



# ARRAY 1

PEMROGRAMAN DASAR (Minggu ke-9)

## 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
- Latihan

# 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  $5 \times 5$  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

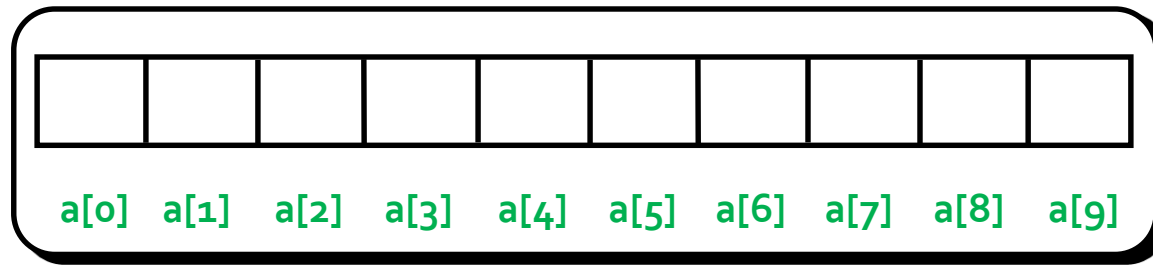
- Array merupakan variabel kompleks dengan tipe data yang sama, menggunakan nama yang sama, dan memiliki suatu index tertentu.
- Atau merupakan sekumpulan nilai (elemen) dengan tipe data yang sama. Dimana masing-masing elemen Array bisa diakses dengan menggunakan indeks yang unik

# 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 jumlahElemen-1

# VISUALISASI ARRAY

*length = 10*

<i>index</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<i>value</i>	12	49	-2	26	5	17	-6	84	72	3

element 0      element 4      element 9



## 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_elemen];
```

atau

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

- 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 :

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

## Mengisi Data pada Array

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

			27			15			
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

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

		12	27			15			
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

# 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

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

- 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";`
- `System.out.println(cars[0]);`

## Mendapatkan Panjang Array

- Anda bisa mendapatkan panjang array dengan menggunakan

**`namaArray.length`**

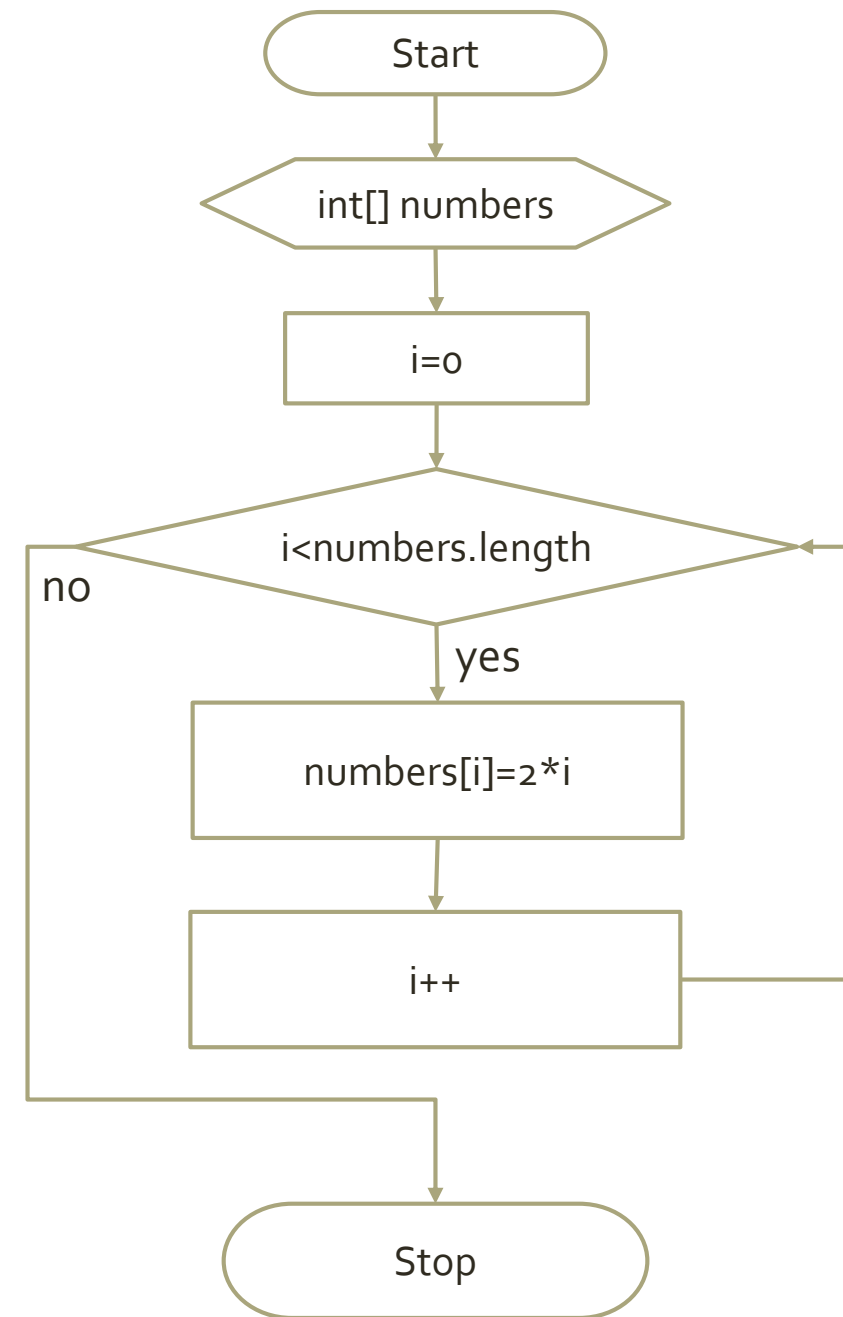
- Contoh penggunaan Panjang array:
  - Berapa indeks dari elemen terakhir dari sebuah array?
  - Berapa indeks dari elemen tengah dari sebuah array?

# 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;  
    }
```

