

# ARRAY



ALGORITMA DAN PEMROGRAMAN  
**Bagian V**

Renny P. Kusumawardani, S.T., M.T., SCJP



## Pengenalan



## Deklarasi & Konstruksi



## Operasi Dasar



## Inisialisasi



## Operasi pada Array: Latihan Soal

# Contents

# PENGENALAN



DEKLARASI  
&  
KONSTRUKSI

OPERASI  
DASAR

INISIALISASI

OPERASI  
PADA  
ARRAY:  
LATIHAN  
SOAL

# Apa itu Array?

- Dalam bahasa pemrograman Java, array adalah suatu objek yang menyimpan nilai beberapa variabel dengan tipe data yang sama
- Array dapat dipergunakan untuk menyimpan data bertipe primitif maupun objek, namun array itu sendiri merupakan suatu objek
- Contoh:

```
int[] n = {0, 2, 3, 5, 9};  
String[] nama = {"Belajar", "Dasar-Dasar", "Pemrograman"}
```

# Array How Tos

- **Deklarasi**

Bagaimana membuat variabel referensi array

- **Konstruksi**

Bagaimana membuat objek array

- **Inisialisasi**

Bagaimana mengisi array dengan elemen-elemen (inisialisasi)

- **Akses dan Manipulasi Data**

Bagaimana mengakses anggota-anggota dari array dan melakukan operasi dengan/pada data-data pada array

Perhatikan video demo ini.  
Adakah praktek coding yang menurut anda kurang tepat?

```
*apples.java X tuna.java
1 class apples{
2     public static void main(String[] args) {
3
4     }
5 }
```

# Coding Example: Introduction to Arrays

# DEKLARASI & KONSTRUKSI



PENGENALAN



DEKLARASI  
&  
KONSTRUKSI



OPERASI  
DASAR



INISIALISASI



OPERASI  
PADA  
ARRAY:  
LATIHAN  
SOAL

# Deklarasi Array

- Array dideklarasikan dengan menyatakan
  1. Tipe data dari elemen array
  2. Nama array
  3. Kurung siku sebelum (recommended!) atau sesudah nama array
- Pada saat deklarasi array, kita belum menyertakan ukuran array  
→ space baru dialokasikan saat array diinstantiasi

# Contoh Deklarasi Array

Array berelemen primitif

```
int[] key;           // cara penulisan yang direkomendasikan  
int key[];
```

Array berelemen objek

```
Thread[] threads; // cara penulisan yang direkomendasikan  
Thread threads[];
```

Array multidimensi

```
String[][][] namaPenghuni;    // array tiga dimensi  
String[][] namaManajer;       // array dua dimensi
```

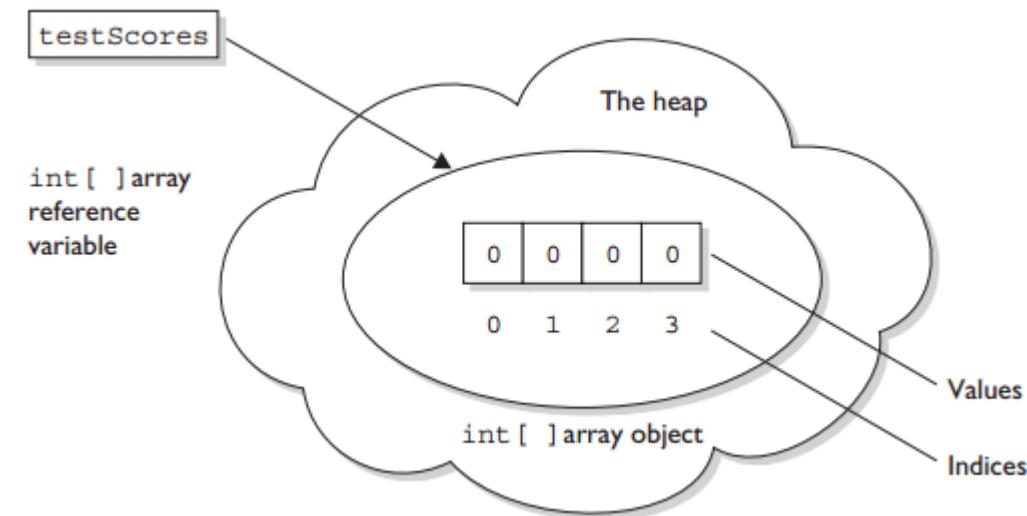
# Mengkonstruksi Array

- Mengkonstruksi array berarti membuat objek array pada heap (lokasi penyimpanan objek di memori)
- Setara dengan menggunakan kata kunci **new**
  - Menciptakan objek array baru
- Untuk melakukan konstruksi, Java perlu mengetahui berapa banyak space yang dibutuhkan pada heap
  - Anda perlu menyebutkan berapa ukuran array yang dibutuhkan
  - Ukuran array adalah jumlah elemen yang dapat disimpan oleh array (kapasitas array)

# Contoh Konstruksi Array

- Contoh: array satu dimensi

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



- Kedua pernyataan (deklarasi dan konstruksi) di atas dapat dinyatakan dalam satu baris sebagai berikut:

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

# Konstruksi Array Multidimensional

- Ingat:  
pada saat konstruksi, Java perlu diberi tahu berapa kapasitas array yang dibutuhkan  
Contoh: **String[] nama = new String[5];**



- Demikian pula untuk array multidimensional  
Contoh: array 2 dimensi  
**int[][] nilaiBarisKolom = new int[3][];**

# Konstruksi Array Multidimensional (2)

- Perhatikan, hanya kapasitas dari kurung siku terdepan yang perlu diberitahukan, yang belakang boleh kosong:

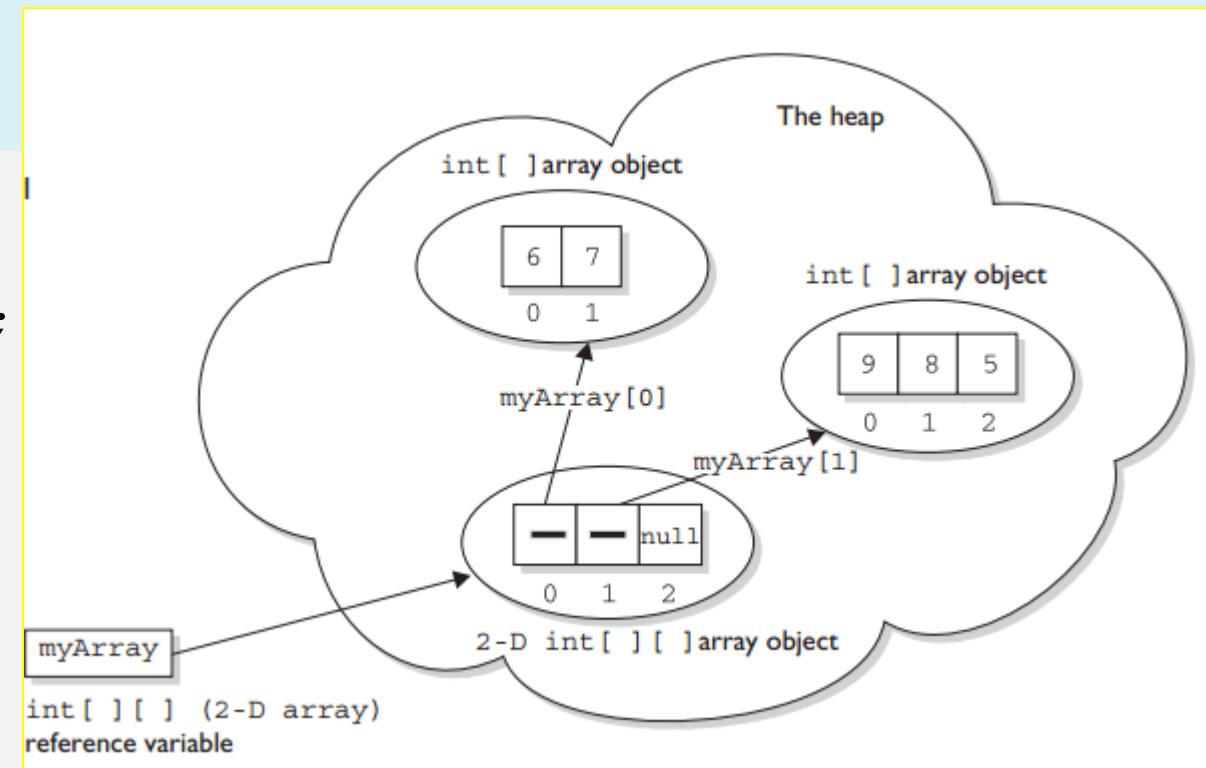
```
int[][] nilaiBarisKolom = int[3][];
```

