

## Array 1

### 1. Kompetensi

1. Mahasiswa mampu memahami dan menjelaskan fungsi array satu dimensi.
2. Mahasiswa mampu membuat program dengan menggunakan konsep array satu dimensi.

### 2. Alat dan Bahan

1. PC/Laptop
2. Text editor (Sublime Text)
3. JDK

### 3. Uraian Teori

Array adalah sekumpulan tempat penyimpanan data yang bertipe sama dan memiliki index. Array dapat diibaratkan sebagai sekumpulan variabel yang bertipe sama dan bernama sama. Array biasanya digunakan untuk menyimpan deret angka. Kemudian untuk membedakan nilai/isi dari variabel tersebut, digunakan index.

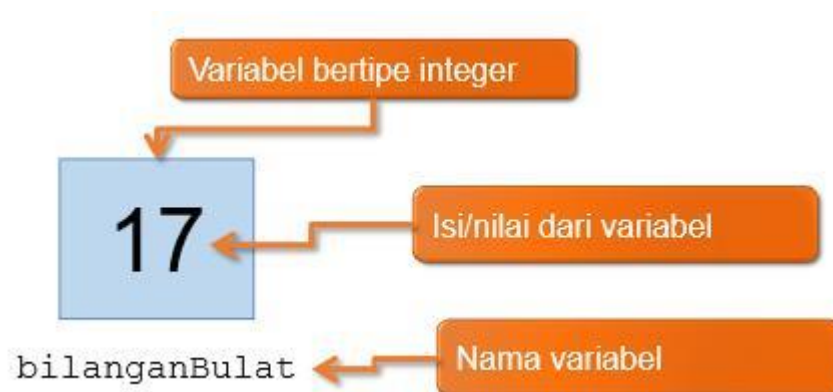
Ada dua tipe array:

- Array satu dimensi
- Array multi dimensi

Pada jobsheet praktikum ini hanya akan dibahas array satu dimensi saja. Array multi dimensi akan dibahas di jobsheet selanjutnya.

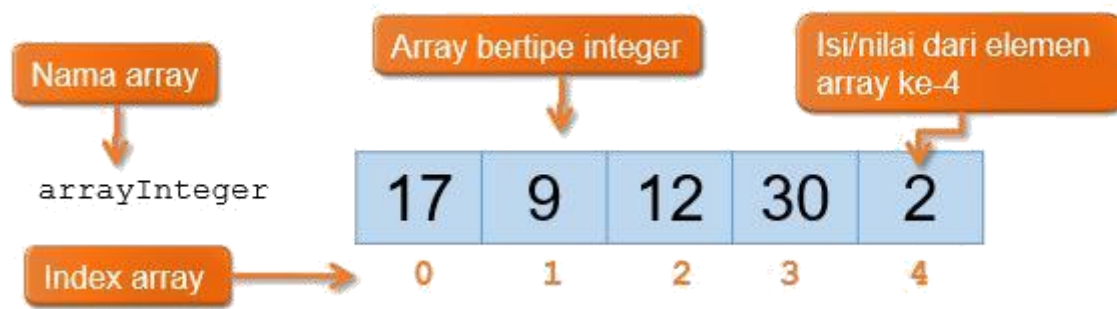
#### 3.1 Array Satu Dimensi

Amati ilustrasi tentang **variabel** berikut ini.



Gambar 1. Variabel

Ilustrasi diatas adalah variabel bertipe integer yang bernama **bilanganBulat** dan berisi nilai 17. Satu buah variabel hanya dapat menyimpan **satu** buah nilai. Sekarang amati ilustrasi tentang **array** berikut ini:



Gambar 2. Array satu dimensi

Ilustrasi diatas adalah **array** bertipe **integer** bernama **arrayInteger** dan memiliki kapasitas 5 buah bilangan integer. Sebuah array dapat menyimpan lebih dari satu nilai (tergantung dari kapasitasnya). Meskipun begitu, nilai-nilai yang disimpan di sebuah array harus bertipe sama. Tiap elemen pada array dinomori dengan **index array**. Index array selalu dimulai dari 0 (nol).

