

Array of Object

TIM AJAR

ALGORITMA DAN STRUKTUR DATA

2023/2024

Capaian Pembelajaran

- Setelah menempuh materi ini, mahasiswa mampu:
 - Memahami konsep penerapan array dalam mengelola banyak object (array of object)
 - Memahami cara deklarasi array of object
 - Memahami cara instansiasi object dalam setiap elemen array
 - Memahami cara mengakses atribut dari object yang ada di dalam array of object
 - Memahami jenis error yang sering muncul saat penerapan array of object
 - Memahami penggunaan looping dalam pengelolaan array

Array of Object

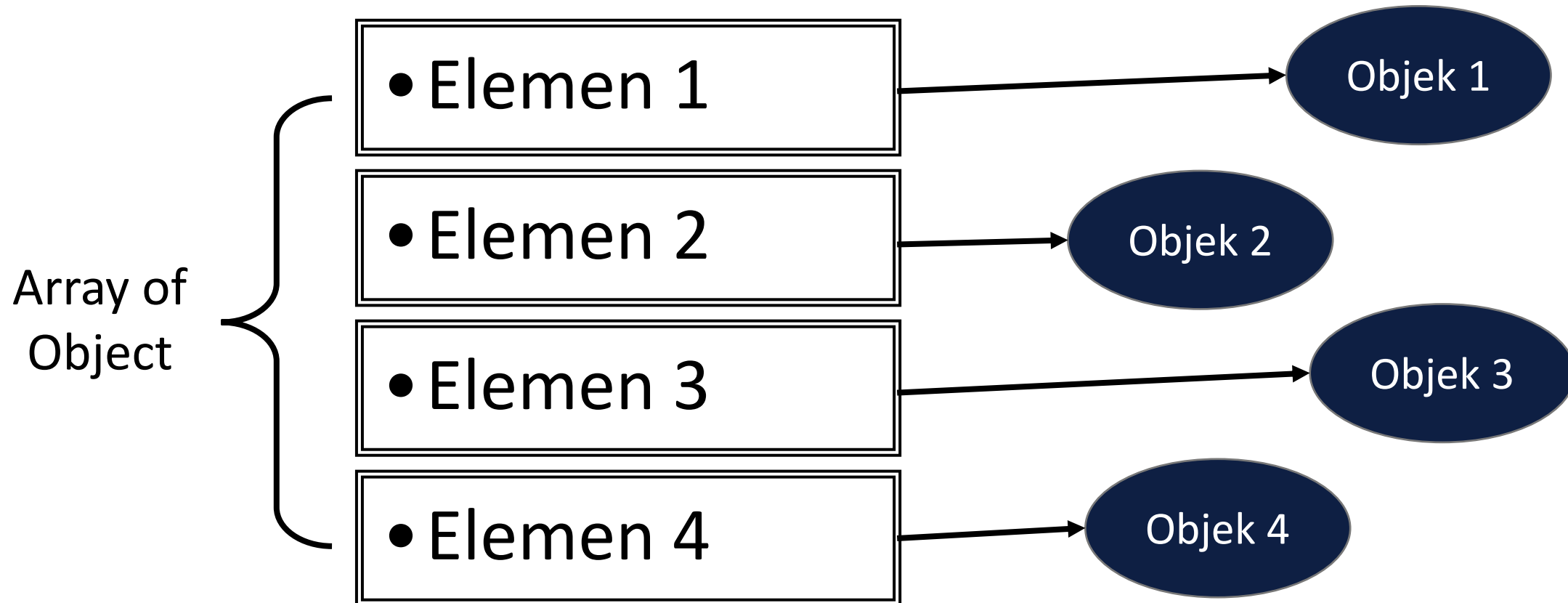
- Array adalah **variable kompleks** yang bisa digunakan sebagai tempat penyimpanan **banyak data** yang **bertipe sama** dan memiliki **index**.
- Selain tipe data primitif seperti int, float, double, dan sebagainya, array pada Java dapat juga menyimpan variabel **objek**.
- Amati kode berikut ini yang merupakan deklarasi class **PersegiPanjang**

```
1  public class PersegiPanjang {  
2      int panjang;  
3      int lebar;  
4  int hitungLuas(){  
5      return panjang*lebar;  
6  }  
7  int hitungKeliling(){  
8      return 2*(panjang+lebar);  
9  }  
10 }
```

- Array of Object digunakan untuk membuat sekumpulan objek yang dibuat dari class **PersegiPanjang**

