



Nama : Deanissa Sherly Sabilla

Kelas : SIB 1B

Absen : 06

JOBSHEET III ARRAY OF OBJECTS

3.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Memahami dan menjelaskan fungsi array yang berisikan variabel object.
2. Mahasiswa mampu menerapkan instansiasi array of objects dalam Java
3. Mahasiswa mampu melakukan operasi terhadap elemen dalam array of objects

3.2 Membuat Array dari Object, Mengisi dan Menampilkan

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat array of objects, kemudian mengisi dan menampilkan array tersebut.

3.2.1 Langkah-langkah Percobaan

1. Buat folder baru dengan nama Praktikum03.
2. Buat class `PersegiPanjang`:

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

3. Buat class `PersegiPanjangDemo` kemudian tambahkan fungsi main sebagai berikut

```
public static void main(String[] args) {
    PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
}
```

4. Kemudian isikan masing-masing atributnya:

```
public static void main(String[] args) {
    PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];

    arrayOfPersegiPanjang[0] = new PersegiPanjang();
    arrayOfPersegiPanjang[0].panjang = 110;
    arrayOfPersegiPanjang[0].lebar = 30;

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

    arrayOfPersegiPanjang[2] = new PersegiPanjang();
    arrayOfPersegiPanjang[2].panjang = 100;
    arrayOfPersegiPanjang[2].lebar = 20;
}
```



5. Cetak ke layar semua atribut dari objek `ppArray`:

```
System.out.println("Persegi panjang ke-1, panjang: " + arrayOfPersegiPanjang[0].panjang +
    ", lebar: " + arrayOfPersegiPanjang[0].lebar);
System.out.println("Persegi panjang ke-2, panjang: " + arrayOfPersegiPanjang[1].panjang +
    ", lebar: " + arrayOfPersegiPanjang[1].lebar);
System.out.println("Persegi panjang ke-3, panjang: " + arrayOfPersegiPanjang[2].panjang +
    ", lebar: " + arrayOfPersegiPanjang[2].lebar);
```

6. Run program dan amati hasilnya.

3.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Persegi panjang ke-1, panjang: 110, lebar: 30
Persegi panjang ke-2, panjang: 80, lebar: 40
Persegi panjang ke-3, panjang: 100, lebar: 20
```

➤ HASIL INPUT :

Class PersegiPanjang :

```
1 public class PersegiPanjang {
2     public int panjang;
3     public int lebar;
4 }
```

Class PersegiPanjangDemo :

```
1 public class PersegiPanjangDemo {
2     public static void main(String[] args) {
3         PersegiPanjang [] arrayOfPersegiPanjang = new PersegiPanjang[3];
4
5         arrayOfPersegiPanjang[0] = new PersegiPanjang();
6         arrayOfPersegiPanjang[0].panjang = 110;
7         arrayOfPersegiPanjang[0].lebar = 30;
8
9         arrayOfPersegiPanjang[1] = new PersegiPanjang();
10        arrayOfPersegiPanjang[1].panjang = 80;
11        arrayOfPersegiPanjang[1].lebar = 40;
12
13        arrayOfPersegiPanjang[2] = new PersegiPanjang();
14        arrayOfPersegiPanjang[2].panjang = 100;
15        arrayOfPersegiPanjang[2].lebar = 20;
16
17        System.out.println("Persegi Panjang ke-1, panjang : " + arrayOfPersegiPanjang[0].panjang +
18            ", lebar : " + arrayOfPersegiPanjang[0].lebar);
19        System.out.println("Persegi Panjang ke-2, panjang : " + arrayOfPersegiPanjang[1].panjang +
20            ", lebar : " + arrayOfPersegiPanjang[1].lebar);
21        System.out.println("Persegi Panjang ke-3, panjang : " + arrayOfPersegiPanjang[2].panjang +
22            ", lebar : " + arrayOfPersegiPanjang[2].lebar);
23    }
24 }
25 }
```

➤ HASIL OUTPUT :

