



Array 1

Array 1 Dimensi

Team Teaching Dasar Pemrograman
2025



Tujuan:

Di akhir pertemuan, mahasiswa diharapkan mampu :

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



Outline

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



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 matriks dituliskan dengan menggunakan variabel berindeks.
- Misalkan sebuah matriks $A[5,5]$ berdimensi 5×5 akan mempunyai elemen matriks berupa: a_{00} s.d a_{44} .
- Dalam pemrograman komputer, implementasi dari variabel berindeks menggunakan array.
- 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 **indeks 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

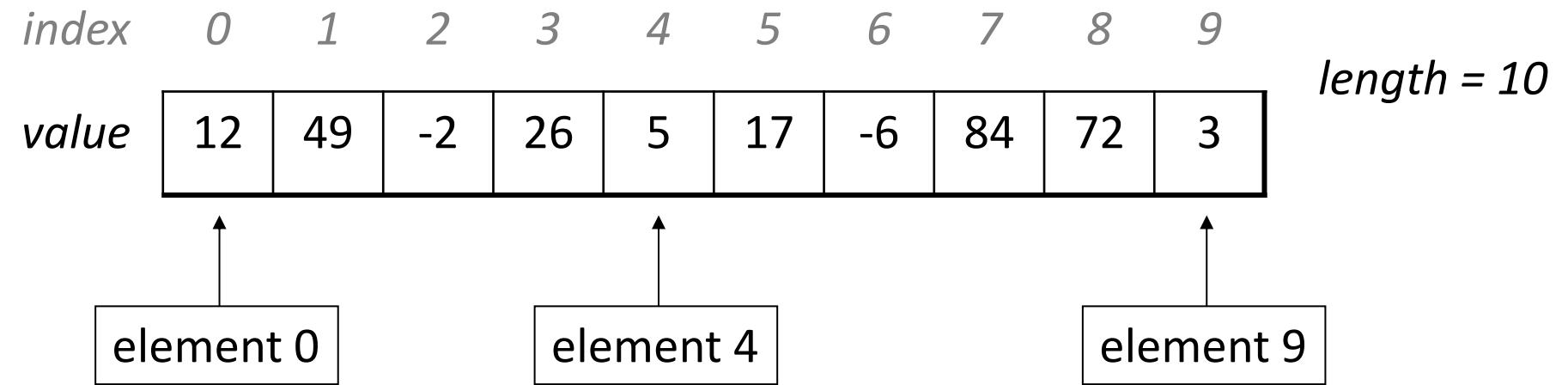
- Misalkan sebuah array bernama `a` dengan jumlah elemen sebanyak 10 elemen, maka elemen-elemen array tersebut dapat digambarkan sbb.:



- 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 namaArray[];
```

atau

```
tipeData[] namaArray;
```

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:

```
tipeData[] namaArray = new tipeData[jumlah_elelen];
```

atau

```
tipeData namaArray[] = new tipeData[jumlah_elelen];
```

- Contoh :

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

atau

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

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 1:

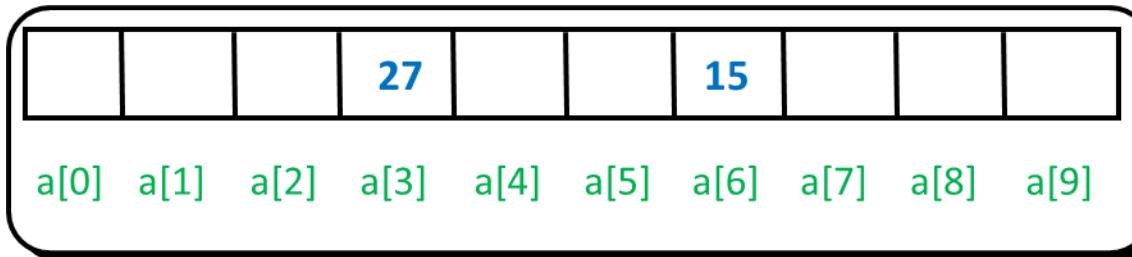
```
String[] cars = {"Volvo", "BMW", "Ford";  
System.out.println(cars[0]); //menampilkan Volvo  
System.out.println(cars[2]); //menampilkan Ford
```

- Contoh 2:

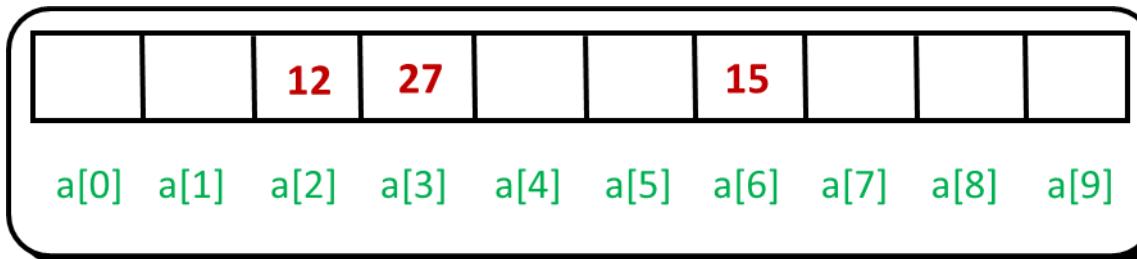
```
int[] nilai = {80, 90, 65, 83, 70};  
System.out.println(nilai[0]); //menampilkan 80  
System.out.println(nilai[4]); //menampilkan 70
```

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.
- Untuk mendeklarasikan dan menginisialisasi array dalam satu langkah, digunakan sintaks berikut:

```
tipeData[] namaArray = {nilai1, nilai2, nilai3, ...};  
atau
```

```
tipeData namaArray[] = {nilai1, nilai2, nilai3, ...};
```

Inisialisasi Array

- Contoh: `int b[]={1, 2, -4, 8};`
- Pada contoh di atas Array memiliki 4 elemen

1	2	-4	8
b[0]	b[1]	b[2]	b[3]

- Contoh: `int b[]={1,2,-4,8,0,0,0,0};`
- Pada contoh di atas Array memiliki 8 elemen

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?

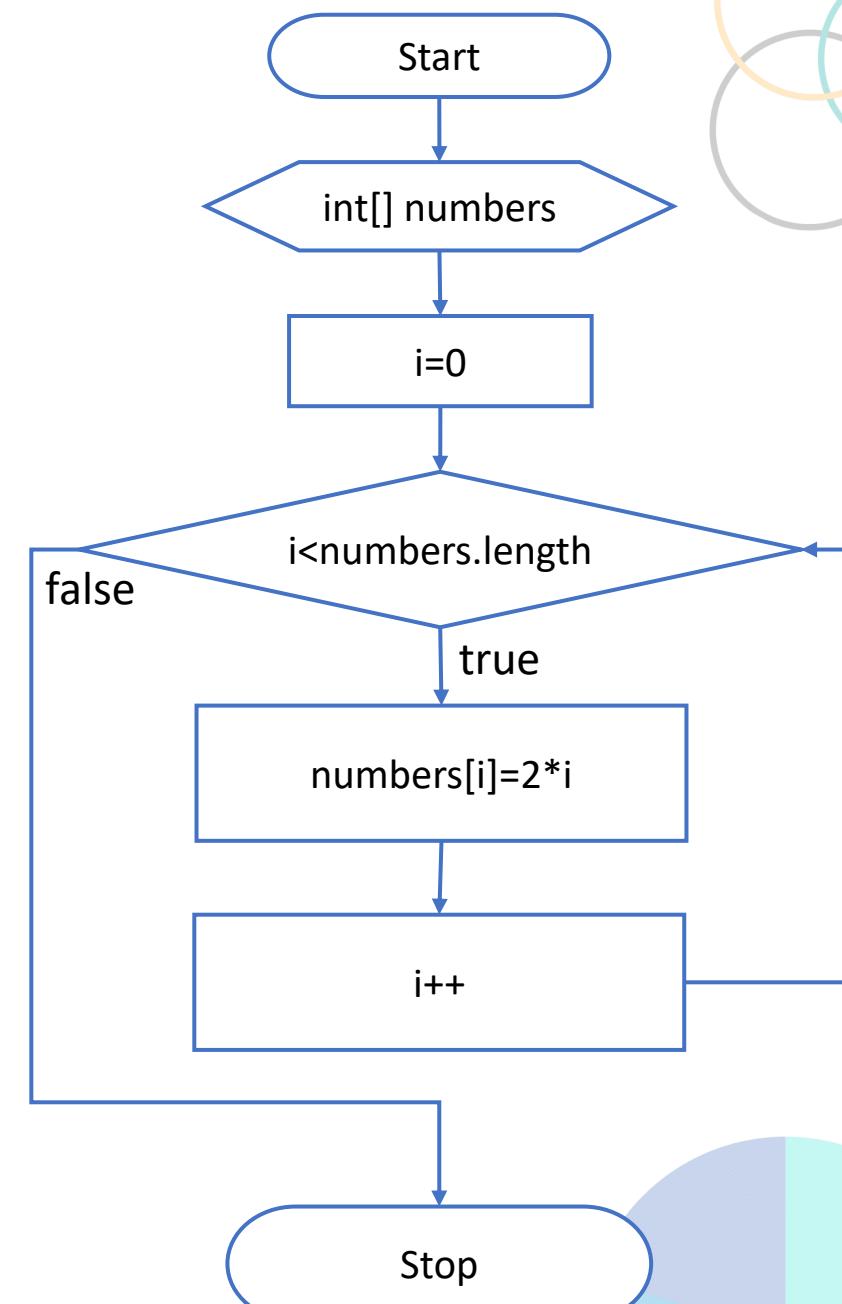
Array - Loop

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

