

Academic Year 2021/2022

Pemrograman Berorientasi Objek

Rofilde Hasudungan

Lecturer

Department of Informatic

Universitas Muhammadiyah Kalimantan Timur

September 27, 2021

Outline

1 Pendahuluan

- Mendeklarasikan dan Membuat Array
- Mengakses Array
- Struktur Array
- Perulangan pada Array

2 Array Multidimensi

- Mendeklarasikan dan Membuat Array Multidimensi
- Mengakses Array Multidimensi
- Perulangan pada Array Multidimensi

3 Array pada Method

- Parameter Array
- Array sebagai return value

Array

Array dalam java dapat diartikan merupakan suatu variable yang memiliki banyak nilai dengan tipe data tertentu yang sama. Contohnya variabel angka yang bertipe data integer berisi angka-angka integer.

Mendeklarasikan dan Membuat Array

Suatu variable array dideklarasikan dengan cara seperti berikut:

```
ElementType[] arrayRefVar
```

Dimana `ElementType` ialah tipe data elemen (nilai/isi) array seperti `int`, `long`, `double`, etc. Kemudian setelah dideklarasikan, selanjutnya untuk membuat array ialah sebagai berikut:

```
arrayRefVar = new ElementType[arraySize]
```

`arraySize` ialah bilangan bulat yang menyatakan ukuran dari array atau disebut juga jumlah elemen array. Ukuran array tidak bisa diubah lagi ketika telah dibuat.

Mengakses Array

Sebagai contoh kita membuat variable array `myList`, kemudian untuk membuat array harus ditentukan jumlah element-nya misalkan 10, sehingga array ini hanya berisi 10 element dengan indeks setiap element ialah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 dimana total indeksnya juga adalah 10.

```
1 double [] myList;  
2  
3 myList = new double [10];
```

Mengakses Array

Indeks element ini digunakan untuk mengakses nilai pada element array atau untuk memberikan nilai pada element array. Perhatikan contoh berikut

```

1  double [] myList;
2  myList = new double [10];
3  myList[0] = 5.6;
4  myList[1] = 4.5;
5  myList[2] = 3.3;
6  myList[3] = 13.2;
7  myList[4] = 4.0;
8  myList[5] = 34.33;
9  myList[6] = 34.0;
10 myList[7] = 45.45;
11 myList[8] = 99.993;
12 myList[9] = 11123;
13 double hasil = myList[0] + myList[7];
14 System.out.println("Element ke 9 = " + myList[9]);
    
```

Struktur Array

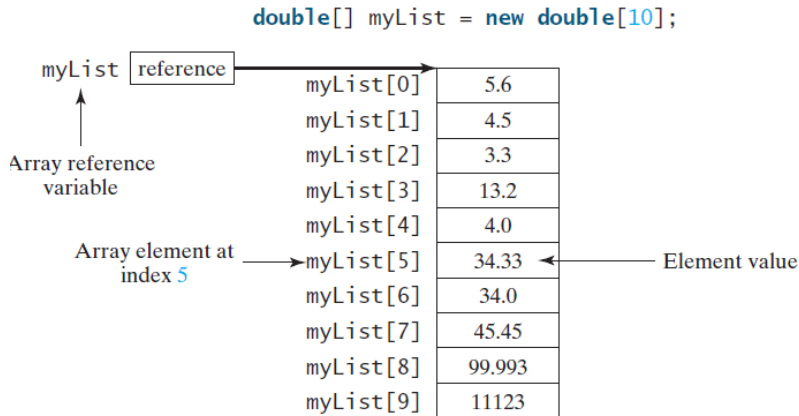


Figure: Variable Array

Looping Element Array

Kita dapat menelusuri (mengakses) nilai dari element array dengan menggunakan perulangan. Hal ini tentunya karena dalam kasus tertentu seperti jumlah element yang sangat banyak kita tidak mungkin mengaksesnya manual seperti contoh sebelumnya. Perhatikan contoh berikut (lanjutan dari contoh sebelumnya) :

```
1 for(int i=0; i < 10; i++) {  
2     System.out.println("Element ke-" + i + " = " + myList[i]);  
3 }
```

Dengan menggunakan perulangan for kita dapat melakukan perulangan dari 0 (menandakan indeks pertama) dan <10 (9) (indeks terakhir).

