

# 3

## Array 2 Dimensi

### Tujuan Instruksional Khusus:

- *Praktikan ini bertujuan untuk memberi pemahaman Abstract Data Type Array 2 dimensi.*

### Teori

Array 2 dimensi pada dasarnya adalah pengembangan dari array 1 dimensi, dimana tipe data masing-masing itemnya bertipe array 1 dimensi. Secara sederhana array 2 dimensi dapat didefinisikan sebagai berikut :

tipeData [ ][ ]dt.

### ADT Array 1 Dimensi

Applikasi array 2 dimensi banyak digunakan pada representasi masalah matrik. Matrik sebagai representasi data dapat digambarkan ADT sebagai berikut :

<b>Matrik</b>
Int nBaris Int nKolom double [ ][ ]itemData
Matrik(int) Matrik(double [ ][ ]dt) Larik getBaris(int) Larik getKolom(int) Matrik tambah(Matrik) Matrik kali(Matrik) Tranpos () Boolean kesamaan(Matrik)

	Program Latihan Praktikum 3.1
1	<b>public class</b> Matrik{
2	// Data
3	<b>private</b> <b>int</b> nBaris, nKolom;
4	<b>private</b> <b>double</b> [][]itemDt;
5	
6	<b>public</b> Matrik( <b>int</b> nBrs, <b>int</b> nKlm){
7	nBaris = nBrs;
8	nKolom = nKlm;
9	itemDt = <b>new</b> <b>double</b> [nBaris][nKolom];
10	}
11	
12	<b>public</b> Matrik( <b>double</b> [][]dt){
13	nBaris = dt.length;
14	nKolom = dt[0].length;
15	this.itemDt = <b>new</b> <b>double</b> [nBaris][nKolom];
16	<b>for</b> ( <b>int</b> i=0; i<nBaris; i++){
17	<b>for</b> ( <b>int</b> j=0; j<nKolom; j++){
18	this.setItem(i,j,dt[i][j]);
19	}
20	}
21	}
22	/**
23	* Fungsi untuk mendapatkan jumlah baris
24	* @return jumlah baris
25	*/
26	<b>public</b> <b>int</b> getNBaris(){ <b>return</b> nBaris;}
27	<b>public</b> <b>int</b> getNKolom(){ <b>return</b> nKolom;}
28	<b>public</b> <b>double</b> getItem( <b>int</b> idB, <b>int</b> idK){
29	<b>return</b> this.itemDt[idB][idK];
30	}
31	<b>public</b> <b>void</b> setItem( <b>int</b> idB, <b>int</b> idK, <b>double</b> dt){
32	this.itemDt[idB][idK] = dt;
33	}
34	
35	<b>public</b> Matrik tambah(Matrik x){
36	Matrik y = <b>null</b> ;
37	<b>if</b> ((this.nBaris == x.getNBaris())&&
38	(this.nKolom == x.getNKolom())){
39	y = <b>new</b> Matrik(x.getNBaris(),x.getNKolom());
40	<b>for</b> ( <b>int</b> i=0; i<this.nBaris; i++){
41	<b>for</b> ( <b>int</b> j=0; j<this.nKolom; j++){
42	y.setItem(i,j,
43	this.itemDt[i][j]+x.getItem(i,j));
44	}
45	}
46	}
47	<b>return</b> y;
48	}
49	
50	<b>public</b> <b>void</b> cetak( <b>String</b> kom){
51	System.out.println(kom);
52	<b>for</b> ( <b>int</b> i=0; i<this.nBaris; i++){

```
53         for (int j=0; j<this.nKolom; j++){
54             System.out.print(this.itemDt[i][j]+"
55 ");
56         }
57         System.out.println();
58     }
59 }
60
61 public static void main (String[] args) {
62     // implementasi untuk ADT_Matrik
63     Matrik A,B,C;
64
65     double [][] X = {{1, 2, 3},{2, 14, 5},{16, 8, 13}};
66     double [][] Y = {{10, 12, 0},{5, 1, 5},{3, 1, 10}};
67
68     A = new Matrik(X);
69     B = new Matrik(Y);
70
71     A.cetak("Matrik A");
72     B.cetak("Matrik B");
73
74     C = A.tambah(B);
75     C.cetak("Matrik C = A+B");
76
77     Larik lb,lk;
78     lb = D.getBaris(1);
79     lb.cetak("Larik baris ke-1");
80     lk = D.getKolom(1);
81     lk.cetak("Larik kolom ke-1");
82 }
}
```

### Latihan 3.

1. Jalankan program di atas, amati hasilnya.

### Tugas 3.1.

1. Tambahkan method untuk mencari determinan matrik dengan header

```
double determinan()
```

2. Tambahkan method untuk transpose matrik dengan header

```
Matrik tranposeMatrik()
```

3. Tambahkan method untuk perkalian matrik dengan header

```
Matrik kali(Matrik m)
```

Interface pemanggilan method di atas sebagai berikut :

Misalkan dideklarasikan Matrik A,B, C;

$C = A.kali(B)$

4. Tambahkan method untuk mengambil isi baris tertentu dan isi kolom tertentu dari matrik dan hasilnya dikeluarkan berupa larik dengan header sebagai berikut :

```
Larik getKolom(int idK)
```

```
Larik getBaris(int idK)
```

Untuk dapat menjalankan ini tambahkan program Larik pada praktikum 1 diatas program ini atau anda susun dalam bentuk paket (minta petunjuk pada asisten). Karena larik pada program 1 tipe itemnya integer maka agar kompatibel dengan program matrik ada dua cara :

