

ARRAY

OBJEKTIF :

1. Mahasiswa Mampu Memahami *Array* pada Java.
 2. Mahasiswa Mampu Menggunakan *Software* IntelliJ IDEA dalam Pembuatan Program dengan *Array*.
-

6.1 DEKLARASI DAN PENCIPTAAN ARRAY

`Array` merupakan sebuah bentuk yang dapat menyimpan banyak data atau *item* dengan tipe data yang sama dalam sebuah variabel. *Item-item* pada `array` merupakan data sederhana atau komposit. `Array` pada Java dideklarasikan dengan menggunakan simbol kurung siku `[]`. `Array` merupakan cara yang lebih efektif dan sederhana dalam mengelompokkan informasi yang berhubungan. Berikut ini adalah sintaks umum deklarasi `array`:

```
tipe_data nama_variabel[];
```

atau

```
tipe_data[] nama_variabel;
```

`tipe_data` merupakan deklarasi tipe basis dari `array` tersebut. Tipe basis menentukan tipe data bagi masing-masing elemen yang membentuk `array`. Dengan demikian, `tipe_data` untuk `array` digunakan untuk menentukan tipe data yang dimuat oleh `array`. Pada setiap `array` terdapat indeks yang memudahkan untuk mengakses data yang disimpan. Indeks merupakan ukuran yang terdapat di dalam simbol kurung siku. Indeks tersebut dimulai dari angka 0 atau berupa teks. Berikut ini merupakan contoh deklarasi `array` di Java:

```
int numbers[];  
char[] letters;  
long grid[][];
```

Pada Java, saat mendeklarasikan `array` kita tidak perlu mendeklarasikan ukuran `array`. Namun, jika akan menyatakan ukuran `array`, maka dilakukan secara eksplisit saat pendeklarasian menggunakan operator `new()` atau dengan mendaftarkan *item-item* untuk `array`. Operator `new()` berfungsi untuk membuat *object* atau `array` baru.

Selain itu, pada Java penempatan kurung siku pada deklarasi `array` dilakukan setelah tipe data atau setelah nama variabel. Berikut ini merupakan contoh pendeklarasian `array` dengan mendeklarasikan ukuran indeksinya dan menambahkan operator `new()` serta dengan memberikan daftar *item* yang termasuk di dalam `array`.

```
String[] nama = new String[5];
nama[0] = "Nana";
nama[1] = "Tia";
nama[2] = "Adit";
nama[3] = "Dika";
nama[4] = "Ayana";
System.out.println(nama[2]);
```

6.2 ARRAY 1 DIMENSI

Array 1 dimensi merupakan array tunggal yang tidak memiliki array lain di dalam indeksinya dan terdiri dari 1 simbol array saja. Array 1 dimensi merupakan daftar variable bertipe serupa. Kita diharuskan mendeklarasikan terlebih dahulu variable array dari tipe yang dikehendaki. Berikut ini merupakan contoh dari array 1 dimensi:

```
int monthDays[];
```

monthDays adalah variable array, namun belum ada ukuran array yang diciptakan. Nilai dari monthDays adalah nilai *null* yang merepresentasikan array tanpa nilai. Untuk membuat monthDays dengan ukuran array bilangan bulat aktual dan fisik, kita harus mendeklarasikan menggunakan operator `new()`. Operator `new()` merupakan operator khusus untuk membuat array baru atau *object*. Bentuk umum `new()` untuk array 1 dimensi adalah sebagai berikut:

```
nama_variabel = new() tipe_data[ukuran_array]
```

Pada potongan program tersebut, `tipe_data` mendeklarasikan tipe data yang dialokasikan, sedangkan `ukuran_array` menspesifikasikan jumlah elemen array. Dengan demikian, untuk menggunakan `new()`, kita harus menspesifikasikan `tipe_data` dan jumlah elemen untuk dialokasikan. Elemen-elemen di array dialokasikan oleh `new()` akan secara otomatis membuat indeks dari array dimulai dari nol. Berikut ini adalah contoh deklarasi dengan operator `new()`:

```
monthDays = new() int[12];
```

Setelah pernyataan di atas dieksekusi, `monthDays` akan membuat ukuran array berisi 12 bilangan bertipe integer, kedua belas angka tersebut dimulai dengan indeks nol.

Proses untuk memperoleh array terdapat dua langkah, yaitu:

1. Deklarasi variable dari tipe array yang dikehendaki.
2. Alokasi memori yang akan menyimpan array menggunakan operator `new()` dan memberikannya ke variabel array.

