

PRAKTIKUM 1

ARRAY

A. TUJUAN PEMBELAJARAN

1. Mengetahui mengenai konsep array.
2. Mengetahui cara membuat array dengan data primitif dan data objek.
3. Mengetahui cara menyimpan data dan mengakses data yang tersimpan di array.

B. DASAR TEORI

Misalkan kita memiliki tiga data yang berbeda dan kita simpan dalam variabel yang berbeda, maka kita akan mendefinisikan seperti berikut ini:

```
int number1;  
int number2;  
int number3;  
  
number1 = 1;  
number2 = 2;  
number3 = 3;
```

Bagaimana jika terdapat banyak data yang bertipe data sama dan memiliki tujuan yang berbeda? Bagaimana cara menyimpannya? Di Java dan bahasa pemrograman lainnya, ada satu kemampuan dimana kita dapat menggunakan satu nama variabel untuk menyimpan sekumpulan data bertipe data sama dan memanipulasinya secara lebih efisien. Tipe data ini disebut dengan Array. Ilustrasi sebuah array dapat dilihat pada Gambar 1.2.

Array pada Gambar 1.1 bernama number. Array number tersebut berukuran 3. Sehingga dapat diisi maksimal 3 elemen data. Data yang tersimpan dalam array disebut dengan elemen dan dalam memori tersimpan dalam alamat memori yang selalu berurutan. Elemen pertama tersimpan dalam indeks ke-0, elemen kedua tersimpan dalam indeks ke-1 dan seterusnya. Elemen terakhir akan tersimpan dalam indeks ke-(n-1). Dalam hal ini n adalah ukuran array.

	0	1	2
number:	1	2	3

Gambar 1.2 Array Dimensi 1

Array adalah obyek yang digunakan untuk menyimpan lebih dari satu data dengan tipe yang sama. Tipe dari array bisa tipe data primitif atau class. Terdapat 3 langkah untuk membuat array:

- Mendeklarasikan variabel array
- Meng-create array beserta ukurannya.
- Memberikan sebuah nilai pada setiap element array.

Ketiga langkah tersebut bisa dikombinasikan, dalam satu statement, kita dapat mendeklarasikan sekaligus membuat array. Atau membuat array sekaligus menginisialisasi elemen array.

Mendeklarasikan variabel array

Mendeklarasikan array adalah membuat referensi array. Ketika array dideklarasikan, array ini masih belum memiliki ukuran. *Blank space* tidak berpengaruh terhadap peletakan tanda []. Cara mendeklarasikan array adalah sebagai berikut:

- `int[] ints;`
- `double[] dubs;`
- `Dimension[] dims;`
- `float[][] twoDee;`

Berikut ini adalah contoh mendeklarasikan array yang ilegal:

```
int [5] scores;
```

Membuat Array

Pada saat pendeklarasian array:

```
int nilai[];
```

yang terjadi adalah 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 sebagai sebuah obyek.

Format penulisan pembuatan array adalah sebagai berikut:

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

Karena array adalah sebuah obyek, maka cara create array dengan menggunakan operator `new`. Besar array ditentukan pada saat runtime

- `int[] ints ;`
- `ints = new int[25] ;`

Pada saat array dibuat, isi array secara otomatis akan diinisialisasi dengan nilai default value.

Memberikan sebuah nilai pada setiap element array

Untuk mengakses data/elemen di array kita menggunakan indeks. Data ke-1 disimpan di array indeks ke-0. Isi dari array : nilai primitif

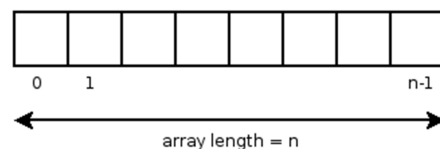
```
int scores = new int[3];
scores[0] = 75;
scores[1] = 80;
scores[2] = 100;
```

Mendeklarasikan dan membuat array.

