

Modul 3 : ADT Array

3.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum = **170 menit**, dengan rincian sebagai berikut (misalkan):

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan tugas / Studi Kasus
- 50 menit **Pengayaan**

3.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Memahami konsep array 1 dimensi dan 2 dimensi
2. Memahami cara membuat array 1 dimensi
3. Memahami cara menyimpan dan mengakses data yang tersimpan di dalam array

3.3 Alat & Bahan

1. Komputer
2. Java IDE

3.4 Dasar Teori

Array adalah object yang digunakan untuk menyimpan banyak data dengan tipe data yang sama. Tipe data dari array dapat berupa int, float, string ataupun dapat berupa class. Array dapat dibentuk mulai dari 1 dimensi, 2 dimensi sampai "n" dimensi. Sebuah array dapat diibaratkan sebuah tabel yang memiliki indeks dan isi dari array tersebut. Seluruh isi dari array memiliki tipe data yang sama dan dapat diisi sesuai kebutuhan. Sedangkan indeks dari array, adalah penomoran yang secara default diberikan kepada array tersebut. Pengisian indeks array selalu dilakukan mulai dari 0, 1, 2 dst.

Indeks	→	0	1	2	3	4
Isi array	→	9	4	2	5	4

Gambar 0.1 Array 1 Dimensi

Pada Gambar 0.1 array 1 dimensi memiliki 5 data di dalamnya, sehingga indeks terkecil dimulai dari 0 dan indeks terbesar adalah 4. Indeks merepresentasikan isi data dalam array misalkan array dengan indeks 3 memiliki data yang bernilai 5.

Indeks						
		↓				
Indeks	→	0	1	2	3	4
0		9	5	8	2	7
1		5	8	2	7	9
2		2	9	5	7	8
3		5	8	7	9	2
4		7	2	9	8	5

Gambar 0.2 Array 2 Dimensi

Pada Gambar 0.2 tentang array 2 dimensi, sedikit berbeda. Array 2 dimensi memiliki hal seperti tabel dan kolom dengan indeks yang merepresentasikan tabel dan kolom tersebut. Indeks dimulai dari 0,0 hingga 4,4. Contoh indeks data 2,4 memiliki data yang bernilai 9.

Mendeklarasikan variabel array

Terdapat 3 langkah untuk membuat array:

1. Mendeklarasikan variabel array
2. Men-create array beserta ukurannya
3. Memberikan Sebuah nilai pada setiap elemen array

Untuk mendeklarasikan variabel array dapat dibuat dengan beberapa format. Array 1 dimensi dan 2 ataupun “n” dimensi memiliki beberapa pemberian format yang mirip, namun pada beberapa hal tidak sama.

Array 1 dimensi dapat dibuat dengan format berikut

“tipe_data [ukuran] nama_array;”

“tipe_data” adalah tipe data yang dipergunakan dalam sebuah array, seluruh data pada array akan memiliki tipe data yang sama.

“ukuran” adalah panjang ukuran dari sebuah array tersebut

“nama_array” adalah nama variabel yang dipergunakan untuk array tersebut

Contoh:

Int [] angka;

String [] alphabet;

Double [] jarijari;

Sedangkan untuk array 2 dan “n” dimensi dengan format berikut

Untuk 2 dimensi *“tipe_data [ukuran_baris] [ukuran_kolom] nama_array;”*

Untuk “n” dimensi *“tipe_data [ukuran] [ukuran] nama_array;”*

Contoh :

Int [] [] angka;

Int [] [] [] angka;

“....” Dapat ditambahkan array / [] sesuai dengan kebutuhan akan data yang akan dipergunakan.

Membuat Array (Men-create array dengan ukurannya)

Karena array adalah sebuah *object*, maka array dibuat dengan operator *“new”*. Sehingga panjang data yang dimiliki array ditentukan pada saat runtime. Contoh :

```
int [ ] desimal;  
desimal = new int[10]
```

Pada saat array dibuat, isi dari array diinisialisasi dengan default *value*. Default *value* isi dari array adalah *“0”* untuk tipe data primitif numerik, *“\u0000”* untuk tipe data char, *“false”* untuk tipe data boolean