Maka dari itu, pada Java semua array dialokasikan secara dinamis. Dengan konsep dinamis array ini, maka akan lebih mempermudah dalam melakukan array. Setelah mengalokasikan array, kita dapat mengakses elemen spesifik di array dengan menspesifikasikan indeks di dalam pasangan kurung siku. Semua array dimulai dengan indeks nol. Berikut ini merupakan contoh pemberian nilai 29 ke indeks ke satu `monthDays`, kita menggunakan perintah berikut:

```
monthDays[1] = 29;
```

Berikut ini adalah menampilkan nilai yang disimpan di indeks pertama, yaitu:

```
System.out.println(monthDays[1]);
```

Berikut ini adalah penciptaan `array` yang berisi jumlah hari maksimum di masing-masing bulan.

```
package com.integratedlaboratory.program;

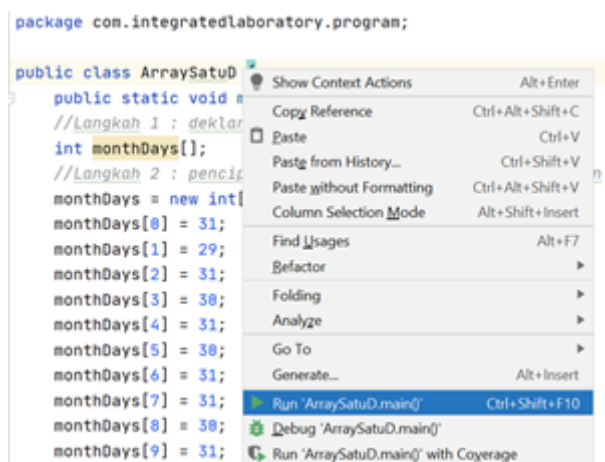
public class ArraySatuD {
    public static void main(String args[]){

        int monthDays[] = new int[12];;

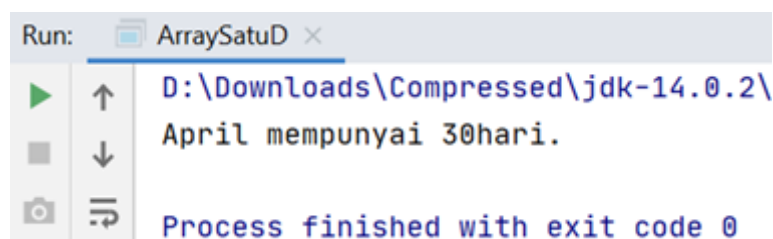
        monthDays[0] = 31;
        monthDays[1] = 29;
        monthDays[2] = 31;
        monthDays[3] = 30;
        monthDays[4] = 31;
        monthDays[5] = 30;
        monthDays[6] = 31;
        monthDays[7] = 31;
        monthDays[8] = 30;
        monthDays[9] = 31;
        monthDays[10] = 30;
        monthDays[11] = 31;

        System.out.println("April mempunyai " + monthDays[3] + "hari. ");
    }
}
```

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:



6.2.1 PEMERIKSAAN BATAS

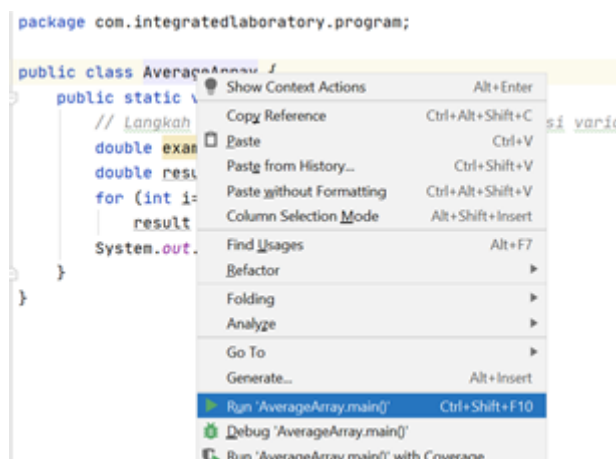
Java akan memeriksa batas yang menyimpan atau mengacu nilai di luar *range* dari `array`. Sistem *run-time* akan memeriksa untuk menjamin indeks berada di dalam *range* yang benar. Berdasarkan contoh, Java akan memeriksa nilai masing-masing indeks ke `monthDays` untuk menjamin indeks berada di antara 0 dan 11. Jika kita mencoba mengakses elemen di luar batas-batas indeks ini, maka akan diperoleh kesalahan *run-time*. Berikut ini adalah contoh penggunaan `array` untuk mencari rata-rata:

```
package com.integratedlaboratory.program;

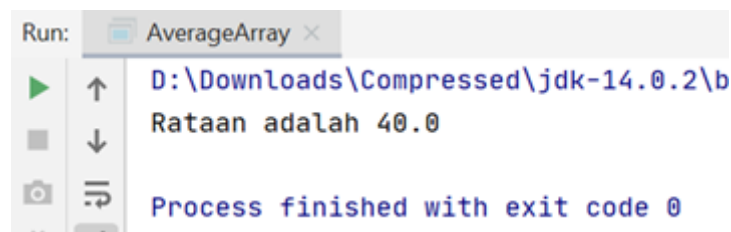
public static void main(String args[]){
    // Langkah 1: deklarasi, penciptaan, inisialisasi variabel array
    double exam_point[] = {20,30,40,50,60};
    double result = 0;
    for (int i=0;i<5;i++)
        result = result + exam_point[i];
    System.out.println("Rataan adalah "+result/5);
}
```

