



Array 1

Array 1 Dimensi

Team Teaching Dasar Pemrograman 2023



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 matriks 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 indeks tertentu.
- Atau merupakan sekumpulan nilai (elemen) dengan tipe data yang sama. Dimana masing-masing elemen Array bisa diakses dengan menggunakan indeks yang unik



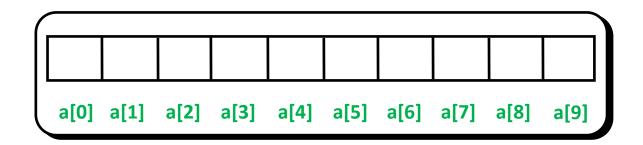
Sifat Array

- 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.:

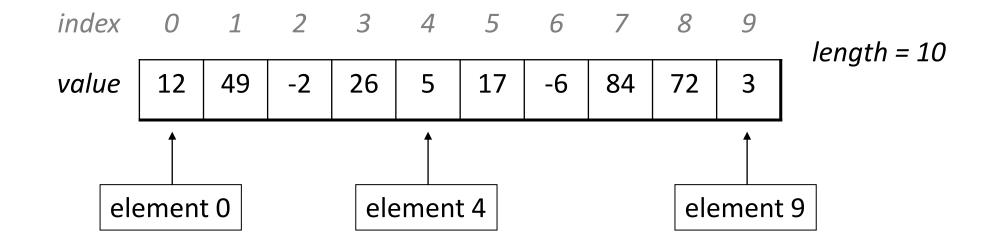


- Kotak kosong menunjukkan elemen dari Array
- Masing- masing elemen memiliki penomoran 0-9 (indeks)
- Indeks array dimulai dari 0 dan diakhiri dengan jumlah Elemen-1



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.:



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

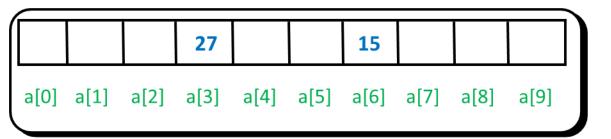
• Contoh:

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

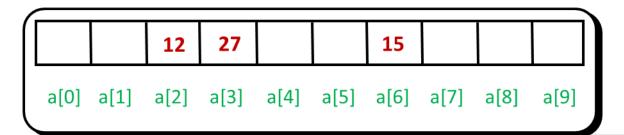


Mengisi Data pada Array

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



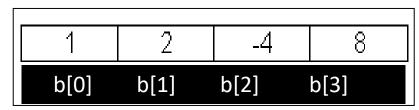
• statement a[2] = a[3] - a[6]; menghasilkan:





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 di atas Array memiliki 4 elemen



• 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?



index

value

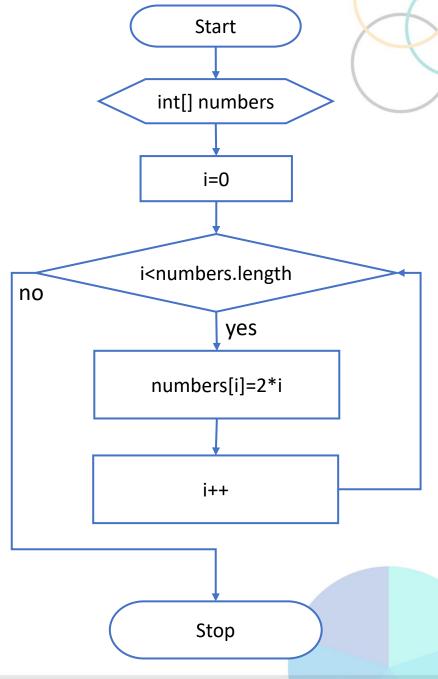
Array - Loop

- 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;
    }

0     1    2    3    4    5    6    7

0     2     4     6     8     10    12    14</pre>
```

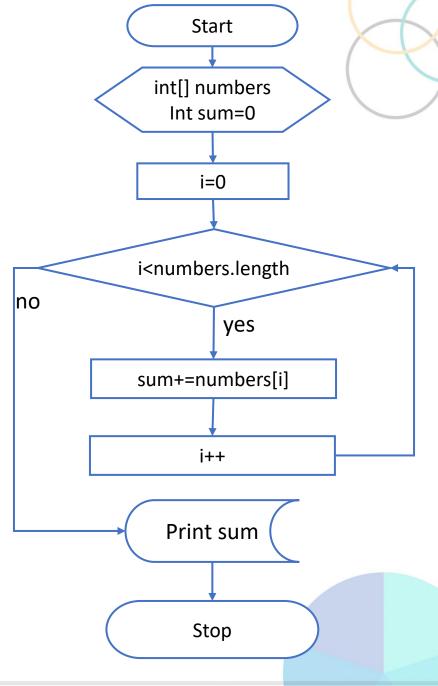




Array-Loop(contoh)

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);</pre>
```





