

Pemrograman Berbasis Obyek

Arrays

Oleh Politeknik Elektronika Negeri Surabaya

2017



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

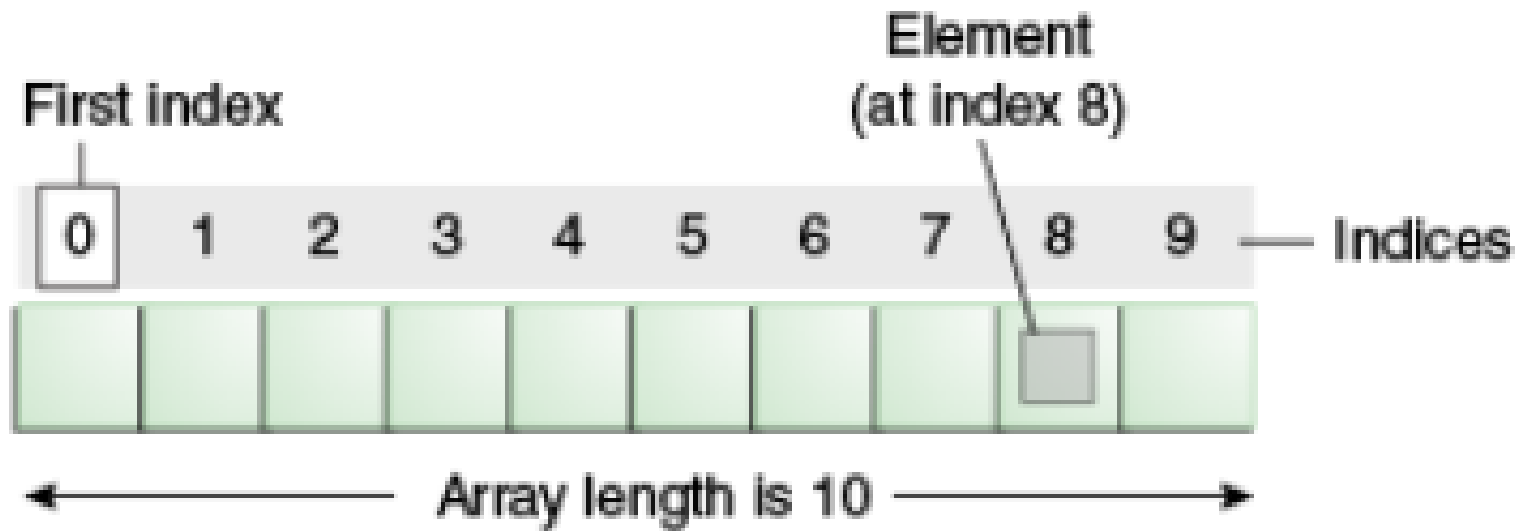
Konten

- Deklarasi array
- Membuat array
- Mengisi array
- Mendeklarasikan dan membuat array
- Inisialisasi array
- Array multi dimensi
- Deklarasi array multi dimensi
- Membuat array multi dimensi
- Mengetahui total elemen array
- Merubah total elemen array
- Mengkopi array
- Referensi array

Definisi Array

- Array adalah suatu kumpulan data pada suatu variabel.
- Pada deklarasi variabel biasa, kita hanya dapat memberikan 1 data pada 1 variabel.
- Tapi dengan konsep array, kita dapat memberikan lebih dari 1 data pada 1 variabel.

An array of ten elements



- Array adalah container object yang menyimpan sejumlah nilai bertipe data sama dengan ukuran yang sudah pasti.
- Panjang array dibuat ketika array dibuat.
- Setelah dibuat maka panjang array sudah pasti dan tidak bisa di ubah.

- Tiap item dalam array disebut element
- Tiap item dapat diakses melalui nomor index

Deklarasi Array

- Cara mendeklarasikan array:

```
tipe_array nama_array[];
```

```
tipe_array[] nama_array;
```

- Contoh :

```
int nilai[];
```

```
int [] nilai[];
```

```
int nilai[][];
```

```
char[] huruf;
```

```
char[][] huruf;
```



- *blank space* tidak berpengaruh terhadap peletakan tanda [].
- Contoh:

```
tipe_array nama_array [];
```

```
tipe_array [] nama_array;
```

```
tipe_array []nama_array;
```


Membuat Array

- Pada saat pendeklarasian array:

```
int nilai[];
```

- Yang terjadi:
 - hanya memberitahu kepada kompiler bahwa kita mendeklarasikan sebuah variabel bernama nilai yang berbentuk array dan bertipe int.
 - belum memesan tempat di memori untuk menampung elemen-elemen array.
- Agar dapat memesan tempat di memori untuk menampung elemen-elemen array, kita perlu membuat array.
- Caranya adalah dengan memakai operator *new*, karena di dalam Java suatu array adalah dianggap suatu obyek.
- Format penulisan:

```
nama_array = new tipe_array[total_elemen_array];
```