Kegiatan mendeklarasikan dan membuat array dapat dilakukan dengan sekali langkah, yaitu :

```
int[] ints = new int[25] ;
```

Statemen diatas membuat array bernama `ints` yang berukuran 25 dan bertipe data `int`. Ilustrasi kegiatan ini dapat dilihat pada Gambar 1.3. Data disimpan mulai dari indeks 0 sampai 24. Pada saat array di buat, isi array diinisialisasi dengan default value, nilai default untuk `int` adalah 0. Ilustrasi penggunaan memori `ints` dapat dilihat di Gambar 1.2. Nilai `n-1` dalam hal ini adalah `25-1 = 24`, jadi elemen ke-25 akan ditaruh dalam indeks ke-24.



Gambar 1.3 Cara penyimpanan data menggunakan indeks

Inisialisasi Array

Ketika dibuat, isi array juga bisa langsung diinisialisasi dengan cara sebagai berikut:

```
float[] diameters = {1.1f, 2.2f, 3.3f, 4.4f,5.5f}
Point[] markup = {new Point(1,5), new Point(3,3), new Point(2,3)} ;
```

Besar Array

Untuk mengetahui besar dari array gunakan properti `.length` setelah nama array.

```
long squares ;
squares = new long[6000] ;

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

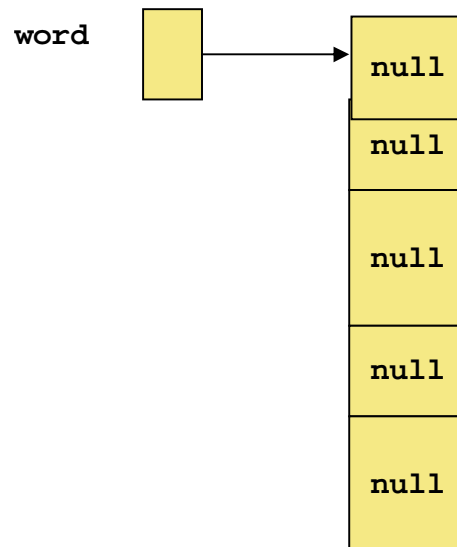
Arrays of Objects

Elemen dari array dapat berupa obyek. Contoh berikut ini menyiapkan ruang untuk menyimpan 5 objek String.

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

Pada saat menginisialisasi array dengan tipe obyek, nilai default dari elemen array adalah null. Ilustrasi kegiatan ini dapat dilihat pada Gambar 1.4.

Array words



Gambar 1.4 Array dengan tipe String pada saat awal dicreate

Selanjutnya bila ingin mengisi array words lakukan dengan cara berikut:

```
words[0] = new String("friendship");
words[1] = "loyalty";
words[2] = "honor";
```

Obyek String dapat dibuat menggunakan literal. Contoh dibawah ini, membuat obyek array dengan menginisialisasi array, array yang dibuat diberi nama verbs, memiliki panjang 4 dan diisi dengan 4 elemen berupa obyek String, yang dibuat menggunakan string literal.

```
String[] verbs = {"play", "work", "eat", "sleep"};
```

Contoh lain membuat array object

```
Student students[] = new Students[3];
students[0] = new Student();
students[1] = new Student();
students[2] = new Student();
```

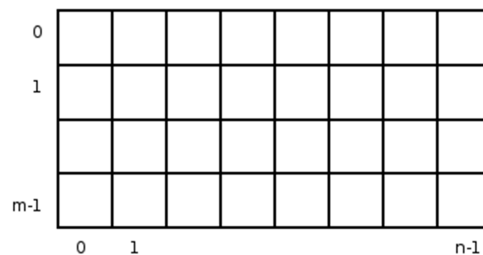
dan