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 array[] = {33,4,5,23,1,5,6};
  //inisisaliasai array = menentukan jumlah elemennya
  //serta menentukan nilai dari setiap elemen

for (int i : array) {
    System.out.println(i);
} // menampilkan setiap elemen array
}
```

```
4
5
23
1
5
6
BUILD SUCCESSFU
```



Array - For Each

Contoh:

```
import java.util.Scanner;
public class Motor {
   public static void main(String[] args) {
       // membuat array motor
       String[] motor = new String[5];
       // membuat scanner
       Scanner scan = new Scanner(System.in);
       // mengisi data ke array
       for( int i = 0; i < motor.length; i++ ){</pre>
           System.out.print("Motor ke-" + i + ": ");
           motor[i] = scan.nextLine();
       System.out.println("-----");
       // menampilkan semua isi array
       for( String b : motor ) {
           System.out.println(b);
```

• Output:

```
Motor ke-0: Yamaha
Motor ke-1: Honda
Motor ke-2: Suzuki
Motor ke-3: Vario
Motor ke-4: Beat
------
Yamaha
Honda
Suzuki
Vario
Beat
```





Perbedaan 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

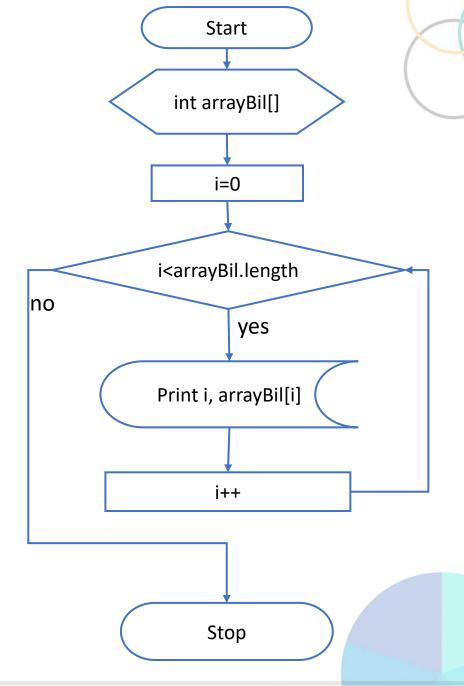
- Contoh: int $b[4] = \{ 1, 2, -4, 8, 9 \};$
 - ERROR karena nilai dimensi lebih kecil dari jumlah elemen.

• Contoh inisialisasi array setelah didefinisikan yang salah:

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



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





```
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]);
        }
    }
}</pre>
```