Ilustrasi Array of Object

Bayangkan terdapat 4 item (elemen) pada sebuah array dengan jenisnya adalah objek



Deklarasi dan Instantiasi Array of Object #1

- Untuk membuat array dari objek PersegiPanjang diatas, caranya hampir sama seperti membuat array biasa:

```
PersegiPanjang[] ppArray = new PersegiPanjang[7];
```

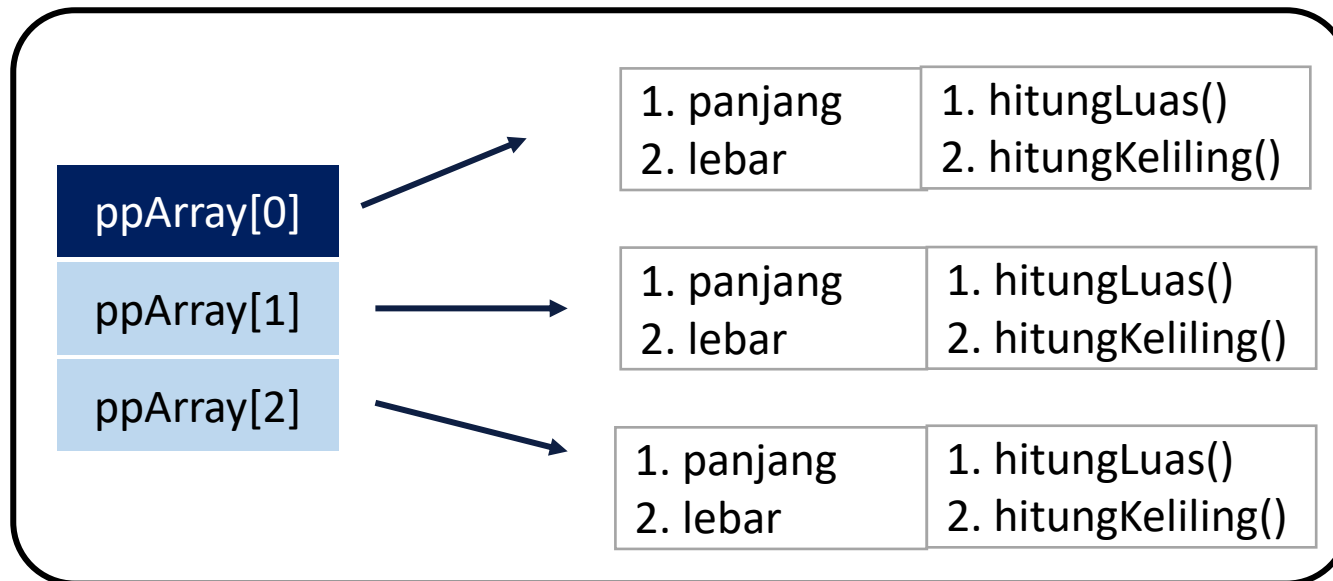
- Kode diatas akan membuat array **ppArray** yang dapat menampung **tujuh objek PersegiPanjang**.
- Namun perlu diperhatikan bahwa array tersebut **masih kosong**, dimana objek yang ada di dalamnya belum terbentuk (**belum terinstantiasi**)
- Untuk mengisi elemen array, kita harus **menginstantiasi objek persegi panjang** pada array tersebut, kita harus membuat objeknya terlebih

```
ppArray[0] = new PersegiPanjang();  
ppArray[0].panjang = 120;  
ppArray[0].lebar = 80;
```

Deklarasi dan Instantiasi Array of Object #2

- Untuk menampilkan ke layar isi atribut dari objek yang berada didalam array, caranya hampir sama, contoh:

```
System.out.println("Panjang: " + ppArray[0].panjang);  
System.out.println("Panjang: " + ppArray[0].lebar);
```



Jika array of objek
pArray[0] dibuat juga
pada index 1 dan 2

Deklarasi dan Instantiasi *Array of Object* #3

- Perhatikan kode berikut ini,

```
public class ArrayObjects
{
    public static void main(String[] args)
    {
        PersegiPanjang[] ppArray = new PersegiPanjang[7];

        ppArray[0] = new PersegiPanjang();
        ppArray[0].panjang = 120;
        ppArray[0].lebar = 80;

        System.out.println("Panjang: " + ppArray[0].panjang);
        System.out.println("Panjang: " + ppArray[0].lebar);
    }
}
```

```
1  public class PersegiPanjang {
2      int panjang;
3      int lebar;
4      int hitungLuas(){
5          return panjang*lebar;
6      }
7      int hitungKeliling(){
8          return 2*(panjang+lebar);
9      }
10 }
```