```
heet3 389f6b7b\bin' 'PersegiPanjangDemo'
Persegi Panjang ke-1, panjang : 110, lebar : 30
Persegi Panjang ke-2, panjang : 80, lebar : 40
Persegi Panjang ke-3, panjang : 100, lebar : 20
PS C:\Users\TOSHIBA\jobsheet3>
```



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!

- Tidak, pada class yang akan dibuat sebagai array of object tidak harus memiliki atribut dan method. Dalam pemrograman Java, array dapat menyimpan referensi objek dari class, termasuk class yang tidak memiliki atribut atau method.

2. Apa yang dilakukan oleh kode program berikut?

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

- Kode tersebut merupakan sebuah deklarasi dan instansiasi array bernama arrayOfPersegiPanjang yang dapat menampung tiga objek dari class PersegiPanjang.

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

- Tidak, jika tidak ada konstruktor yang didefinisikan secara eksplisit dalam sebuah kelas, Java akan memberikan konstruktor default secara implisit kepada kelas tersebut.
- Pemanggilan konstruktor pada baris program arrayOfPersegiPanjang[0] = new PersegiPanjang(); menggunakan konstruktor default yang diberikan oleh Java jika tidak ada konstruktor lain yang didefinisikan dalam kelas PersegiPanjang.

```
arrayOfPersegiPanjang[0] = new PersegiPanjang();
```

4. Apa yang dilakukan oleh kode program berikut?

```
arrayOfPersegiPanjang[0].panjang = 110;  
arrayOfPersegiPanjang[0].lebar = 30;
```

- Kode program tersebut mengatur nilai panjang dan lebar dari objek yang terdapat pada indeks 0 dari arrayOfPersegiPanjang.

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

- Menerapkan sistem OOP, Pemisahan kelas PersegiPanjang dan PersegiPanjangDemo dalam dua file yang terpisah karena pemisahan kepentingan (separation of concerns).
 1. Kelas PersegiPanjang: Ini adalah kelas yang mewakili entitas persegi panjang. Fokus utamanya adalah mendefinisikan itu sendiri, seperti panjang dan lebar.
 2. Kelas PersegiPanjangDemo: Ini adalah kelas yang bertanggung jawab untuk penggunaan kelas PersegiPanjang. Fokusnya adalah menginstansiasi objek PersegiPanjang, mengatur nilai-nilai, dan menunjukkan bagaimana menggunakan kelas tersebut.

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

3.3.1 Langkah-langkah Percobaan

1. Import scanner pada class `PersegiPanjangDemo`.

```
import java.util.Scanner;
```

Note: Letakkan kode import dibawah kode package (jika ada).

2. Pada praktikum 3.2 poin nomor 4, modifikasi kode program sebagai berikut. Buat objek `Scanner` untuk menerima input, kemudian lakukan looping untuk menerima informasi panjang dan lebar:

```
Scanner sc = new Scanner(System.in);
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];

for (int i = 0; i < 3; i++) {
    arrayOfPersegiPanjang[i] = new PersegiPanjang();

    System.out.println("Persegi panjang ke-" + (i + 1));
    System.out.print("Masukkan panjang: ");
    arrayOfPersegiPanjang[i].panjang = sc.nextInt();
    System.out.print("Masukkan lebar: ");
    arrayOfPersegiPanjang[i].lebar = sc.nextInt();
}
```

3. Pada praktikum 3.2 poin nomor 5, modifikasi kode program sebagai berikut. Lakukan looping untuk mengakses elemen array dan menampilkan informasinya ke layar:

```
for (int i = 0; i < 3; i++) {
    System.out.println("Persegi panjang ke-" + (i + 1) + ", panjang: "
        + arrayOfPersegiPanjang[i].panjang + ", lebar: " + arrayOfPersegiPanjang[i].lebar);
}
```

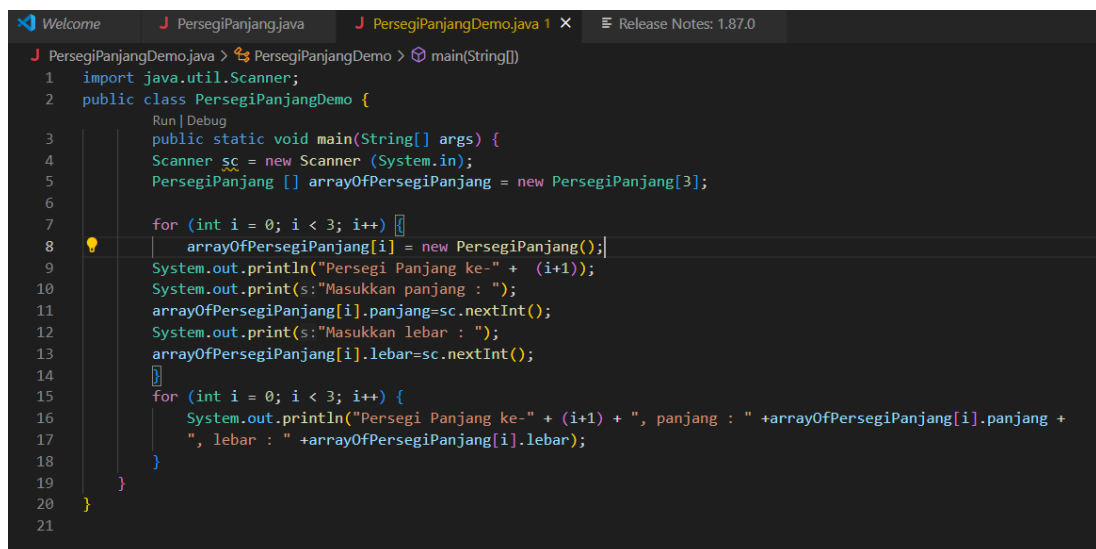
4. Run program dan amati hasilnya.

3.3.2 Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.