```
Indeks Value

0 33

1 4

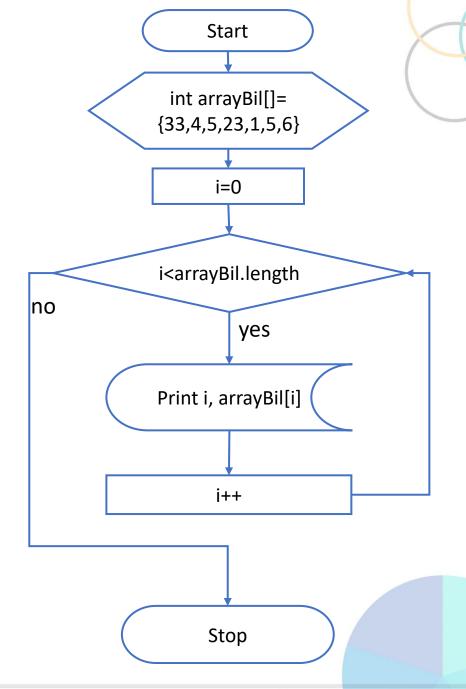
2 5

3 23

4 1

5 5

6 6
```





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

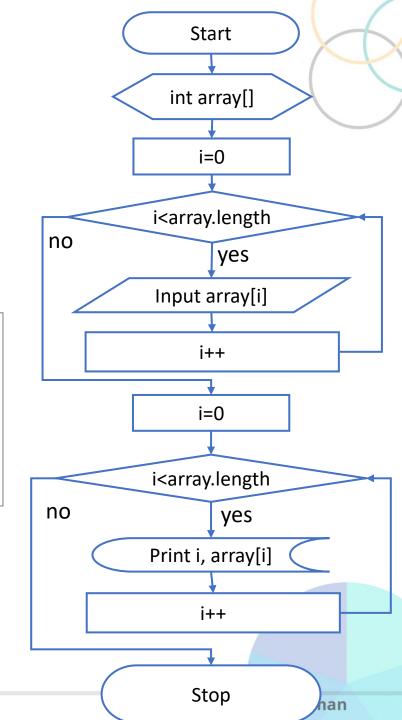
```
import java.util.Scanner;

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(s:"Masukkan angka: ");
            array[i] = sc.nextInt();
        }

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

```
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
```





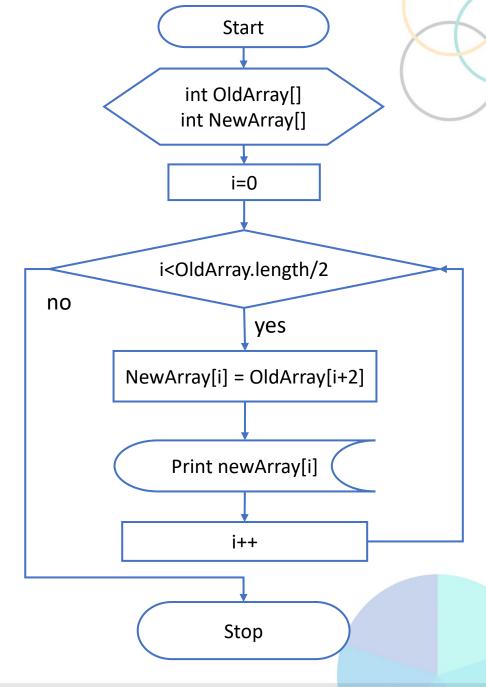
Membuat salinan isi array

```
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]);
      }
   }
}</pre>
```

Output:

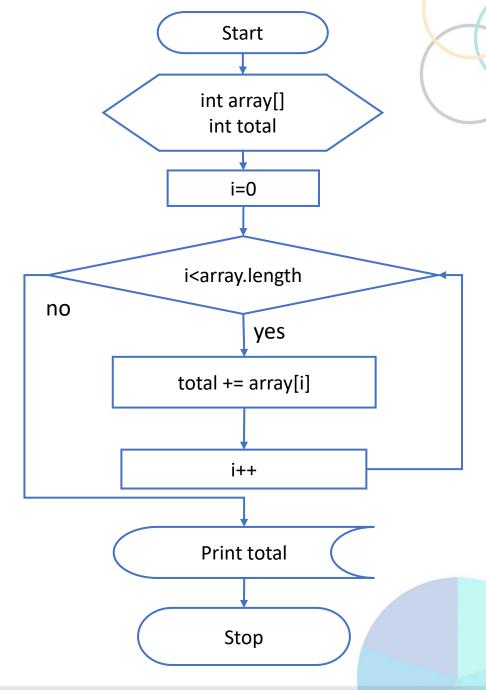
6 7





Penjumlahan Array

```
Hasil penjumlahan isi array = 71
```







MATERI PENGAYAAN

Searching dan Sorting





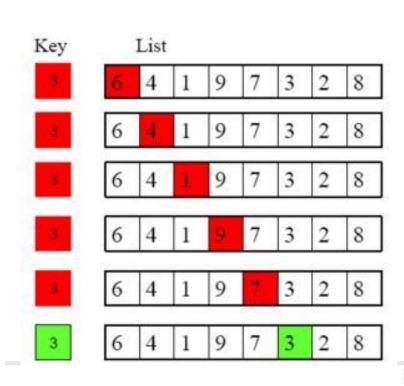
Searching

- Salah satu hal yang sering dilakukan pada operasi array adalah pencarian atau searching
- Pencarian dilakukan untuk menemukan nilai tertentu pada elemen 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

