



# Array 1

Array 1 Dimensi



### Tujuan:

Di akhir pertemuan, mahasiswa diharapkan mampu:

- Memahami konsep array 1 dimensi
- Memberikan contoh penggunaan array 1 dimensi
- Menyelesaikan studi kasus searching dan sorting sederhana



- Outline
- Pengantar
- Deklarasi dan Instansiasi
- Mengakses dan mengisi data array
- Panjang Elemen Array
- Searching
- Sorting
- Latihan





### Pengantar

- Bagaimana kita menyimpan data berjumlah besar yang memiliki tipe data dan nilai yang sejenis?
- Contoh: Menyimpan data nilai semua mahasiswa Polinema
  - Kita perlu mendeklarasikan ribuan variable
  - Setiap nama variable harus unik
  - Operasi yang dilakukan tiap variable nilai sama. Contoh:
    - Operasi mencetak nilai
    - Konversi nilai angka menjadi nilai huruf



### Pengantar

- Dalam matematika, terutama terkait dengan matriks yang memiliki elemen matriks. Elemen matrik dituliskan dengan menggunakan variabel berindeks.
- Misalkan sebuah matriks A[5,5] berdimensi 5x5 akan mempunyai elemen matriks berupa:  $a_{00}$  s.d  $a_{44}$ .
- Dalam pemrograman komputer, implementasi dari variabel berindeks menggunakan array. Sehingga array dapat berdimensi satu atau lebih dari satu.



### Definisi

- Array merupakan variabel kompleks dengan tipe data yang sama, menggunakan nama yang sama, dan memiliki suatu index tertentu.
- Atau merupakan sekumpulan nilai (elemen) dengan tipe data yang sama. Dimana masing-masing elemen Array bisa diakses dengan menggunakan indeks yang unik



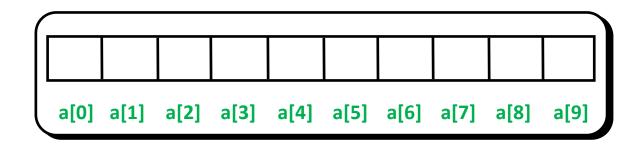
## Sifat Array

- Homogen
  - Seluruh elemen di dalam struktur array mempunyai tipe data yang sama.
- Random Access
  - Setiap elemen di dalam struktur array dapat dicapai secara individual, langsung ke lokasi elemen yang diinginkan, tidak harus melalui elemen pertama.
- Merupakan variabel referensi.



### VISUALISASI ARRAY

• Misalkan sebuah array bernama a dengan jumlah elemen sebanyak 10 elemen, maka elemen-elemen array tersebut dapat digambarkan sbb.:

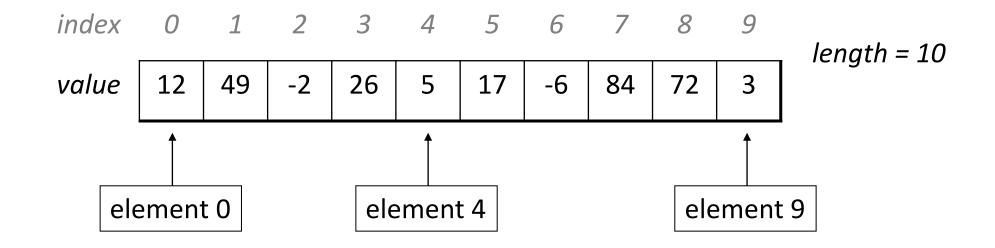


- Kotak kosong menunjukkan elemen dari Array
- Masing- masing elemen memiliki penomoran 0-9 (indeks)
- Indeks array dimulai dari 0 dan diakhiri dengan jumlah Elemen-1



### VISUALISASI ARRAY







## Deklarasi Array Satu Dimensi

Deklarasi

```
tipeData namaArrray[];
     atau

tipeData[] namaArrray;
Contoh: int a[]; int[] a;
```

- tipeData adalah tipe data dari array yang akan dibuat.
- namaArray adalah nama dari array yang akan dibuat.



