

## 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];  
}
```

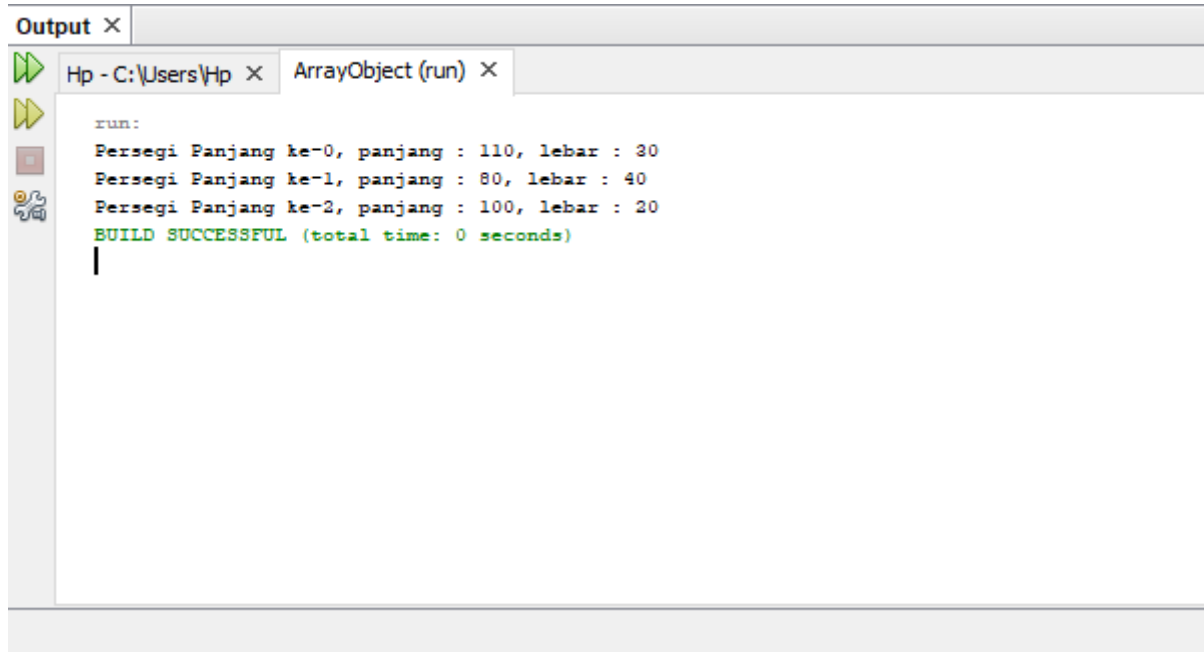
4. Kemudian isikan masing-masing atributnya:

```
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);
```

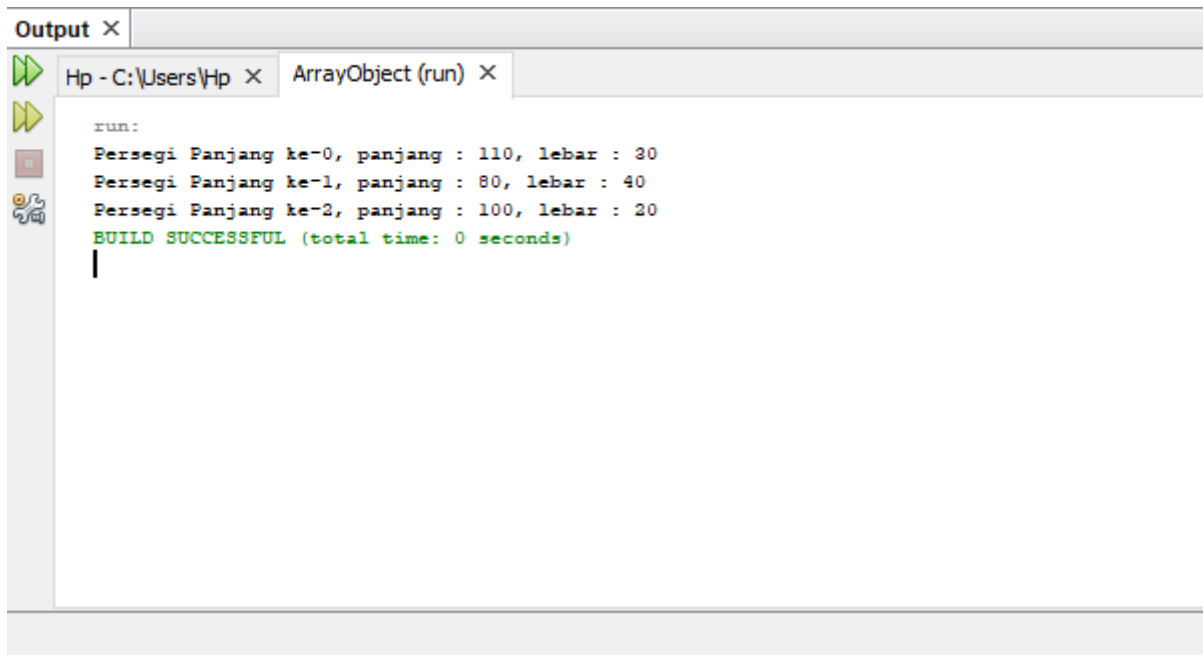
6. Jalankan dan amati hasilnya.



### 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

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method?Jelaskan!

Iya, setiap apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method. Dikarenakan atribut dan method selalu berkaitan. Dimana atribut dan method ini sendiri menjadi pemeran utama setelah menemukan object untuk menjalankan suatu class.

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

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

Tidak, karena konstruktor digunakan untuk membuat proses awal dalam mempersiapkan objek dan memberi nilai awal, dan juga untuk memanggil object.

3. Apa yang dimaksud dengan kode berikut ini:

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

Itu adalah kode untuk memanggil class persegiPanjang, dengan deklarasi dalam kurung yaitu angka 3, mengartikan data yang akan dijalankan atau di proses terdapat 3 atribut.

4. Apa yang dimaksud dengan kode berikut ini:

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

Itu adalah kode atribut guna memasukkan jumlah Panjang dan lebar yang nantinya akan dijalankan dalam method.

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

Karena Ketika class berada dalam satu project di kelas main akan terjadi eror dan tidak bisa dijalankan.

### 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.