### 3.2 Cara Mendeklarasikan Array

Secara umum, cara mendeklarasikan array adalah sebagai berikut:

```
tipe[] namaArray = new tipe[kapasitas];
```

- **Tipe** adalah tipe data dari array yang akan dibuat.
- **namaArray** adalah nama dari array yang akan dibuat.
- **Kapasitas** adalah banyaknya nilai yang dapat disimpan didalam array yang akan dibuat.

Untuk mengakses (mengisi/membaca) sebuah elemen dari array, kita hanya perlu menuliskan nama array tersebut, kemudian diikuti dengan index yang dituju didalam tanda kurung kotak [ ].

Contoh program berikut ini membuat array bernama **bil**, bertipe **integer**, jumlah elemen 4, kemudian mengisinya dengan beberapa nilai, kemudian menampilkan isi elemennya ke layar.

```
class MyArray
{
    public static void main(String[] args)
    {
        int[] bil = new int[4];
        bil[0] = 5;
        bil[1] = 12;
        bil[2] = 7;
        bil[3] = 20;
        System.out.println(bil[1]);
        System.out.println(bil[0]);
        System.out.println(bil[3]);
    }
}
```

Deklarasi array

Mengisi array

Menampilkan isi array elemen ke 1, 0 dan 3

Hasil:



```
E:\Java>java MyArray
12
5
20
```

Ada beberapa alternatif cara untuk mendeklarasikan array.

1. Deklarasi array tanpa mengalokasikan jumlah elemennya:

```
int[] myArray;
```

2. Deklarasi array dengan mengalokasikan 10 elemen:

```
int[] myArray = new int[10];
```

3. Mengubah jumlah elemen array dengan 50:

```
myArray = new int[50];
```

4. Deklarasi array dan mengisinya secara langsung:

```
int[] myArray = {10, 20, 30, 40};
```

```
String[] myArray2 = {"Malang", "Surabaya"};
```

5. Mengetahui jumlah elemen

```
array: myArray.length
```

### 3.3 Menggunakan Perulangan Pada Array Satu Dimensi

Untuk mengakses (mengisi/membaca) nilai dari sebuah array, kita dapat menggunakan perulangan.

```
class MyArray
{
    public static void main(String[] args)
    {
        int[] bil = new int[4];

        bil[0] = 5;
        bil[1] = 12;
        bil[2] = 7;
        bil[3] = 20;

        for(int i = 0; i < 4; i++)
        {
            System.out.println(bil[i]);
        }
    }
}
```

Hasil:

```
E:\Java>java MyArray
5
12
7
20
```



Kita juga bisa menggunakan perulangan untuk menerima input dan menyimpannya kedalam array. Contoh, program yang meminta input sebanyak 5 bilangan kemudian menampilkan kembali 5 bilangan tersebut.

```
import java.util.Scanner;

class MyArray
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int[] bil = new int[5];

        for(int i = 0; i < 5; i++)
        {
            System.out.print("Masukkan bilangan ke-" + i + ": ");
            bil[i] = sc.nextInt();
        }

        for(int i = 0; i < 5; i++)
        {
            System.out.println("Isi bilangan ke-" + i + " adalah: " + bil[i]);
        }
    }
}
```

Hasil:

```
E:\Java>java MyArray
Masukkan bilangan ke-0: 5
Masukkan bilangan ke-1: 10
Masukkan bilangan ke-2: 25
Masukkan bilangan ke-3: 100
Masukkan bilangan ke-4: 15
Isi bilangan ke-0 adalah: 5
Isi bilangan ke-1 adalah: 10
Isi bilangan ke-2 adalah: 25
Isi bilangan ke-3 adalah: 100
Isi bilangan ke-4 adalah: 15
```

### 3.4 Input Jumlah Elemen Array

Kita juga dapat membuat program yang meminta input berapa banyak jumlah elemen array. Langkahnya secara umum adalah:

1. Deklarasikan array tanpa mengalokasikan jumlah elemennya.
2. Buat input yang menerima jumlah elemen array kemudian simpan di sebuah variabel.
3. Set jumlah elemen array menggunakan variabel yang diinputkan tadi.