Perintah:

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:



6.3 ARRAY MULTI DIMENSI

Pada Java, array multi dimensi merupakan array yang di dalamnya terdapat array. Array multi dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang akan diakses melalui banyak indeks atau simbol. Array multi dimensi ditandai dengan memiliki lebih dari satu pasangan kurung siku. Sintaks untuk deklarasi array multi dimensi sebagai berikut:

```
tipe_data[][] nama_variabel = new tipe_data[][];
```

Contoh:

Berikut ini merupakan array dua dimensi bertipe int bernama `int2Darr` dideklarasikan dan diciptakan dengan perintah sebagai berikut:

```
int int2Darr[][] = new int[4][5];
```

Perintah tersebut mendeklarasikan dan mengalokasikan array 4 kali 5 dan memberikan ke variabel `int2Darr`. Secara internal, matriks ini diimplementasikan sebagai `array` dari `array` bertipe `int`. Secara konseptual `array` ini digambarkan sebagai berikut:



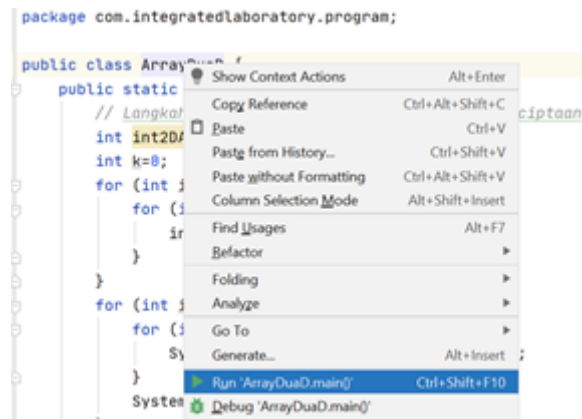
Program berikut memberi angka ke masing-masing elemen di array dari kiri ke kanan, puncak ke bawah, kemudian menampilkan masing-masing elemen ini:

```
package com.integratedlaboratory.program;

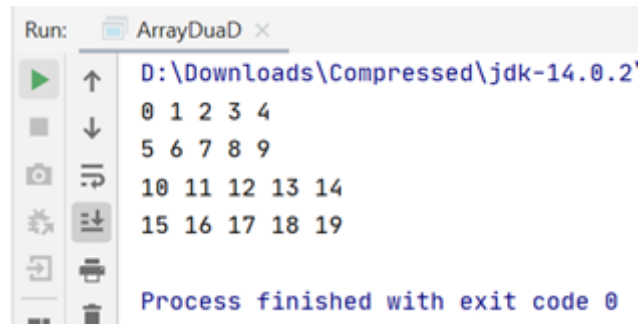
public class ArrayDuaD {
    public static void main(String args[]){
        // Langkah 1: deklarasi variabel array dan penciptaan
        int int2Darr[][] = new int[4][5];
        int k=0;
        for (int i=0;i<4;i++) {
            for (int j=0;j<5;j++){
                int2Darr[i][j] = k++;
            }
        }
        for (int i=0;i<4;i++) {
            for (int j=0;j<5;j++){
                System.out.print(int2Darr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Perintah:

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:



Ketika kita mengalokasikan array multi dimensi, kita dapat mendeklarasikan ukuran dimensi pertama (paling kiri) dan dimensi lainnya pada bagian sebelah kanan.

Dengan mengalokasikan dimensi pertama `int2DArr` saat dideklarasikan, kemudian dapat mengalokasikan dimensi kedua secara manual. Berikut ini merupakan contoh pendeklarasian array dua dimensi dengan ukuran yang berbeda:

```
//Deklarasi dan alokasi dimensi pertama
int int2DArrA = new() int[4][];

//Alokasi dimensi kedua secara manual
int2DArrA[0] = new() int[5];
int2DArrA[1] = new() int[5];
int2DArrA[2] = new() int[5];
int2DArrA[3] = new() int[5];
```

Pada kasus di atas, dalam fitur Java mengalokasikan dimensi kedua dilakukan secara individu atau satu per satu. Kelebihan dari melakukan alokasi dimensi secara manual satu per satu adalah dapat mengalokasikan jumlah elemen berbeda untuk masing-masing dimensi. Berikut ini adalah contoh :

```
package com.integratedlaboratory.program;

public class Array2DBeda {
    public static void main(String args[]) {
        // Langkah 1: deklarasi variabel array dan penciptaan
        int int2DUnequalArr[][] = new int[4][];
        int2DUnequalArr[0] = new int[1];
        int2DUnequalArr[1] = new int[2];
        int2DUnequalArr[2] = new int[3];
        int2DUnequalArr[3] = new int[4];
        int k = 0;
        for (int i = 0; i < 4; i++) {
```

```

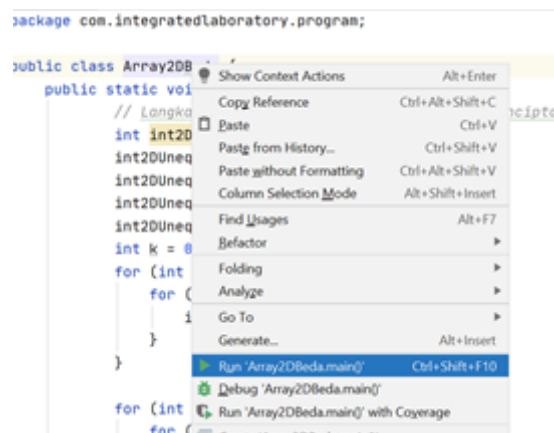
        for (int j = 0; j < i + 1; j++) {
            int2DUnequalArr[i][j] = k++;
        }
    }

    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < i + 1; j++) {
            System.out.print(int2DUnequalArr[i][j] + " ");
        }
        System.out.println();
    }
}
}

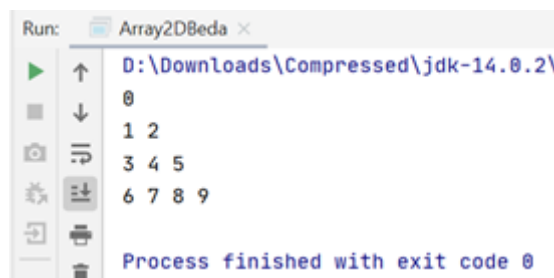
```

Perintah:

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:



Berikut ini merupakan contoh gambaran dari program yang mendeklarasikan array dua dimensi di mana ukuran dimensi kedua tidak sama:

[0] [0]			
[1] [0]	[1] [1]		
[2] [0]	[2] [1]	[2] [2]	
[3] [0]	[3] [1]	[3] [2]	[3] [3]

Penggunaan array multi dimensi yang berbeda jumlah indeks nya tidak direkomendasikan pada kebanyakan aplikasi karena bertentangan dengan harapan *user* mengenai pengertian umum array multi dimensi. Akan tetapi, teknik menggunakan array multi dimensi akan sangat efektif jika digunakan untuk situasi yang memerlukan array dua dimensi dalam jumlah isi yang sangat besar dan dengan isi jarang, sehingga banyak elemen di *array tidak digunakan. Karena, pada situasi ini array tidak seragam, sehingga merupakan solusi yang ideal.

6.3.1 INISIALISASI

Kita dapat melakukan inisialisasi array multi dimensi cukup dengan mengapit masing-masing inisialisasi dimensi dengan satu pasangan kurung kurawal. Berikut ini adalah contoh dari pendeklarasian array 2 dimensi :

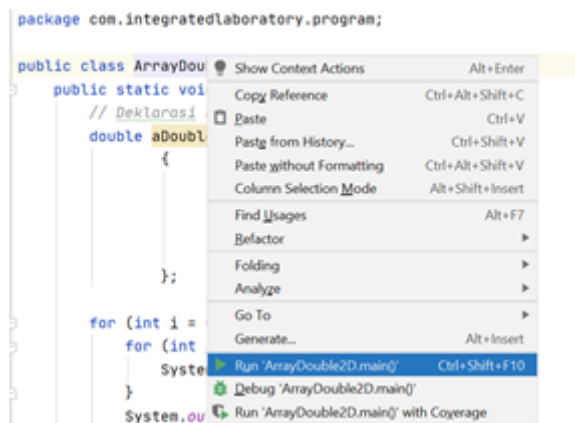
```
package com.integratedlaboratory.program;