Memberikan nilai / isi array pada setiap elemen array

Mengisikan nilai pada array dapat dilakukan dengan cara sebagai berikut

Cara ke-1

```
double ratarata = new double[3];  
ratarata [0] = 21.32;  
ratarata [1] = 11.06;  
ratarata [2] = 80.36;
```

Cara ke-2

```
int [ ] angka = {4, 1, 5, 9, 0}
```

0	1	2
21.32	11.06	80.36

Gambar 0.3 Array yang didapatkan dari cara ke-1

Sehingga dari cara ke-1 didapatkan isi data seperti pada Gambar 0.3, sedangkan pada cara ke-2 didapatkan isi data dengan indeksinya seperti pada gambar Gambar 0.4 yang tiap indeksnya menyesuaikan dengan data yang diisikan.

0	1	2	3	4
4	1	5	9	0

Gambar 0.4 Array yang didapatkan dari cara ke-2

Pada Cara ke-1 indeks diberikan nilai isi array satu persatu. Sedangkan pada Cara ke-2 isi dari array diberikan untuk beberapa indeks secara langsung. Baik cara ke-1 maupun cara ke-2 adalah untuk pengisian nilai array pada array 1 dimensi. Sedangkan untuk array 2 ataupun “n” dimensi dapat dilakukan dengan cara ke-3 dan ke-4 berikut.

Cara ke-3

```
double ratarata = new double[2][2];
ratarata [0] [0] = 21.32;
ratarata [0] [1] = 11.06;
ratarata [0] [2] = 80.36;
ratarata [1] [0] = 40.26;
ratarata [1] [1] = 85.55;
ratarata [1] [2] = 13.41;
ratarata [2] [0] = 37.51;
ratarata [2] [1] = 74.30;
ratarata [2] [2] = 88.21;
```

Pada Cara ke-3 akan menghasilkan array seperti Gambar 0.5, array dimasukkan sesuai indeks yang diberikan. Indeks masing-masing merepresentasikan baris dan kolom dari data yang dimasukkan. Tiap data yang dimasukkan harus memiliki indeks baris dan kolom.

0	1	2
21.32	40.26	37.51
11.06	85.55	74.30
80.36	13.41	88.21

Gambar 0.5 Array yang didapatkan dari Cara ke-3

Cara ke-4

```
int [ ] angka = {4, 33, 1, 14}, {40, 12, 9, 66}, {21, 8, 12, 42}, {9, 37, 6, 1};
```

Pada cara ke 4, menghasilkan seperti Gambar 0.6. Tiap data pada cara ke-4 yang berada dalam “{ }” akan dimasukkan pada baris indeks ke-0 dan akan masuk pada kolom ke-0 sampai ke-4.

	0	1	2	3
0	4	33	1	14
1	40	12	9	66
2	21	8	12	42
3	9	37	6	1

Gambar 0.6 Array yang didapatkan dari Cara ke-4

Mengukur Besaran Array

Untuk mengukur besar dari array yang dimiliki dapat menggunakan fungsi. Misalkan jika deklarasi dari array memiliki ukuran sebesar 4000, namun isi dari array tidak penuh. Untuk mengetahui berapa data yang ada pada sebuah array yang berukuran tersebut.

"nama_array.length"

Fungsi *".length"* ditaruh dibelakang nama array dari array yang akan diukur panjangnya.

Implementasi dapat dilihat pada kode berikut untuk array 1 dimensi.

```
long diameter;
diameter = new long[4000] ;
for (int i=0 ; i<squares.length ;i++)
{ diameter[i] = i * i ;
}
```

Mengakses Nilai Array

Mengakses nilai array adalah sesuatu yang dilakukan untuk melihat / mengambil data yang ada pada sebuah array. Data pada dapat dilihat berdasar indeks yang dimiliki. Tidak harus urut dari indeks terkecil ke indeks yang paling besar untuk dapat melihat isi dalam array. Misalkan seperti pada array 1 dimensi berikut untuk memanggil array yang dimiliki pada Gambar 0.3.

```
System.out.println("Elemen pada Indeks 2 adalah angka ke-3, nominalnya adalah: " + anArray [2]);
```

Akan menghasilkan 80.36, karena angka tersebut adalah angka ke 3. Namun yang dipanggil adalah indeks yang kedua berdasarkan fungsi *"anArray[2]"*.