- Arti perintah di atas:  
tolong buatkan suatu array dengan tipe data integer yang dapat menampung tiga elemen, yang masing-masing elemennya juga berupa array bertipe data int

# Konstruksi Array Multidimensional (3)

- Contoh:

```
int[][] myArray = new int[3][ ];  
  
myArray[0] = new int[2];  
myArray[0][0] = 6;  
myArray[0][1] = 7;  
  
myArray[1] = new int[3];  
myArray[1][0] = 9;  
myArray[1][1] = 8;  
myArray[1][2] = 5;
```



# OPERASI DASAR



PENGENALAN



DEKLARASI  
&  
KONSTRUKSI



OPERASI  
DASAR



INISIALISASI



OPERASI  
PADA  
ARRAY:  
LATIHAN  
SOAL

# Panjang Array

- So you have declared an array to be able to contain n elements
- Pada operasi yang dilakukan pada array, seringkali kita membutuhkan informasi mengenai berapa panjang array
- Apakah anda harus menghafalkan nilai n dan mengetikkannya secara manual? → rentan terhadap kesalahan!
- Gunakan sintaks: **nama\_array.length**

Contoh:

- Anda memiliki array String bernama **namaMahasiswa** yang memiliki kapasitas 12 elemen
- Cara mengakses panjang array tersebut: **namaMahasiswa.length** → 12

# Mengakses Elemen Array

- Cara mengakses elemen spesifik dari array adalah dengan menggunakan **indeks**
  - indeks menyatakan pada urutan keberapa elemen array yang ingin diakses
  - Contoh:  
**nama [3]** mengakses elemen keempat dari array nama
- Indeks array dimulai dari 0!
  - Dengan demikian, indeks terakhir dari suatu array yang dapat diakses adalah panjang array – 1
  - Contoh:

```
String[] nama = new String[5];
```

nama	nama [0]	nama [1]	nama [2]	nama [3]	nama [4]
------	----------	----------	----------	----------	----------

# Mengakses Seluruh Elemen pada Array

# Contoh Pengaksesan Seluruh Elemen pada Array

- Contoh:

Sebagai sekretaris himpunan, anda bertugas menuliskan daftar nama kakak-kakak kelas anda untuk acara syukuran wisuda. Anda ingin menuliskan nama-nama tersebut dengan disertai gelar baru mereka, yaitu S.Kom.

- Dengan sintaks #2:

```
String[] daftarWisudawan = new String[50];  
...  
for (String nama : daftarWisudawan) {  
    System.out.println(nama + "S.Kom.");  
}
```

Perhatikan video demo ini.  
Adakah praktek coding yang menurut anda kurang tepat?

```
*apples.java X tuna.java
1 class apples{
2     public static void main(String[] args) {
3
4     }
5 }
6
7
8
```

# Coding Example: Enhanced for Loop

# INISIALISASI



PENGENALAN



DEKLARASI  
&  
KONSTRUKSI



OPERASI  
DASAR



INISIALISASI



OPERASI  
PADA  
ARRAY:  
LATIHAN  
SOAL

# Inisialisasi Array

- Mengisikan nilai pada array
- Posisi pada array diakses dengan menggunakan indeks  
→ Indeks adalah suatu penomoran berurut terhadap anggota-anggota array

Contoh:

```
Animal[] pets = new Animal[3];
```

→ Menghasilkan satu objek array pada heap. Array ini mengandung tiga elemen referensi terhadap objek bertipe Animal, namun, saat ini ketiga Animal tersebut masih null

```
pets[0] = new Animal();  
pets[1] = new Animal();  
pets[2] = new Animal();
```

→ Pada masing-masing baris, menginstantiasi (menciptakan suatu instans) objek bertipe Animal, kemudian meletakkan referensi terhadap objek Animal tersebut pada elemen array pets yang ditunjuk oleh indeks

# Contoh (Lain) Inisialisasi Array

- Array String

```
String[] namaDosen = new String[3];  
namaDosen[0] = "Bu Tyas";  
namaDosen[1] = "Bu Hanim";  
namaDosen[2] = "Bu Anisah";
```

- Array integer

```
int[] jumlahKucingPeliharaan = int[3];  
jumlahKucingPeliharaan[0] = 1;  
jumlahKucingPeliharaan[1] = 0;  
jumlahKucingPeliharaan[2] = 2;
```

# Deklarasi, Konstruksi, dan Inisialisasi Array Dalam Baris Tunggal

- Anda dapat melakukan deklarasi, konstruksi, dan inisialisasi suatu array dalam satu baris
- Sintaks:

```
tipeData[] namaArray = {nilaiElemen0, nilaiElemen1, ..., nilaiElemenN-1}
```

- Secara otomatis Java akan membuat suatu array dengan kapasitas sebesar N dan dengan nilai-nilai awal elemen nilaiElemen0, nilaiElemen1, ... dan nilaiElemenN-1
- Contoh:

```
int[] jumlahSaudara = {1, 4, 2, 1, 0, 3}
```

Pernyataan di atas akan menyebabkan terkonstruksinya sebuah array bernama jumlahSaudara yang dapat menampung 6 elemen nilai bertipe data int. Nilai-nilai awal dari array tersebut adalah 1, 4, 2, 1, 0, 3

# Inisialisasi Array melalui Perulangan

- Jika kita menginginkan array diinisialisasi dengan nilai yang sama untuk setiap elemen
  - bisa mempergunakan sintaks perulangan
- Sintaks

```
for (int index = 0; index < panjang_array; index++) {  
    nama_array[index] = nilai_inisialisasi;  
}
```

# Contoh Inisialisasi Array Melalui Perulangan

Contoh:

Inisialisasi nilai kuliah Bahasa Pemrograman di awal perkuliahan:

```
long[] nilaiBP = new long[43];  
  
for (int i = 0; i < nilaiBP.length; i++) {  
    nilaiBP[i] = 10;  
}
```

