



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**:

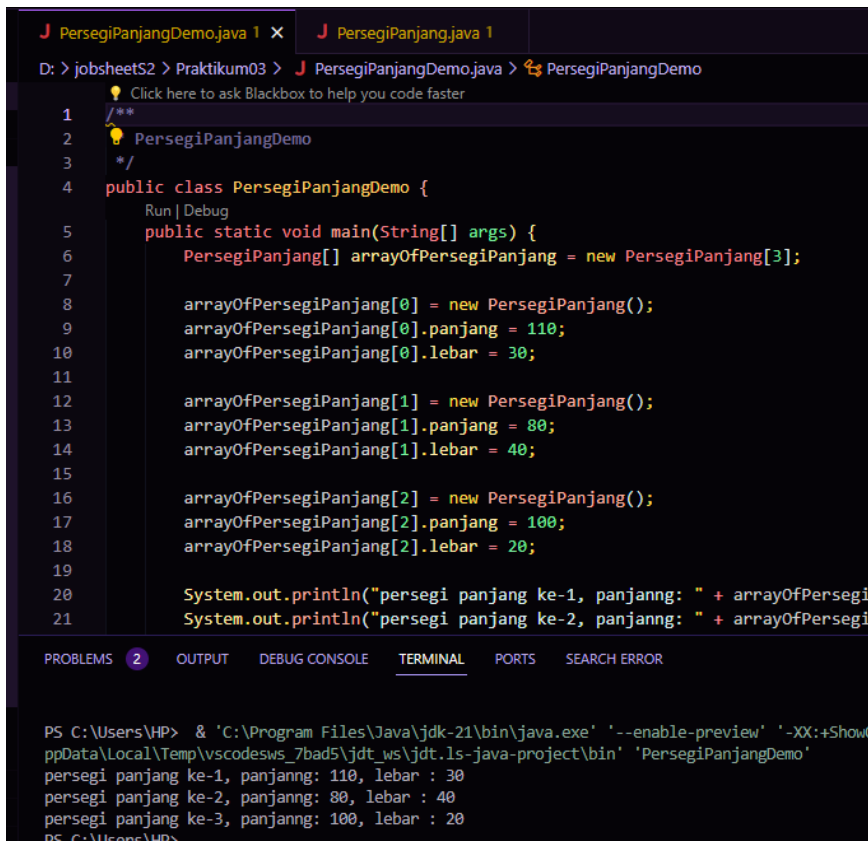
```
System.out.println("Persegi panjang ke-1, panjang: " + arrayOfPersegiPanjang[0].panjang +
    ", lebar: " + arrayOfPersegiPanjang[0].lebar);
System.out.println("Persegi panjang ke-2, panjang: " + arrayOfPersegiPanjang[1].panjang +
    ", lebar: " + arrayOfPersegiPanjang[1].lebar);
System.out.println("Persegi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



```

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

```

PS C:\Users\HP> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowC
ppData\Local\Temp\vscodesws_7bad5\jdt_ws\jdt.ls-java-project\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\HP>
    
```

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, sebuah class yang akan dijadikan array of object tidak wajib memiliki atribut atau method. Secara umum, array of object bisa dibuat dari class mana pun, bahkan class yang tidak memiliki atribut atau method sama sekali. Atribut Panjang dan lebar

2. Apa yang dilakukan oleh kode program berikut?

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

Kode tersebut membuat sebuah array dengan nama arrayOfPersegiPanjang yang dapat menampung objek dari tipe PersegiPanjang. Array tersebut memiliki panjang 3, artinya dapat menampung 3 objek `PersegiPanjang`.

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

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

Tidak, Pada baris program arrayOfPersegiPanjang[0] = new PersegiPanjang(); terdapat pemanggilan konstruktor default. Apa yang dilakukan oleh kode program berikut?



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

Kode program melakukan inisialisasi dan pengisian nilai pada array arrayOfPersegiPanjang.

Pada baris pertama, arrayOfPersegiPanjang[0] diinisialisasi sebagai objek dari class PersegiPanjang. Kemudian, pada baris-baris selanjutnya, nilai panjang dan lebar dari objek arrayOfPersegiPanjang[0] diatur dengan nilai masing-masing 110 dan 30.