```

package arrayobject;
import java.util.Scanner;

public class ArrayObject {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        persegiPanjang[] ppArray = new persegiPanjang[3];
        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();
        }
        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);
        }
    }
}

```

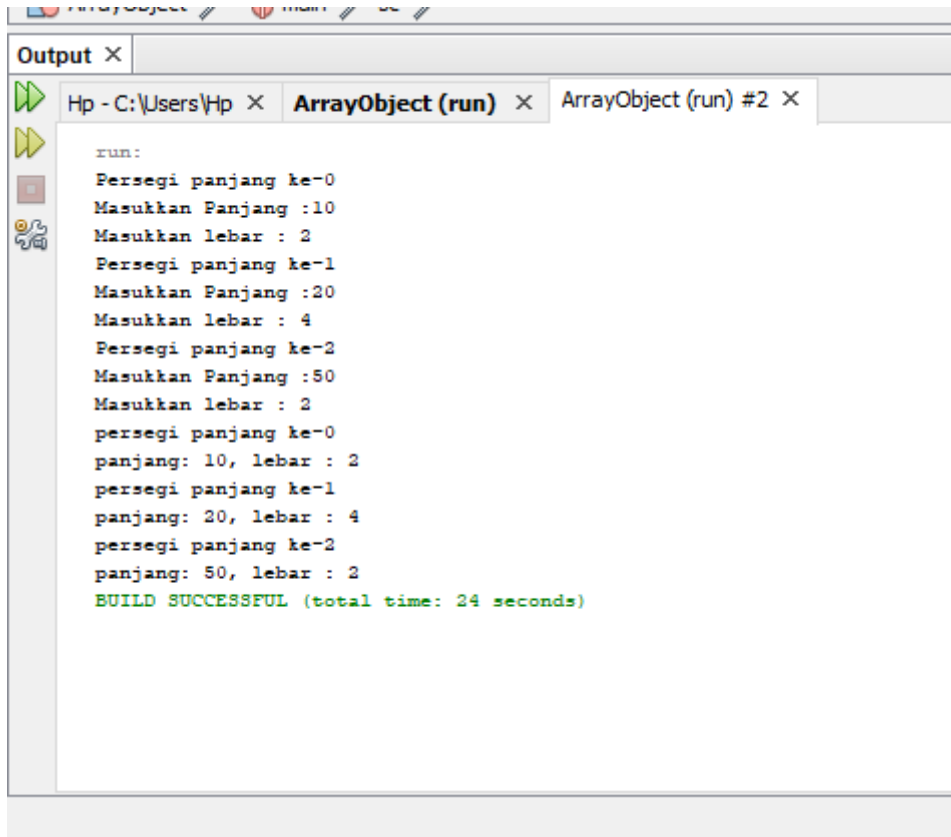
### 3.3.2 Verifikasi Hasil Percobaan

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

```



```
Output X
Hp - C:\Users\Hp x  ArrayObject (run) x  ArrayObject (run) #2 x

run:
Persegi panjang ke-0
Masukkan Panjang :10
Masukkan lebar : 2
Persegi panjang ke-1
Masukkan Panjang :20
Masukkan lebar : 4
Persegi panjang ke-2
Masukkan Panjang :50
Masukkan lebar : 2
persegi panjang ke-0
panjang: 10, lebar : 2
persegi panjang ke-1
panjang: 20, lebar : 4
persegi panjang ke-2
panjang: 50, lebar : 2
BUILD SUCCESSFUL (total time: 24 seconds)
```

### 3.3.3 Pertanyaan

1. Apakah array of object dapat diimplementasikan pada array 2 Dimensi?

Iya, 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?

```
Persegi[] pgArray = new Persegi[100];
pgArray[5].sisi = 20;
```

Karena jumlah yang akan dihasilkan dapat melampaui jumlah yang dapat dicangkep oleh tipe data integer itu sendiri.

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, ketika saya coba melakukan instansi tidak terjadi error dan program masih bisa dijalankan.

### 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:

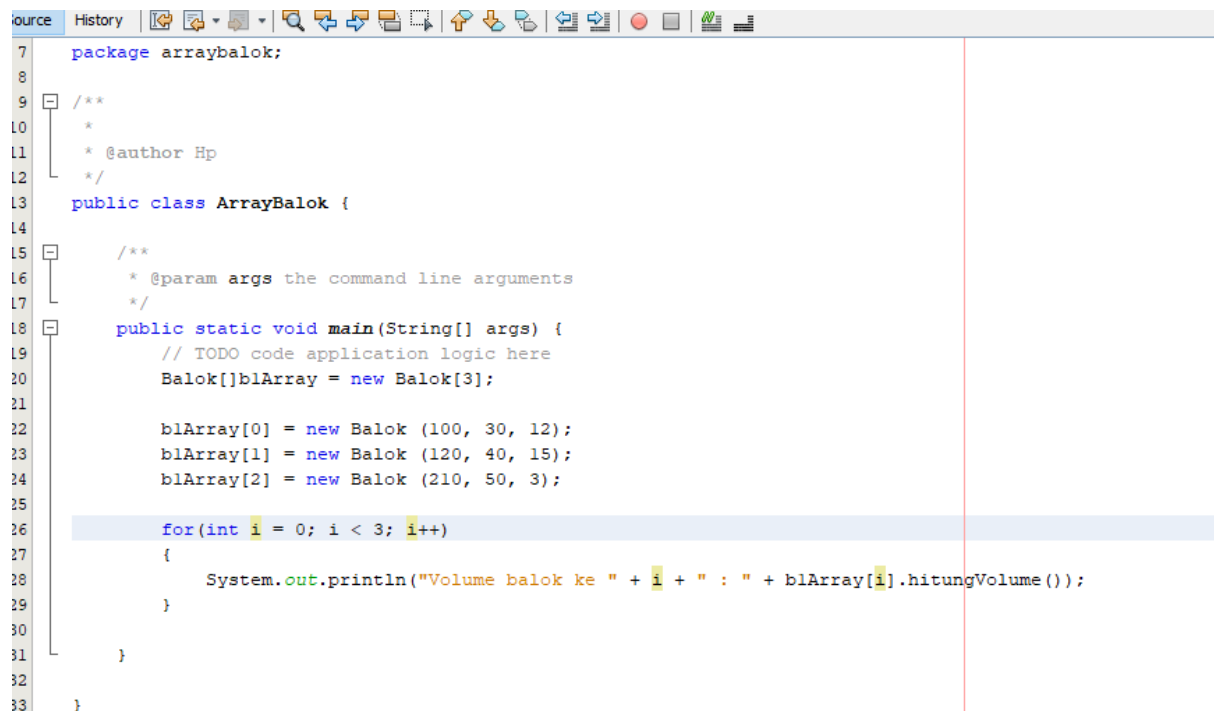
```
blArray[0] = new Balok(100, 30, 12);
blArray[1] = new Balok(120, 40, 15);
blArray[2] = new Balok(210, 50, 25);
```

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.