```
Persegi panjang ke-1
Masukkan panjang: 2
Masukkan lebar: 3
Persegi panjang ke-2
Masukkan panjang: 4
Masukkan lebar: 5
Persegi panjang ke-3
Masukkan panjang: 2
Masukkan lebar: 4
Persegi panjang ke-1, panjang: 2, lebar: 3
Persegi panjang ke-2, panjang: 4, lebar: 5
Persegi panjang ke-3, panjang: 2, lebar: 4
```

➤ HASIL INPUT :



```
1 import java.util.Scanner;
2 public class PersegiPanjangDemo {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         PersegiPanjang [] arrayOfPersegiPanjang = new PersegiPanjang[3];
6
7         for (int i = 0; i < 3; i++) {
8             arrayOfPersegiPanjang[i] = new PersegiPanjang();
9             System.out.println("Persegi Panjang ke-" + (i+1));
10            System.out.print(s:"Masukkan panjang : ");
11            arrayOfPersegiPanjang[i].panjang=sc.nextInt();
12            System.out.print(s:"Masukkan lebar : ");
13            arrayOfPersegiPanjang[i].lebar=sc.nextInt();
14        }
15        for (int i = 0; i < 3; i++) {
16            System.out.println("Persegi Panjang ke-" + (i+1) + ", panjang : " +arrayOfPersegiPanjang[i].panjang +
17            ", lebar : " +arrayOfPersegiPanjang[i].lebar);
18        }
19    }
20 }
21
```

➤ HASIL OUTPUT :

```
Persegi Panjang ke-1
Masukkan panjang : 20
Masukkan lebar : 2
Persegi Panjang ke-2
Masukkan panjang : 22
Masukkan lebar : 10
Persegi Panjang ke-3
Masukkan panjang : 23
Masukkan lebar : 12
Persegi Panjang ke-1, panjang : 20, lebar : 2
Persegi Panjang ke-2, panjang : 22, lebar : 10
Persegi Panjang ke-3, panjang : 23, lebar : 12
```

3.3.3 Pertanyaan

1. Tambahkan method cetakInfo() pada class PersegiPanjang kemudian modifikasi kode program pada langkah no 3.

➤ HASIL INPUT :

Class PersegiPanjang

```

J PersegiPanjang.java > PersegiPanjang > cetakInfo()
1 public class PersegiPanjang {
2     public int panjang;
3     public int lebar;
4
5     public void cetakInfo() {
6         System.out.println("panjang: " + panjang + ", lebar: " + lebar);
7     }
8 }
    
```

Class PersegiPanjangDemo

```

J PersegiPanjangDemo.java > PersegiPanjangDemo > main(String[])
1 import java.util.Scanner;
2 public class PersegiPanjangDemo {
3     Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         PersegiPanjang [] arrayOfPersegiPanjang = new PersegiPanjang[3];
7
8         for (int i = 0; i < 3; i++) {
9             arrayOfPersegiPanjang[i] = new PersegiPanjang();
10            System.out.println("Persegi Panjang ke-" + (i+1));
11            System.out.print(s:"Masukkan panjang : ");
12            arrayOfPersegiPanjang[i].panjang=sc.nextInt();
13            System.out.print(s:"Masukkan lebar : ");
14            arrayOfPersegiPanjang[i].lebar=sc.nextInt();
15        }
16        for (int i = 0; i < 3; i++) {
17            System.out.println("Persegi Panjang ke-" + (i+1));
18            arrayOfPersegiPanjang[i].cetakInfo();
19        }
20    }
21 }
    
```

HASIL OUTPUT :

```

va\jdt_ws\jobsheet3_389f6b7b\b
Persegi Panjang ke-1
Masukkan panjang : 22
Masukkan lebar : 12
Persegi Panjang ke-2
Masukkan panjang : 25
Masukkan lebar : 11
Persegi Panjang ke-3
Masukkan panjang : 14
Masukkan lebar : 4
Persegi Panjang ke-1
panjang: 22, lebar: 12
Persegi Panjang ke-2
panjang: 25, lebar: 11
panjang: 14, lebar: 4
PS C:\Users\TOSHIBA\jobsheet3>
    
```



2. Misalkan Anda punya **array baru** bertipe array of PersegiPanjang dengan nama **myArrayOfPersegiPanjang**. Mengapa kode berikut menyebabkan error?