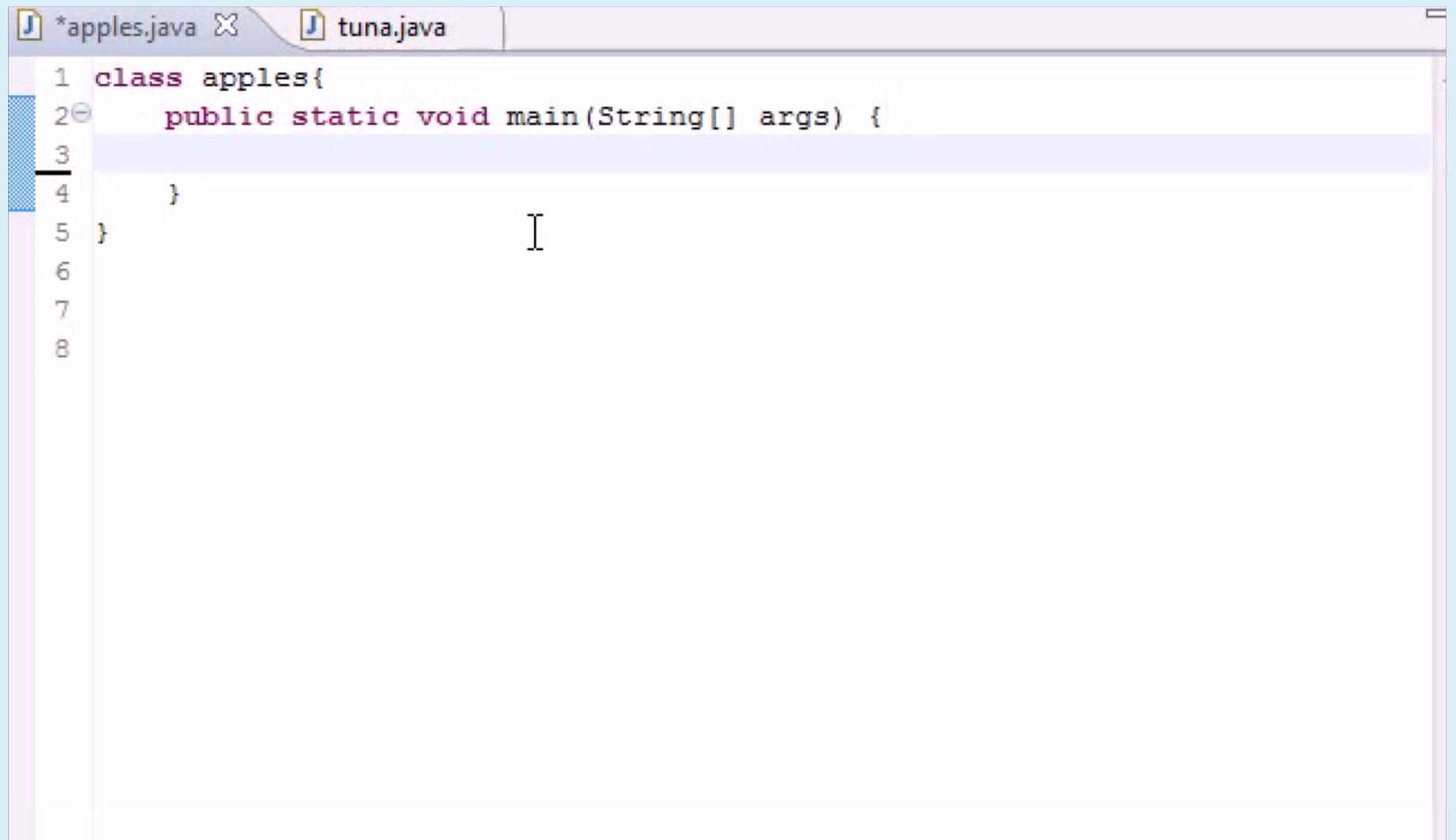
# Inisialisasi Array menggunakan java.util.Arrays

- Anda juga dapat melakukan inisialisasi array dengan menggunakan method **static fill(array, nilai\_inisialisasi)** pada kelas **java.util.Arrays**
- Contoh:

Sebagai bendahara angkatan, anda membuat suatu program untuk mencatat nama teman-teman anda yang sudah membayar iuran atau belum. Hal ini anda catat pada suatu array bertipe data boolean. Jika sudah, anda beri tanda elemen dengan indeks no. urut teman anda dengan nilai true. Sebagai inisialisasi, seluruh elemen ingin anda isi dengan false.

```
import java.util.Arrays;  
...  
boolean[] sudahBayar = new boolean[43];  
Arrays.fill(sudahBayar, false);  
...
```

Perhatikan video demo ini.  
Adakah praktek coding yang menurut anda kurang tepat?



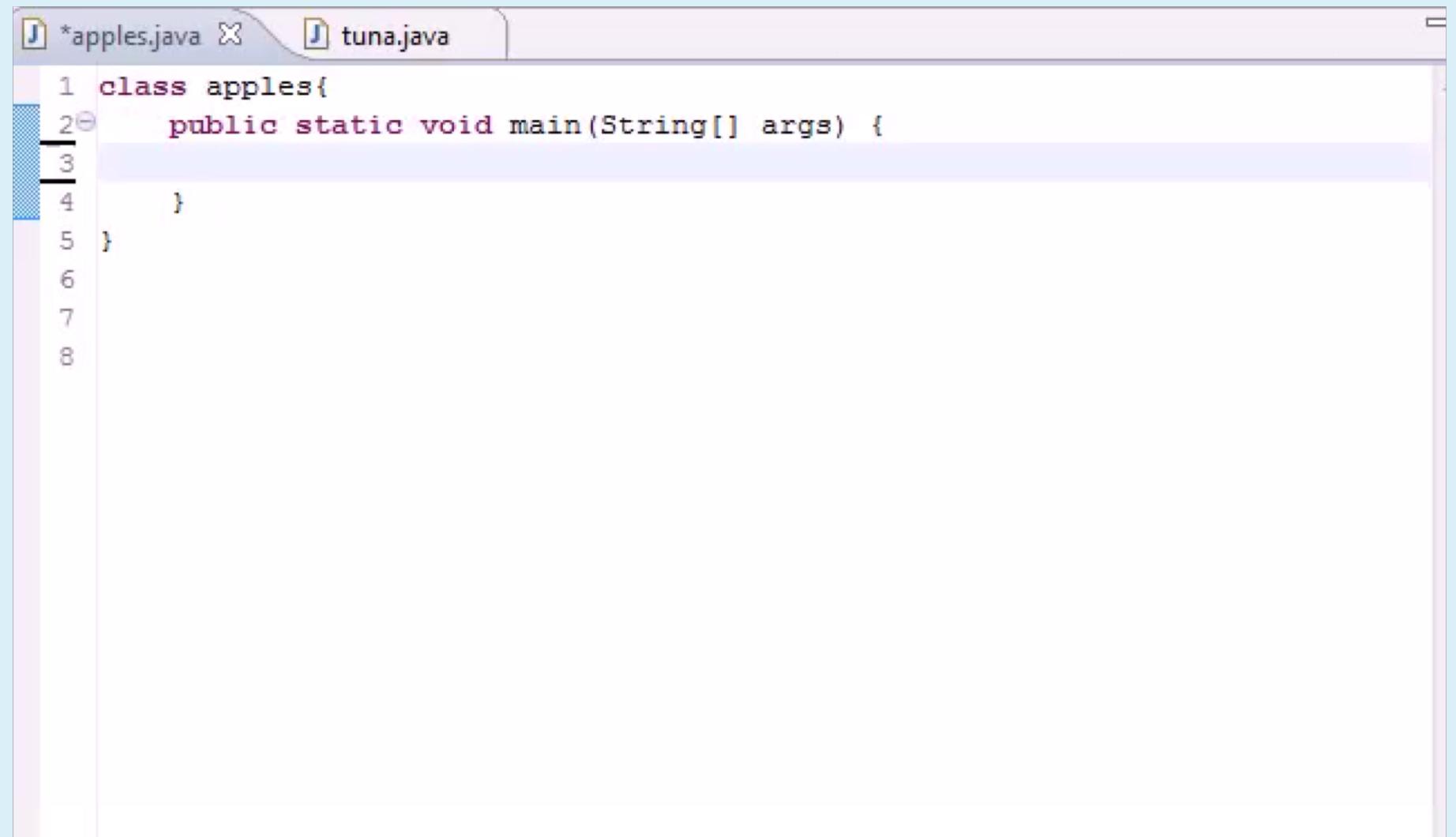
The screenshot shows a Java code editor with two tabs at the top: '\*apples.java' and 'tuna.java'. The 'apples.java' tab is active, displaying the following code:

```
1 class apples{
2     public static void main(String[] args) {
3
4     }
5 }
```