public class ArrayDouble2D {
    public static void main(String args[]) {
        // Deklarasi dan inisialisasi
        double aDoubleMat[][] =
        {
            {0 * 0, 1 * 0, 2 * 0, 3 * 0},
            {0 * 1, 1 * 1, 2 * 1, 3 * 1},
            {0 * 2, 1 * 2, 2 * 2, 3 * 2},
            {0 * 3, 1 * 3, 2 * 3, 3 * 3}
        };

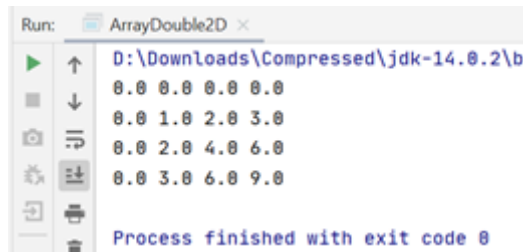
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.print(aDoubleMat[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Perintah:

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:



```
Run: ArrayDouble2D
D:\Downloads\Compressed\jdk-14.0.2\b
0.0 0.0 0.0 0.0
0.0 1.0 2.0 3.0
0.0 2.0 4.0 6.0
0.0 3.0 6.0 9.0
Process finished with exit code 0
```

Masing-masing baris di array diinisialisasi sebagai spesifik daftar inisialisasi. Berikut ini adalah contoh dari pendeklarasian array 3 dimensi:

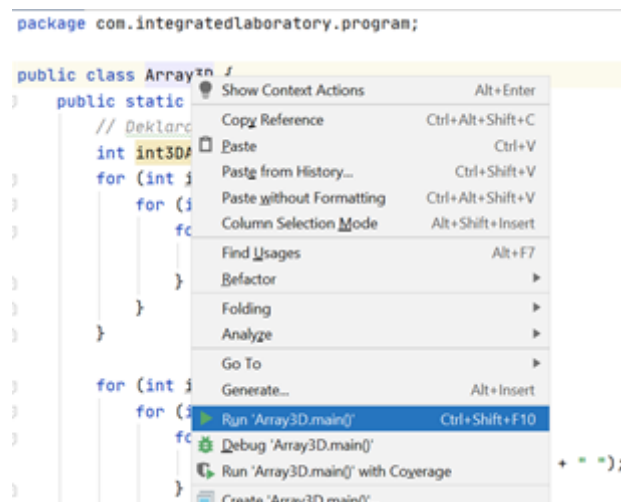
```
package com.integratedlaboratory.program;

public class Array3D {
    public static void main(String args[]){
        // Deklarasi dan alokasi
        int int3DArr[][][] = new int[3][4][5];
        for (int i=0;i<3;i++) {
            for (int j=0;j<4;j++){
                for (int k=0;k<5;k++) {
                    int3DArr[i][j][k] = i * j * k;
                }
            }
        }

        for (int i=0;i<3;i++) {
            for (int j=0;j<4;j++){
                for (int k=0;k<5;k++) {
                    System.out.print(int3DArr[i][j][k] + " ");
                }
                System.out.println();
            }
            System.out.println();
        }
    }
}
```

Perintah:

Tekan tombol Ctrl+Shift+F10 untuk melakukan Run pada IntelliJ IDEA atau dengan melakukan klik kanan pada file java seperti berikut:



Hasil program:

Run: Array3D x

▶

↑

■

📷

⚙️

🔄

📊

📌

↓

🔄

🗑️

D:\Downloads\Compressed\jdk-14.0.2'
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12

0 0 0 0 0
0 2 4 6 8
0 4 8 12 16
0 6 12 18 24

Process finished with exit code 0

REFERENSI :

- [1] Hariyanto, Bambang. 2017. *Esensi-Esensi Bahasa Pemrograman Java Revisi Kelima*. Bandung: Informatika.
- [2] Muhardian, Ahmad. 2016. "Belajar Java: Menggunakan Array untuk Menyimpan Banyak Hal", <https://www.petanikode.com/java-array/>. Diakses pada 17 Juli 2020.