```
PersegiPanjang[] myArrayOfPersegiPanjang = new PersegiPanjang[100];
myArrayOfPersegiPanjang[5].panjang = 20;
```

- Kode tersebut error karena pada saat pembuatan array baru myArrayOfPersegiPanjang, hanya array itu sendiri yang dibuat. Artinya, terdapat 100 slot kosong yang bertipe PersegiPanjang, namun objek-objek PersegiPanjang belum dibuat dan dialokasikan pada setiap slot array.
- Ketika mencoba mengakses elemen ke-5 dari array myArrayOfPersegiPanjang (myArrayOfPersegiPanjang[5]) dan kemudian mencoba untuk mengatur nilai panjang (myArrayOfPersegiPanjang[5].panjang), itu menyebabkan NullPointerException. Ini karena elemen pada indeks ke-5 masih null, yang berarti tidak ada objek PersegiPanjang yang diinisialisasi di dalamnya.

```
Exception in thread "main" java.lang.NullPointerException: Cannot assign field "panjang" because "myArrayOfPersegiPanjang[5]" is null
    at PersegiPanjang.cetakInfo(PersegiPanjang.java:9)
    at PersegiPanjangDemo.main(PersegiPanjangDemo.java:17)
PS C:\Users\TOSHIBA\jobsheet3>
```

3.4 Constructor Berparameter

Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

3.4.1 Langkah-langkah Percobaan

1. Pada class PersegiPanjang, tambahkan constructor berparameter

```
public PersegiPanjang(int p, int l) {
    panjang = p;
    lebar = l;
}
```

2. Run program kemudian amati hasilnya.
3. Modifikasi class PersegiPanjangDemo sehingga instansiasi dilakukan menggunakan constructor berparameter

```
Scanner sc = new Scanner(System.in);
PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
int panjang, lebar;

for (int i = 0; i < 3; i++) {
    System.out.println("Persegi panjang ke-" + (i + 1));
    System.out.print("Masukkan panjang: ");
    panjang = sc.nextInt();
    System.out.print("Masukkan lebar: ");
    lebar = sc.nextInt();

    arrayOfPersegiPanjang[i] = new PersegiPanjang(panjang, lebar);
}
```

4. Run program dan amati hasilnya.

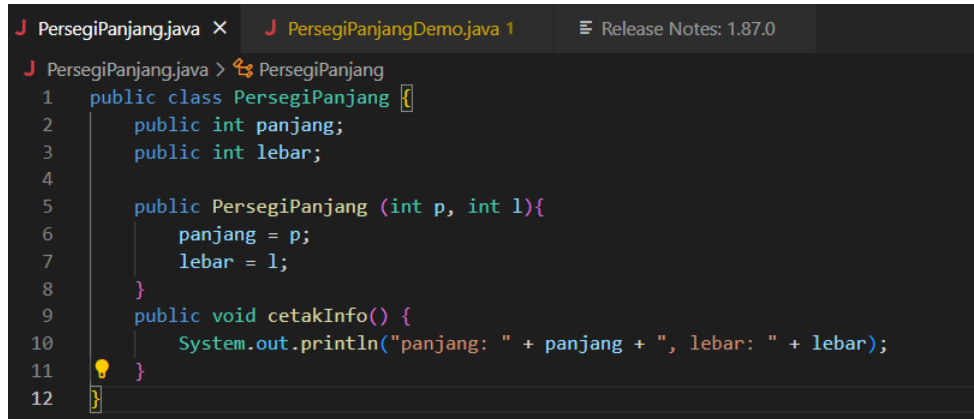
3.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Persegi panjang ke-1
Masukkan panjang: 2
Masukkan lebar: 3
Persegi panjang ke-2
Masukkan panjang: 4
Masukkan lebar: 5
Persegi panjang ke-3
Masukkan panjang: 2
Masukkan lebar: 4
Persegi panjang ke-1, panjang: 2, lebar: 3
Persegi panjang ke-2, panjang: 4, lebar: 5
Persegi panjang ke-3, panjang: 2, lebar: 4
```