Sedangkan untuk array 2 dimensi berdasarkan Gambar 0.6 Menggunakan kode berikut

```
System.out.println("Elemen pada Indeks baris ke 0 dan kolom ke 2, nominalnya adalah: " + anArray [0] [2]);
```

Akan menghasilkan angka 1, karena yang diakses dari sebuah array adalah pada baris ke-[0] dan ke-[1] dari fungsi *"anArray [0] [2]"*.

3.5 Prosedur Praktikum

Prosedur dari praktikum ini adalah tuliskan dan jalankan kode berikut, kemudian tampilkan pada sub-bab hasil percobaan, analisislah pada sub-bab analisis hasil serta berilah kesimpulan untuk masing-masing percobaan yang diberikan pada sub-bab 0.5 ini.

Percobaan ke-1 : Membuat array, menyimpan array dan mengakses array 1 dimensi

```
1 class ArrayDemo {
2     public static void main(String[] args) {
3         int[] anArray;
4         anArray = new int[10];
5         // initialize each element array
6         anArray[0] = 10;
7         anArray[1] = 20;
8         anArray[2] = 30;
9         anArray[3] = 40;
```

```

10     anArray[4] = 50;
11     anArray[5] = 60;
12     anArray[6] = 70;
13     anArray[7] = 80;
14     anArray[8] = 90;
15     anArray[9] = 100;
16
17     System.out.println("Index Element 0: " + anArray[0]);
18     System.out.println("Index Element 1: " + anArray[1]);
19     System.out.println("Index Element 2: " + anArray[2]);
20     System.out.println("Index Element 3: " + anArray[3]);
21     System.out.println("Index Element 4: " + anArray[4]);
22     System.out.println("Index Element 5: " + anArray[5]);
23     System.out.println("Index Element 6: " + anArray[6]);
24     System.out.println("Index Element 7: " + anArray[7]);
25     System.out.println("Index Element 8: " + anArray[8]);
26     System.out.println("Index Element 9: " + anArray[9]);
27 }
28 }

```

Apakah hanya dapat dilakukan akses pada satu indeks array saja? Misalkan langsung pada indeks ke 5 / 9 dsb?

Jika pada `anArray = new int[10];` angka 10 anda rubah dengan 15, maka apa isi data dari indeks ke 10 sampai indeks ke 14?

Jika tipe data diubah ke string, double, float atau boolean. Maka jelaskan yang terjadi pada indeks ke 0 sampai 9? Dan jika isi data berjumlah 15, maka apa yang terjadi pada indeks ke 10 sampai 14 tentang masing-masing tipe data tersebut?

Percobaan ke-2 : Mengurutkan data dan menyisipkan data pada array

```

1  import java.util.Arrays;
2  public class MainClass {
3      public static void main(String args[]) throws Exception {
4          int array[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };
5          Arrays.sort(array);
6          printArray("Sorted array", array);
7          int index = Arrays.binarySearch(array, 1);
8          System.out.println("Didn't find 1 @ " + index);
9          int newIndex = -index - 1;
10         array = insertElement(array, 1, newIndex);
11         printArray("With 1 added", array);
12     }
13
14     private static void printArray(String message, int array[]) {
15         System.out.println(message + ": [length: " + array.length + "]");
16         for (int i = 0; i < array.length; i++) {
17             if (i != 0){
18                 System.out.print(", ");
19             }
20             System.out.print(array[i]);
21         }

```

```

22     System.out.println();
23 }
24
25     private static int[] insertElement(int original[],
26     int element, int index) {
27         int length = original.length;
28         int destination[] = new int[length + 1];
29         System.arraycopy(original, 0, destination, 0, index);
30         destination[index] = element;
31         System.arraycopy(original, index, destination, index + 1, length - index);
32         return destination;
33     }
34 }

```

Berapakah isi maksimal dari array tersebut?

Proses penyisipan seperti apakah yang terjadi? Disisipkan di depan, tengah atau belakang?

Percobaan ke-3 : Membandingkan dua buah array

```

1  public class Main {
2      public static void main(String[] args) throws Exception {
3          int[] array = {1,2,3,4,5,6};
4          int[] array1 = {1,2,3,4,5,6};
5          int[] array2 = {1,2,3,4};
6          System.out.println("Variabel array = array 1?" +Arrays.equals(array, array 1));
7          System.out.println("Variabel array = array 2?" +Arrays.equals(array, array 2));
8          System.out.println("Variabel array 1 = array 2?" +Arrays.equals(array1, array 2));
9      }
10 }

```

Bagaimanakah proses pembandingan antara tiap-tiap array yang dibandingkan?