Looping Element Array

Penggunaan perulangan pada array seperti yang ditunjukkan pada contoh slide sebelumnya sangatlah bermanfaat dan mudah diimplementasikan. Namun demikian, bahasa java memiliki mekanisme yang lebih baik untuk menangani perulangan pada array yang disebut dengan for-each

```
1 for (double value: myList) {  
2     System.out.println("Value = " + value);  
3 }
```

Dengan menggunakan perulangan seperti ini, kita tidak perlu untuk menentukan indeks pertama dan terakhir seperti pada contoh sebelumnya.

Contoh: Menghitung Jumlah Element Array

```
1 public class CalculateSumOfArray {
2     public static void main(String[] args) {
3         final int JUMLAH_MAHASISWA = 6;
4         double total_nilai=0, rata2;
5         double[] nilai_mahasiswa;
6         nilai_mahasiswa = new double [JUMLAH_MAHASISWA];
7         nilai_mahasiswa[0] = 80.5;
8         nilai_mahasiswa[1] = 57;
9         nilai_mahasiswa[2] = 73;
10        nilai_mahasiswa[3] = 67;
11        nilai_mahasiswa[4] = 87;
12        nilai_mahasiswa[5] = 56;
13        for(double nilai: nilai_mahasiswa) {
14            total_nilai = total_nilai + nilai;
15        }
16        rata2 = total_nilai / JUMLAH_MAHASISWA;
17        System.out.println("Total Nilai Mahasiswa = " + total_nilai);
```

```
18     System.out.println("Rata-Rata Nilai Mahasiswa = " + rata2);  
19 }  
20 }
```

Array Multidimensi

Sederhananya array multidimensi ialah array didalam array. Mengapa array multidimensi diperlukan? Banyak sekali data terutama data yang kompleks seperti tabel atau matrix. Tabel atau matriks ialah array dua dimensi, dimana dimensi pertama ialah kolom dan dimensi kedua ialah baris.

Indeks. Array dengan banyak dimensi memiliki banyak indeks. Seperti tabel yang memiliki 2 dimensi, maka setiap dimensi memiliki indeksnya masing-masing.

Contoh Tabel

Distance Table (in miles)							
	Chicago	Boston	New York	Atlanta	Miami	Dallas	Houston
Chicago	0	983	787	714	1375	967	1087
Boston	983	0	214	1102	1763	1723	1842
New York	787	214	0	888	1549	1548	1627
Atlanta	714	1102	888	0	661	781	810
Miami	1375	1763	1549	661	0	1426	1187
Dallas	967	1723	1548	781	1426	0	239
Houston	1087	1842	1627	810	1187	239	0

Figure: Matrix jarak antar kota

Mendeklarasikan dan Membuat Array multidimensi

Mendeklarasikan dan membuat array multidimensi sama dengan membuat array satu dimensi seperti penjelasan sebelumnya. Namun yang perlu diperhatikan ialah indeksnya harus sama dengan jumlah dimensi.

```
1 final int KOLOM = 5;
2 final int BARIS = 5;
3 int matrix[][];
4 matrix = new int[5][5];
```