```
MyDate date[] = { new MyDate(1,1,2012),
                  new MyDate(1,2,2012),
                  new MyDate(1,3,2012)};
```

Membuat array dua dimensi

Array dimensi dua dapat dianggap seperti tabel elemen dengan m baris dan n kolom.

Gambar 1.5 adalah ilustrasi array dua dimensi.



Gambar 1.5 Array 2 dimensi

Contoh pembuatan array dengan 4 baris dan 5 kolom adalah sebagai berikut:

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

Berikut ini adalah array dua dimensi yang memiliki ukuran tidak beraturan:

```
int twoDim[][] = new int[4][];
twoDim[0] = new int[2];
twoDim[1] = new int[4];
twoDim[2] = new int[6];
twoDim[3] = new int[8];
```

Merubah ukuran 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.

Isi dari suatu array dapat kita kopi pada array yang lain dengan memanfaatkan method `arraycopy()` pada class `System`. Format penulisannya sebagai berikut :

```
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 lengkap dari program untuk mengkopi array adalah sebagai berikut:

```
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 lain dari program mengkopi isi array adalah sebagai berikut:

```
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));
    }
}
```

Output program ini adalah: `caffein`

Suatu array juga dapat me-refer (merujuk) ke array yang lain, dengan kata lain merujuk pada alamat memori yang sama. Sebagai contoh adalah program berikut ini :

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

Di baris ketiga, kita meng-*assign* array nilai ke array result. 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 lengkap dari program untuk merefer array adalah sebagai berikut:

```
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]);
    }
}
```

Selanjutnya diberikun contoh program untuk mengakses elemen array dengan menggunakan 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]);
        }
    }
}
```

Keluaran program ini adalah sebagai berikut:

```
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
```

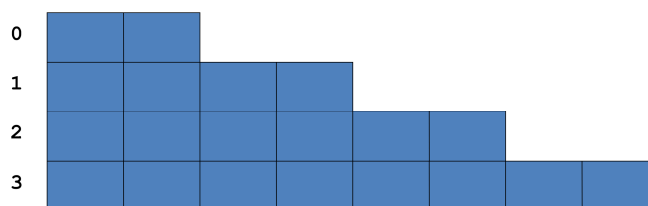
Versi lain penggunaan perulangan for untuk mengakses elemen array adalah sebagai berikut:

```
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);
        }
    }
}
```

Membuat array dua dimensi bukan persegi panjang

Dijava membuat array dua dimensi, untuk setiap baris dapat memiliki jumlah kolom yang berbeda. Sebagai contoh pada gambar 1.7, baris 0 mempunyai 2 kolom, baris 1 mempunyai 4 kolom, baris 2 mempunyai 6 kolom, baris 3 mempunyai 8 kolom.

```
int twoDim[][] = new int[4][]
twoDim[0] = new int[2];
twoDim[1] = new int[4];
twoDim[2] = new int[6];
twoDim[3] = new int[8];
```



Gambar 1.7 Array Dimensi 2 bukan persegi panjang

C. TUGAS PENDAHULUAN

Buatlah review mengenai array, berikan 1 contoh cara membuat array dengan tipe data primitif dan 1 contoh cara mendeklarasikan array dengan tipe objek.

D. PERCOBAAN

Percobaan 1 : Cara membuat array, menyimpan data di array dan mengakses data dari array. Data yang digunakan adalah data dengan tipe primitif.

```
class ArrayDemo {
    public static void main(String[] args) {
        // declares an array of integers
        int[] anArray;

        // allocates memory for 10 integers
        anArray = new int[10];

        // initialize first element
        anArray[0] = 100;
        // initialize second element
        anArray[1] = 200;
        // etc.
        anArray[2] = 300;
        anArray[3] = 400;
        anArray[4] = 500;
        anArray[5] = 600;
        anArray[6] = 700;
        anArray[7] = 800;
        anArray[8] = 900;
        anArray[9] = 1000;

        System.out.println("Element at index 0: " + anArray[0]);
        System.out.println("Element at index 1: " + anArray[1]);
        System.out.println("Element at index 2: " + anArray[2]);
        System.out.println("Element at index 3: " + anArray[3]);
        System.out.println("Element at index 4: " + anArray[4]);
        System.out.println("Element at index 5: " + anArray[5]);
        System.out.println("Element at index 6: " + anArray[6]);
        System.out.println("Element at index 7: " + anArray[7]);
        System.out.println("Element at index 8: " + anArray[8]);
        System.out.println("Element at index 9: " + anArray[9]);
    }
}
```