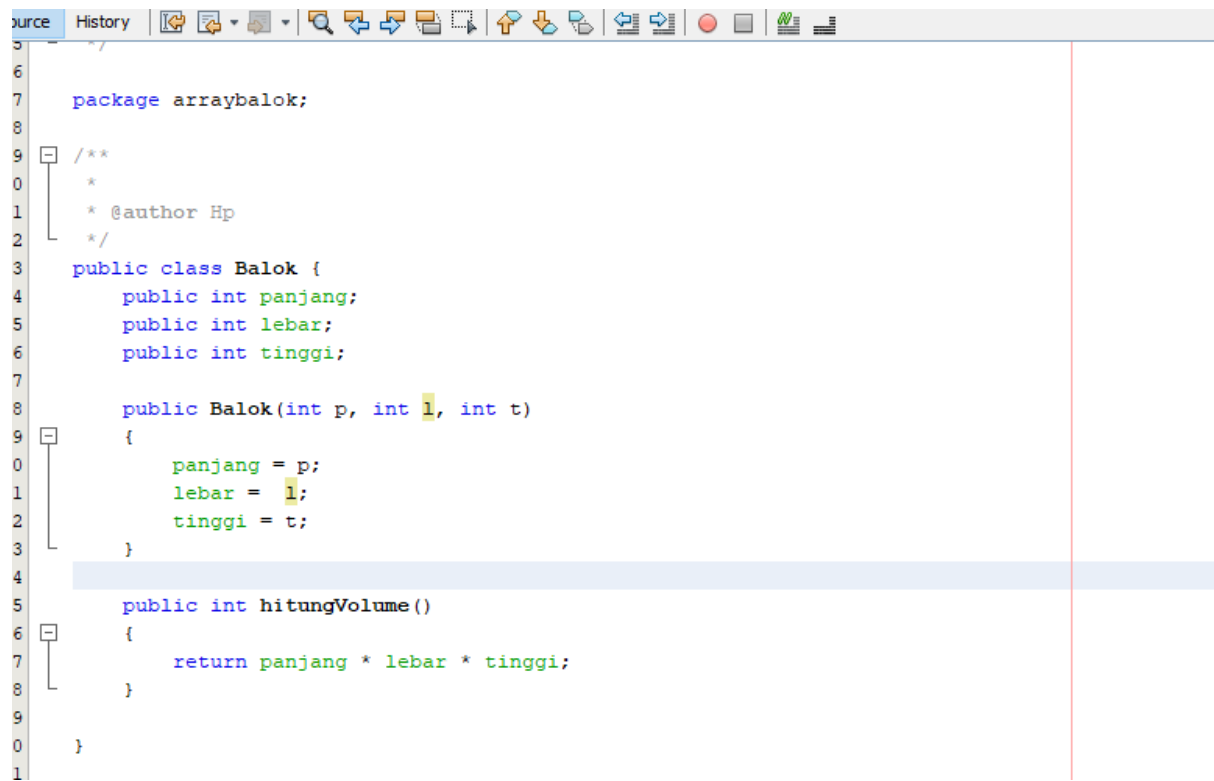
ArrayBalok:



```
source History | [Icons]
7 package arraybalok;
8
9 /**
10  *
11  * @author Hp
12  */
13 public class ArrayBalok {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         // TODO code application logic here
20         Balok[] blArray = new Balok[3];
21
22         blArray[0] = new Balok (100, 30, 12);
23         blArray[1] = new Balok (120, 40, 15);
24         blArray[2] = new Balok (210, 50, 3);
25
26         for(int i = 0; i < 3; i++)
27         {
28             System.out.println("Volume balok ke " + i + " : " + blArray[i].hitungVolume());
29         }
30     }
31 }
32
33 }
```



Class balok:

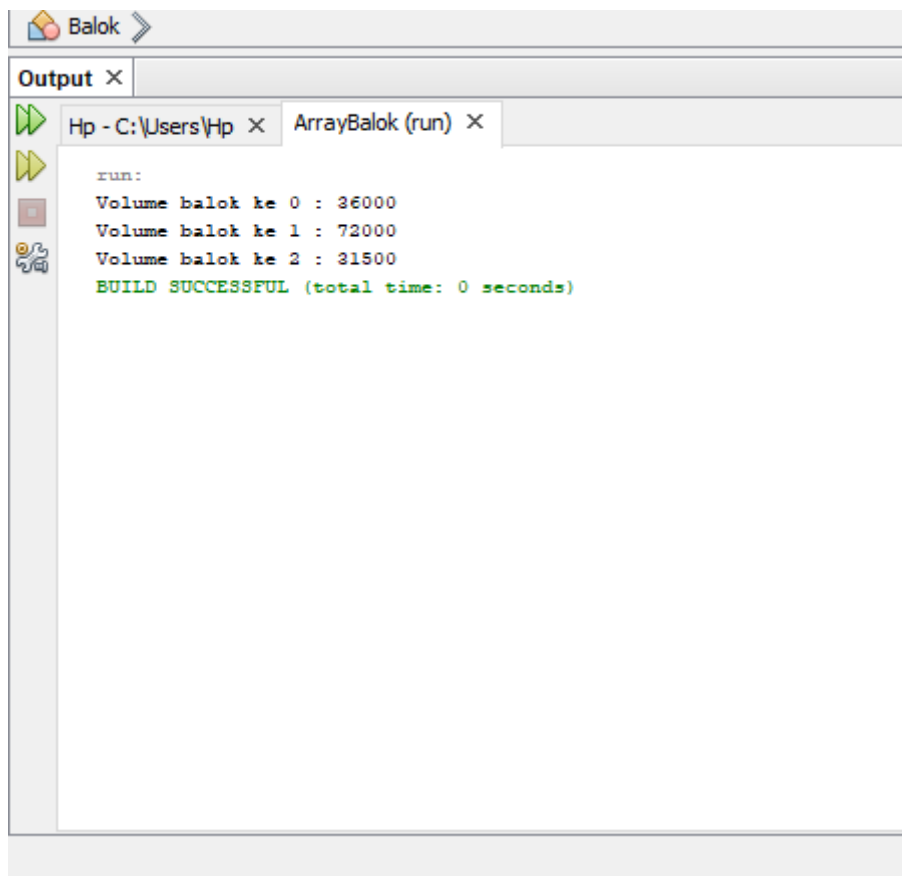


```
5
6
7 package arraybalok;
8
9 /**
10  *
11  * @author Hp
12  */
13 public class Balok {
14     public int panjang;
15     public int lebar;
16     public int tinggi;
17
18     public Balok(int p, int l, int t)
19     {
20         panjang = p;
21         lebar = l;
22         tinggi = t;
23     }
24
25     public int hitungVolume()
26     {
27         return panjang * lebar * tinggi;
28     }
29 }
30
31
```