```
int[] numbers = new int[8];
for (int i = 0; i < numbers.length; i++) {
    numbers[i] = 2 * i;
}
```

index 0 1 2 3 4 5 6 7

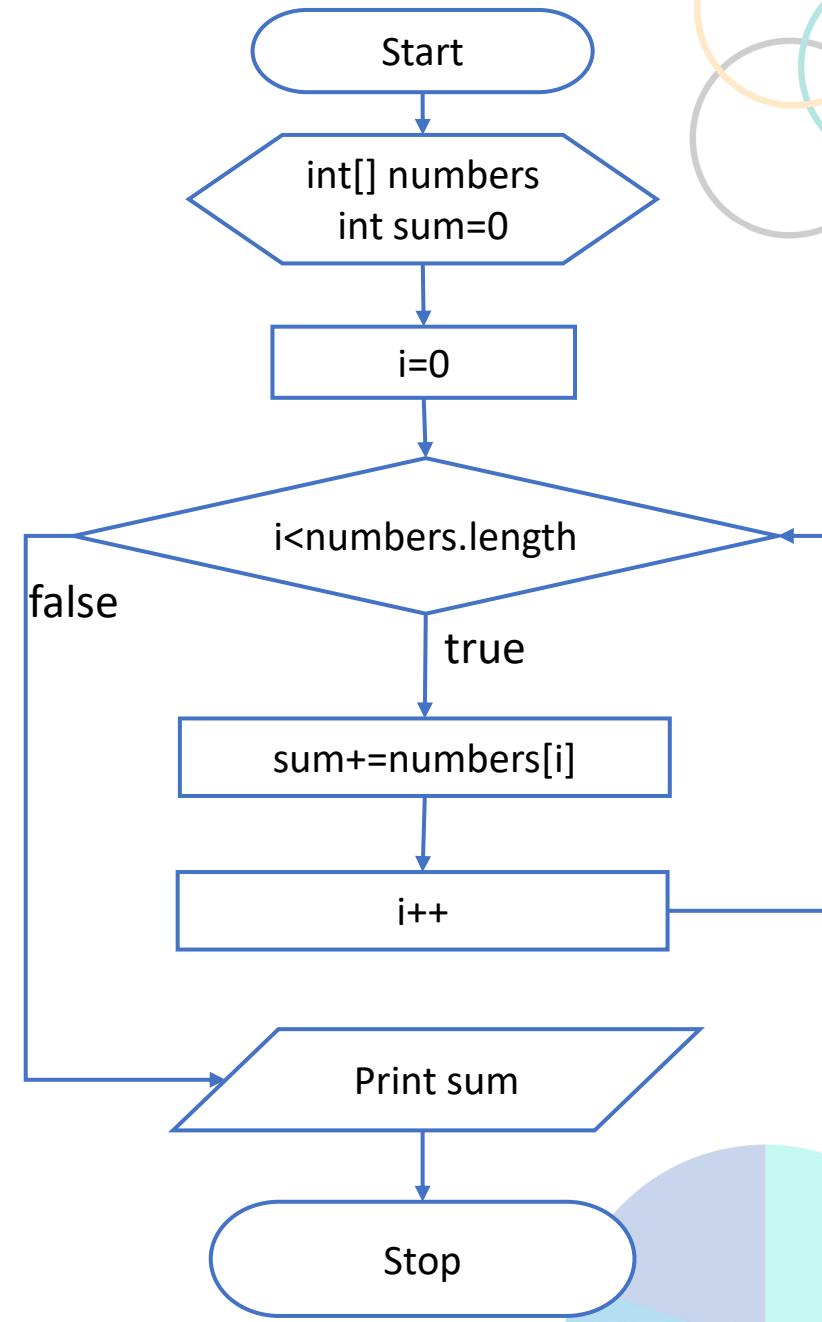
value	0	2	4	6	8	10	12	14
-------	---	---	---	---	---	----	----	----



Array-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];
}
System.out.println(sum);
```





Array - Loop

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 - Loop

- Mengakses semua elemen array dengan menggunakan perulangan "for-each".
- Contoh:

```
public static void main(String[] args) {  
  
    int bilangan[] = {31, 37, 41, 43, 47, 53, 59, 61, 67};  
    //deklarasi dan inisialisasi array  
  
    for (int i : bilangan) {  
        System.out.println(i);  
        //menampilkan setiap elemen array bilangan  
    }  
}
```

Output:

```
31  
37  
41  
43  
47  
53  
59  
61  
67
```

Array - For Each

Contoh:

```
public static void main(String[] args) {  
  
    //membuat array motor  
    String[] motor = new String[6];  
  
    Scanner sc = new Scanner(System.in);  
  
    //mengisi data ke dalam array  
    for (int i = 0; i < motor.length; i++) {  
        System.out.print("Masukkan nama motor ke-"+i+": ");  
        motor[i] = sc.nextLine();  
    }  
    System.out.println(x:-----);  
    System.out.println(x:"Nama motor yang sudah dimasukkan");  
  
    //menampilkan isi array  
    for (String b : motor){  
        System.out.println(b);  
    }  
}
```

- **Output:**

Masukkan nama motor ke-0: Yamaha Mio
Masukkan nama motor ke-1: Honda Beat
Masukkan nama motor ke-2: Suzuki Satria
Masukkan nama motor ke-3: Honda Scoopy
Masukkan nama motor ke-4: Kawasaki Ninja
Masukkan nama motor ke-5: Yamaha Nmax

Nama motor yang sudah dimasukkan:
Yamaha Mio
Honda Beat
Suzuki Satria
Honda Scoopy
Kawasaki Ninja
Yamaha Nmax