Percobaan 2 : Mengurutkan data pada array.

```
import java.util.*;

public class array{
    public static void main(String[] args){
        int num[] = {50,20,45,82,25,63};
        int l = num.length;
        int i,j,t;
        System.out.print("Given number : ");
        for (i = 0;i < l;i++ ){
            System.out.print(" " + num[i]);
        }
        System.out.println("\n");
        System.out.print("Accending order number : ");
    }
}
```

```

Arrays.sort(num);
    for(i = 0;i < l;i++){
        System.out.print(" " + num[i]);
    }
}
}

```

Percobaan 3 : Mengurutkan data dan menyisipkan data pada array

```

import java.util.Arrays;

public class MainClass {
    public static void main(String args[]) throws Exception {
        int array[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };
        Arrays.sort(array);
        printArray("Sorted array", array);
        int index = Arrays.binarySearch(array, 1);
        System.out.println("Didn't find 1 @ "
            + index);
        int newIndex = -index - 1;
        array = insertElement(array, 1, newIndex);
        printArray("With 1 added", array);
    }
    private static void printArray(String message, int array[]) {
        System.out.println(message
            + ": [length: " + array.length + "]");
        for (int i = 0; i < array.length; i++) {
            if (i != 0){
                System.out.print(", ");
            }
            System.out.print(array[i]);
        }
        System.out.println();
    }
    private static int[] insertElement(int original[],
        int element, int index) {
        int length = original.length;
        int destination[] = new int[length + 1];
        System.arraycopy(original, 0, destination, 0, index);
        destination[index] = element;
        System.arraycopy(original, index, destination, index
            + 1, length - index);
        return destination;
    }
}

```

Percobaan 4 : Untuk mengetahui apakah dua array sama atau tidak.

```

import java.util.Arrays;

public class Main {
    public static void main(String[] args) throws Exception {
        int[] ary = {1,2,3,4,5,6};
        int[] ary1 = {1,2,3,4,5,6};
    }
}

```

```

int[] ary2 = {1,2,3,4};
System.out.println("Is array 1 equal to array 2?? "
+Arrays.equals(ary, ary1));
System.out.println("Is array 1 equal to array 3?? "
+Arrays.equals(ary, ary2));
    }
}

```

Percobaan 5 : Mengetahui panjang baris dan kolom pada array dimensi dua.

```

public class Main {
    public static void main(String args[]) {
        String[][] data = new String[2][5];
        System.out.println("Dimension 1: " + data.length);
        System.out.println("Dimension 2: " + data[0].length);
    }
}

```

Percobaan 6 : array dua dimensi

```

public class twoDimension{
    public static void main(String[] args) {
        int[][] a2 = new int[10][5];
        for (int i=0; i<a2.length; i++) {
            for (int j=0; j<a2[i].length; j++) {
                a2[i][j] = i;
                System.out.print(" " + a2[i][j]);
            }
            System.out.println("");
        }
    }
}

```

Percobaan 7 : array dua dimensi non persegi panjang

```

public class twoDimension2{
    public static void main(String[] args) {

        int myArray[][] = { {1, 2 }, { 3 }, { 4, 5, 6 } };
        for ( int i = 0; i < myArray.length; i++ )
        {
            System.out.println("Row number " + i + " has " + myArray[i].length + "
columns." );
        }//end for

        for ( int i = 0; i < myArray.length; i++ ) {
            for ( int j = 0; j < myArray[i].length; j++ ) {
                System.out.print(myArray[i][j]+" ");
            }
            System.out.println("");
        }
    }
}

```

```
}
```

E. LATIHAN

Latihan 1 : Terdapat array dengan data 23, 6, 47, 35, 2, 14, lakukan seperti di bawah ini !