A cursor is positioned at the end of the closing brace of the main method. The code editor has a light blue background and uses standard Java syntax highlighting.

# Coding Example: Creating an Array Table

Perhatikan video demo ini.  
Adakah praktek coding yang menurut anda kurang tepat?



The screenshot shows a Java code editor with two tabs at the top: '\*apples.java' and 'tuna.java'. The 'apples.java' tab is active. The code in 'apples.java' is:

```
1 class apples{  
2     public static void main(String[] args) {  
3     }  
4 }  
5  
6  
7  
8
```

# Coding Example: Summing Elements of Arrays

# OPERASI PADA ARRAY

## Latihan 1 Array Berdimensi Tunggal



PENGENALAN



DEKLARASI  
&  
KONSTRUKSI



OPERASI  
DASAR



INISIALISASI



OPERASI  
PADA  
ARRAY:  
LATIHAN  
SOAL

# 1. Deklarasi, Konstruksi, dan Inisialisasi Array

- Buatlah suatu array String berkapasitas 10
- Isilah masing-masing elemen pada array tersebut dengan nama teman-teman Anda
- Cetaklah array String anda dengan rapi sesuai format sebagai berikut:

**Nama ke-1: nama\_teman1**

**Nama ke-2: nama\_teman2**

...

**Nama ke-10: nama\_teman10**

- Buatlah suatu array int berkapasitas sesuai dengan jumlah anggota keluarga anda
- Isilah masing-masing elemen pada array tersebut dengan usia anggota keluarga anda
- Cetaklah masing-masing elemen pada array int anda dengan dipisahkan dengan tab
- Jumlahkanlah seluruh elemen pada array int anda, kemudian cetaklah hasilnya sebagai berikut:

**Jumlah usia angota keluarga saya: . . . tahun.**

(Gunakanlah System.out.printf)

## 2. Menghitung Dot Product dari Dua Vector

- Anda menghitung hasil perkalian skalar dari dua buah vektor dengan menggunakan operasi dot product sebagai berikut:

$$\mathbf{x} \cdot \mathbf{y} = \sum_i x_i \cdot y_i$$

- Hitunglah secara manual hasil dari dot product dua buah vektor berikut ini:

x [ 6 | -5 | 3 | 2 | 7 | -8 ]

y [ 4 | 2 | -11 | 9 | 6 | 1 ]

- Tulislah kode untuk menghitung hasil dot product dari kedua vektor tersebut
- Buatlah method untuk menghitung dot product dari dua buah vektor dengan *method signature* sebagai berikut:

```
static int dotProduct(int[] vektor1, int[] vektor2)
```

### 3. Jaccard Similarity Index

- Dalam pemberian rekomendasi kepada pelanggan, anda seringkali ingin mengetahui pola pembelian pelanggan yang karakteristiknya serupa
  - **Bagaimana mengukur kesamaan karakteristik pelanggan?**
- Salah satu cara sederhana untuk mengukur kesamaan karakteristik pelanggan adalah melalui kesamaan barang yang dibeli
- Untuk merepresentasikan pembelian suatu pelanggan, anda dapat menggunakan tabel sebagai berikut:

ID	nama	item1	item2	item3	item4	item5	item6
1	Bambang	1	0	0	0	1	1
2	Cahyono	0	1	0	1	1	0
3	Dedi	1	1	0	0	1	1
4	Eko	0	0	1	0	0	0

# 3. Jaccard Similarity Index (2)

- Pada tabel di atas, 1 berarti pelanggan bersangkutan membeli item produk tersebut dan 0 berarti tidak

Task:

- Carilah pengertian dan formula penghitungan dari Jaccard Similarity Index
- Hitunglah secara manual Jaccard Similarity Index dari setiap pasang dari keempat pelanggan di atas?
  - Pasangan pelanggan yang mana yang paling serupa? Pasangan pelanggan yang mana yang paling tidak mirip?
- Representasikan pola pembelian dari masing-masing pelanggan dalam bentuk array
- Buatlah method untuk menghitung Jaccard Similarity Index dengan method signature sebagai berikut:

```
int jaccardSimilarity(int vektor1, int vektor2)
```

- Dengan menggunakan method tersebut, hitunglah Jaccard Similarity Index dari setiap pasang pelanggan

## 4. Mengurutkan Nilai Elemen Array

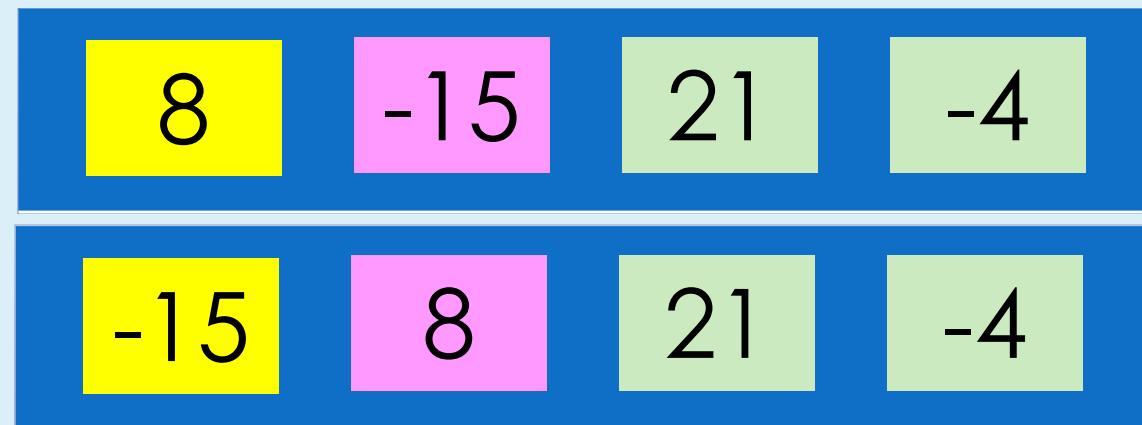
- Jika anda memiliki sekumpulan nilai, bagaimana anda akan mengurutkan kumpulan nilai tersebut?
- Misalkan kumpulan nilai tersebut anda simpan dalam suatu array sebagai berikut:

```
int [] myArray = {8, 15, 21, -4, 11, 37, 25}
```

# Mengurutkan Nilai Elemen Array – Metode 1

- Salah satu susunan langkah yang dapat anda pergunakan adalah sebagai berikut:
  1. Mulai dari elemen array pertama sampai elemen array kedua dari terakhir:
    - Bandingkan nilai elemen tersebut dengan nilai elemen array di sebelahnya (yang indeksnya lebih besar)
    - Apabila nilai elemen tersebut lebih besar dari nilai elemen array di sebelahnya, maka pertukarkan posisi kedua elemen tersebut
  2. Apabila pada langkah 1 tidak terjadi pertukaran tempat sama sekali, artinya array telah terurut, dan iterasi dapat dihentikan. Namun, apabila masih terjadi pertukaran tempat, kita masih perlu mengulangi langkah 1.