Percobaan ke-4 : Array dua dimensi

```

1  public class Array2Dimensi{
2      public static void main(String[] args) {
3          int[][] a2 = new int[10][5];
4          for (int i=0; i<a2.length; i++) {
5              for (int j=0; j<a2[i].length; j++) {
6                  a2[i][j] = i;
7                  System.out.print(" " + a2[i][j]);
8              }
9              System.out.println("");
10             }
11         }
12     }

```

Berapa tabel dan kolom kah array tersebut?

Apakah array tersebut sudah memiliki isi? Jika sudah isinya apa? Kalau belum isikanlah.

Percobaan ke-5 : Menghitung panjang baris dan kolom pada array dua dimensi

```

1  public class Main {
2      public static void main(String args[]) {
3          String[][] data = new String[4][5];
4          System.out.println("Dimension 1: " + data.length);
5          System.out.println("Dimension 2: " + data[0].length);

```

6	}
7	}

Isikanlah data dengan beberapa data, kemudian hitunglah isi data pada array tersebut.

3.6 Hasil Percobaan

Pada sub-bab ini jelaskan tentang apa yang anda dapatkan setelah anda menjalankan masing-masing kode program tersebut.

Percobaan ke-1

Percobaan ke-2

Percobaan ke-3

Percobaan ke-4

Percobaan ke-5

3.7 Analisis Hasil

Pada sub bab ini, jelaskan tentang fenomena yang terjadi pada array yang dijalankan pada tiap percobaan.

Percobaan ke-1

Percobaan ke-2

Percobaan ke-3

Percobaan ke-4

Percobaan ke-5

3.8 Kesimpulan

Percobaan ke-1

Percobaan ke-2

Percobaan ke-3

Percobaan ke-4

Percobaan ke-5

3.9 Latihan

Terdapat array dengan data 30, 87, 90, 3, 1, 50, 23, 4, 25, 23, 40, 35, 47, 2, 33.

1. Buatlah Kodenya dalam bahasa Java.
2. Jalankan dan pastikan tidak ada errornya.

3.10 Tugas

Tugas 1

Berdasar latihan yang telah anda buat maka:

1. Urutkan data pada array tersebut.
2. Hitung rata-rata data tersebut.
3. Hitung nilai maksimal dan minimalnya.
4. Tampilkan data yang bilangan ganjil saja dan prima saja.
5. Dari data tersebut, buatlah array 2 dimensi dengan format 3 baris dan 5 kolom.