- Lakukan casting (minta petunjuk pada asisten)
- Anda edit tipe data itemDt pada larik bertipe float

	Program Latihan Praktikum 3.2
1	<code>package ADT_Larik;</code>
2	
3	<code>/**</code>
4	<code> * ADT Matrik</code>
5	<code> * @author achmad ridok</code>
6	<code> */</code>
7	<code>public class Matrik{</code>
8	<code>    private int nBaris, nKolom;</code>
9	<code>    private double [][]itemDt;</code>
10	<code>    /**</code>
11	<code>     * constructor untuk membuat suatu matrik</code>
12	<code>     * @param nBrs : banyaknya baris</code>
13	<code>     * @param nKlm : banyaknya kolom</code>
14	<code>     */</code>
15	<code>    public Matrik(int nBrs, int nKlm){</code>
16	<code>        nBaris = nBrs;</code>
17	<code>        nKolom = nKlm;</code>
18	<code>        itemDt = new double[nBaris][nKolom];</code>
19	<code>    }</code>
20	<code>    /**</code>
21	<code>     * constructor untuk membuat matrik dari array 2 dimensi</code>
22	<code>     * @param A : array dua dimensi</code>
23	<code>     */</code>
24	<code>    public Matrik(double [][]A){</code>
25	<code>        this(A.length,A[0].length); // panggil constructor</code>
26	<code>        this.nBaris = A.length;</code>

```
27         this.nKolom = A[0].length;
28
29         for (int i=0; i<nBaris; i++){
30             for (int j=0; j<nKolom; j++){
31                 this.itemDt[i][j] = A[i][j];
32             }
33         }
34     }
35     /**
36      * Fungsi untuk mendapatkan jumlah baris
37      * @return jumlah baris
38      */
39     public int getNBaris(){ return nBaris;}
40     public int getNKolom(){ return nKolom;}
41     public double getItem(int idB, int idK){
42         return this.itemDt[idB][idK];
43     }
44     public void setItem(int idB, int idK, double dt){
45         this.itemDt[idB][idK] = dt;
46     }
47     /**
48      * fungsi tambah antara dua matrik A dan B
49      * @param A : Matrik
50      * @param B : Matrik
51      * @return Matrik hasil
52      */
53     public static Matrik tambah(Matrik A, Matrik B){
54         // tambahkan bagian ini
55     }
56
57     /**
58      * fungsi static perkalian antara vektor dengan matrik
59      * Syarat : lebar L sama dengan jumlah baris M
60      * @param L : Vector (Larik)
61      * @param M : Matrik
62      * @return Vector (Larik) berdimensi nKolom dari M
63      */
64     public static Larik VektorKaliMatrik(Larik L, Matrik M){
65
66         Larik lHasil = null;
67         Larik lKolom = null;
68         if (L.getSize() == M.getNBaris()){
69             lHasil = new Larik(M.getNKolom());
70             for (int i=0; i<M.getNKolom(); i++){
71                 lKolom = M.getKolom(i);
72                 double hasil = Larik.LarikKaliLarik(L,
73 lKolom);
74                 System.out.println(hasil);
75                 lHasil.isiItem(i, hasil);
76             }
77         }
78     }
79     return lHasil;
80 }
81
```

```
82      /**
83      * fungsi static determinan matriks
84      * @return double determinan
85      */
86      public static double determinan(){
87          // lengkapi bagian ini
88      }
89
90      /**
91      * fungsi static tranpos suatu matrik
92      * @param A : Matrik
93      * @return Matrik tranpos
94      */
95      public static Matrik tranpos(Matrik A){
96          // lengkapi bagian ini
97      }
98
99      /**
100     * fungsi untuk mendapatkan vektor baris dari matrik
101     * @param idBaris : indek baris yang akan diekstrak
102     * @return Larik representasi baris
103     */
104     public Larik getBaris(int idBaris){
105         // lengkapi bagian ini
106     }
107
108     /**
109     * fungsi untuk mendapatkan vektor kolom suatu matrik
110     * @param idKolom : id kolom yang akan diekstrak
111     * @return Larik representasi kolom
112     */
113
114     public Larik getKolom(int idKolom){
115         Larik l = new Larik(this.nBaris);
116         for (int i=0; i<this.nBaris; i++){
117             double itemKolom = this.getItem(i, idKolom);
118             l.setItem(i, itemKolom);
119         }
120         return l;
121     }
122
123     /**
124     * procedure cetak
125     * @param kom
126     */
127     public void cetak(String kom){
128         System.out.println(kom);
129         for (int i=0; i<this.nBaris; i++){
130             for (int j=0; j<this.nKolom; j++){
131                 System.out.printf("%.2f
132 ",this.itemDt[i][j]);
133             }
134             System.out.println();
135         }
136     }
```

**Tugas 3.2.** Lengkapi program di atas sehingga dengan potongan program berikut akan menghasilkan keluaran sebagaimana disamping

<pre>Matrik A,B,C; double [][]data1 = {{1,2,3},{3,4,7}}; double [][]data2 = {{4,5,1},{6,1,9}}; A = new Matrik(data1); B = new Matrik(data2); A.cetak("A"); B.cetak("B"); C = Matrik.tambah(A,B); C.cetak("C"); System.out.println("Determinan C : "+C.determinan()); Matrik CT = Matrik.tranpos(C); CT.cetak("Tranpos"); Larik l1 = C.getBaris(1); l1.cetak("Baris ke 1 dari C"); Larik l2 = Matrik.VektorKaliMatrik(l1,CT); l2.cetak("Hasil kali C.L1");</pre>	<pre>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 Determinan C : Bukan matriks persegi Tranpos 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</pre>
---	---