### Instansiasi Array Satu Dimensi

- Instansiasi objek array:
  - Ketika sebuah array dideklarasikan, hanya referensi dari array yang dibuat. untuk alokasi memori dilakukan dengan menggunakan kunci kata new
  - Cara Instansiasi variabel array:

```
namaArray = new tipeData[jumlahElemen];
contoh: a = new int[10];
```



### Array Satu Dimensi

 Deklarasi dan instansiasi objek array dapat digabungkan dalam sebuah instruksi sbb.:



### Mengakses Elemen Array

• Merujuk ke nomor indeks.

### namaArray[indeks]

- Contoh:
  - Mengakses sebuah variabel array a dengan indeks i, dapat dituliskan:
     a[i]
  - Indeks i hanya dapat bernilai 0 atau positif dengan nilai maksimumnya adalah: (jumlah\_elemen - 1).



### Mengakses Elemen Array

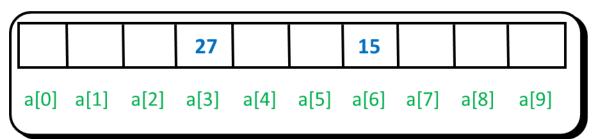
• CONTOH:

```
String[] cars = {"Volvo", "BMW", "Ford"};
System.out.println(cars[0]); //menampilkan Volvo
System.out.println(cars[2]); //menampilkan Ford
```

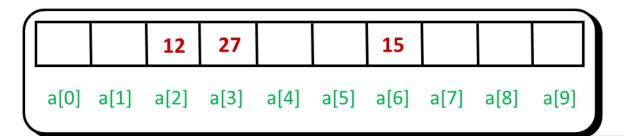


### Mengisi Data pada Array

- Mengisi data ke elemen array dilakukan dengan menggunakan assignment operator.
- Contoh: a[6] = 15; a[3] = 27;



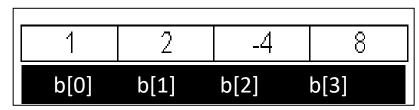
• statement a[2] = a[3] - a[6]; menghasilkan:





### Inisialisasi Array

- Array dapat diinisialisasi secara eksplisit pada saat didefinisikan dan bisa tidak diberikan nilai dimensinya.
- Contoh: int b[]={1, 2, -4, 8};
- Pada contoh diatas Array memiliki 4 element



• Contoh; int  $b[]=\{1, 2, -4, 8, 0, 0, 0, 0\};$ 

1	2	-4	8	0	0	0	0
b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]



## Inisialisasi Array (contoh)

```
• boolean results[] = { true, false, true, false };
• String[] cars = {"Volvo", "BMW", "Ford"};
• int[] myNum = {10, 20, 30, 40};
• double []grades = {100, 90, 80, 75};
• String days[] = { "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"};
```



### Mengganti Elemen Array

```
• String[] cars = {"Volvo", "BMW", "Ford"};
• cars[0] = "Opel";
```



## Mendapatkan Panjang Array

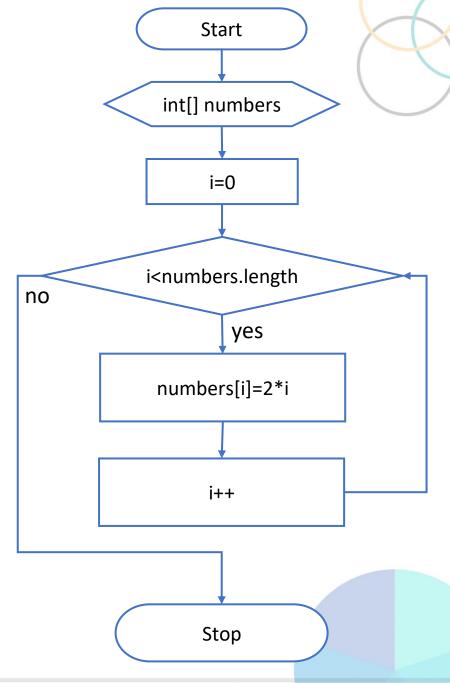
 Anda bisa mendapatkan panjang array dengan menggunakan namaArray.length

- Contoh penggunaan Panjang array:
  - Berapa indeks dari elemen terakhir dari sebuah array?
  - Berapa indeks dari elemen tengah dari sebuah array?