Coba amati program berikut ini:

```
import java.util.Scanner;

class MyArray2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int N;
        int[] arr;

        System.out.print("Masukkan jumlah elemen array: ");
        N = sc.nextInt();

        arr = new int[N];

        for(int i = 0; i < N; i++)
        {
            System.out.print("Masukkan array ke-" + i + ": ");
            arr[i] = sc.nextInt();
        }

        for(int i = 0; i < N; i++)
        {
            System.out.println("Array ke-" + i + " adalah: " + arr[i]);
        }
    }
}
```

Deklarasi array tanpa alokasi jumlah elemen

Input jumlah elemen array, simpan ke variabel N

Set jumlah elemen array menggunakan variabel N

Hasilnya:

```
Masukkan jumlah elemen array: 3
Masukkan array ke-0: 40
Masukkan array ke-1: 20
Masukkan array ke-2: 70
Array ke-0 adalah: 40
Array ke-1 adalah: 20
Array ke-2 adalah: 70
```

#### 4. Langkah Praktikum

Ikuti langkah-langkah praktikum berikut ini.

##### Praktikum 1

1. Buka editor Sublime Text.
2. Buat file baru beri nama "MyArray.java".
3. Buat array bertipe integer dengan nama bil dengan kapasitas 4 elemen.

```
int[] bil = new int[4];
```

4. Isi masing-masing elemen array bil tadi dengan angka 5, 12, 7, 20.

```
bil[0] = 5;
bil[1] = 12;
bil[2] = 7;
bil[3] = 20;
```



5. Tampilkan ke layar semua isi elemennya:

```
System.out.println(bil[0]);
System.out.println(bil[1]);
System.out.println(bil[2]);
System.out.println(bil[3]);
```

6. Cocokkan dan amati hasilnya dengan gambar berikut ini:

```
5
12
7
20
```

## Praktikum 2

1. Buat file baru beri nama "ArrayLoop.java".
2. Buat array bertipe integer dengan nama usia, dengan kapasitas 5 elemen.

```
int[] usia = new int[5];
```

3. Isi array usia tadi dengan nilai 25, 19, 20, 20.

```
usia[0] = 25;
usia[1] = 19;
usia[2] = 20;
usia[3] = 20;
```

4. Dengan perulangan, tampilkan semua isi array usia.

```
for(int i = 0; i < 4; i++)
{
    System.out.println(usia[i]);
}
```

5. Cocokkan dan amati hasilnya dengan gambar berikut ini:

```
25
19
20
20
```

## Praktikum 3

1. Buat file baru beri nama "ArrayInputLoop.java".
2. Import dan deklarasikan Scanner untuk keperluan input.

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

```
Scanner sc = new Scanner(System.in);
```

3. Buat array bertipe integer dengan nama nilaiUAS, dengan kapasitas 6 elemen.

```
int[] nilaiUAS = new int[6];
```



4. Menggunakan perulangan, buat input untuk mengisi elemen dari array nilaiUAS.

```
for(int i = 0; i < 6; i++)
{
    System.out.print("Masukkan nilai UAS ke-" + i + ": ");
    nilaiUAS[i] = sc.nextInt();
}
```

5. Menggunakan perulangan, tampilkan semua isi elemen dari array nilaiUAS.

```
for(int i = 0; i < 6; i++)
{
    System.out.println("Nilai UAS ke-" + i + " adalah: " +
        nilaiUAS[i]);
}
```