- Contoh :

```
int nilai[];
```

```
nilai = new int[5];
```

- Contoh :

```
int nilai[] = new int[5];
```

- Pada saat baris kedua dieksekusi, kompiler Java akan menyediakan *space* di memori sebanyak 5 elemen dengan tipe masing-masing adalah int.
- Identifikasi elemen array sering disebut dengan indeks array, yang berawal dari 0 sampai `total_element_array-1`.



Mengisi Array

- Array harus diisi dengan nilai yang sesuai dengan tipe array-nya.
- Beritahukan kepada kompiler Java elemen manakah dari array yang akan diisi.
- Format :

```
nama_array[elemen_array] = nilai;
```

- Contoh :

```
int nilai[];
```

```
nilai = new int[2];
```

```
nilai[1] = 100;
```

- Di baris ketiga, kita berusaha mengakses elemen array pada indeks ke-1 dan memberinya dengan nilai 100.



Contoh:

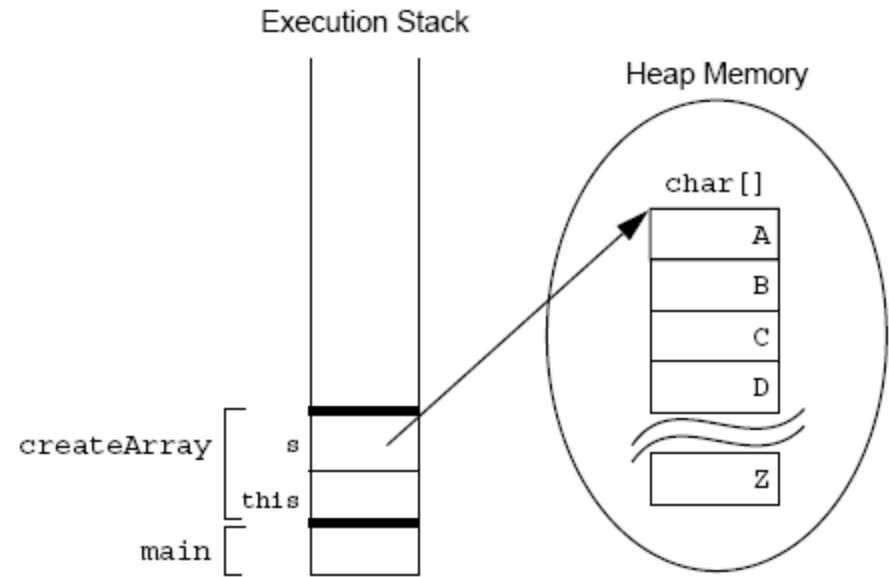
```
public class Array1 {  
    public static void main(String args[]) {  
        int nilai[];  
        nilai = new int[2];  
  
        nilai[0] = 50;  
        nilai[1] = 100;  
        System.out.println("Nilai indeks ke-0 = " + nilai[0]);  
        System.out.println("Nilai indeks ke-1 = " + nilai[1]);  
    }  
}
```

Contoh:

```
public char[] createArray() {
    char[] s;

    s = new char[26];
    for ( int i=0; i<26; i++ ) {
        s[i] = (char) ('A' + i);
    }

    return s;
}
```



Mendeklarasikan dan Membuat Array

- Dimungkinkan melakukan deklarasi dan pembuatan array hanya pada satu baris *statement*.

- Format:

```
tipe_array nama_array[] = new  
    tipe_array[total_elemen_array];
```

- Contoh :

```
int nilai[] = new int[5];
```



Contoh:

```
public class Array2 {  
    public static void main(String args[]) {  
        int nilai[] = new int[2];  
  
        nilai[0] = 50;  
        nilai[1] = 100;  
        System.out.println("Nilai indeks ke-0 = " + nilai[0]);  
        System.out.println("Nilai indeks ke-1 = " + nilai[1]);  
    }  
}
```

Inisialisasi Array

- Pemberian nilai awal bagi setiap elemen array yang dibuat.
- Pada saat membuat array dengan sejumlah elemen tertentu, kompiler Java akan memberikan **nilai default** kepada setiap elemen array selama kita tidak memberikan nilai awal pada saat pembuatan array.