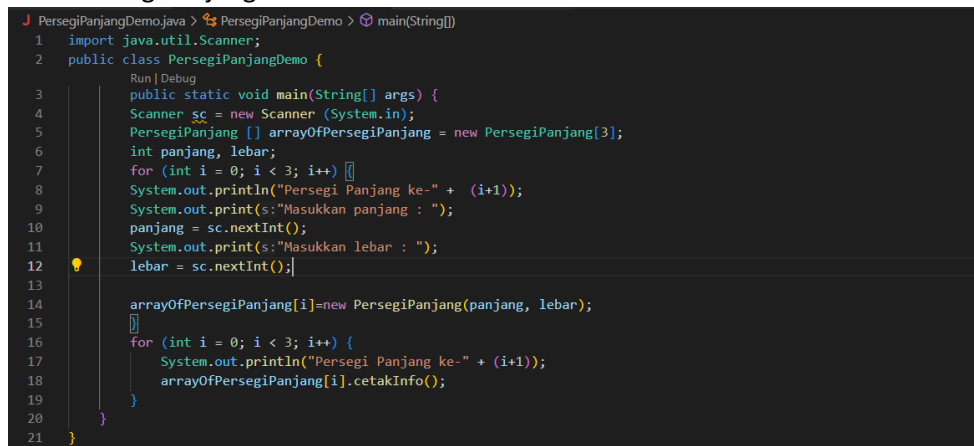
➤ HASIL INPUT :

Class PersegiPanjang



```
1 public class PersegiPanjang {
2     public int panjang;
3     public int lebar;
4
5     public PersegiPanjang (int p, int l){
6         panjang = p;
7         lebar = l;
8     }
9     public void cetakInfo() {
10        System.out.println("panjang: " + panjang + ", lebar: " + lebar);
11    }
12 }
```

Class PersegiPanjangDemo



```
1 import java.util.Scanner;
2 public class PersegiPanjangDemo {
3     Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner (System.in);
6         PersegiPanjang [] arrayOfPersegiPanjang = new PersegiPanjang[3];
7         int panjang, lebar;
8         for (int i = 0; i < 3; i++) {
9             System.out.println("Persegi Panjang ke-" + (i+1));
10            System.out.print(s:"Masukkan panjang : ");
11            panjang = sc.nextInt();
12            System.out.print(s:"Masukkan lebar : ");
13            lebar = sc.nextInt();
14
15            arrayOfPersegiPanjang[i]=new PersegiPanjang(panjang, lebar);
16        }
17        for (int i = 0; i < 3; i++) {
18            System.out.println("Persegi Panjang ke-" + (i+1));
19            arrayOfPersegiPanjang[i].cetakInfo();
20        }
21    }
22 }
```




➤ **HASIL OUTPUT :**

```
6b7b\bin\PersegiPanjangDemo
Persegi Panjang ke-1
Masukkan panjang : 22
Masukkan lebar : 12
Persegi Panjang ke-2
Masukkan panjang : 20
Masukkan lebar : 9
Persegi Panjang ke-3
Masukkan panjang : 11
Masukkan lebar : 2
Persegi Panjang ke-1
panjang: 22, lebar: 12
Persegi Panjang ke-2
panjang: 20, lebar: 9
Persegi Panjang ke-3
panjang: 11, lebar: 2
PS C:\Users\TOSHIBA\jobsheet3>
fileToExceptionMessage: 'en'
```

3.4.3 Pertanyaan

1. Mengapa terjadi compile error pada langkah no 2?
 - Karena menambahkan konstruktor public PersegiPanjang(int p, int l) ke dalam kelas PersegiPanjang tanpa menambahkan penggunaan konstruktor tersebut di dalam kelas main. Namun, masalahnya adalah konstruktor default (PersegiPanjang()) yang disediakan oleh Java tidak lagi tersedia, karena kelas tersebut sekarang memiliki konstruktor khusus (public PersegiPanjang(int p, int l)).
2. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya
 - Ya, sebuah class dapat memiliki lebih dari 1 constructor dalam bahasa pemrograman Java. Konstruktor dalam Java dapat di-overload, yang berarti dapat mendefinisikan beberapa konstruktor dengan parameter yang berbeda atau jumlah parameter yang berbeda.