- Urutkan data pada array
- Hitung rata-rata dari data tersebut.
- Dapatkan data minimum dan maximum.
- Tampilkan dari data tersebut yang merupakan bilangan ganjil.
- Tampilkan dari data tersebut yang merupakan bilangan prima.

Latihan 2 :

Terdapat array dua dimensi dengan tipe String, kolom 1 untuk menyimpan warna, kolom 2 untuk menyimpan sifat dari warna tersebut. Buatlah program seperti di bawah ini.

Input :

Warna yang anda sukai : Ungu

Output :

Karakter Anda :

Warna Ungu adalah pribadi dengan tipe yang benar-benar luar biasa. Selalu optimis, dalam menghadapi masa depan kamu tidak pernah ragu-ragu, apa yang dikerjakan kamu adalah yang terbaik. Kamu pandai benar dalam mengikuti perkembangan jaman. Dalam bercinta, hanya merekalah yang kuat mental yang bisa mendekati dan menjadi kekasih kamu.

Warna Biru

Warna kesukaan kamu Biru, biasanya termasuk tipe pemurung, selalu menyenangkan dan selalu bertindak pasif dalam segala hal. Selalu mengharapkan kedamaian dan ketenangan. Kamu memiliki kesulitan dalam pergaulan. Demikian pula dalam bercinta karena kamu sering sekali menyembunyikan perasaan.

Warna Hitam

Kamu termasuk tipe orang yang sangat lincah dalam hal-hal tertentu saja. Jika berada dilingkungan yang tidak disukai, maka kamu akan menjadi murung. Kamu selalu tampil menarik, rapi, cukup banyak lawan jenis berusaha mengejar dan merebut cinta kamu.

Warna Merah

Kamu sangat berwibawa dan juga senang melindungi orang yang lemah. Walau sering kali bergaul dan bercanda tapi kamu bisa menahan diri. Banyak orang mengatakan cinta, tapi kamu selalu berpikir dan berpikir lagi. Kamu termasuk tipe yang sulit jatuh cinta.

Warna Hijau

Jika warna favorit kamu hijau, maka kamu adalah tipe yang sangat romantik, menyukai keindahan, menyenangkan alam dengan udara yang sejuk. Kamu adalah seseorang yang selalu memegang prinsip. Dalam hal bercinta kamu mengidam-idamkan calon teman hidup yang penuh toleransi dan dapat dipercaya.

Warna Kuning

Kesukaan kamu warna kuning menandakan bahwa kamu memiliki sifat optimis. Kamu tipe periang dan senang bergaul, tidak memiliki penampilan yang loyo. Sifat tolong-menolong selalu ada dalam diri kamu, karena menolong merupakan suatu kewajiban mutlak bagi kamu. Kamu orang yang tidak pernah meremehkan siapapun juga, walaupun seseorang itu dungu atau bloon.

Warna Putih

Kamu adalah orang yang dilahirkan ke dunia dengan sempurna, jika menyukai warna putih, banyak orang mengagumi kamu karena sifat angun, sifat idealis dan moral kamu yang teramat tinggi. Tak pernah angkuh, senang menolong siapa saja yang membutuhkan bantuan kamu.

Warna Ungu

Warna Ungu adalah pribadi dengan tipe yang benar-benar luar biasa. Selalu optimis, dalam menghadapi masa depan kamu tidak pernah ragu-ragu, apa yang dikerjakan kamu adalah yang terbaik. Kamu pandai benar dalam mengikuti perkembangan jaman. Dalam bercinta, hanya merekalah yang kuat mental yang bisa mendekati dan menjadi kekasih kamu.

F. LAPORAN RESMI

Kerjakan hasil percobaan(D) dan latihan(E) di atas dan tambahkan analisa.