- Misal :

```
int nilai[];
```

```
nilai = new int[5];
```

- Pada saat membuat array di baris kedua, tidak diberikan nilai awal (inisialisasi) pada elemen array.
- Pada saat baris kedua selesai dieksekusi, maka kompiler Java akan memberikan nilai default kepada setiap elemen array.
- Nilai default akan bergantung pada tipe array.
- Karena pd contoh diatas dideklarasikan array dengan tipe int, maka setiap elemen dari array akan diberikan nilai 0.



Contoh:

```
public class InitArray {  
    public static void main(String args[]) {  
        int nilai[] = new nilai[5];  
        System.out.println("Nilai indeks ke-2 = " + nilai[2]);  
    }  
}
```

Mendeklarasikan, Membuat, dan Menginisialisasi Array

- Format

```
tipe_array nama_array[] = {nilai_indeks_0,           nilai_indeks_1,  
    ..., nilai_indeks_n};
```

- Contoh :

```
int nilai[] = {70, 65, 85};
```

Contoh:

```
public class Array2 {  
    public static void main(String args[]) {  
        int nilai[] = {50, 100};  
  
        System.out.println("Nilai indeks ke-0 = " + nilai[0]);  
        System.out.println("Nilai indeks ke-1 = " + nilai[1]);  
    }  
}
```

Array multi dimensi

- In the Java programming language, a multidimensional array is simply an array whose components are themselves arrays.
- This is unlike arrays in C or Fortran.
- A consequence of this is that the rows are allowed to vary in length

Deklarasi array multi dimensi

- Cara **mendeklarasikan** array multi dimensi tidak jauh berbeda dengan cara mendeklarasikan array berdimensi tunggal
- Cukup dengan menambahkan tanda [].
- Contoh:

int nilai[]; → berdimensi 1

char huruf[][]; → berdimensi 2

double total[][][]; → berdimensi 3

String[][] nama; → berdimensi 2

float[] panjang[]; → berdimensi 2

short[][] tinggi[]; → berdimensi 3

Membuat array multi dimensi

- Pembuatan array multi dimensi identik dengan pembuatan array dimensi tunggal, yaitu dengan menambahkan tanda[].
- Contoh:

```
int x[ ][ ] = new int[3][4];
```

berdimensi 2, dengan 3 elemen di dimensi ke-1 dan 4 elemen di dimensi ke-2

```
double[ ][ ][ ] balok = new double[3][4][2];
```

berdimensi 3, dengan 3 elemen di dimensi ke-1, 4 elemen di dimensi ke-2 dan 2 elemen di dimensi ke-3



Pembuatan array multi dimensi yang tidak beraturan.

- Diperkenankan untuk membuat suatu array yang berdimensi tidak beraturan.
- Contoh :

```
int x[ ][ ] = new int[3][ ];
```

```
x[0] = new int[2];
```

```
x[1] = new int[3];
```

```
x[2] = new int[1];
```


Multidimensional Arrays

- Arrays of arrays:

```
int twoDim [] [] = new int [4] [] ;  
twoDim[0] = new int [5] ;  
twoDim[1] = new int [5] ;
```

```
int twoDim [] [] = new int [] [4] ; illegal
```

Mengetahui total elemen array

- Untuk mengetahui panjang dari suatu array yang telah dibuat, dapat memakai properti *length*.
- Contoh :

```
int x[][] = new int[3][];
```

```
x[0] = new int[2];
```

```
x[1] = new int[3];
```

```
x[2] = new int[1];
```

```
x.length → 3 }
```

```
x[0].length → 2
```

```
x[1].length → 3
```

```
x[2].length → 1
```

Contoh

```
class MultiDimArrayDemo {  
    public static void main(String[] args) {  
        String[][] names = {  
            {"Mr. ", "Mrs. ", "Ms. "},  
            {"Smith", "Jones"}  
        };  
  
        // Mr. Smith  
        System.out.println(names[0][0] + names[1][0]);  
        // Ms. Jones  
        System.out.println(names[0][2] + names[1][1]);  
    }  
}
```

- The output from this program is:

Mr. Smith

Ms. Jones



Merubah total elemen array ??

- Tidak diperkenankan merubah total elemen array.
- Diperkenankan membuat array baru dengan nama object yang sama
- Tetapi total elemen array yang sebelumnya akan tergantikan dengan yang baru.
- Contoh, membuat sebuah array dengan total elemen 2 dan kemudian mengisinya dengan suatu nilai.

```
byte posisi[] = new byte[2];
```

```
posisi[0] = 55;
```

```
posisi[1] = 68;
```

```
byte posisi[] = new byte[5];
```

- Yang terjadi : total elemen array yang sebelumnya dan juga nilai dari setiap elemennya akan hilang dan tergantikan dengan array yang baru.