## Array \( \sqrt{100p} \)

- Kita dapat menggunakan panjang array, bersama dengan indeksnya, untuk melakukan beberapa operasi menggunakan loop.
- Misalnya, kita dapat menginisialisasi array secara efisien.

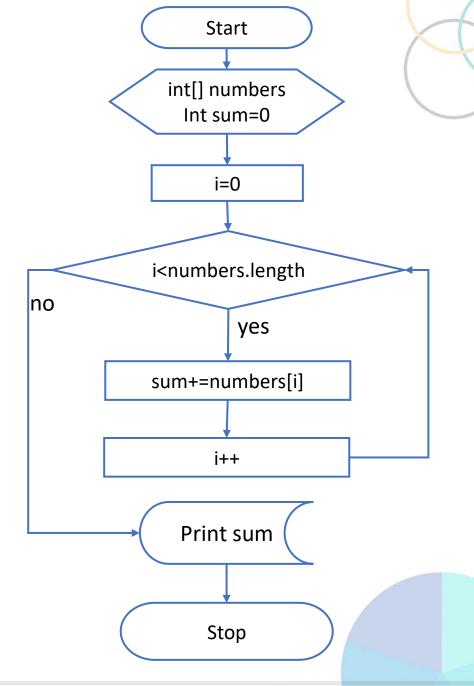




## Array $\bigcirc$ loop(contoh)

Menjumlahkan semua elemen array

```
// assume that the user has created int[]
numbers
int sum = 0;
for (int i = 0; i < numbers.length;
i++) {
    sum += numbers[i];
}
println(sum);</pre>
```









### For each loop

- Bentuk lain dari loop for yang digunakan untuk menelusuri array
- for-each loop mengurangi kode secara signifikan dan tidak ada penggunaan indeks atau lebih tepatnya penghitung dalam loop.
- Sintaks:

```
for(tipeDataArray tempVar : namaArray) {
          //statement
}
```



## Array V loop (Contoh)



 Mengakses Semua elemen array dengen menggunakan perulangan "for-each"

```
public static void main(String[] args) {
    int array[] = \{33,4,5,23,1,5,6\};
    //inisisaliasai array = menentukan jumlah elemennya
    //serta menentukan nilai dari setiap elemen
                                                                   33
    for (int i : array) {
        System.out.println(i);
                                                                   23
    } // menampilkan setiap elemen array
```

BUILD SUCCESSFUL



### Perbedaan deklarasi dan inisialisasi dengan atau tanpa ARRAY



## Penggunaan Array

- 1. Deklarasikan variabel referensi array
- 2. Instansiasi elemen array
- 3. Inisialisasi array (*Jika diperlukan*)
- 4. Memanipulasi elemen array



## Contoh Inisialisasi Array yang salah

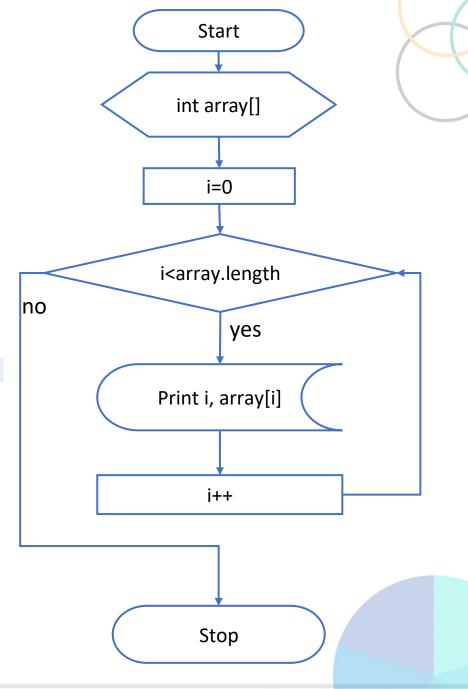
- Contoh: int  $b[4] = \{ 1, 2, -4, 8, 9 \};$ 
  - ERROR karena nilai dimensi lebih kecil dari jumlah element.