HASIL

```
run:
Panjang: 120
Panjang: 80
BUILD SUCCESSFUL (total time: 0 seconds)
```

Error NullPointerException

- Sebelum mengisi atribut pada objek didalam array, sebelumnya harus dibuat objek-nya terlebih dahulu.
- Perhatikan bahwa kode dibawah ini akan memunculkan error **NullPointerException** pada saat program dijalankan:

```
PersegiPanjang[] ppArray = new PersegiPanjang[7];  
  
ppArray[1].panjang = 70;
```

```
run:  
Exception in thread "main" java.lang.NullPointerException  
    at arrayobjects.ArrayObjects.main(ArrayObjects.java:22)  
C:\Users\rrism\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1  
BUILD FAILED (total time: 0 seconds)
```

- Hal tersebut dikarenakan pada index ppArray ke-1, belum dibuat objek PersegiPanjang. Berikut ini kode yang benar:

```
PersegiPanjang[] ppArray = new PersegiPanjang[7];  
ppArray[1] = new PersegiPanjang();  
ppArray[1].panjang = 70;
```

Pembuatan objek
PersegiPanjang pada
ppArray index ke-1

Error ArrayIndexOutOfBoundsException

- Error yang ditimbulkan karena pengaksesan elemen array di luar batas (indeks array di luar 0 s.d. length-1)

```
1 package week2;
2
3 public class PersegiPanjang {
4     int panjang;
5     int lebar;
6     int hitungLuas(){
7         return panjang*lebar;
8     }
9     int hitungKeliling(){
10        return 2*(panjang+lebar);
11    }
12
13    Run | Debug
14    public static void main(String[] args) {
15        PersegiPanjang[] pArr = new PersegiPanjang[3];
16        pArr[3].lebar = 12;
17    }
```

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
    at week2.PersegiPanjang.main(PersegiPanjang.java:15)
```

Looping untuk membuat objek ke semua indeks array

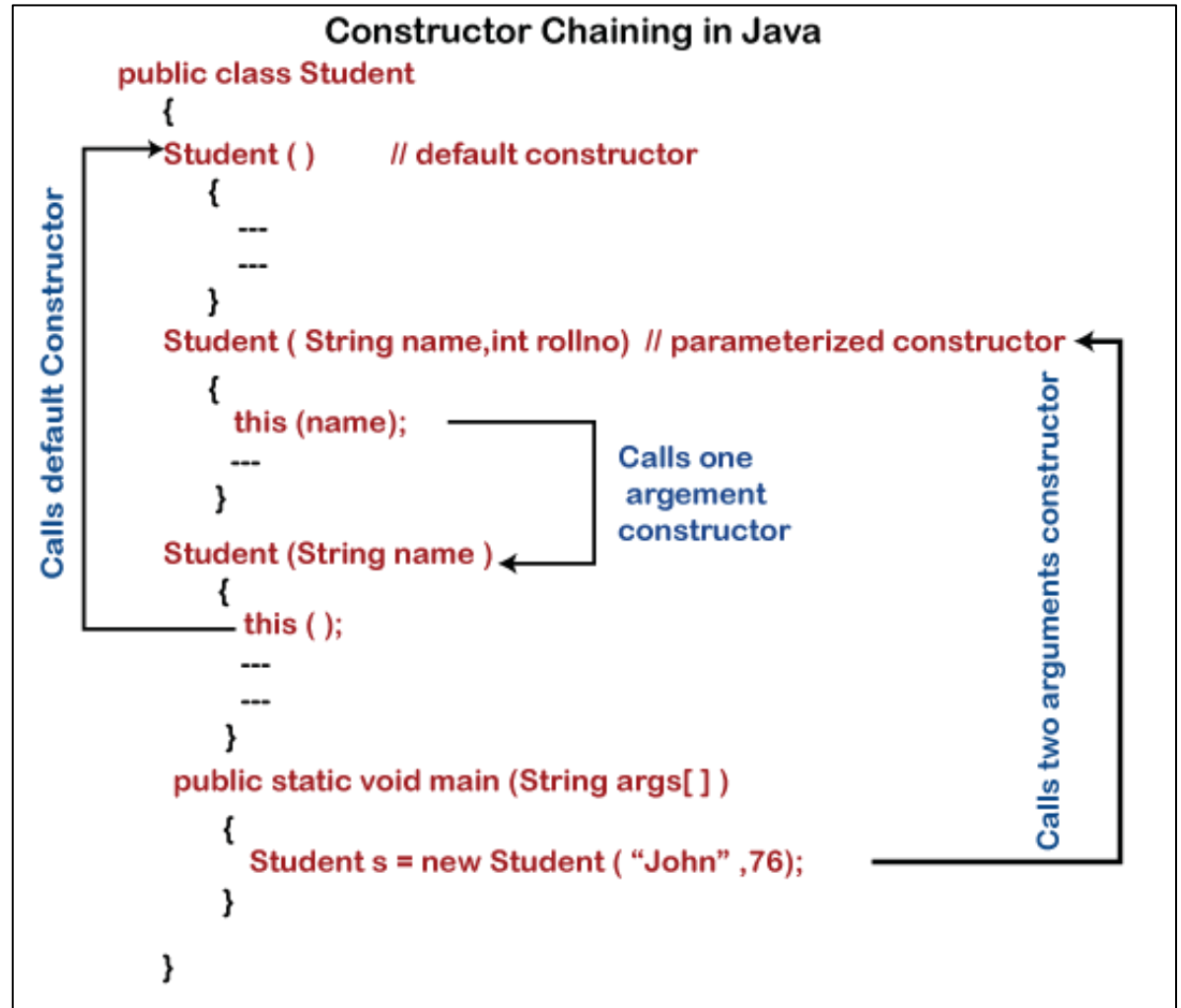
- Kita bisa gunakan *looping* untuk secara otomatis membuat objek dan mengisikannya ke semua indeks array yang ada:

```
for(int i = 0; i < 7; i++)  
{  
    ppArray[i] = new PersegiPanjang();  
}
```

```
13  ✓    public static void main(String[] args) {  
14      PersegiPanjang[] pArr = new PersegiPanjang[3];  
15  ✓    for(int i=0;i<pArr.length;i++){  
16      pArr[i] = new PersegiPanjang();  
17      pArr[i].panjang = (int)(Math.random()*10);  
18      pArr[i].lebar = (int)(Math.random()*10);  
19      }  
20  ✓    for(PersegiPanjang obj:pArr){  
21      System.out.println("P = "+obj.panjang+", L = "+obj.lebar);  
22      }  
23      }
```

Menggunakan Konstruktor #1

- Kita dapat juga menggunakan konstruktor untuk membuat objek didalam array sekaligus mengisi atributnya.
- Konstruktor adalah *method* yang bernama sama dengan nama class dan tidak memiliki return type.
- Konstruktor ini akan dipanggil pertama kali saat kita membuat objek dari class tersebut.



Menggunakan Konstruktor #2

- Perhatikan class PersegiPanjang yang telah dilengkapi konstruktor berikut ini:

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

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

Konstruktor yang menerima parameter p dan l

Atribut panjang diisi dengan parameter p

Atribut lebar diisi dengan parameter l

- Kemudian pada fungsi main:

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

ppArray[0] = new PersegiPanjang(100, 20);
ppArray[1] = new PersegiPanjang(120, 70);
ppArray[2] = new PersegiPanjang(80, 40);
```

Membuat objek PersegiPanjang menggunakan konstruktor

Menggunakan Konstruktor #3

- Perhatikan kode selengkapnya berikut,

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

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

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

        ppArray[0] = new PersegiPanjang(100, 20);
        ppArray[1] = new PersegiPanjang(120, 70);
        ppArray[2] = new PersegiPanjang(80, 40);

        System.out.println("Lebar persegi panjang ke-1: " + ppArray[1].lebar);
    }
}
```

HASIL

```
run:
Lebar persegi panjang ke-1: 70
BUILD SUCCESSFUL (total time: 0 seconds)
```

Latihan



1. Berikanlah tiga contoh penerapan array of object dalam beberapa bidang yang Anda ketahui, serta tentukan atribut dan method yang dimiliki pada penerapan tersebut (Sebutkan atribut dan method dalam bentuk list)!
2. Buatlah class diagram pada jawaban no 1! (Ada 3 class diagram)
3. Terdapat sebuah class "MataKuliah" yang memiliki,
 - Atribut: kode mata kuliah, nama mata kuliah, dosen pengampu, kuota kelas, dan daftar mahasiswa
 - Method: mengganti kode mata kuliah, mengganti nama mata kuliah, mengganti, dosen pengampu, menambah kuota kelas, mengurangi kuota kelas, dan menambah mahasiswa ke dalam kelas

Buatlah class diagram berdasarkan kasus tersebut