Mengkopi elemen array

- Isi dari suatu array dapat di kopi pada array yang lain dengan memanfaatkan method `arraycopy()` pada class `System`.
- Format :

```
System.arraycopy(array1, p1, array2, p2, n);
```

dimana :

array1 = array asal/sumber pengkopian

array2 = array tujuan pengkopian

p1 = posisi indeks awal pengkopian pada array asal

p2 = posisi indeks awal pengkopian pada array tujuan

n = banyaknya elemen array yang akan dikopi



- Contoh :

```
int lama[ ] = {1, 2, 3, 4, 5, 6};
```

```
int baru[ ] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
```

```
System.arraycopy(lama, 2, baru, 5, 3);
```

- Elemen-elemen array baru akan bernilai masing 10, 9, 8, 7, 6, 3, 4, 5, 2, 1

Contoh:

```
public class CopyArray {  
    public static void main(String args[]) {  
        int lama[] = {1, 2, 3, 4, 5, 6};  
        int baru[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};  
  
        System.arraycopy(lama, 2, baru, 5, 3);  
  
        System.out.println("Isi array baru :");  
        for(int i=0; i<baru.length; i++) {  
            System.out.print(baru[i] + " ");  
        }  
    }  
}
```

Contoh arraycopy

```
class ArrayCopyDemo {  
    public static void main(String[] args) {  
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e', 'i', 'n', 'a', 't', 'e', 'd' };  
        char[] copyTo = new char[7];  
        System.arraycopy(copyFrom, 2, copyTo, 0, 7);  
        System.out.println(new String(copyTo));  
    }  
}
```

- The output from this program is:
 caffein



Referensi array

- Suatu array juga dapat me-refer (merujuk) ke array yang lain, dengan kata lain merujuk pada alamat memori yang sama.

- Contoh :

```
int nilai[] = {10, 20, 30};
```

```
int result[];
```

```
result = nilai;
```

- Di baris ketiga, assign array result ke array nilai.

Akibatnya, array result akan me-refer (merujuk) pada array nilai, sehingga kedua array tersebut merujuk alamat memori yang sama.

Pada saat nilai elemen pada array result diubah, misalnya dengan :

```
result[1] = 50;
```

maka nilai dari indeks ke-1 pada array nilai (nilai[1]) juga ikut berubah menjadi 50.



Contoh

```
public class ArrayReference {  
    public static void main(String args[]) {  
        int nilai[] = {10, 20, 30};  
        int result[];  
        result = nilai;  
        result[1] = 50;  
  
        System.out.println("Indeks ke-1 array nilai berisi " + nilai[1]);  
    }  
}
```

Mengakses elemen array dengan for

```
class ForDemo {  
    public static void main(String[] args){  
        int x [] = {1,2,3,4,5,6,7,8,9,10};  
        for(int i=0; i<10; i++){  
            System.out.println("Count is: " + x[i]);  
        }  
    }  
}
```

- The output of this program is:

Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10



Mengakses elemen array dengan for

```
class EnhancedForDemo {  
    public static void main(String[] args){  
        int[] numbers = {1,2,3,4,5,6,7,8,9,10};  
        for (int item : numbers) {  
            System.out.println("Count is: " + item);        }  
    }  
}
```

Tugas

1. Apakah yang dimaksud dengan array?
2. Buatlah contoh mendeklarasikan, memberi nilai dan mengakses elemen array!
3. Bagaimana cara mengetahui panjang array?
4. Uraikan pengetahuan anda mengenai array multidimensi!
5. Bagaimana cara mengkopi isi array?
6. Apakah yang dimaksud dengan referensi array. Beri contoh!



1. Oracle Java Documentation, The Java™ Tutorials, <https://docs.oracle.com/javase/tutorial/>, Copyright © 1995, Oracle 2015.
2. Tita Karlita, Yuliana Setrowati, Rizky Yuniar Hakkun, Pemrograman Berorientasi Obyek, PENS-2012
3. Sun Java Programming, Sun Educational Services, Student Guide, Sun Microsystems, 2001.
bridge to the future
4. John R. Hubbard, Programming With Java, McGraw-Hill, ISBN: 0-07-142040-1, 2004.
5. Patrick Niemeyer, Jonathan Knudsen, Learning Java, O'reilly, CA, ISBN: 1565927184, 2000.
6. Philip Heller, Simon Roberts, Complete Java 2 Certification Study Guide, Third Edition, Sybex, San Francisco, London, ISBN: 0-7821-4419-5, 2002.
7. Herbert Schildt, The Complete Reference, Java™ Seventh Edition, Mc Graw Hill, Osborne, ISBN: 978-0-07-163177-8, 2007