Perbedaan dengan atau tanpa ARRAY

```
int number1 = 1;  
int number2 = 2;          << tanpa array  
int number3 = 3;  
  
int number[] = {1, 2, 3}; << dengan 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 };`
- Contoh inisialisasi array setelah didefinisikan yang salah :
`int b[5];`
`b[5]={0,0,0,0,0};`

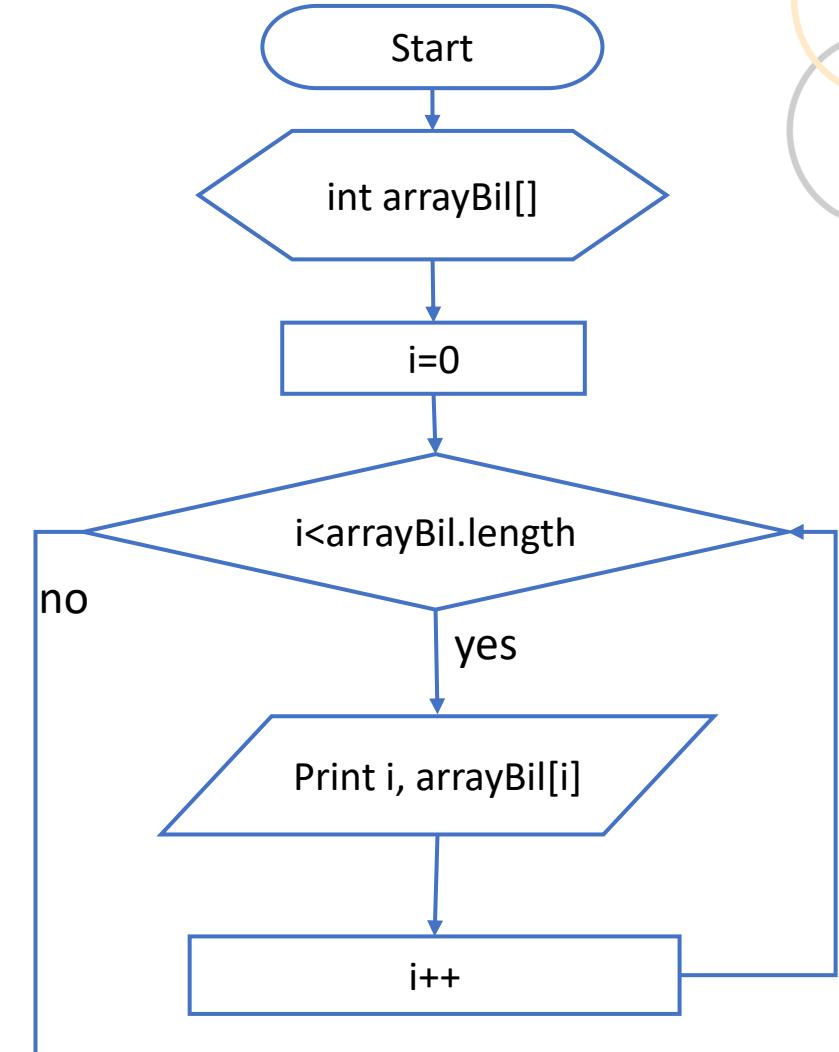
Contoh

```
public class ArrayBil1{
    Run | Debug
    public static void main(String[] args) {
        int arrayBil[];          //deklarasi array
        arrayBil = new int[10];   //instansiasi array