Mendeklarasikan dan membuat array multidimensi

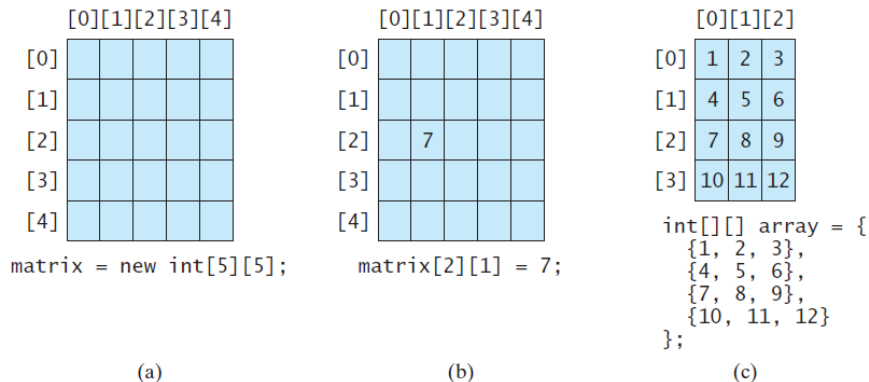


FIGURE 7.1 The index of each subscript of a two-dimensional array is an **int** value, starting from **0**.

Figure: Representasi nilai element pada array multidimensi

Mengisi Value element array multidimensi

```
1 final int KOLOM = 5;
2 final int BARIS = 5;
3 int matrix[][];
4 matrix = new int[5][5];
5 matrix[0][0] = 7;
6 matrix[0][1] = 5;
7 matrix[0][2] = 2;
8 matrix[0][3] = 1;
9 matrix[0][4] = 9;
10 matrix[1][0] = 3;
11 //...
```

Perulangan pada Array Multidimensi

Untuk melakukan perulangan terhadap element array multidimensi ialah sama dengan perulangan pada satu dimensi yang telah dibahas sebelumnya. Namun, jika pada array satu dimensi menggunakan satu perulangan, maka pada array multidimensi tergantung pada jumlah dimensi. Berikut ialah contoh perulangan pada array dua dimensi menggunakan for menggunakan contoh code sebelumnya:

```

1  for(int i = 0; i < BARIS; i++) {
2      for(int j = 0; j < KOLOM; j++) {
3          // Code
4          // contoh
5          System.out.println(matrix[i][j]);
6      }
7      // Code
8  }
```

Perulangan pada Array Multidimensi

Sedangkan dengan menggunakan perulangan for-each, contoh perulangan dua dimensi ialah sebagai berikut

```
1 for(int[] baris: matrix) {
2     for(value: baris) {
3         // Code
4         // Contoh
5         System.out.println(value);
6     }
7     // Code
8 }
```

Contoh: Perulangan pada Array Multidimensi

```

1 public class ExMuldimentionArray {
2     public static void main(String[] args) {
3         final int BARIS = 5;
4         final int KOLOM = 5;
5         int jumlah = 0;
6         // DEKLARASI ARRAY MULTIDIMENSI
7         int matrix[][];
8         // MEMBUAT ARRAY MULTIDIMENSI DENGAN UKURAN 5X5
9         matrix = new int[5][5];
10        // MENGISI ELEMENT ARRAY 5X5
11        matrix[0][0] = 7;
12        matrix[0][1] = 5;
13        matrix[0][2] = 2;
14        matrix[0][3] = 1;
15        matrix[0][4] = 9;
16        matrix[1][0] = 3;
17        matrix[1][1] = 9;
    
```

```

18     matrix[1][2] = 1;
19     matrix[1][3] = 2;
20     matrix[1][4] = 3;
21     matrix[2][0] = 4;
22     matrix[2][1] = 5;
23     matrix[2][2] = 6;
24     matrix[2][3] = 7;
25     matrix[2][4] = 8;
26     matrix[3][0] = 9;
27     matrix[3][1] = 0;
28     matrix[3][2] = 4;
29     matrix[3][3] = 7;
30     matrix[3][4] = 9;
31     matrix[4][0] = 3;
32     matrix[4][1] = 4;
33     matrix[4][2] = 7;
34     matrix[4][3] = 9;
35
36     // Perulangan for
37     System.out.println("Perulangan dengan for: ");
38     for(int i=0; i< BARIS; i++) {

```

```

39     for(int j=0; j< BARIS; j++) {
40         System.out.print(matrix[i][j] + " ");
41     }
42     System.out.println();
43 }
44
45 // Perulangan for-each
46 System.out.println("Perulangan dengan for-each: ");
47 for(int[] baris : matrix) {
48     for(int value : baris) {
49         System.out.print(value + " ");
50     }
51     System.out.println();
52 }
53
54 // MENGHITUNG JUMLAH
55 System.out.print("JUMLAH NILAI ELEMENT = ");
56 for(int[] baris : matrix) {
57     for(int value : baris) {
58         jumlah = jumlah + value;
59     }