Hasil

Tugas 2

Tuliskan dan jalankan program berikut

```
1 public class Matrik{
2     private int nBaris, nKolom;
3     private double [][]itemDt;
4     /**
5      * constructor untuk membuat suatu matrik
6      * @param nBrs : banyaknya baris
7      * @param nKlm : banyaknya kolom
8      */
9     public Matrik(int nBrs, int nKlm){
10         nBaris = nBrs;
11         nKolom = nKlm;
12         itemDt = new double[nBaris][nKolom];
13     }
14     /**
15      * constructor untuk membuat matrik dari array 2 dimensi
16      * @param A : array dua dimensi
17      */
18     public Matrik(double [][]A){
19         this(A.length,A[0].length); // panggil constructor
20         this.nBaris = A.length;
21         this.nKolom = A[0].length;
22
23         for (int i=0; i<nBaris; i++){
24             for (int j=0; j<nKolom; j++){
25                 this.itemDt[i][j] = A[i][j];
26             }
27         }
28     }
29     /**
30      * Fungsi untuk mendapatkan jumlah baris
31      * @return jumlah baris
32      */
33     public int getNBaris(){ return nBaris;}
34     public int getNKolom(){ return nKolom;}
35     public double getItem(int idB, int idK){
36         return this.itemDt[idB][idK];
37     }
38     public void setItem(int idB, int idK, double dt){
39         this.itemDt[idB][idK] = dt;
40     }
41     /**
42      * fungsi tambah antara dua matrik A dan B
43      * @param A : Matrik
44      * @param B : Matrik
45      * @return Matrik hasil
46      */
47     public static Matrik tambah(Matrik A, Matrik B){
48         // tambahkan bagian ini
49     }
```

```

50
51  /**
52   * fungsi static perkalian antara vektor dengan matrik
53   * Syarat : lebar L sama dengan jumlah baris M
54   * @param L : Vector (Larik)
55   * @param M : Matrik
56   * @return Vector (Larik) berdimensi nKolom dari M
57   */
58  public static Larik VektorKaliMatrik(Larik L, Matrik M){
59      Larik IHasil = null;
60      Larik IKolom = null;
61      if (L.getSize() == M.getNBaris()){
62          IHasil = new Larik(M.getNKolom());
63          for (int i=0; i<M.getNKolom(); i++){
64              IKolom = M.getKolom(i);
65              double hasil = Larik.LarikKaliLarik(L, IKolom);
66              System.out.println(hasil);
67              IHasil.isiltem(i, hasil);
68          }
69      }
70      return IHasil;
71  }
72
73  /**
74   * fungsi static tranpos suatu matrik
75   * @param A : Matrik
76   * @return Matrik tranpos
77   */
78  public static Matrik tranpos(Matrik A){
79      // lengkapi bagian ini
80  }
81
82  /**
83   * fungsi untuk mendapatkan vektor baris dari matrik
84   * @param idBaris : indek baris yang akan diekstrak
85   * @return Larik representasi baris
86   */
87  public Larik getBaris(int idBaris){
88      // lengkapi bagian ini
89  }
90
91  /**
92   * fungsi untuk mendapatkan vektor kolom suatu matrik
93   * @param idKolom : id kolom yang akan diekstrak
94   * @return Larik representasi kolom
95   */
96  public Larik getKolom(int idKolom){
97      Larik l = new Larik(this.nBaris);
98      for (int i=0; i<this.nBaris; i++){
99          double itemKolom = this.getItem(i, idKolom);
100         l.isiltem(i, itemKolom);

```

```

101         return l;
102     }
103
104     /**
105     * procedure cetak
106     * @param kom
107     */
108     public void cetak(String kom){
109         System.out.println(kom);
110         for (int i=0; i<this.nBaris; i++){
111             for (int j=0; j<this.nKolom; j++){
112                 System.out.printf("%.2f ",this.itemDt[i][j]);
113             }
114             System.out.println();
115         }
116     }
117

```

Setelah program diatas dijalankan, kemudian sisipkan program dibawah sehingga program dapat berjalan

```

1  Matrik A,B,C;
2  double [][]data1 = {{1,2,3},{3,4,7}};
3  double [][]data2 = {{4,5,1},{6,1,9}};
4  A = new Matrik(data1);
5  B = new Matrik(data2);
6  A.cetak("A"); B.cetak("B");
7  C = Matrik.tambah(A,B);
8  C.cetak("C");
9  Matrik CT = Matrik.tranpos(C);
10 CT.cetak("Tsanpos");
11 Larik l1 = C.getBaris(1);
12 l1.cetak("Baris ke 1 dari C");
13 Larik l2 =
14 Matrik.VektorKaliMatrik(l1,CT);
15 l2.cetak("Hasil kali C.L1");

```

Setelah program tersebut disisipkan harapannya adalah keluar hasil berikut

```

A
1.00 2.00 3.00
3.00 4.00 7.00
B
4.00 5.00 1.00
6.00 1.00 9.00
C
5.00 7.00 4.00
9.00 5.00 16.00
Transpose
5.00 9.00
7.00 5.00
4.00 16.00
Baris ke 1 dari C
9.00 5.00 16.00

```

144.0 362.0 Hasil kali C.L1 144.00 362.00
--

- Apakah maksud dari A?
- Apakah maksud dari B?
- Apakah maksud dari C?
- Apakah pengertian dan hasil dari Transpose?
- Apakah maksud dari Baris ke 1 dari C?
- Apakah maksud dari Hasil kali C.L1?

3.11 DAFTAR PUSTAKA

- Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms Using Java 6 edition", Wiley, USA, 2014.
- John R. Hubbard, "Scaum's Outline of Data Structures With Java second Edition", McGraw-Hill, New york, 2007.
- Robert Lafore, "Data Structures and Algorithm in Java second Edition", Sams Publishing, Indiana, 2003