Contoh inisialisasi array setelah didefinisikan yang salah :

```
int b[5];
b[5]={0,0,0,0,0};
```



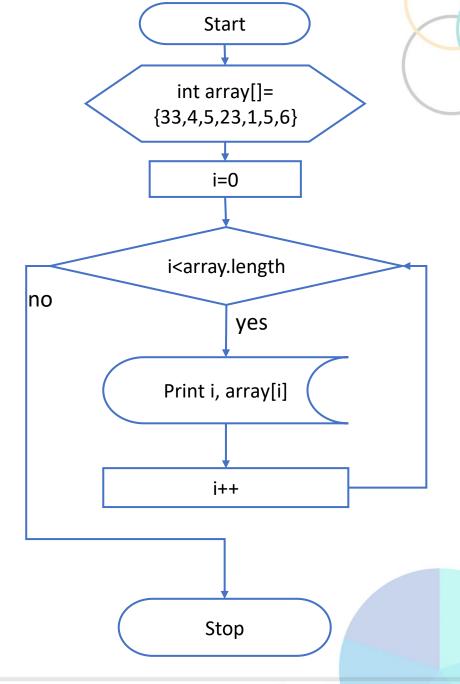
```
public class sampleArray2 {
    public static void main(String[] args) {
        int array[]; //deklarasi Array
        array = new int[10]; //intansiasi Array
        System.out.printf("%s%5s\n", "Index ", "Value");
        //menambah setiap elemen array dan ditampilkan
        for (int i=0; i<array.length;i++) {
            System.out.printf("%2d%5d\n",i,array[i]);
            run-single:
            Index Value
                 0
            BUILD SUCCESSFUL
```





```
public static void main(String[] args) {
    int array[] = {33,4,5,23,1,5,6};
    //inisisaliasai array = menentukan jumlah elemennya
    //serta menentukan nilai dari setiap elemen
    System.out.printf("%s%5s\n", "Index ", "Value");
    for (int i=0; i<array.length;i++)</pre>
        System.out.printf("%2d%5d\n",i,array[i]);
    } // menampilkan setiap elemen array
                                            run-single:
                                            Index Value
                                                33
                                                23
```

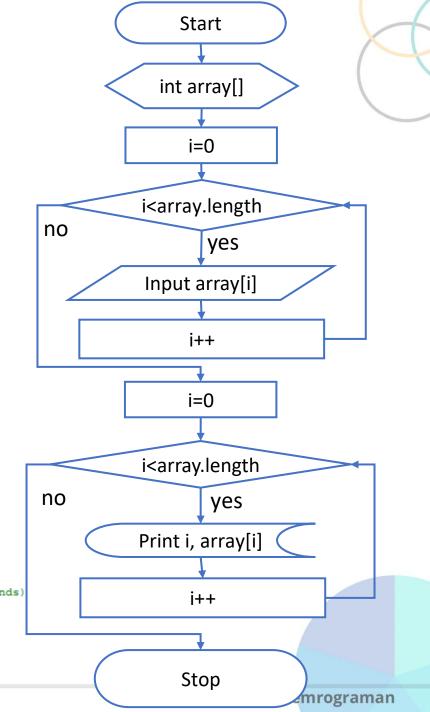
BUILD SUCCESSFUL





 Contoh, program yang meminta input sebanyak 5 bilangan kemudian menampilkan kembali 5 bilangan tersebut.

```
run-single:
public static void main(String[] args) {
                                                           masukkan angka :
    Scanner input = new Scanner(System.in);
    int array[];
                   //deklarasi Array
                                                           masukkan angka :
    array = new int[5]; //intansiasi Array
                                                           masukkan angka :
                                                           masukkan angka :
    for (int i=0; i<array.length;i++)
                                                           masukkan angka :
        System.out.println("masukkan angka: ");
                                                                0 dengan elemen
        array[i]=input.nextInt();
                                                                 1 dengan elemen
                                                                 2 dengan elemen
                                                                  dengan elemen
                                                                 4 dengan elemen
    for (int i=0; i<array.length;i++)
                                                           BUILD SUCCESSFUL (total time: 8 seconds)
        System.out.printf("array %2d dengan elemen %5d\n",i,array[i]);
    } // menampilkan setiap elemen array
```