<i>index</i>	0	1	2	3	4	5	6	7
<i>value</i>	0	2	4	6	8	10	12	14



# Array ♥ loop (contoh)

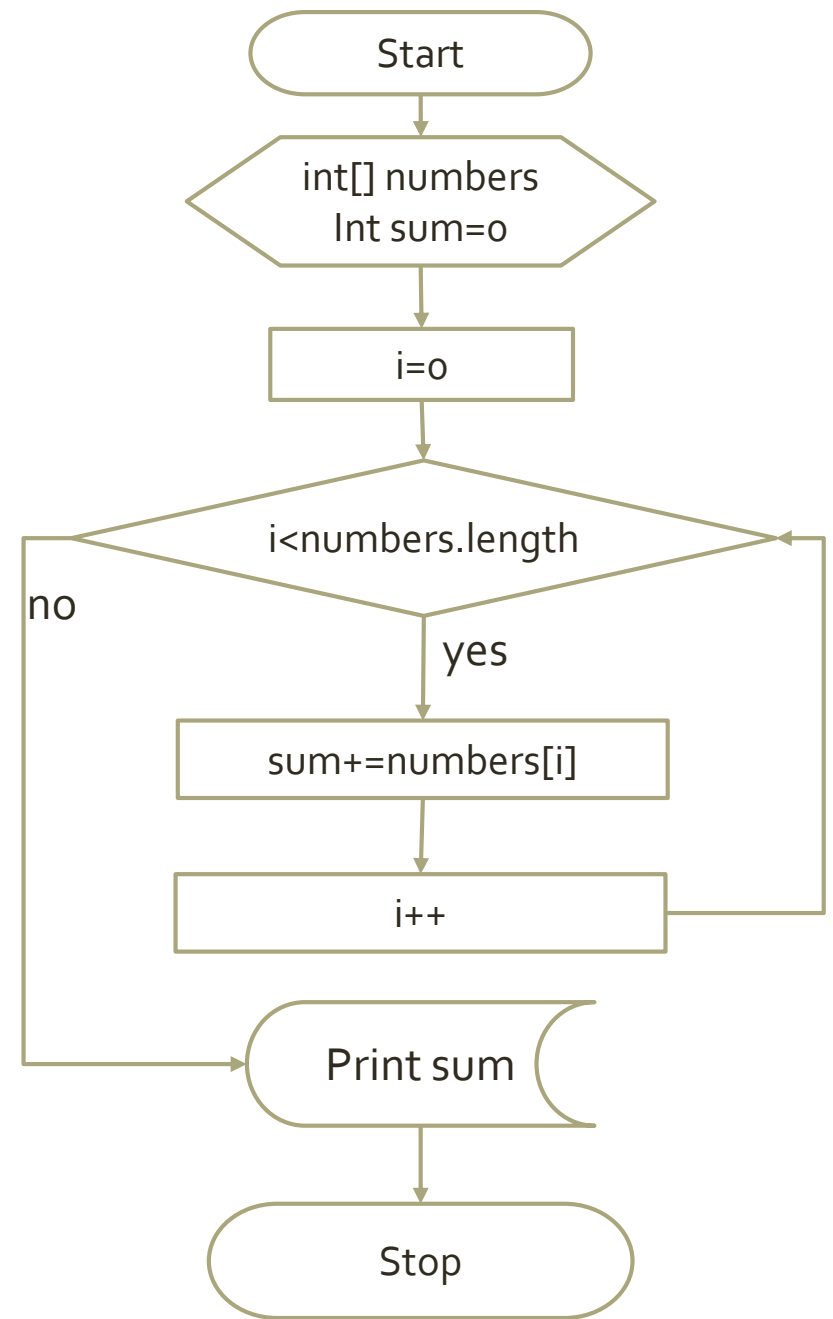
## Menjumlahkan semua elemen array

// assume that the user has created `int[] numbers`

```
int sum = 0;
```

```
for (int i = 0; i < numbers.length; i++) {  
    sum += numbers[i];  
}
```

```
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 (Contoh)

Mengakses Semua elemen array dengan menggunakan perulangan "**for-each**"

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

-----  
33  
4  
5  
23  
1  
5  
6  
BUILD SUCCESSFUL  
|

## Perbedaan dengan atau tanpa ARRAY

```
int number1 = 1;  
int number2 = 2;    << not with array  
int number3 = 3;
```

```
int number[] = {1, 2, 3}; << with array
```

## Penggunaan Array

1. Deklarasikan variabel referensi array
2. Instansiasi elemen array
3. Inisialisasi array (*Jika diperlukan*)
4. Manipulasi 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};
```