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

Karena melakukan pemanggilan array, dan karena menerapkan oop

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
```

```
/**
 * PersegiPanjangDemo
 */
import java.util.Scanner;
public class PersegiPanjangDemo {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];

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

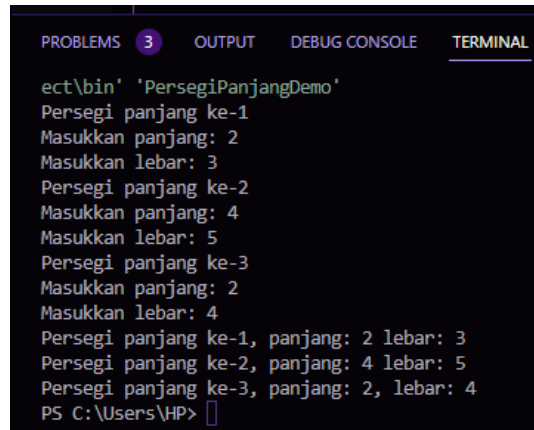
            System.out.println("Persegi panjang ke-" + (i + 1));
            System.out.print("Masukkan panjang: ");
            arrayOfPersegiPanjang[i].panjang = sc.nextInt();
            System.out.print("Masukkan lebar: ");
            arrayOfPersegiPanjang[i].lebar = sc.nextInt();
        }
        System.out.println("Persegi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

```

Outputnya :



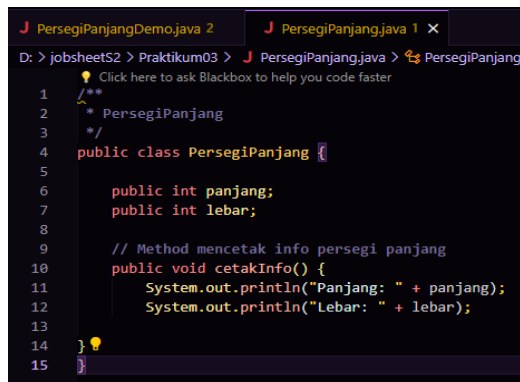
```

ect\bin' 'PersegiPanjangDemo'
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
PS C:\Users\HP>

```

3.3.3 Pertanyaan

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



```

1  /**
2   * PersegiPanjang
3   */
4  public class PersegiPanjang {
5
6      public int panjang;
7      public int lebar;
8
9      // Method mencetak info persegi panjang
10     public void cetakInfo() {
11         System.out.println("Panjang: " + panjang);
12         System.out.println("Lebar: " + lebar);
13     }
14 }
15

```

Menambahkan fungsi cetakinfo

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 menyebabkan error karena pada baris `myArrayOfPersegiPanjang[5].panjang = 20;`, elemen ke-5 dari array `myArrayOfPersegiPanjang` belum diinisialisasi sebagai objek dari class `PersegiPanjang`.

Yng hrus dilakukan biar gk eror hrus di instansiasi

3.4 Constructor Berparameter

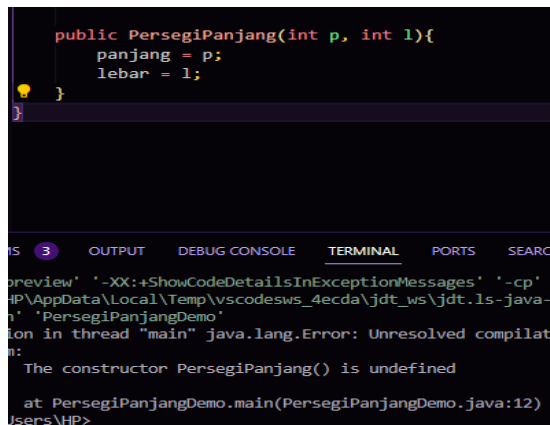
Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

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.



Hasilnya akan eror

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.

5. `import java.util.Scanner;`



```

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

```

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

```

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
ect\bin' 'PersegiPanjangDemo'
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
PS C:\Users\HP>

```

Outputnya :



3.4.3 Pertanyaan

1. Mengapa terjadi compile error pada langkah no 2?

Compile error terjadi karena class `PersegiPanjang` membutuhkan konstruktor dengan dua parameter bertipe `int` untuk panjang (`p`) dan lebar (`l`), tetapi tidak ada konstruktor default yang didefinisikan secara eksplisit dalam class tersebut.

2. Apakah suatu class dapat memiliki lebih dari 1 constructor? Jika iya, berikan contohnya

Ya, suatu kelas dapat memiliki lebih dari satu konstruktor

```
// Konstruktor kedua dengan dua parameter
public Person(String name, int age) {
    this.name = name;
    this.age = age;
}
```

```
// Metode getter untuk nama
public String getName() {
    return name;
}

/ Constructor Default
public Rectangle() {
    width = 0;
    height = 0;
}
```

3. Tambahkan method hitungLuas() dan hitungKeliling()