        System.out.printf(format:"%s%6s\n",...args:"Indeks","Value");
        //menampilkan array
        for (int i = 0; i < arrayBil.length; i++){
            System.out.printf(format:"%2d%6d\n",i,arrayBil[i]);
        }
    }
}
```

Output:

Indeks	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

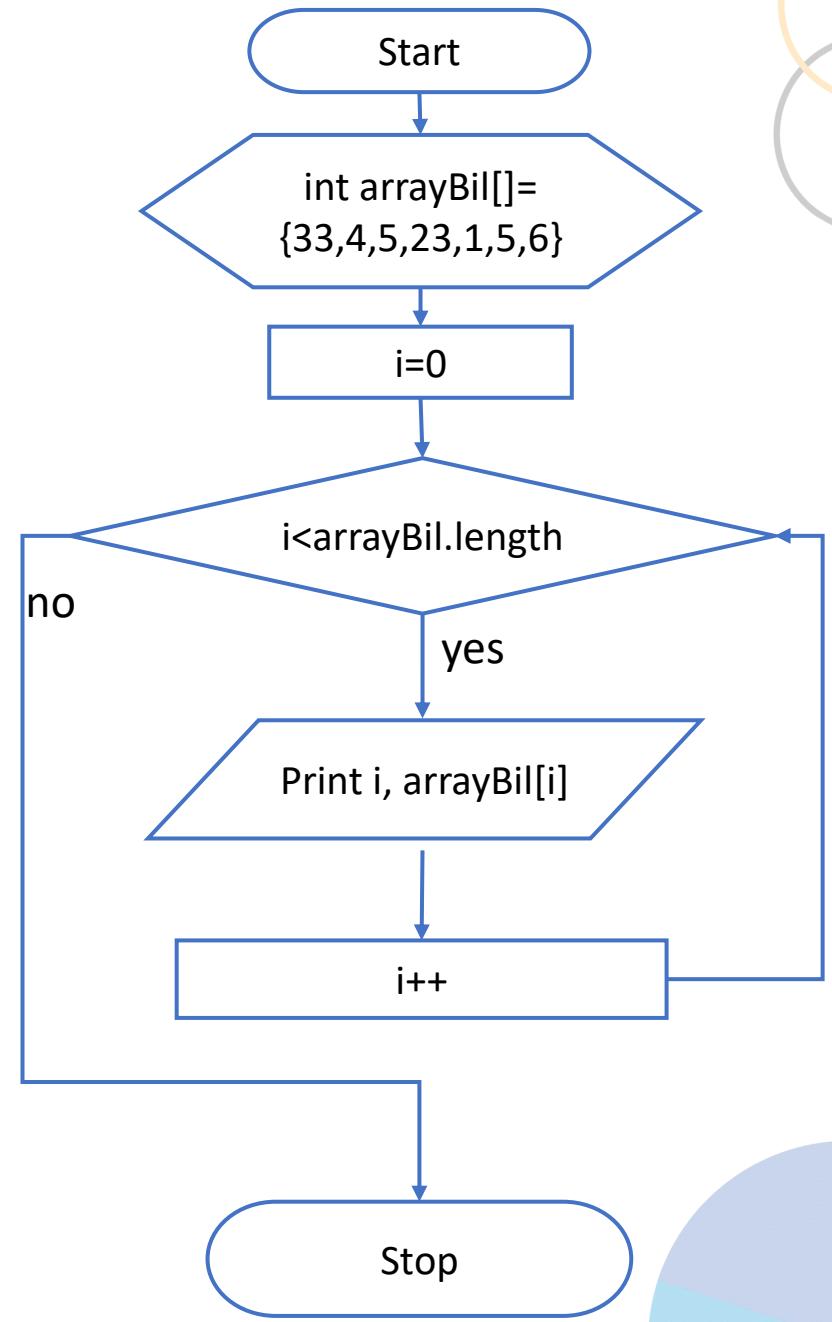


Contoh

```
public class ArrayBil2{  
    Run | Debug  
    public static void main(String[] args) {  
        int arrayBil[] = {33,4,5,23,1,5,6};  
        //inisialisasi array: menentukan jumlah elemen array dan  
        //menentukan nilai dari setiap elemennya  
  
        System.out.printf(format:"%s%6s\n",...args:"Indeks","Value");  
        //menampilkan array  
        for (int i = 0; i < arrayBil.length; i++){  
            System.out.printf(format:"%2d%6d\n",i,arrayBil[i]);  
        }  
    }  
}
```

Output:

Indeks	Value
0	33
1	4
2	5
3	23
4	1
5	5
6	6



Contoh

- Program meminta input sebanyak 5 bilangan kemudian menampilkan kembali 5 bilangan tersebut.

```
import java.util.Scanner;

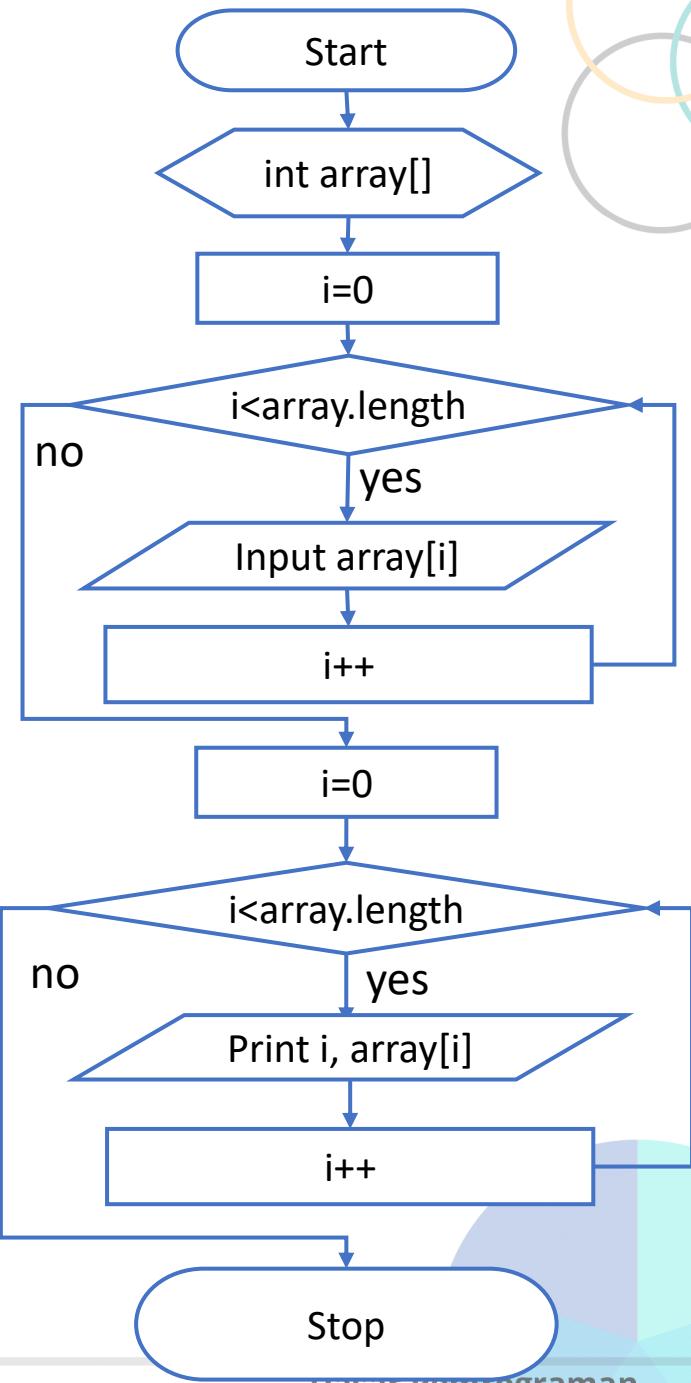
public class ArrayInput {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int array[];
        array = new int[5];