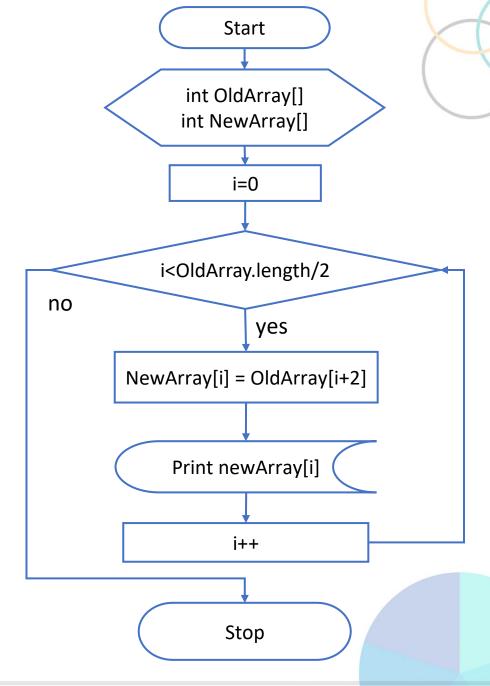


Membuat Salinan isi array

```
public static void main(String[] args) {
  int[] OldArray = {1, 3, 6, 7, 9};
  int[] NewArray = new int[5];

  for (int i = 0; i <OldArray.length/2; i++) {
     NewArray[i] = OldArray[i+2];
     System.out.print(NewArray[i]+" ");
  }

  run-single:
  6 7 BUILD SUCCESSFUL</pre>
```



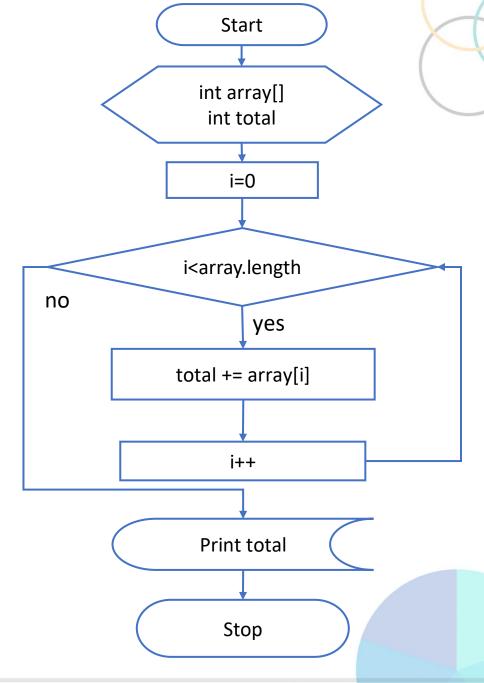


Penjumlahan Array

```
public class sampleArray5 {
    public static void main(String[] args) {
        int array[] = {33,4,5,23,1,5,6};
        int total = 0;

        for (int i=0; i<array.length;i++)
        {
            total +=array[i];
        }//menambahkan setiap nilai dari elemen array ke total
        System.out.println(total);
    }
}
run-single:
77</pre>
```

BUILD SUCCESSFUL







## MATERI PENGAYAAN

Searching dan Sorting





## Searching

- Salah satu hal yang sering dilakukan pada operasi array adalah pencarian atau searching
- Pencarian dilakukan untuk menemukan nilai tertentu pada elemen didalam array
- Salah satu algoritma searching yang paling mudah adalah Linier Search



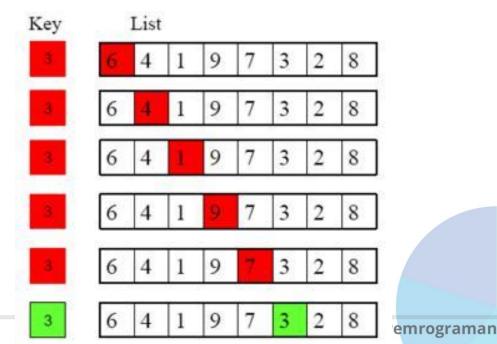
## Searching



 Misalkan pada sebuah array, ingin mencari dimana posisi index dari sebuah array.

 Pada Linear Search, dibandingkan "key" atau angka yang ingin dicari, dengan tiap elemen yang ada didalam array.