```

> jobsheetS2 > Praktikum03 > J PersegiPanjang.java > PersegiPanjang > hitungLuas()
Click here to ask Blackbox to help you code faster
1  /**
2   * PersegiPanjang
3   */
4  public class PersegiPanjang {
5
6      public int panjang;
7      public int lebar;
8
9      // Konstruktor dengan dua parameter
10     public PersegiPanjang(int panjang, int lebar) {
11         this.panjang = panjang;
12         this.lebar = lebar;
13     }
14     // Method untuk mencetak informasi persegi panjang
15     public void cetakInfo() {
16         System.out.println("Panjang: " + panjang);
17         System.out.println("Lebar: " + lebar);
18     }
19
20     // Method untuk menghitung luas persegi panjang
21     public int hitungLuas() {
22         return panjang * lebar;
23     }
24
25     // Method untuk menghitung keliling persegi panjang
26     public int hitungKeliling() {
27         return 2 * (panjang + lebar);
28     }
29 }

```

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

```

5. import java.util.Scanner;
6.
7. public class PersegiPanjangDemo12 {
8.     public static void main(String[] args) {
9.         Scanner input12 = new Scanner(System.in);
10.        PersegiPanjang[] arrayOfPersegiPanjang = new PersegiPanjang[3];
11.        int panjang, lebar;
12.
13.        for (int i = 0; i < 3; i++) {
14.            System.out.println("Persegi panjang ke-" + (i+1));
15.            System.out.print("Masukkan panjang: ");
16.            panjang = input12.nextInt();
17.            System.out.print("Masukkan lebar: ");
18.            lebar = input12.nextInt();
19.
20.            arrayOfPersegiPanjang[i] = new PersegiPanjang(panjang,
21.                lebar);
22.        }
23.    }
24. }

```



```

22.
23.     for (int i = 0; i < 3; i++) {
24.         System.out.print("Persegi panjang ke-" + (i+1) + ",
panjang: "
25.             + arrayOfPersegiPanjang[i].panjang + ", lebar: " +
arrayOfPersegiPanjang[i].lebar);
26.
27.         System.out.println(", Luas: " +
arrayOfPersegiPanjang[i].hitungLuas() +
28.             ", Keliling: " +
arrayOfPersegiPanjang[i].hitungKeliling());
29.     }
30. }
31.}
32.

```

```

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

```

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

```

34. import java.util.Scanner;
35.
36. public class PersegiPanjangDemo12 {
37.     public static void main(String[] args) {
38.         Scanner input12 = new Scanner(System.in);
39.
40.         // Menentukan panjang array dari user
41.         System.out.print("Masukkan jumlah persegi panjang: ");
42.         int jumlahPersegi = input12.nextInt();
43.         PersegiPanjang[] arrayOfPersegiPanjang = new
PersegiPanjang[jumlahPersegi];
44.
45.         for (int i = 0; i < jumlahPersegi; i++) {
46.             System.out.println("Persegi panjang ke-" + (i + 1));
47.             System.out.print("Masukkan panjang: ");
48.             int panjang = input12.nextInt();
49.             System.out.print("Masukkan lebar: ");
50.             int lebar = input12.nextInt();

```



```

51.
52.         arrayOfPersegiPanjang[i] = new PersegiPanjang(panjang,
lebar);
53.     }
54.
55.     for (int i = 0; i < jumlahPersegi; i++) {
56.         System.out.print("Persegi panjang ke-" + (i + 1) + ",
panjang: "
57.             + arrayOfPersegiPanjang[i].getPanjang() + ", lebar:
" + arrayOfPersegiPanjang[i].getLebar());
58.
59.         // Menampilkan method hitungLuas() dan hitungKeliling()
60.         System.out.println(", Luas: " +
arrayOfPersegiPanjang[i].hitungLuas() +
61.             ", Keliling: " +
arrayOfPersegiPanjang[i].hitungKeliling());
62.     }
63. }
64.}
65.

```

Outputnya :

```

Masukkan jumlah persegi panjang: 3
Persegi panjang ke-1
Masukkan panjang: 2
Masukkan lebar: 5
Persegi panjang ke-2
Masukkan panjang: 7
Masukkan lebar: 6
Persegi panjang ke-3
Masukkan panjang: 4
Masukkan lebar: 6
Persegi panjang ke-1, panjang: 2, lebar: 5Luas: 10, Keliling: 14
Persegi panjang ke-2, panjang: 7, lebar: 6Luas: 42, Keliling: 26
Persegi panjang ke-3, panjang: 4, lebar: 6Luas: 24, Keliling: 20

```