Contoh :

// Constructor pertama tanpa parameter

```
public Orang() {
    nama = "bebi";
    umur = 19;
}
```

// Constructor kedua dengan dua parameter

```
public Orang(String nama, int umur) {
    this.nama = nama;
    this.umur = umur;
}
```

3. Tambahkan method `hitungLuas()` dan `hitungKeliling()`

➤ **HASIL INPUT :**

```
PersegiPanjang.java > PersegiPanjang > hitungKeliling()
1  public class PersegiPanjang {
2      public int panjang;
3      public int lebar;
4
5      public PersegiPanjang (int p, int l){
6          panjang = p;
7          lebar = l;
8      }
9      public int hitungLuas() {
10         return panjang * lebar;
11     }
12     public int hitungKeliling() {
13         return 2 * (panjang + lebar);
14     }
15     public void cetakInfo() {
16         System.out.println("panjang: " + panjang + ", lebar: " + lebar);
17     }
18 }
```

4. Lakukan modifikasi kode program untuk menampilkan luas dan keliling dengan contoh output sebagai berikut:

```
Persegi panjang ke-1
Masukkan panjang: 2
Masukkan lebar: 4
Persegi panjang ke-2
Masukkan panjang: 10
Masukkan lebar: 20
Persegi panjang ke-3
Masukkan panjang: 5
Masukkan lebar: 7
Persegi panjang ke-1, panjang: 2, lebar: 4, luas: 8, keliling: 12
Persegi panjang ke-2, panjang: 10, lebar: 20, luas: 200, keliling: 60
Persegi panjang ke-3, panjang: 5, lebar: 7, luas: 35, keliling: 24
```

➤ **HASIL OUTPUT :**

```
va\jdt_ws\jobsheet3_389f6b7b\bin' 'f
Persegi Panjang ke-1
Masukkan panjang : 2
Masukkan lebar : 4
Persegi Panjang ke-2
Masukkan panjang : 10
Masukkan lebar : 20
Persegi Panjang ke-3
Masukkan panjang : 5
Masukkan lebar : 7
Persegi Panjang ke-1
panjang: 2, lebar: 4
Luas: 8
Keliling: 12
Persegi Panjang ke-2
panjang: 10, lebar: 20
Luas: 200
Keliling: 60
Persegi Panjang ke-3
panjang: 5, lebar: 7
Luas: 35
Keliling: 24
PS C:\Users\TOSHIBA\jobsheet3>
```

5. Modifikasi kode program pada praktikum 3.3 agar length array ditentukan dari user melalui input dengan Scanner

➤ **HASIL INPUT :**

```
J PersegiPanjangDemo.java > PersegiPanjangDemo > main(String[])
1 import java.util.Scanner;
2 public class PersegiPanjangDemo {
    Run | Debug
3     public static void main(String[] args) {
4         Scanner sc = new Scanner (System.in);
5         int panjang, lebar;
6         System.out.print(s:"Masukkan jumlah persegi panjang : ");
7         int jumlah = sc.nextInt();
8         PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[jumlah];
9
10        for (int i = 0; i < jumlah; i++) {
11            System.out.println("Persegi Panjang ke-" + (i+1));
12            System.out.print(s:"Masukkan panjang : ");
13            panjang = sc.nextInt();
14            System.out.print(s:"Masukkan lebar : ");
15            lebar = sc.nextInt();
16            arrayOfPersegiPanjang[i]=new PersegiPanjang(panjang, lebar);
17        }
18
19        for (int i = 0; i < arrayOfPersegiPanjang.length; i++) {
20            System.out.println("Persegi Panjang ke-" + (i+1));
21            arrayOfPersegiPanjang[i].cetakInfo();
22            System.out.println("Luas: " + arrayOfPersegiPanjang[i].hitungLuas());
23            System.out.println("Keliling: " + arrayOfPersegiPanjang[i].hitungKeliling());
24        }
25    }
}
```

➤ **HASIL OUTPUT :**

```
rs\TOSHIBA\AppData\Roaming\Code\User
rsegiPanjangDemo'
Masukkan jumlah persegi panjang : 2
Persegi Panjang ke-1
Masukkan panjang : 40
Masukkan lebar : 10
Persegi Panjang ke-2
Masukkan panjang : 30
Masukkan lebar : 5
Persegi Panjang ke-1
panjang: 40, lebar: 10
Luas: 400
Keliling: 100
Persegi Panjang ke-2
panjang: 30, lebar: 5
Luas: 150
Keliling: 70
PS C:\Users\TOSHIBA\jobsheet3>
```