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 1, step 1

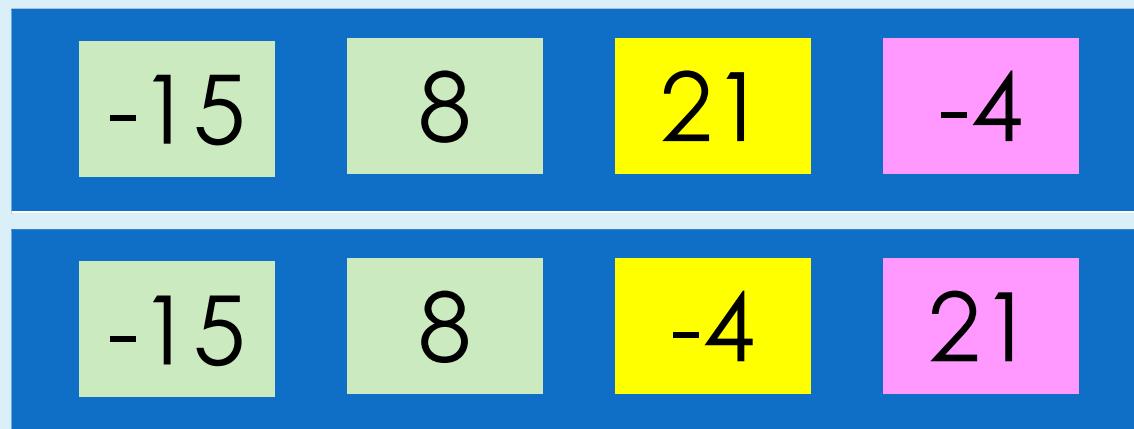


Terjadi pertukaran posisi!

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 1, step 2

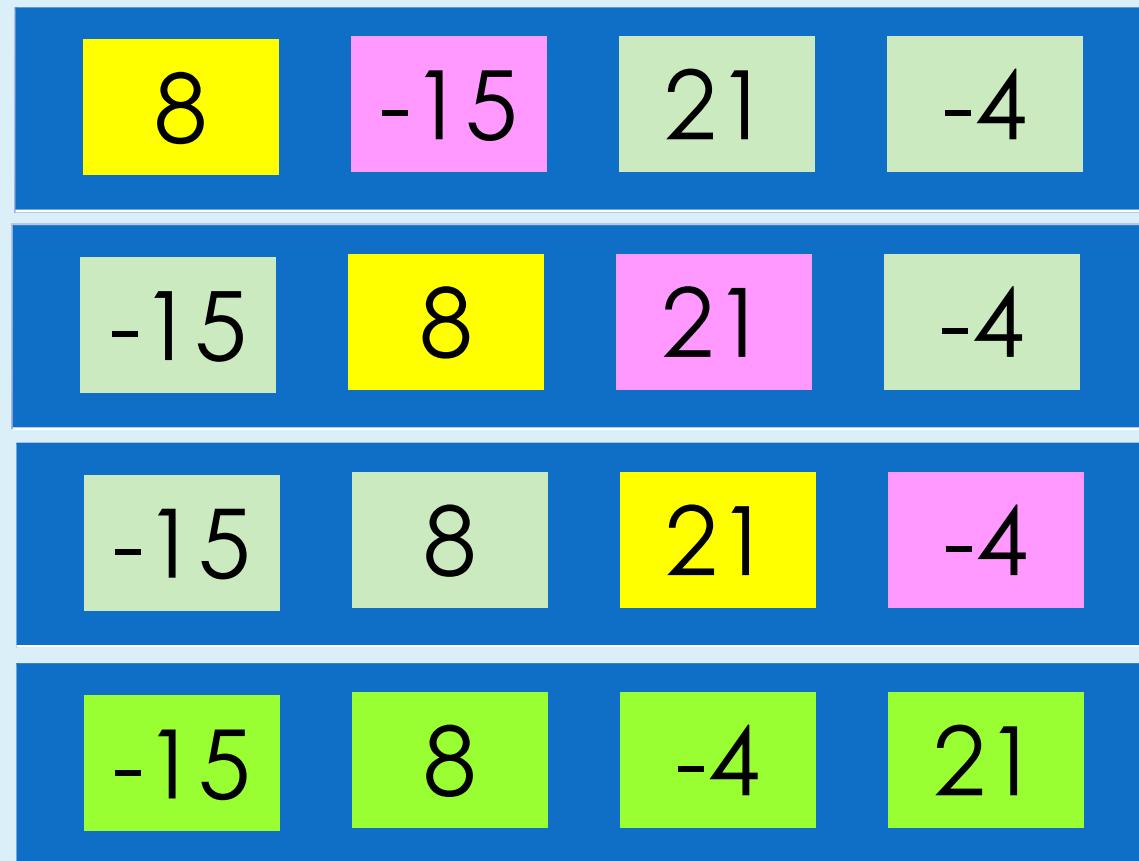


# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 1, step 3



Terjadi pertukaran posisi!

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 1, review



Apakah pada iterasi ini terjadi pertukaran posisi?