# Contoh

```
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	0
---	---

1	0
---	---

2	0
---	---

3	0
---	---

4	0
---	---

5	0
---	---

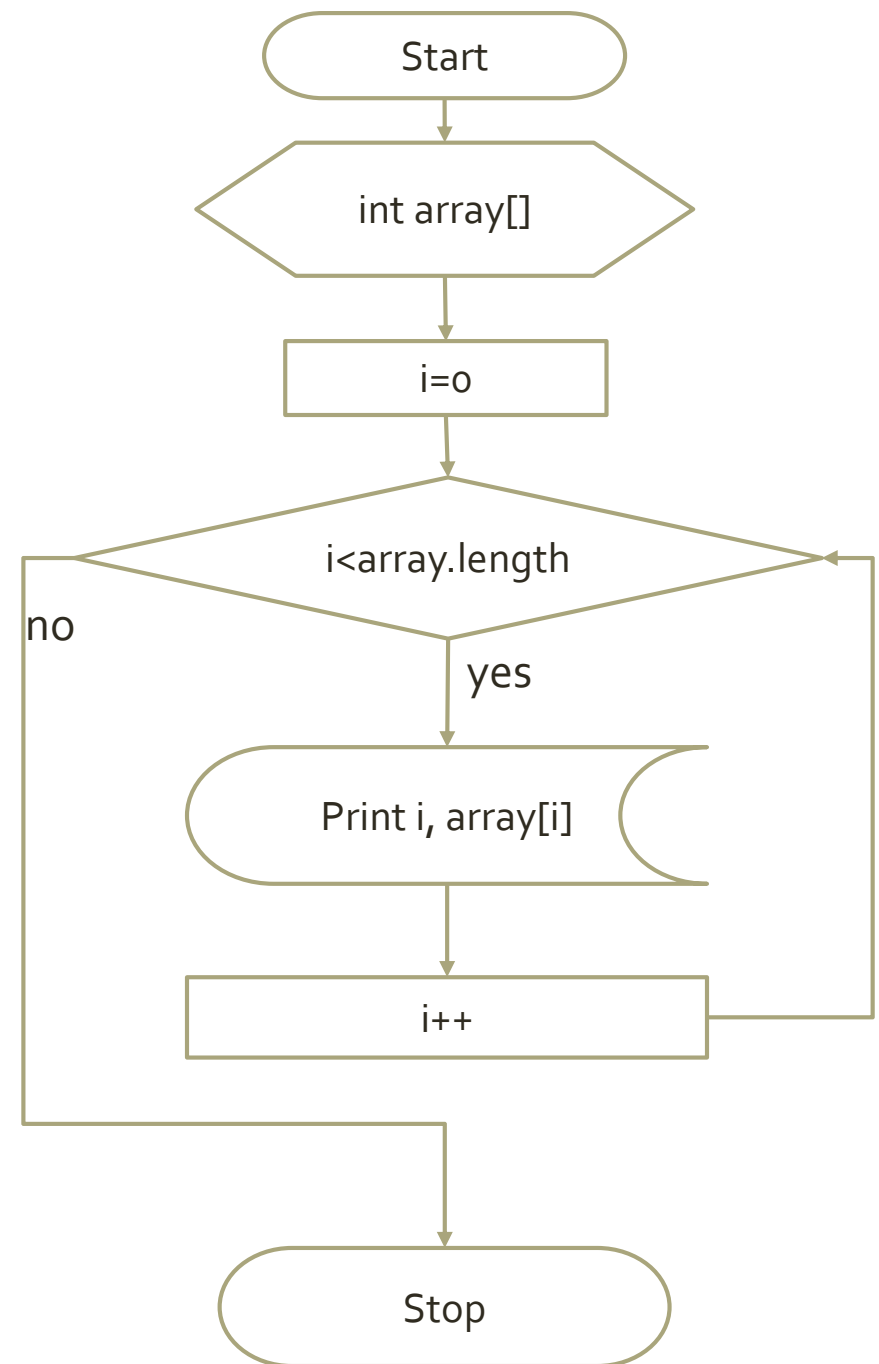
6	0
---	---

7	0
---	---

8	0
---	---

9	0
---	---

BUILD SUCCESSFUL