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

        for (int i = 0; i < array.length; i++){
            System.out.println("Array "+i+" dengan elemen "+array[i]);
        }
    }
}
```

Output:

```
Masukkan angka: 2
Masukkan angka: 5
Masukkan angka: -3
Masukkan angka: 12
Masukkan angka: 9
Array 0 dengan elemen 2
Array 1 dengan elemen 5
Array 2 dengan elemen -3
Array 3 dengan elemen 12
Array 4 dengan elemen 9
```



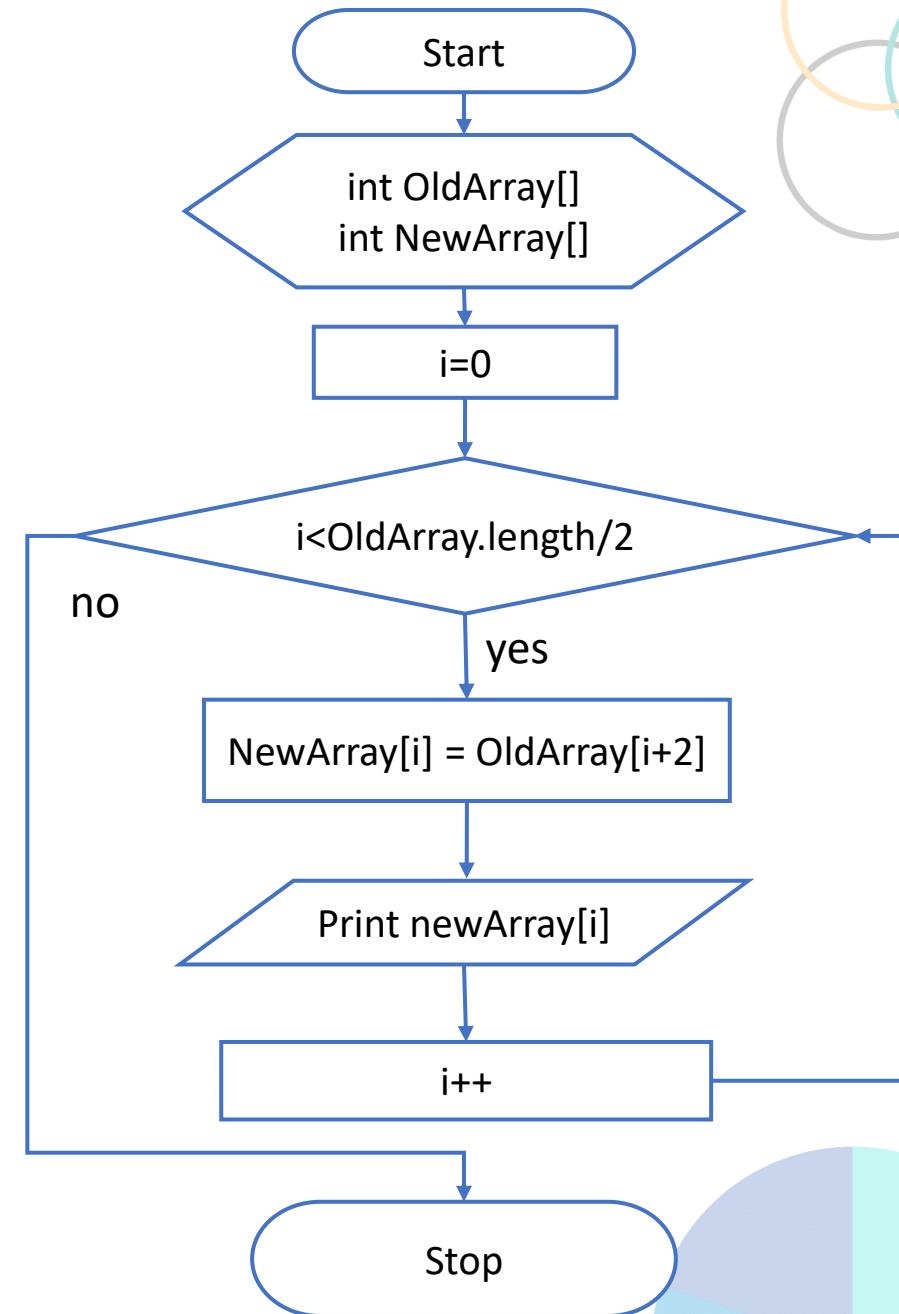
Contoh

- Membuat salinan isi array