**YA**

→ Lanjutkan ke iterasi berikutnya

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 2, step 1

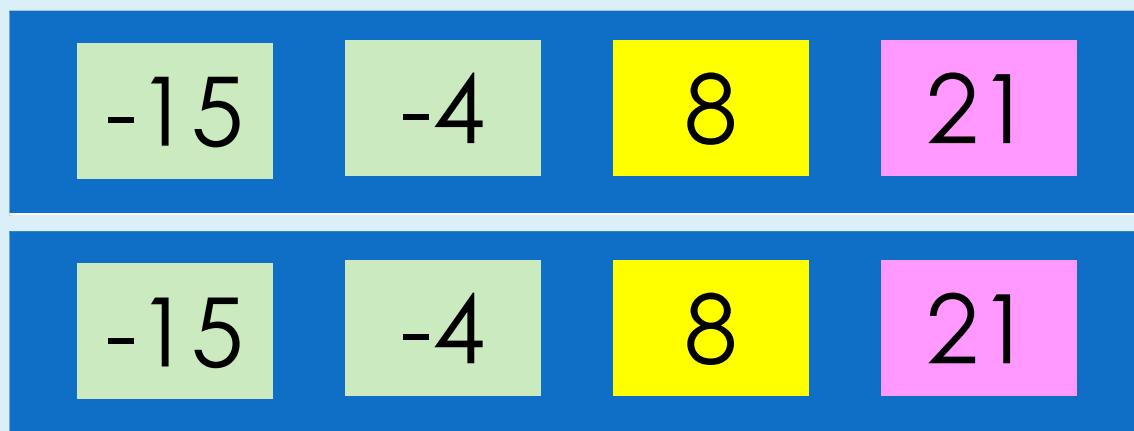


# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 2, step 2

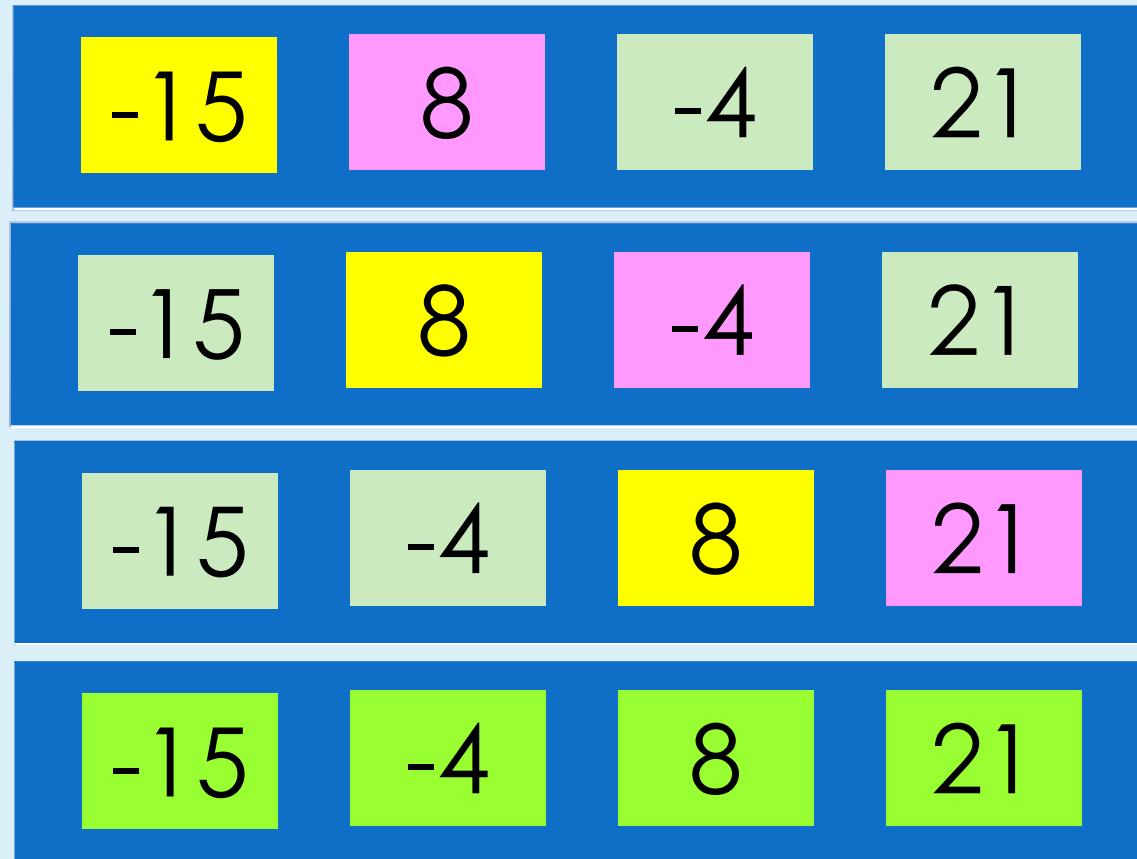


Terjadi pertukaran posisi!

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 2, step 3



# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 2, review

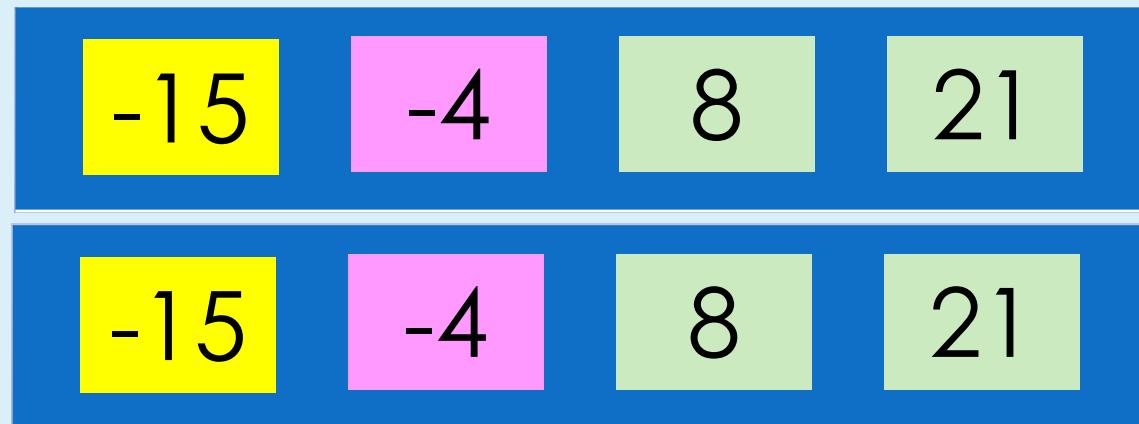


Apakah pada iterasi ini terjadi pertukaran posisi?