### 3.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

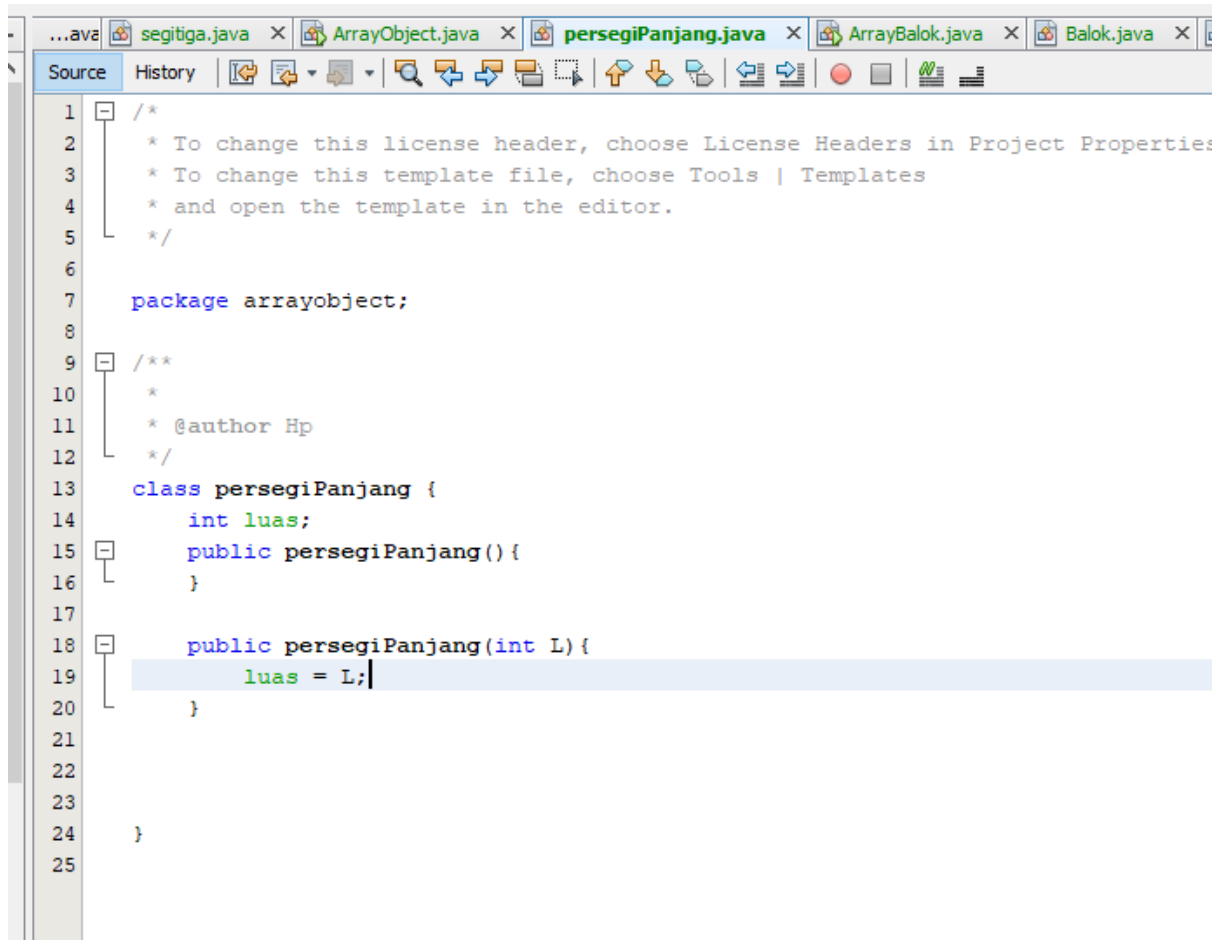
```
run:
Volume balok ke 0: 36000
Volume balok ke 1: 72000
Volume balok ke 2: 262500
```



### 3.4.3 Pertanyaan

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

Iya bisa, sebuah konstruktor dapat berjumlah lebih dari 1 dalam satu kelas.



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

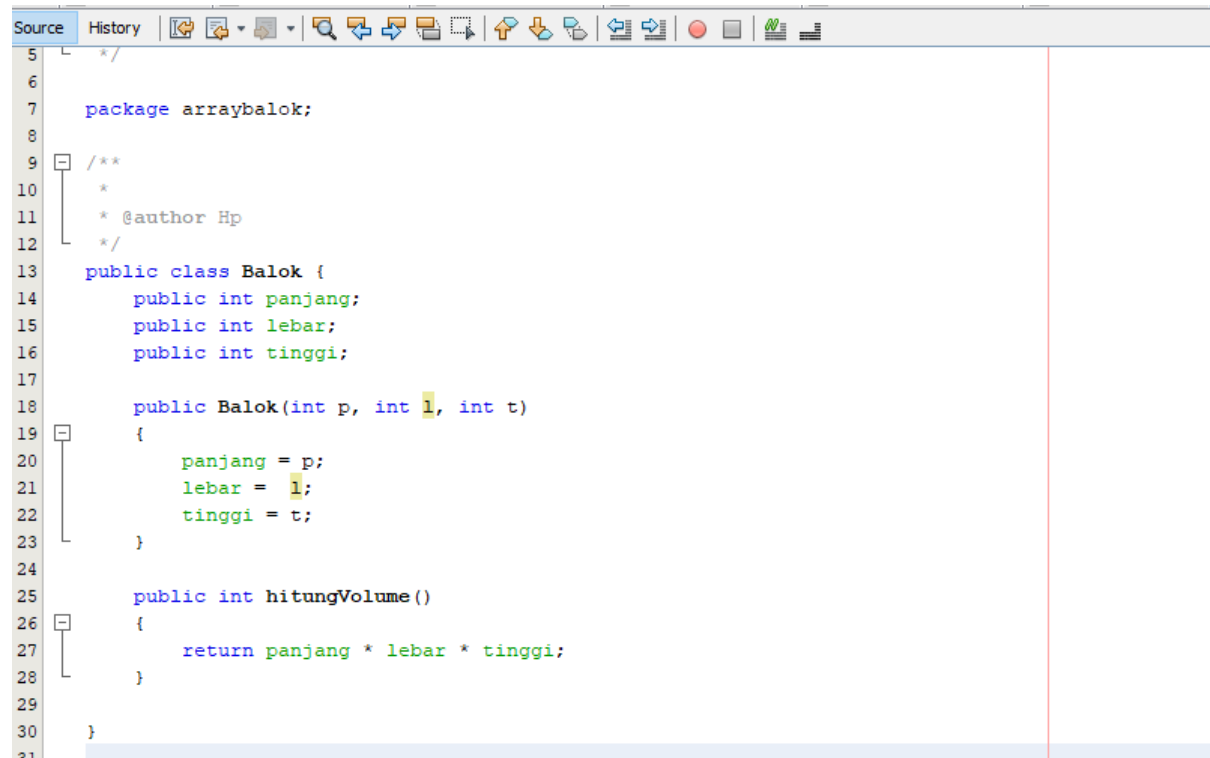
sgArray ke-1 alas: 20, tinggi: 10

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()`.