6. Cocokkan dan amati hasilnya dengan gambar berikut ini:

```
Masukkan nilai UAS ke-0: 100
Masukkan nilai UAS ke-1: 70
Masukkan nilai UAS ke-2: 60
Masukkan nilai UAS ke-3: 90
Masukkan nilai UAS ke-4: 40
Masukkan nilai UAS ke-5: 75
Nilai UAS ke-0 adalah: 100
Nilai UAS ke-1 adalah: 70
Nilai UAS ke-2 adalah: 60
Nilai UAS ke-3 adalah: 90
Nilai UAS ke-4 adalah: 40
Nilai UAS ke-5 adalah: 75
```

#### Praktikum 4

Pada praktikum ini, kita akan membuat program untuk menyeleksi kelulusan mahasiswa. Program akan menerima input sebanyak 10 nilai mahasiswa. Mahasiswa dengan nilai akhir lebih dari 70 akan dinyatakan lulus.

1. Buat file baru beri nama "SeleksiKelulusan".
2. Import dan deklarasikan Scanner untuk keperluan input.

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

```
Scanner sc = new Scanner(System.in);
```

3. Buat array nilaiAkhir bertipe integer dengan kapasitas 10.

```
int[] nilaiAkhir = new int[10];
```

4. Menggunakan perulangan, buat input untuk mengisi array nilaiAkhir.

```
for(int i = 0; i < 10; i++)
{
    System.out.print("Masukkan nilai Mahasiswa ke-" + i + ": ");
    nilaiAkhir[i] = sc.nextInt();
}
```

5. Menggunakan perulangan dan percabangan, lakukan seleksi isi array nilaiAkhir dengan kondisi jika isi elemen tertentu pada array nilaiAkhir lebih dari 70 maka dinyatakan lulus.





```
for(int i = 0; i < 10; i++)
{
    if(nilaiAkhir[i] > 70)
    {
        System.out.println("Mahasiswa ke-" + i + " lulus!");
    }
}
```

6. Cocokkan dan amati hasilnya:

```
Masukkan nilai Mahasiswa ke-0: 40
Masukkan nilai Mahasiswa ke-1: 72
Masukkan nilai Mahasiswa ke-2: 50
Masukkan nilai Mahasiswa ke-3: 69
Masukkan nilai Mahasiswa ke-4: 100
Masukkan nilai Mahasiswa ke-5: 30
Masukkan nilai Mahasiswa ke-6: 30
Masukkan nilai Mahasiswa ke-7: 50
Masukkan nilai Mahasiswa ke-8: 70
Masukkan nilai Mahasiswa ke-9: 80
Mahasiswa ke-1 lulus!
Mahasiswa ke-4 lulus!
Mahasiswa ke-9 lulus!
```

## 5. Pertanyaan

Jawablah pertanyaan berikut ini.

1. Apa yang dimaksud dengan array?
2. Apa yang dimaksud dengan elemen array?
3. Bolehkah elemen array diisi dengan tipe data yang berbeda-beda?
4. Bagaimana cara mendeklarasikan array "harga" dengan tipe integer dan berkapasitas 100 elemen?
5. Pada Praktikum 1, bagaimana cara menampilkan elemen ke-0 dan ke-2 pada array bil?
6. Modifikasi hasil program pada Praktikum 4, agar hanya menampilkan mahasiswa yang **tidak lulus** saja.
7. Modifikasi hasil prgoram pada Praktikum 4, agar menampilkan semua mahasiswa, dan ditandai mana yang lulus dan tidak lulus.

```
Masukkan nilai Mahasiswa ke-0: 40
Masukkan nilai Mahasiswa ke-1: 72
Masukkan nilai Mahasiswa ke-2: 50
Masukkan nilai Mahasiswa ke-3: 69
Masukkan nilai Mahasiswa ke-4: 100
Masukkan nilai Mahasiswa ke-5: 30
Masukkan nilai Mahasiswa ke-6: 30
Masukkan nilai Mahasiswa ke-7: 50
Masukkan nilai Mahasiswa ke-8: 70
Masukkan nilai Mahasiswa ke-9: 80
Mahasiswa ke-0 tidak lulus!
Mahasiswa ke-1 lulus!
Mahasiswa ke-2 tidak lulus!
Mahasiswa ke-3 tidak lulus!
Mahasiswa ke-4 lulus!
Mahasiswa ke-5 tidak lulus!
Mahasiswa ke-6 tidak lulus!
Mahasiswa ke-7 tidak lulus!
Mahasiswa ke-8 tidak lulus!
Mahasiswa ke-9 lulus!
```





## 6. Tugas

Kerjakan tugas sesuai dengan instruksi berikut ini.