# Contoh

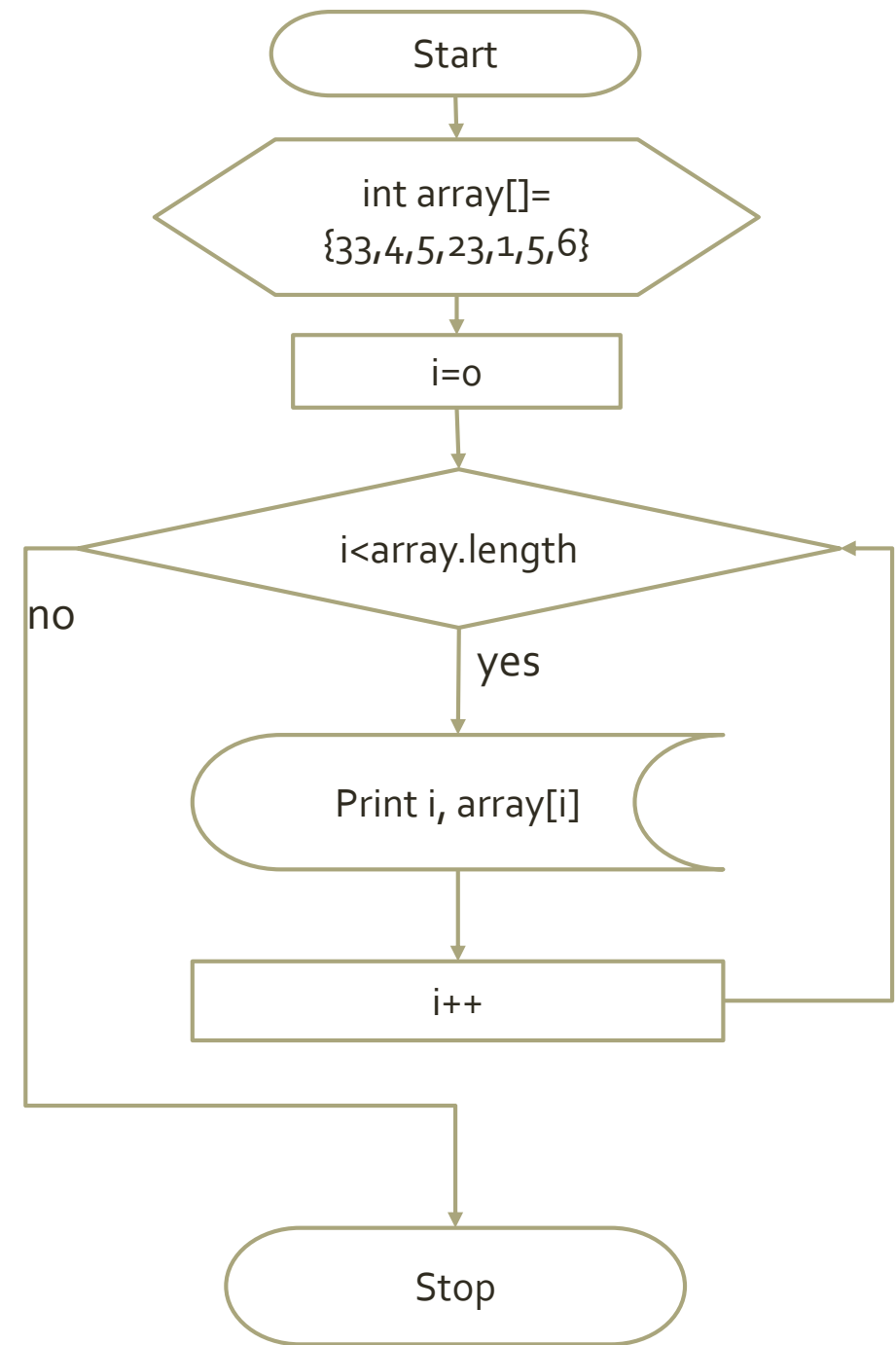
```
public static void main(String[] args) {  
  
    int array[] = {33,4,5,23,1,5,6};  
    //inisialisasi array = menentukan jumlah elemennya  
    //serta menentukan nilai dari setiap elemen  
  