### Class Balok



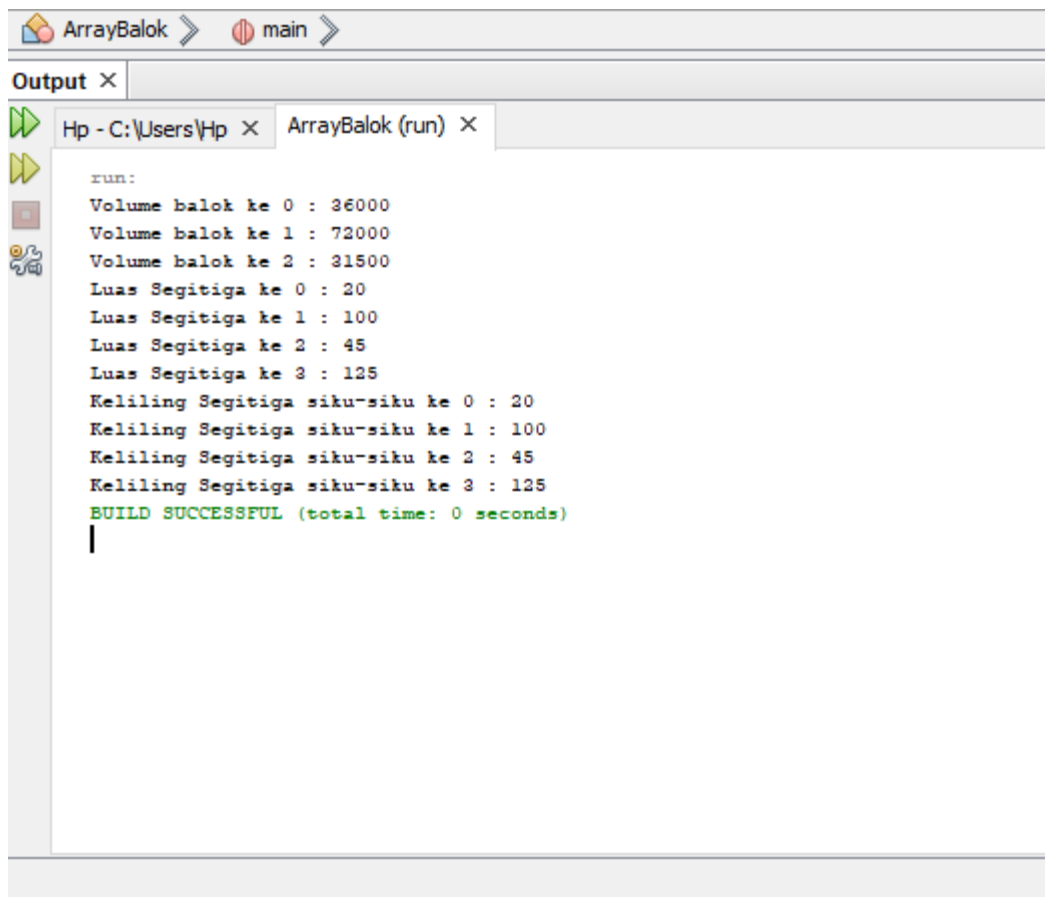
```
5  /**
6
7  package arraybalok;
8
9  /**
10 *
11 * @author Hp
12 */
13 public class Balok {
14     public int panjang;
15     public int lebar;
16     public int tinggi;
17
18     public Balok(int p, int l, int t)
19     {
20         panjang = p;
21         lebar = l;
22         tinggi = t;
23     }
24
25     public int hitungVolume()
26     {
27         return panjang * lebar * tinggi;
28     }
29
30 }
31
```

### Class segitiga

```
Source History
7 package arraybalok;
8
9 /**
10  *
11  * @author Hp
12  */
13 class segitiga {
14     public int alas;
15     public int tinggi;
16
17     public segitiga(int a, int t)
18     {
19         alas = a;
20         tinggi = t;
21     }
22     public int hitungLuas()
23     {
24         return alas * tinggi * 1/2;
25     }
26     public int hitungKeliling()
27     {
28         return (hitungLuas() - alas - tinggi) + alas + tinggi ;
29     }
30 }
31
```

## ArrayBalok

```
Source History
7 package arraybalok;
8
9 public class ArrayBalok {
10
11     /**
12      * @param args the command line arguments
13      */
14     public static void main(String[] args) {
15         // TODO code application logic here
16         Balok[] b1Array = new Balok[3];
17         segitiga[] b2Array = new segitiga[4];
18
19         b1Array[0] = new Balok (100, 30, 12);
20         b1Array[1] = new Balok (120, 40, 15);
21         b1Array[2] = new Balok (210, 50, 3);
22
23         b2Array[0] = new segitiga (10, 4);
24         b2Array[1] = new segitiga (20, 10);
25         b2Array[2] = new segitiga (15, 6);
26         b2Array[3] = new segitiga (25, 10);
27
28         for(int i = 0; i < 3; i++)
29         {
30             System.out.println("Volume balok ke " + i + " : " + b1Array[i].hitungVolume());
31         }
32
33         for(int i = 0; i < 4; i++)
34         {
35             System.out.println("Luas Segitiga ke " + i + " : " + b2Array[i].hitungLuas());
36         }
37
38         for(int i = 0; i < 4; i++)
39         {
40             System.out.println("Keliling Segitiga siku-siku ke " + i + " : " + b2Array[i].hitungKeliling());
41         }
42     }
43 }
44
45 }
46
```



The screenshot shows an IDE window with a tab labeled 'ArrayBalok' and a sub-tab 'main'. Below the tabs is an 'Output' window. The output window contains the following text:

```
run:
Volume balok ke 0 : 36000
Volume balok ke 1 : 72000
Volume balok ke 2 : 31500
Luas Segitiga ke 0 : 20
Luas Segitiga ke 1 : 100
Luas Segitiga ke 2 : 45
Luas Segitiga ke 3 : 125
Keliling Segitiga siku-siku ke 0 : 20
Keliling Segitiga siku-siku ke 1 : 100
Keliling Segitiga siku-siku ke 2 : 45
Keliling Segitiga siku-siku ke 3 : 125
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

ArrayTugas1 - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Source History

