk;
    }
}

```

```

Masukkan Nama Mahasiswa ke-0 :pino
Masukkan Jenis Kelamin (L/P):L
Masukkan nim: 1234
Masukkan Ipik: 34,3

Masukkan Nama Mahasiswa ke-1 :nabel
Masukkan Jenis Kelamin (L/P):L
Masukkan nim: 345
Masukkan Ipik: 34,7

Masukkan Nama Mahasiswa ke-2 :fahmi
Masukkan Jenis Kelamin (L/P):P
Masukkan nim: 2345
Masukkan Ipik: 89,0

```

```

Nama :pino
Jenis Kelamin :L
Nim:1234
Ipik:34.3
Nama :nabel
Jenis Kelamin :L
Nim:345
Ipik:34.7
Nama :fahmi
Jenis Kelamin :P
Nim:2345
Ipik:89.0

```

```

PS D:\alvino\Semester 2\Prak algoritma & struktur data\jobsheet3>

```

3.

```

public static double hitungRataRataIpik(mahasiswa[] mhs) {
    double totalIpik = 0;
    for (mahasiswa m : mhs) {
        totalIpik += m.hasilIpik();
    }
    return totalIpik / mhs.length;
}

public static mahasiswa findMahasiswaBerprestasi(mahasiswa[]
mhs) {
    mahasiswa mahasiswaBerprestasi = mhs[0];
    for (int i = 1; i < mhs.length; i++) {
        if (mhs[i].hasilIpik() > mahasiswaBerprestasi.hasilIpik())
        {
            mahasiswaBerprestasi = mhs[i];
        }
    }
}

```

```
        }  
    }  
    return mahasiswaBerprestasi;  
}
```

Main:

```
double rataRataIpk = mahasiswa.hitungRataRataIpk(mhs);  
    mahasiswa mahasiswaBerprestasi =  
mahasiswa.findMahasiswaBerprestasi(mhs);  
  
    System.out.println("Rata-rata IPK: " + rataRataIpk);  
    System.out.println();  
    System.out.println("Mahasiswa dengan IPK Tertinggi:");  
    System.out.println("Nama          : " +  
mahasiswaBerprestasi.hasilNama());  
    System.out.println("Jenis Kelamin : " +  
mahasiswaBerprestasi.hasilJenisKelamin());  
    System.out.println("Nim          : " +  
mahasiswaBerprestasi.hasilNim());  
    System.out.println("Ipk          : " +  
mahasiswaBerprestasi.hasilIpk());
```

Masukkan Nama Mahasiswa ke-0 :pino
Masukkan Jenis Kelamin (L/P):L
Masukkan nim: 123
Masukkan Ipk: 3,4

Masukkan Nama Mahasiswa ke-1 :vivi
Masukkan Jenis Kelamin (L/P):P
Masukkan nim: 1234
Masukkan Ipk: 3,6

Masukkan Nama Mahasiswa ke-2 :kiki
Masukkan Jenis Kelamin (L/P):L
Masukkan nim: 1234
Masukkan Ipk: 3,8

Nama :pino
Jenis Kelamin :L
Nim:123
Ipk:3.4

Nama :vivi
Jenis Kelamin :P
Nim:1234
Ipk:3.6

Nama :kiki
Jenis Kelamin :L
Nim:1234
Ipk:3.8

Rata-rata IPK: 3.6

Mahasiswa dengan IPK Tertinggi:

Nama : kiki
Jenis Kelamin : L
Nim : 1234
Ipk : 3.8