    System.out.printf("%s%5s\n","Index ","Value");  
  
    for (int i=0; i<array.length;i++)  
    {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

run-single:

Index Value

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

BUILD SUCCESSFUL |

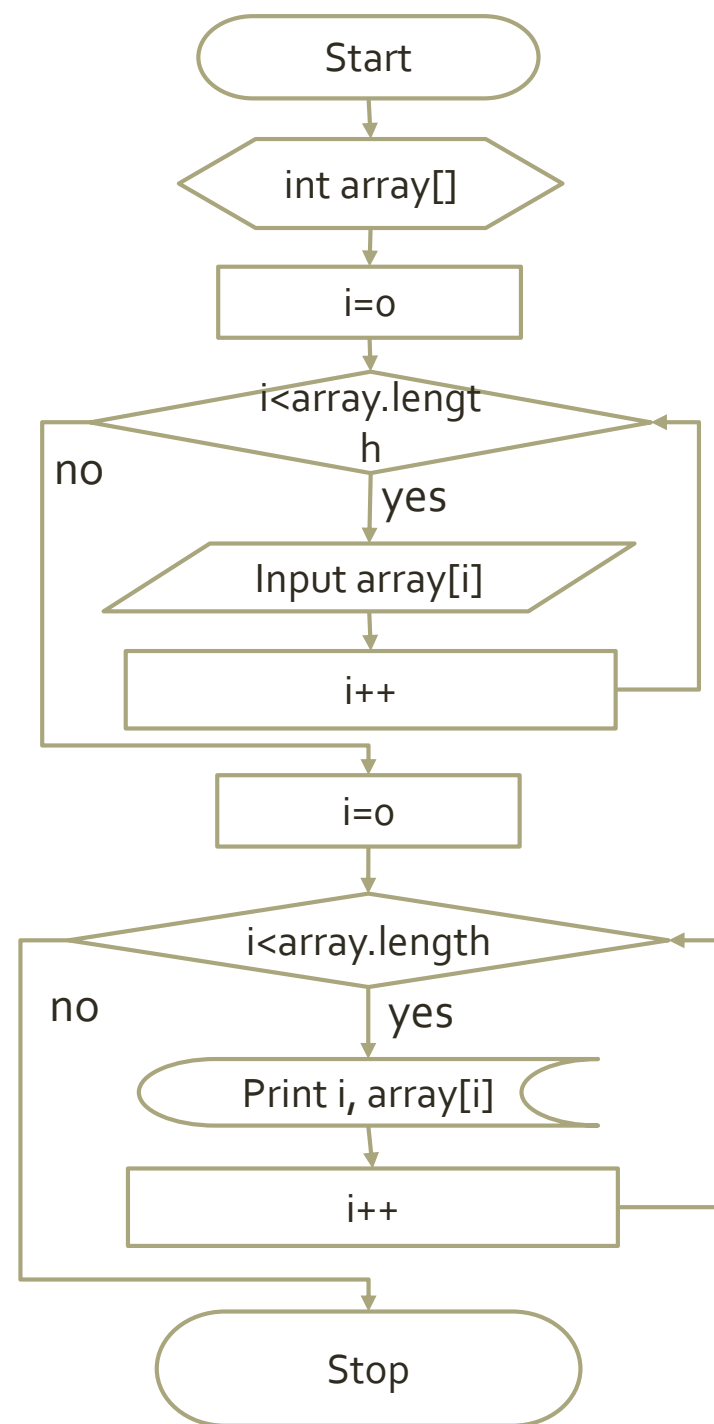


# Contoh

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

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

```
run-single:  
masukkan angka :  
4  
masukkan angka :  
11  
masukkan angka :  
56  
masukkan angka :  
2  
masukkan angka :  
7  
array 0 dengan elemen    4  
array 1 dengan elemen   11  
array 2 dengan elemen   56  
array 3 dengan elemen    2  
array 4 dengan elemen    7  
BUILD SUCCESSFUL (total time: 8 seconds)
```



# TUGAS

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. Nama-nama bulan disimpan dalam array secara berurutan.