```
4  * and open the template in the editor.
5  */
6
7  package arraykubus;
8
9  /**
10   *
11   * @author Hp
12   */
13  class Balok_1 {
14      int panjang, lebar, tinggi;
15
16      Balok_1(int p, int l, int t){
17          panjang = p;
18          lebar = l;
19          tinggi = t;
20      }
21
22      int hitungVolumeBalok(){
23          return panjang * tinggi * lebar;
24      }
25
26      int hitungLuasPermukaanBalok(){
27          return 2 * (panjang * lebar + panjang * tinggi + lebar * tinggi);
28      }
29  }
30
31
```

Output X

Hp - C:\Users\Hp X ArrayNilai (run) #2 X

ArrayNilai (run) running... 1 | 1 | DNS

ArrayTugas1 - NetBeans IDE 7.4

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

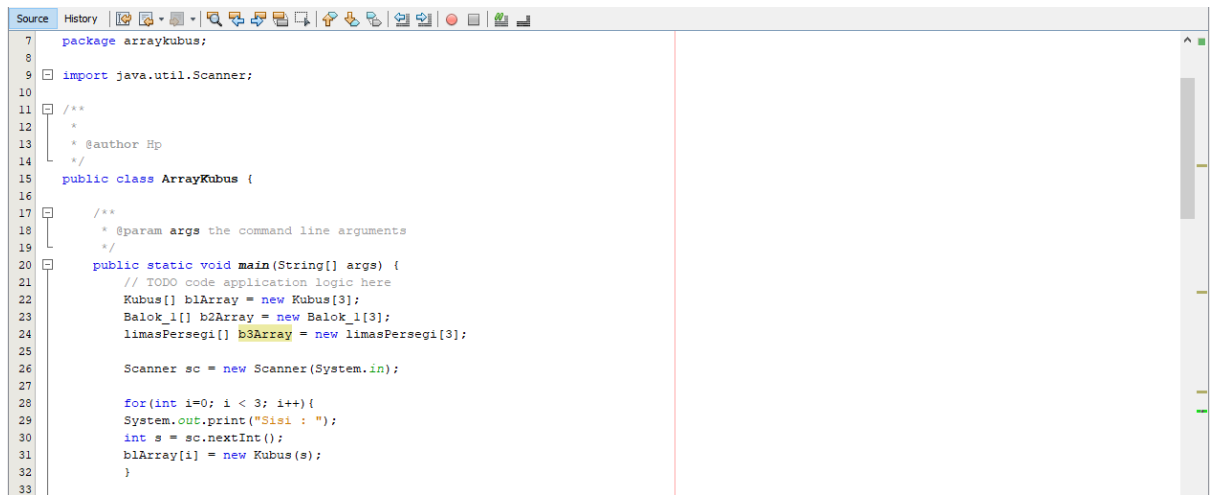
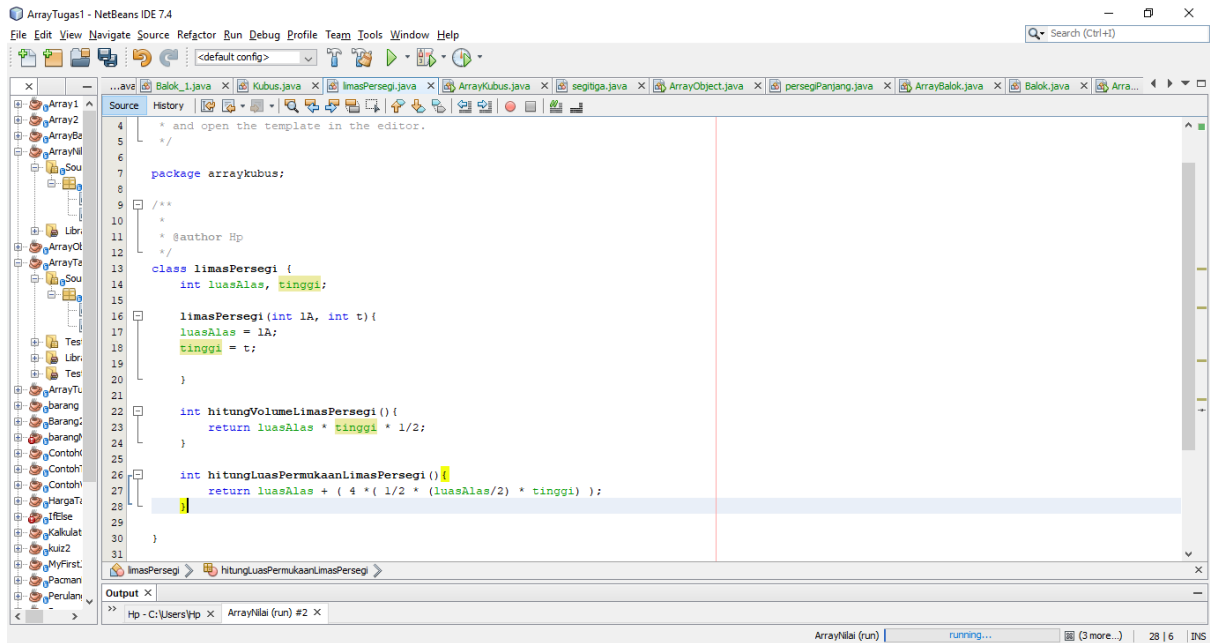
Source History