```

```

60     }
61     System.out.print(jumlah);
62 }
63 }

```

Array sebagai Parameter pada Method

Samahalnya seperti data yang lain, kita dapat memberikan parameter berupa array pada method dengan cara pendeklarasian yang mirip. Perhatikan formatnya sebagai berikut

```

1 modifier returnType methodName(type par0, type[] par1, type[][]
    par2, type[][][] par3, ... ) {
2     //Method body
3
4 }
```

Jika kalian perhatikan pendeklarasian array dan non-array memiliki format yang mirip, hal yang membedakan ialah tanda "[]". Kemudian, penggunaan didalam method sama seperti yang telah dijelaskan pada penjelasan dan contoh sebelumnya.

Contoh: Method untuk Menghitung n buah element array

```

1  import java.util.Scanner;
2
3  public class ExParametersArray {
4
5      public static void main(String[] args) {
6          int jumlahElementArray;
7          int angka[];
8          Scanner input = new Scanner(System.in);
9          System.out.print("Masukan jumlah angka = ");
10         jumlahElementArray = input.nextInt();
11         angka = new int[jumlahElementArray];
12
13         for(int i=0; i < jumlahElementArray; i++) {
14             System.out.print("Nilai ke-" + i + " = ");
15             angka[i] = input.nextInt();
16         }
17

```



```

18     System.out.println("Hasil penjumlahan = " + jumlahNilaiArray(angka
19     ));
20 }
21 public static int jumlahNilaiArray(int[] arr1) {
22     int jumlah=0;
23     for (int angka : arr1) {
24         jumlah += angka;
25     }
26     return jumlah;
27 }
28 }

```

Array sebagai Return Value pada Method

Samahalnya seperti data yang lain, array juga dapat dijadikan return value. Perhatikan format sebagai berikut

```

1  modifier returnType[] methodName(type par0, type[] par1, type
    [][] par2, type[][][] par3, ... ) {
2      //Method body
3
4
5  }
```

Jika kalian perhatikan bagian returnType memiliki format yang mirip, namun dengan tambahan tanda "[]" yang menandakan nilai balik ialah array satu dimensi. Jumlah "[]" dapat ditambah sesuai dengan jumlah dimensi seperti "[][]" untuk dua dimensi, "[][][]" untuk tiga dimensi dan seterusnya.

Contoh: Hasil Operasi Aritmatika dua buah bilangan bulat

```

1 public class ExArrayAsReturnValue {
2     public static void main(String[] args) {
3         int angka1 = 17, angka2 = 3;
4         double[] operasiAritmatika = aritmatikaDuaAngka(angka1, angka2);
5         System.out.println(angka1 + " + " + angka2 + " = " +
6             operasiAritmatika[0]);
7         System.out.println(angka1 + " - " + angka2 + " = " +
8             operasiAritmatika[1]);
9         System.out.println(angka1 + " * " + angka2 + " = " +
10            operasiAritmatika[2]);
11        System.out.println(angka1 + " / " + angka2 + " = " +
12            operasiAritmatika[3]);
13        System.out.println(angka1 + " % " + angka2 + " = " +
14            operasiAritmatika[4]);
15    }
16
17    public static double[] aritmatikaDuaAngka(int a, int b) {

```

```

13     int penjumlahan = a + b;
14     int pengurangan = a - b;
15     int perkalian = a * b;
16     double pembagian = (double) a / b;
17     int modulo = a % b;
18
19     double[] arr1 = new double[5];
20     arr1[0] = (double) penjumlahan;
21     arr1[1] = (double) pengurangan;
22     arr1[2] = (double) perkalian;
23     arr1[3] = pembagian;
24     arr1[4] = (double) modulo;
25
26     return arr1;
27 }
28 }

```