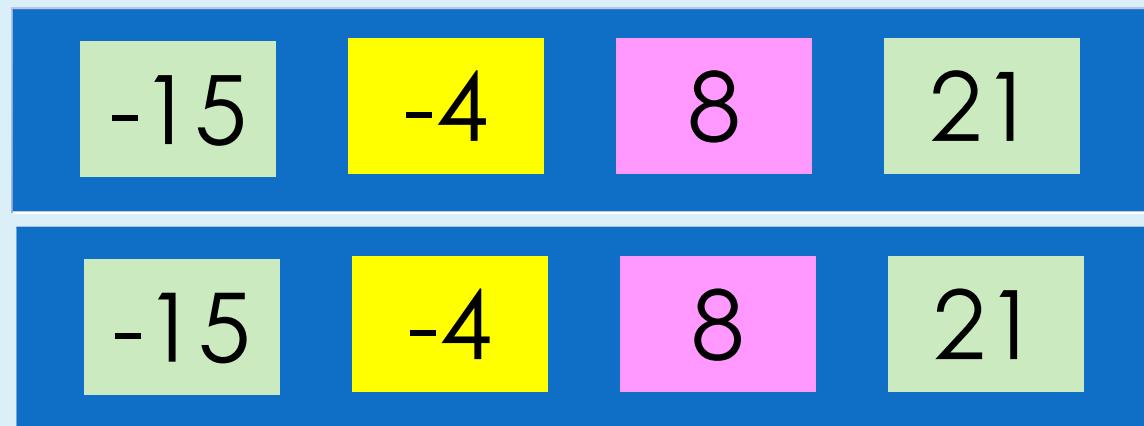
**YA**

→ Lanjutkan ke iterasi berikutnya

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 3, step 1



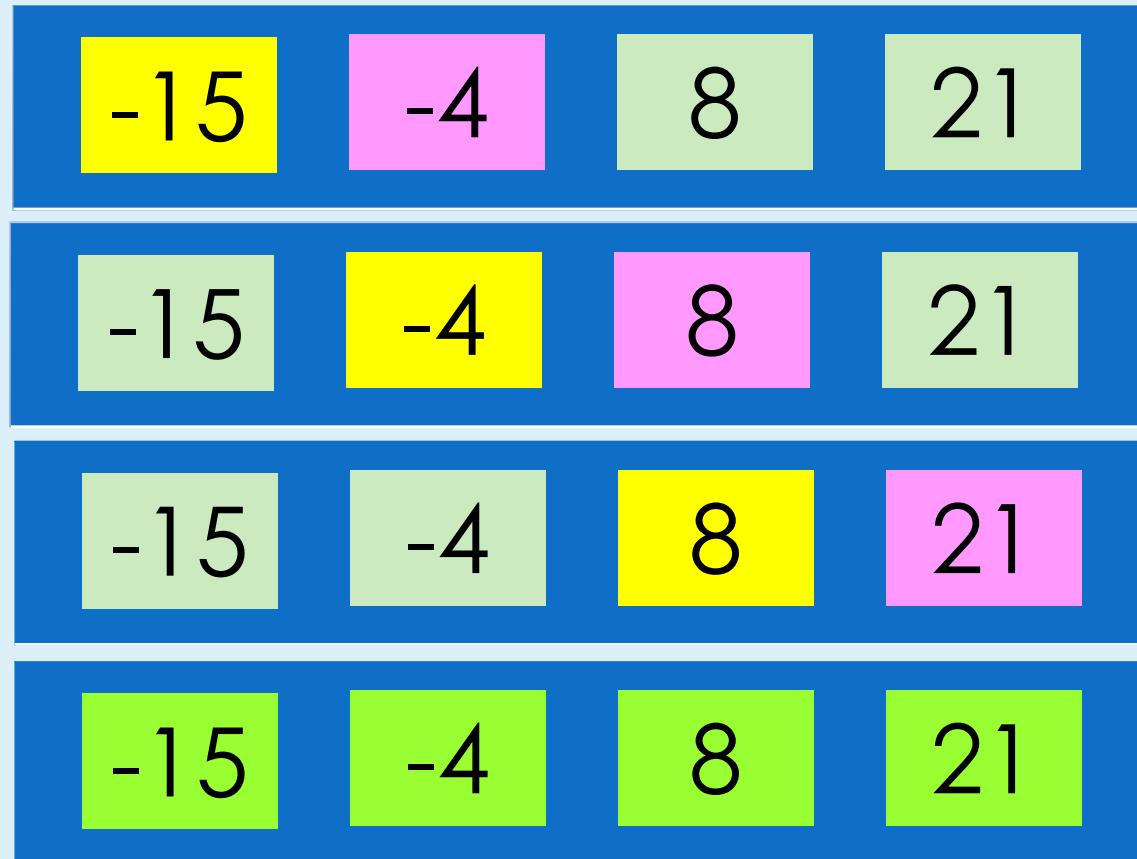
# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 3, step 2



# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 3, step 3



# Mengurutkan Nilai Elemen Array – Visualisasi Metode 1 – Iterasi 3, review



Apakah pada iterasi ini terjadi pertukaran posisi?

**TIDAK**

- Hentikan iterasi
- Array telah terurut

# Mengurutkan Nilai Elemen Array – Metode 1 – How to

- Beberapa trik yang perlu anda ketahui untuk mengimplementasikan pengurutan nilai elemen array dengan metode 1:
  - Mempertukarkan nilai dua elemen array
  - Melakukan operasi secara berurutan pada seluruh elemen array
  - Memberi tanda apabila pada suatu iterasi telah terjadi perubahan (dalam kasus ini, pertukaran nilai elemen array)

# Mengurutkan Nilai Elemen Array – Metode 1 – How to (2)

- Sintaks/pernyataan seperti apa yang anda butuhkan?

## **Activity points!**

- Mempertukarkan nilai dua elemen array
  - ...
- Melakukan operasi secara berurutan pada seluruh elemen array
  - ...
- Memberi tanda apabila pada suatu iterasi telah terjadi perubahan (dalam kasus ini, pertukaran nilai elemen array)
  - ...

# Mengurutkan Nilai Elemen Array – Metode 1 – How to (3)

## Exercise

- Lakukanlah langkah-langkah pengurutan elemen array secara manual menggunakan metode 1 pada array berikut:  
`int[] myArray = {8, 15, 21, -4, 11, 37, 25}`
- Tulislah kode untuk melakukan pengurutan tersebut secara programatik
- Buatlah method untuk melakukan pengurutan array menggunakan metode 1 dengan method signature sebagai berikut

`int[] sortArray1(int[] anArray)`

# Mengurutkan Nilai Elemen Array – Metode 2

- Metode lain yang dapat anda lakukan adalah sebagai berikut:

Mulai dari elemen array kedua sampai elemen array terakhir:

- a. Bandingkan nilai elemen tersebut dengan nilai elemen array di sebelahnya (yang indeksnya lebih kecil)
- b. Apabila nilai elemen tersebut lebih besar dari nilai elemen array di sebelahnya, maka pertukarkan posisi kedua elemen tersebut. Lanjutkan dengan langkah a.
- c. Apabila nilai elemen tersebut lebih kecil, lanjutkan dengan meninjau elemen array berikutnya

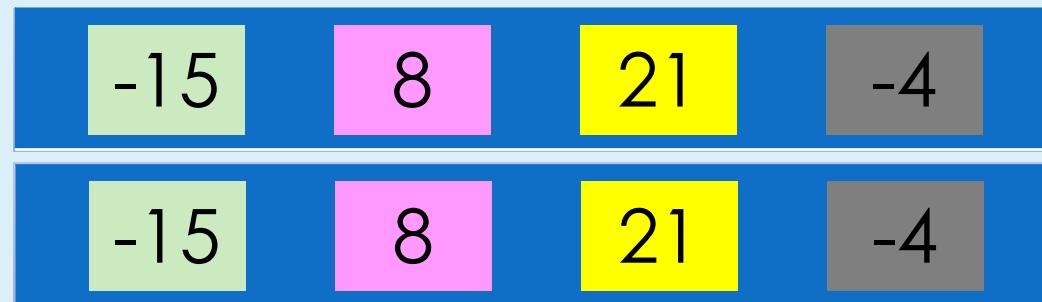
# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Step 1 – Substep 1



**Terjadi pertukaran posisi!**

- Teruskan dengan membandingkan dengan nilai elemen di sebelah kiri
- Namun, dalam kasus ini elemen sudah berpindah ke index 0, sehingga tidak ada elemen di sebelah kirinya

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Step 2 – Substep 1



# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Step 3 – Substep 1



**Terjadi pertukaran posisi!**  
→ Kita teruskan dengan  
membandingkan dengan  
elemen di sebelah kiri (yg  
indeksnya lebih kecil)