```
4  * and open the template in the editor.
5  */
6
7  package arraykubus;
8
9  /**
10   *
11   * @author Hp
12   */
13  class Kubus {
14      int sisi;
15
16      Kubus(int s){
17          sisi = s;
18      }
19
20      int hitungVolumeKubus(){
21          return sisi * sisi * sisi;
22      }
23
24      int hitungLuasPermukaanKubus(){
25          return 6 * (sisi * sisi);
26      }
27  }
28
29
```

Output X

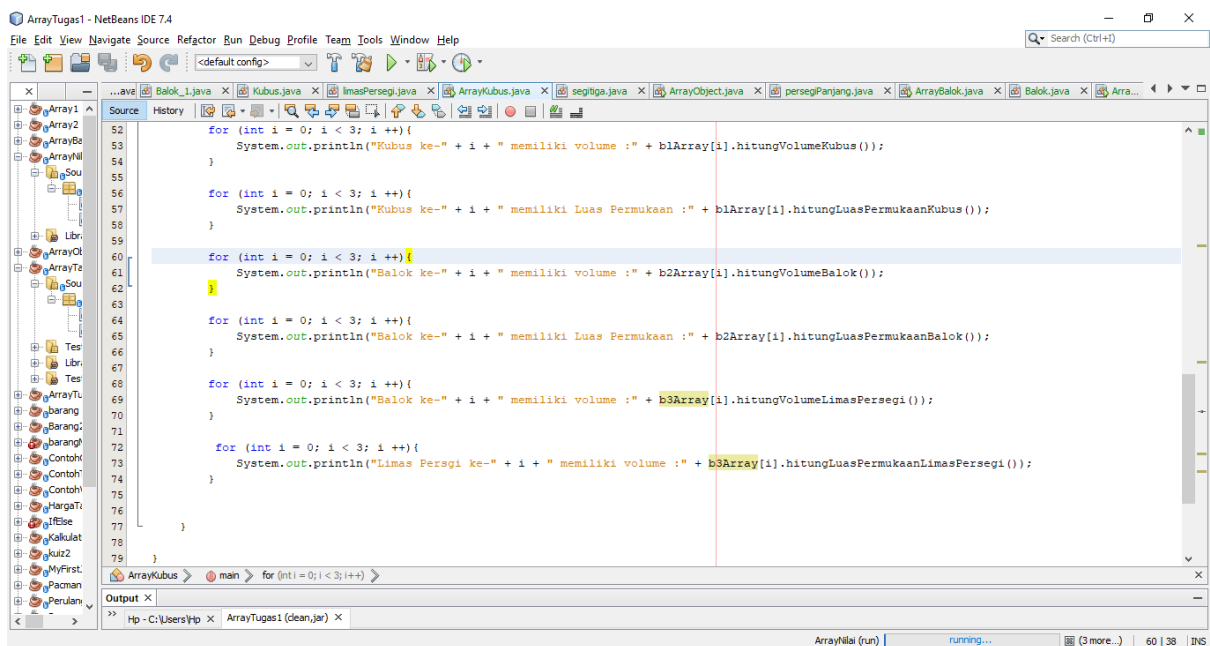
Hp - C:\Users\Hp X ArrayNilai (run) #2 X

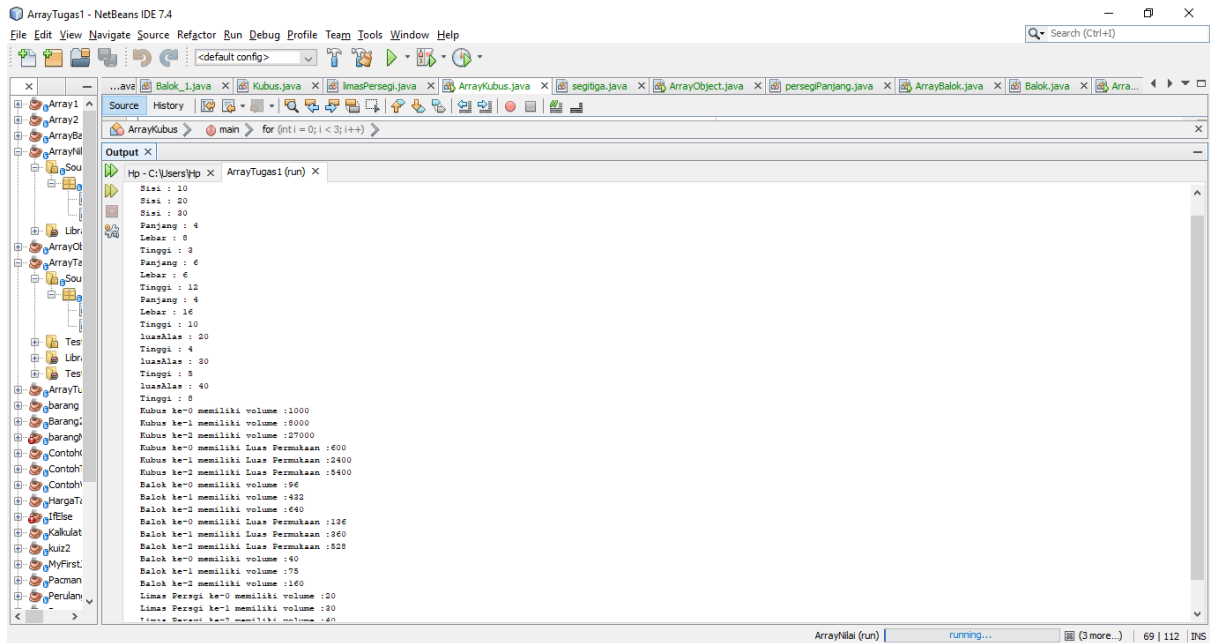
ArrayNilai (run) running... 1 | 1 | DNS





```
Source History
34         for(int i=0; i < 3; i++){
35             System.out.print("Panjang : ");
36             int p = sc.nextInt();
37             System.out.print("Lebar : ");
38             int l = sc.nextInt();
39             System.out.print("Tinggi : ");
40             int t = sc.nextInt();
41             b2Array[i] = new Balok_l(p, l, t);
42         }
43
44         for(int i=0; i < 3; i++){
45             System.out.print("luasAlas : ");
46             int lA = sc.nextInt();
47             System.out.print("Tinggi : ");
48             int t = sc.nextInt();
49             b3Array[i] = new limasPersegi(lA, t);
50         }
51     }
```





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

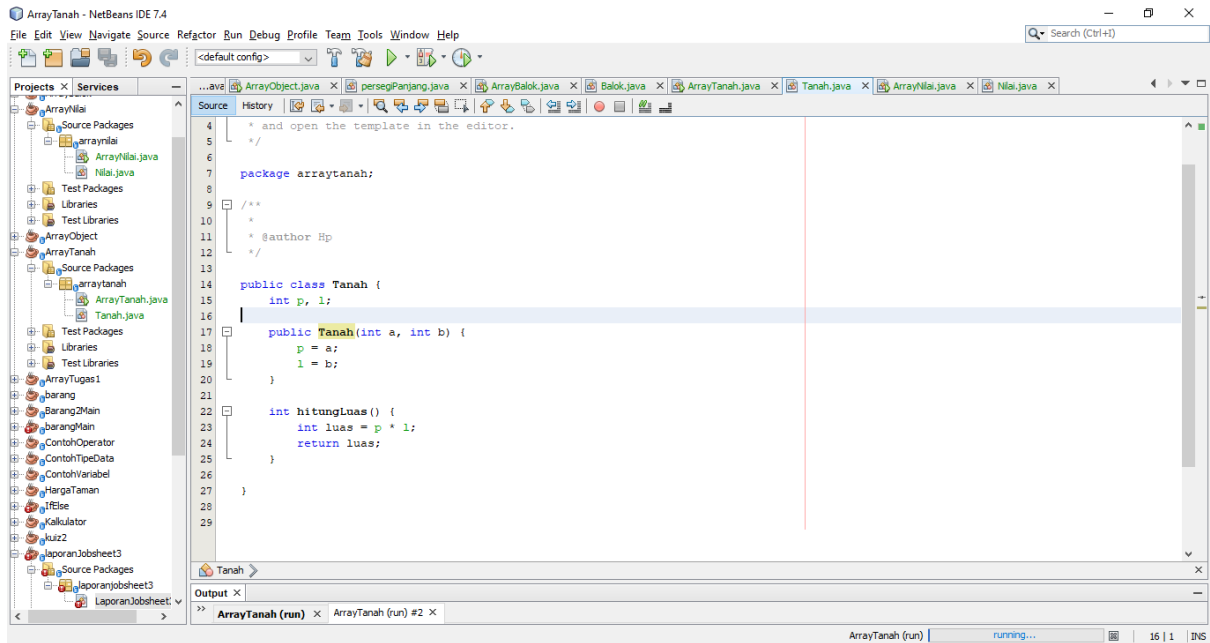
Lebar: 100

Luas Tanah 1: 4000

Luas Tanah 2: 25000

Luas Tanah 3: 12000

Tanah terluas: Tanah 2



## Class main