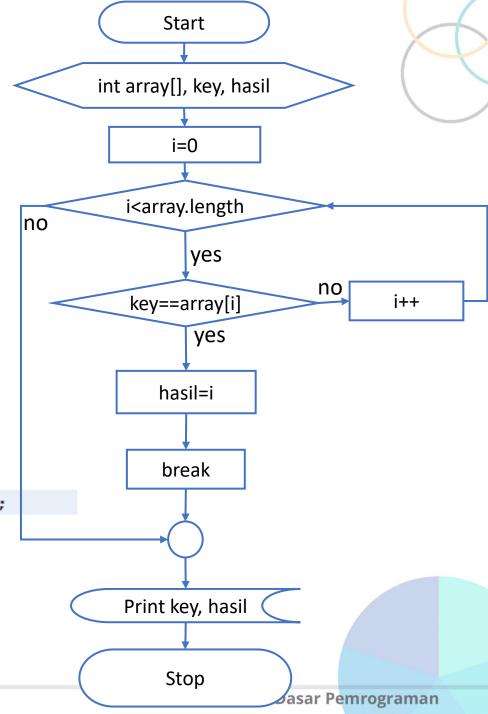




Searching

```
public static void main(String[] args) {
    int 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)
```





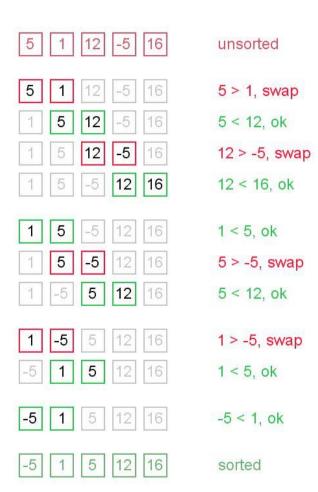
Sorting

- 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

- Di dalam 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.

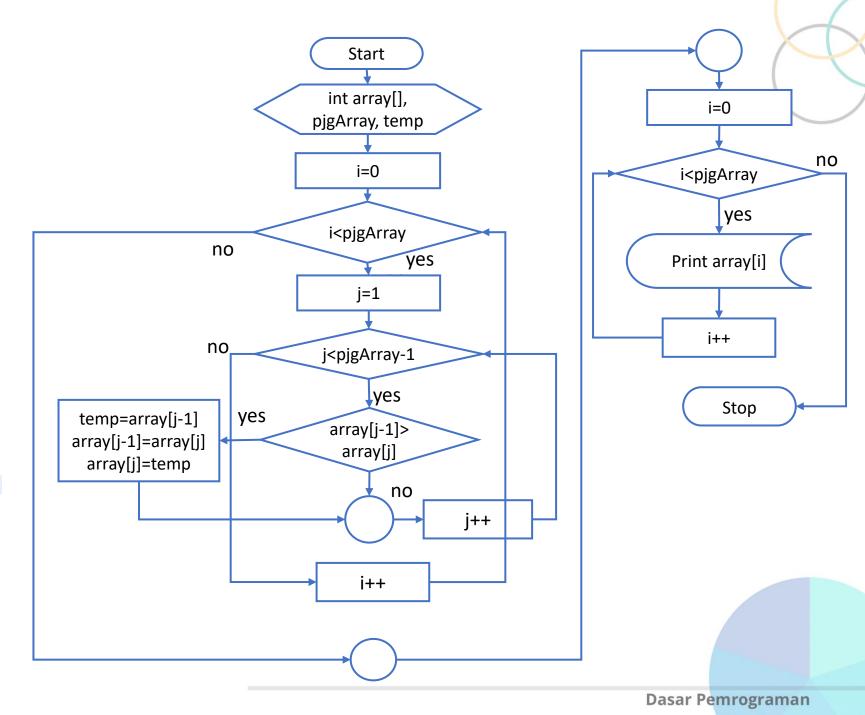






Bubble Sort

```
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++)
            for (int j=1; j<pjgArray-1;j++)
                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
            3 4 6 7 9 8 BUILD
```





LATIHAN INDIVIDU



- 1. Buat *flowchart* pengisian elemen array dengan jumlah elemen 60 menggunakan looping!
- 2. Buat *flowchart* untuk mengisi elemen array dengan jumlah elemen 10, kemudian tampilkan 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 bilangan bulat sejumlah 8 elemen, kemudian menghitung rata-rata dari seluruh elemen array tersebut.



TUGAS KELOMPOK

- 1. Identifikasi sesuai project masing-masing kelompok, fitur apa saja yang membutuhkan penggunaan array 1 dimensi.
- 2. Identifikasi sesuai project tersebut, fitur apa saja yang membutuhkan operasi array 1 dimensi berupa *searching* dan *sorting*.
- 3. Buatlah algoritma dalam bentuk flowchart sesuai kebutuhan yang telah Anda identifikasi berdasarkan tugas pada nomor 1 dan 2