```
public class SalinArray {  
    Run | Debug  
    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.println(NewArray[i]);  
        }  
    }  
}
```

Output:

6
7



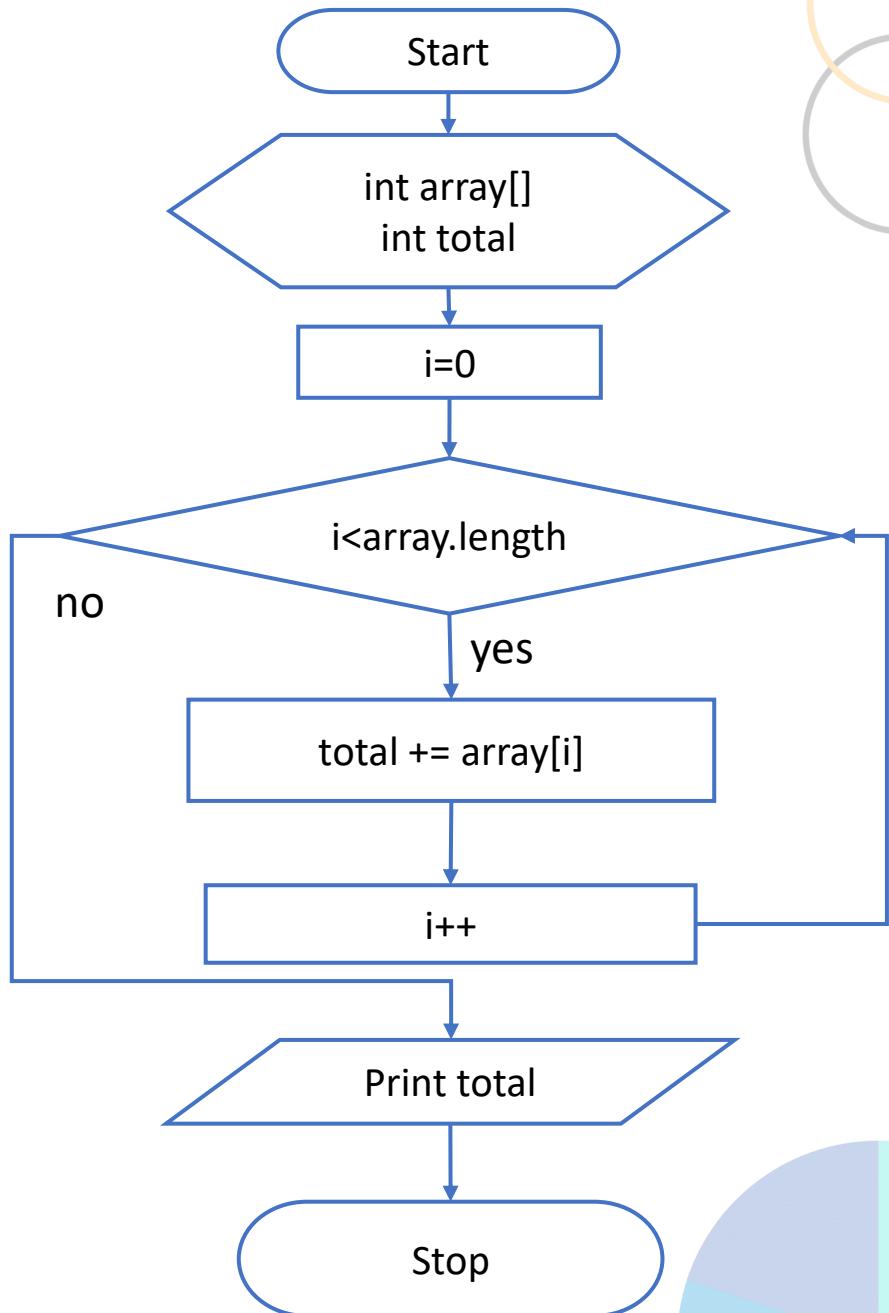
Contoh

- Penjumlahan Array

```
public class PenjumlahanArray {  
    Run | Debug  
    public static void main(String[] args) {  
        int[] array = {5,7,13,17,29};  
        int total = 0;  
  