```

7   package arraytanah;
8
9   import java.util.Scanner;
10
11  public class ArrayTanah {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18          Scanner sc = new Scanner(System.in);
19          System.out.print("Masukkan Jumlah Tanah Yang Akan Dihitung : ");
20          int hitungan = sc.nextInt();
21          System.out.println();
22          Tanah[] tnh = new Tanah[hitungan];
23
24          for (int i = 0; i < tnh.length; i++) {
25              System.out.print("Masukkan panjang tanah ke - " + (i + 1) + " : ");
26              int p = sc.nextInt();
27              System.out.print("Masukkan lebar tanah ke - " + (i + 1) + " : ");
28              int l = sc.nextInt();
29              tnh[i] = new Tanah(p, l);
30              System.out.println();
31          }
32

```

```

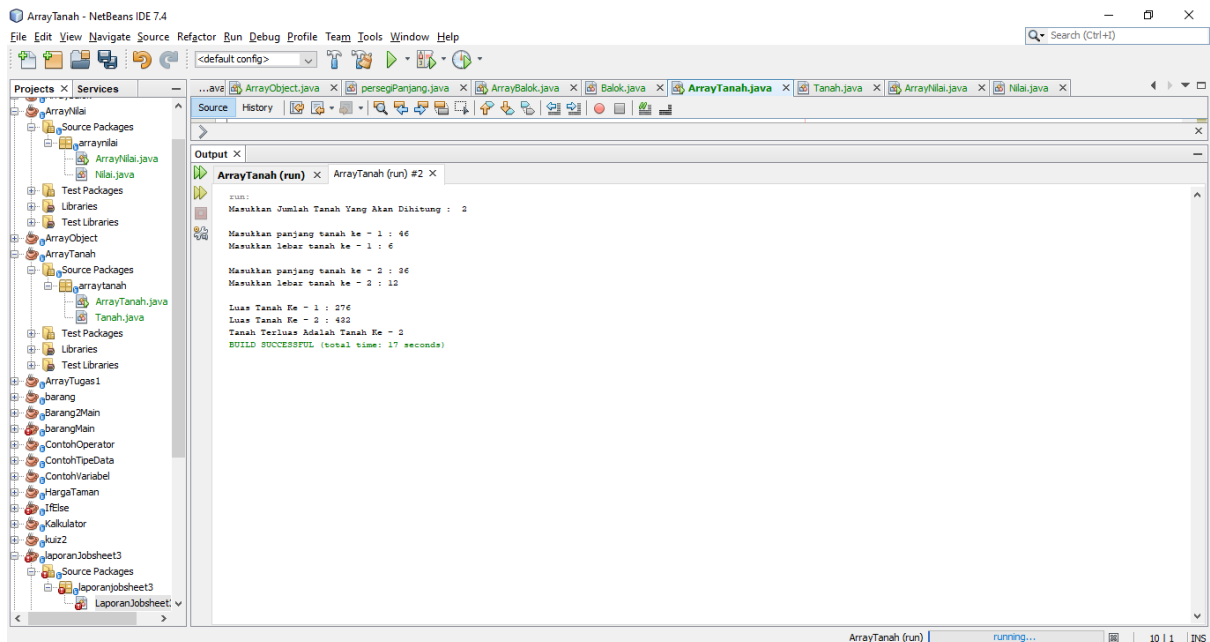
    int a = 0;
    int b = 0;
    for (int i = 0; i < tnh.length; i++) {
        if (tnh[i].hitungLuas() > a) {
            a = tnh[i].hitungLuas();
            b = i + 1;
        }
    }

    for (int i = 0; i < tnh.length; i++) {
        System.out.println("Luas Tanah Ke - " + (i + 1) + " : " + tnh[i].hitungLuas());
    }

    System.out.println("Tanah Terluas Adalah Tanah Ke - " + b);
}
}
}

```

Output :



3. 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:

```
1  /*  
2   * To change this license header, choose License Headers in Project Properties.  
3   * To change this template file, choose Tools | Templates  
4   * and open the template in the editor.  
5   */  
6  
7  package arraynilai;  
8  
9  /**  
10   *  
11   * @author Hp  
12   */  
13  class Nilai {  
14      public String nama;  
15      public int nim;  
16      public String jenisKelamin;  
17      public double ipk;  
18  }  
19  
20
```

```

7 package arraynilai;
8 import java.util.Scanner;
9 public class ArrayNilai {
10     /**
11      * @param args the command line arguments
12      */
13     public static void main(String[] args) {
14         // TODO code application logic here
15         Nilai[] ppArray = new Nilai[3];
16         Scanner sc = new Scanner (System.in);
17
18         System.out.println("MASUKKAN DATA YANG AKAN DIINPUT");
19         System.out.println("=====");
20
21         for(int i = 0; i < 3; i++)
22         {
23             ppArray[i] = new Nilai();
24             System.out.println("Masukkan data mahasiswa Ke-"+ (i+1));
25             System.out.print("Masukkan Nama : ");
26             ppArray[i].nama = sc.next();
27             System.out.print("Masukkan NIM : ");
28             ppArray[i].nim = sc.nextInt();
29             System.out.print("Masukkan Jenis Kelamin : ");
30             ppArray[i].jenisKelamin = sc.next();
31             System.out.print("Masukkan IPK : ");
32             ppArray[i].ipk = sc.nextDouble();
33         }

```

```

34
35         System.out.println("=====");
36
37         for (int i = 0; i < 3; i++)
38         {
39             System.out.println("Data Mahasiswa Ke-" + (i+1));
40             System.out.println("Nama: " + ppArray[i].nama );
41             System.out.println( "NIM : " + ppArray[i].nim);
42             System.out.println( "Jenis Kelamin : " + ppArray[i].jenisKelamin);
43             System.out.println( " Nilai IPK : " + ppArray[i].ipk);
44         }
45     }
46
47 }
48

```

Output X

▶

▶

■

⚙

Hp - C:\Users\Hp X

ArrayNilai (run) #2 X

MASUKKAN DATA YANG AKAN DIINPUT

=====

Masukkan data mahasiswa Ke-1

Masukkan Nama : Amal

Masukkan NIM : 2435364

Masukkan Jenis Kelamin : perempuan

Masukkan IPK : 3.9

Masukkan data mahasiswa Ke-2

Masukkan Nama : Haechan

Masukkan NIM : 2435376

Masukkan Jenis Kelamin : Laki-laki

Masukkan IPK : 3.8

Masukkan data mahasiswa Ke-3

Masukkan Nama : Jihoon

Masukkan NIM : 3343625

Masukkan Jenis Kelamin : Laki-laki

Masukkan IPK : 3.7

=====

Data Mahasiswa Ke-1

Nama: Amal

NIM : 2435364

Jenis Kelamin : perempuan

Nilai IPK : 3.9

Data Mahasiswa Ke-2

Nama: Haechan

NIM : 2435376

Jenis Kelamin : Laki-laki

Nilai IPK : 3.8

Data Mahasiswa Ke-3

Nama: Jihoon

NIM : 3343625

Jenis Kelamin : Laki-laki

Nilai IPK : 3.7

BUILD SUCCESSFUL (total time: 1 minute 6 seconds)