1. **(InputArray)** Buatlah program yang terdapat array dengan jumlah elemen 7, buatlah input untuk mengisi elemen array tersebut, kemudian tampilkan ke layar semua isi array
2. **(ReversedArray)** Buatlah program yang terdapat array dengan jumlah elemen 4, buatlah input untuk mengisi elemen array tersebut, kemudian tampilkan isi array tersebut dengan urutan terbalik.
3. **(ArrayCopy)** Buatlah program yang terdapat array **arr1** dan **arr2**, masing-masing dengan jumlah elemen 4, buat input untuk mengisi array **arr1**, kemudian salin isi dari **arr1** ke **arr2**, kemudian tampilkan ke layar isi dari **arr2**.
4. **(ArrayAddition)** Buatlah program yang terdapat dua array yaitu **arr1** dan **arr2**, masing-masing berkapasitas 3 elemen. Buatlah input untuk memasukkan elemen kedua array tersebut, kemudian menggunakan perulangan, jumlahkan masing-masing elemen dari **arr1** dengan **arr2**, kemudian tampilkan ke layar hasil penjumlahannya
5. **(ArraySum)** Buatlah program yang menerima input jumlah elemen array, inputkan isi arraynya, kemudian jumlahkan semua isi array tersebut dan tampilkan hasil penjumlahannya. Contoh hasil program:

```
Masukkan isi array: 3
Masukkan array ke-0: 5
Masukkan array ke-1: 2
Masukkan array ke-2: 9
Hasil penjumlahan isi elemen array: 16
```

Note:

Untuk cara menerima input jumlah elemen array, lihat bab 3.4

6. **(OddEven)** Buatlah program yang menerima input jumlah elemen array, inputkan isi arraynya, kemudian tampilkan mana yang genap dan mana yang ganjil. Contoh hasil program:

```
Masukkan isi array: 5
Masukkan array ke-0: 10
Masukkan array ke-1: 20
Masukkan array ke-2: 15
Masukkan array ke-3: 30
Masukkan array ke-4: 25
Ganjil:
15
25
Genap:
10
20
30
```

7. **(ArrayMax)** Buatlah program yang menerima input jumlah elemen array, inputkan isi arraynya, kemudian tampilkan bilangan terbesar dari isi elemen arraynya. Contoh hasil program:

```
Masukkan isi array: 4
Masukkan array ke-0: 25
Masukkan array ke-1: 10
Masukkan array ke-2: 55
Masukkan array ke-0: 15
Bilangan terbesar: 55
```

## 7. Materi Pengayaan

### 7.1 Pencarian Pada Array

Salah satu hal yang sering dilakukan pada operasi array adalah pencarian atau *searching*. Pencarian dilakukan untuk menemukan nilai tertentu pada elemen didalam array. Ada banyak algoritma searching, namun yang paling mudah adalah Linear Search.

Misalkan pada sebuah array, kita ingin mencari dimana posisi index dari sebuah array. Pada Linear Search, kita membandingkan “key” atau angka yang ingin kita cari, dengan tiap elemen yang ada didalam array. Amati gambar berikut:



Gambar 2. Linear Search.