- Key yang ingin dicari adalah 3
- Digunakan perulangan untuk membandingkan masing-masing dari elemen array
- angka 3 berada di index ke 5.
- Maka setelah ketemu, looping akan berhenti





## Searching

```
public static void main(String[] args) {
    int array[] = \{6,4,1,9,7,3,2,8\};
    int key= 3;
    int hasil = 0;
    for (int i=0; i<array.length;i++)
       if (key==array[i])
           hasil=i;
           break;
    System.out.println("key "+key+" terdapat di index= "+hasil);
run-single:
key 3 terdapat di index= 5
BUILD SUCCESSFUL (total time: 1 second)
```

```
Start
        int array[], key, hasil
                 i=0
           i<array.length
no
                   yes
            key==array[i
                                  no
                                           i++
                   yes
               hasil=i
                break
           Print key, hasil
                Stop
                              asar Pemrograman
```



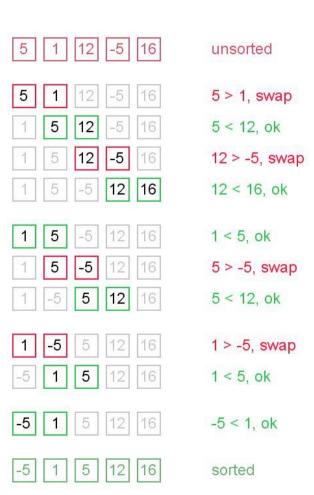
### Sorting

- Sorting adalah proses mengurutkan elemen array dari yang terkecil ke besar (ascending) atau sebaliknya (descending)
- Salah satu algoritma Sorting yang paling mudah adalah BubbleSort



### Sorting

- 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.

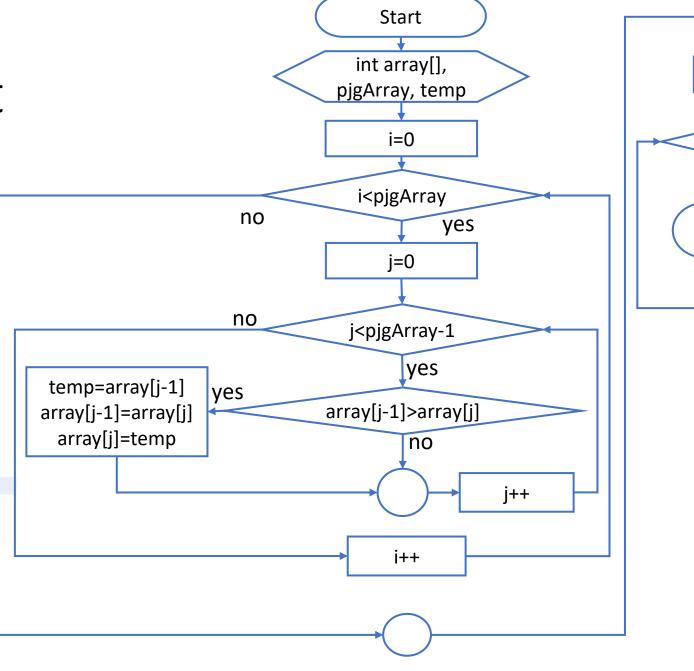




### **Bubble Sort**

```
public class bubbleSort {
   public static void main(String[] args)
        int array[] = \{6,4,1,9,7,3,2,8\};
        int pjgArray = array.length;
        int temp =0;
       for (int i=0; i<pjgArray;i++)</pre>
            for (int j=1; j<pjgArray-1;j++)</pre>
                if (array[j-1]>array[j])
                    temp=array[j-1];
                    array[j-1]=array[j];
                    array[j]=temp;
        System.out.println("hasil urutan");
        for(int i=0; i<pjgArray;i++)</pre>
            System.out.print(array[i]+" ");
         run-single:
        hasil urutan
```

7 9 8 BUILD



i=0

i<pjgArra

Print array[i]

j++

Stop

yes

no



### LATIHAN



- 1. Buat flowchart pengisian variable array dengan Panjang elemen 50 menggunakan looping!
- 2. Buat flowchart untuk mengisi elemen array dengan jumlah elemen 5, kemudian tampilkan isi array tersebut dengan urutan terbalik.
- 3. Buat flowchart yang meminta inputan pengguna berupa angka 1-12. Tampilkan nama bulan sesuai dengan inputan pengguna. Namanama bulan disimpan dalam array secara berurutan.