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Step 3 – Substep 2



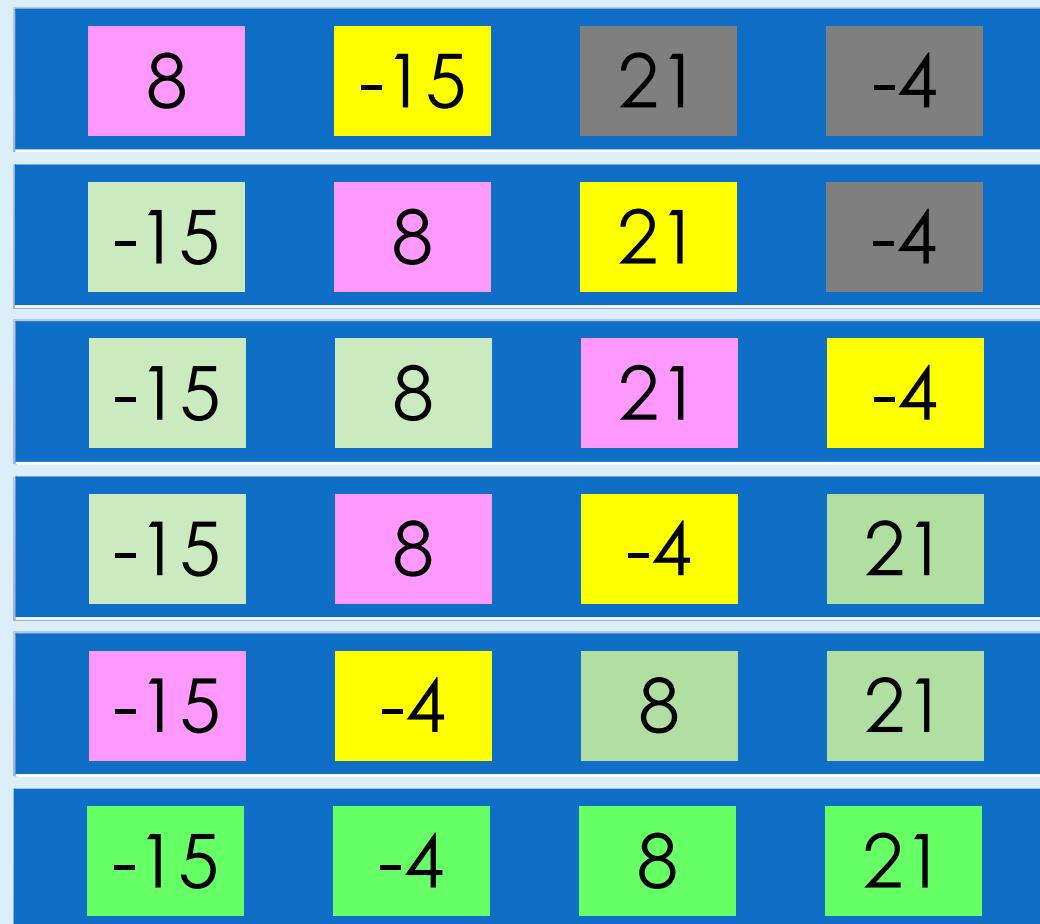
**Terjadi pertukaran posisi!**

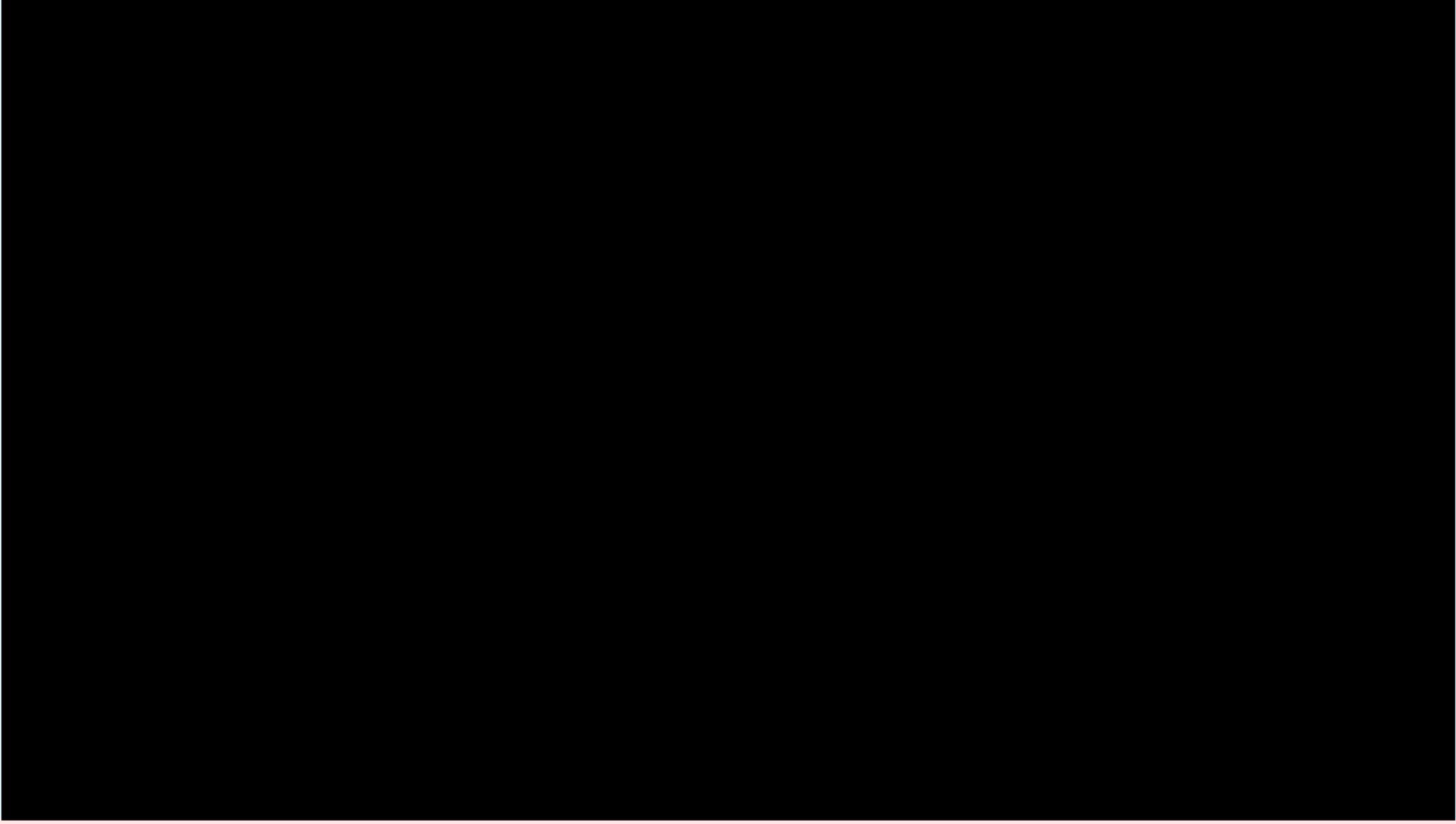
→ Kita teruskan dengan membandingkan dengan elemen di sebelah kiri (yg indeksnya lebih kecil)

# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Step 3 – Substep 3



# Mengurutkan Nilai Elemen Array – Visualisasi Metode 2 – Review





**Sorting Illustration: Romanian Folk Dance**

# Mengurutkan Nilai Elemen Array – Metode 2 – Exercise

- Lakukanlah langkah-langkah pengurutan elemen array secara manual menggunakan metode 2 pada array berikut:

```
int[] myArray = {8, 15, 21, -4, 11, 37, 25}
```

- Tulislah kode untuk melakukan pengurutan tersebut secara programatik
- Buatlah method untuk melakukan pengurutan array menggunakan metode 2 dengan method signature sebagai berikut

```
int[] sortArray2(int[] anArray)
```