        for(int i = 0; i < array.length; i++){  
            total += array[i];  
        }  
        System.out.println("Hasil penjumlahan isi array = "+total);  
    }  
}
```

Output:

Hasil penjumlahan isi array = 71





MATERI PENGAYAAN

Searching

Searching

- Salah satu hal yang sering dilakukan pada operasi array adalah pencarian atau *searching*
- Pencarian dilakukan untuk menemukan nilai tertentu pada elemen di dalam array
- Salah satu algoritma searching yang paling mudah adalah *Linear 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 di dalam 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

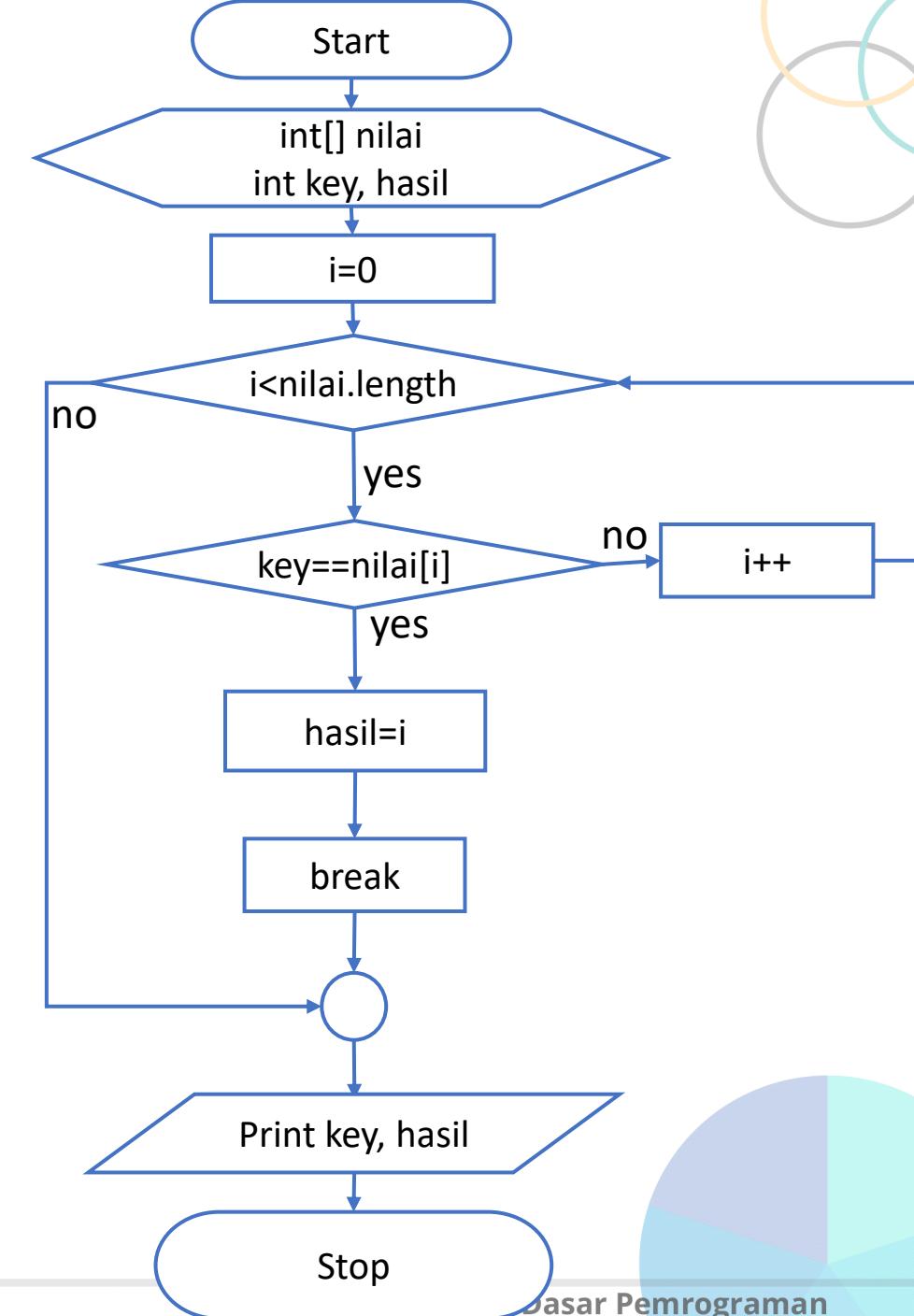


Searching

```
public static void main(String[] args) {  
    int[] nilai = {80, 85, 78, 96, 90, 82, 86};  
    int key = 96;  
    int hasil = 0;  
  
    for (int i = 0; i < nilai.length; i++) {  
        if (key == nilai [i]) {  
            hasil = i;  
            break;  
        }  
    }  
    System.out.println("Key "+key+" ketemu di indeks ke - "+hasil);  
}
```

- Output:

Key 96 ketemu di indeks ke - 3





LATIHAN

1. Buat *flowchart* pengisian elemen array dengan jumlah elemen 30 menggunakan looping, kemudian menampilkan isi array tersebut.
2. Buat *flowchart* untuk mengisi elemen array dengan jumlah elemen 10, kemudian menampilkan isi array tersebut secara terbalik.
3. Buat *flowchart* yang meminta inputan pengguna berupa angka 1-12. Tampilkan nama bulan sesuai dengan inputan pengguna tersebut. Nama-nama bulan disimpan dalam array secara berurutan.
4. Buat *flowchart* untuk mengisi array nilai berupa bilangan bulat sejumlah 10 nilai, kemudian menghitung rata-rata dari seluruh nilai tersebut.
5. Buat *flowchart* untuk membuat sebuah array bilangan bulat sejumlah 10 elemen, lalu menerima input sebuah nilai integer (misal n), periksa isi array, dan cetak ada di lokasi mana saja isi array yang nilainya sama dengan n.