Pada gambar diatas, key atau angka yang ingin kita cari adalah 3. Menggunakan looping kita bandingkan masing-masing elemen dari array. Dan ternyata angka 3 berada di index ke 5. Maka setelah ketemu, looping akan berhenti, dan hasil akhir dari program adalah 5 (index dimana angka 3 berada). Amati contoh program berikut:

```
import java.util.Scanner;

class LinearSearch
{
    public static void main(String[] args)
    {
        int[] arr = {6, 4, 1, 9, 7, 3, 2, 8};
        int key = 3;
        int hasil = 0;

        for(int i = 0; i < arr.length; i++)
        {
            if(key == arr[i])
            {
                hasil = i;
                break;
            }
        }

        System.out.println("Key ada di array ke: " + hasil);
    }
}
```

*arr.length* digunakan untuk mengetahui jumlah elemen array

Jika key ketemu, akhiri looping menggunakan perintah break

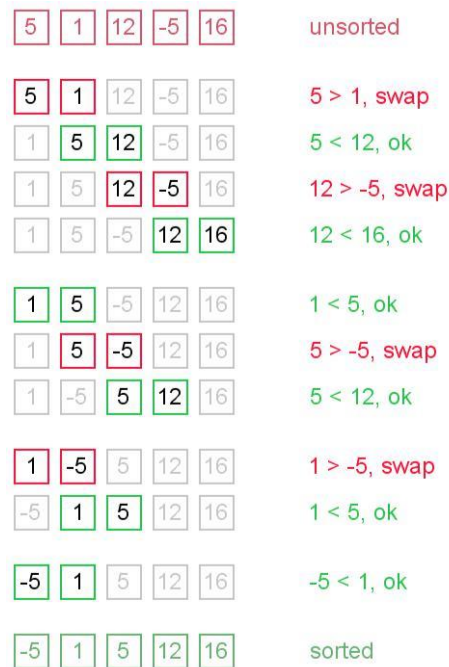
Hasil:

```
E:\Java>java LinearSearch
Key ada di array ke: 5
```

## 7.2 Sorting

Sorting adalah proses mengurutkan elemen array dari yang terkecil ke besar (ascending) atau sebaliknya (descending). Ada banyak algoritma untuk pengurutan, namun yang paling mudah adalah Bubble Sort.

Didalam Bubble Sort, dilakukan looping dari elemen pertama sampai elemen terakhir dari array. Kemudian tiap elemen dibandingkan dengan elemen berikutnya. Jika elemen tersebut lebih besar dari elemen berikutnya, maka akan ditukar. Amati ilustrasi berikut ini:



Gambar 3. Bubble Sort.

Berikut adalah contoh program Bubble Sort:

```
class BubbleSort
{
    public static void main(String[] args)
    {
        int[] arr = {16, 4, 10, 90, 27, 3, 12, 28};
        int n = arr.length;
        int temp = 0;

        for(int i = 0; i < n; i++)
        {
            for(int j = 1; j < (n-i); j++)
            {
                if(arr[j-1] > arr[j])
                {
                    //swap elements
                    temp = arr[j-1];
                    arr[j-1] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        System.out.println("Hasil pengurutan:");

        for(int i = 0; i < n; i++)
        {
            System.out.println(arr[i]);
        }
    }
}
```

Hasilnya:

```
E:\Java>java BubbleSort
Hasil pengurutan:
3
4
10
12
16
27
28
90
```



## 8. Tugas Pengayaan

Kerjakan tugas sesuai dengan instruksi berikut ini.

1. **(SearchInput)** Buatlah program yang menerima input jumlah array, isi array, key yang ingin dicari. Cetak ke layar index posisi elemen dari key yang ingin dicari. Contoh hasil program:

```
Masukkan jumlah elemen array: 4
Array ke 0: 5
Array ke 1: 12
Array ke 2: 25
Array ke 3: 10
Masukkan key yang ingin dicari: 12
Key ada di posisi index ke: 1
```

2. **(SortingInput)** Buatlah program yang menerima input jumlah array, isi array, kemudian urutkan array tersebut, kemudian tampilkan ke layar hasil pengurutannya. Contoh hasil program:

```
Masukkan jumlah elemen array: 4
Array ke 0: 5
Array ke 1: 12
Array ke 2: 25
Array ke 3: 10
Hasil
pengurutan: 5 10
12 25
```