



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : ADT ARRAY 2 DIMENSI
NAMA : CHARISMA PRASETYA PUTERA AMMAL
NIM : 185150700111019
TANGGAL : 13/09/2019
ASISTEN : AFIF MUSYAYYIDIN

A. DEFINISI MASALAH

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 is 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	package ADT_Larik;
2	
3	/**
4	*
5	* @author ASUS
6	*/
7	public class Matrik {
8	private int nBaris, nKolom;
9	private double[][] itemDt;

```

10  /**
11  * constructor untuk membuat suatu matrik
12  * @param nBrs : banyaknya baris
13  * @param nKlm : banyaknya kolom
14  */
15  public Matrik(int nBrs, int nKlm) {
16      nBris = nBrs;
17      nKolom = nKlm;
18      itemDt = new double[nBris][nKolom];
19  }
20  /**
21  * constructor untuk membuat matrik dari array 2
22  dimensi
23  * @param A : array dua dimensi
24  */
25  public Matrik(double[][] A) {
26      this(A.length, A[0].length); // panggil
27  contructor
28      this.nBris = A.length;
29      this.nKolom = A[0].length;
30
31      for (int i = 0; i < nBris; i++) {
32          for (int j = 0; j < nKolom; j++) {
33              this.itemDt[i][j] = A[i][j];
34          }
35      }
36  /**
37  * Fungsi untuk mendapatkan jumlah baris
38  * @return jumlah baris
39  */
40  public int getNBris() {
41      return nBris;
42  }
43
44  public int getNKolom() {
45      return nKolom;
46  }
47
48  public double getItem(int idB, int idK) {
49      return this.itemDt[idB][idK];
50  }
51
52  public void setItem(int idB, int idK, double dt) {
53      this.itemDt[idB][idK] = dt;
54  }
55  /**
56  * fungsi tambah antara dua matrik A dan B
57  *
58  * @param A : Matrik
59  * @param B : Matrik
60  * @return Matrik hasil
61  */
62  public static Matrik tambah(Matrik A, Matrik B) {
63  // tambahkan bagian ini
64  }
65
66  /**

```

```

67      * fungsi static perkalian antara vektor dengan
matrik Syarat : lebar L sama
68      * dengan jumlah baris M
69      *
70      * @param L : Vector (Larik)
71      * @param M : Matrik
72      * @return Vector (Larik) berdimensi nKolom dari M
73      */
74      public static Larik VektorKaliMatrik(Larik L, Matrik
M) {
75          Larik lHasil = null;
76          Larik lKolom = null;
77          if (L.getSize() == M.getNBaris()) {
78              lHasil = new Larik(M.getNKolom());
79              for (int i = 0; i < M.getNKolom(); i++) {
80                  lKolom = M.getKolom(i);
81                  double hasil = Larik.LarikKaliLarik(L,
lKolom);
82                      System.out.println(hasil);
83                      lHasil.isiItem(i, hasil);
84                  }
85              }
86              return lHasil;
87          }
88
89      /**
90      * fungsi static determinan matriks
91      *
92      * @return double determinan
93      */
94      public static double determinan() {
95      // lenkapi bagian ini
96      }
97
98      /**
99      * fungsi static tranpos suatu matrik
100     *
101     * @param A : Matrik
102     * @return Matrik tranpos
103     */
104     public static Matrik tranpos(Matrik A) {
105     // lenkapi bagian ini
106     }
107
108     /**
109     * fungsi untuk mendapatkan vektor baris dari matrik
110     *
111     * @param idBaris : indek baris yang akan diekstrak
112     * @return Larik representasi baris
113     */
114     public Larik getBaris(int idBaris) {
115     // lenkapi bagian ini
116     }
117
118     /**
119     * fugsii untuk mendapatkan vektor kolom suatu matrik
120     *
121     * @param idKolom : id kolom yang akan diekstrak
122     * @return Larik representasi kolom

```

123	*/
124	public Larik getKolom(int idKolom) {
125	Larik l = new Larik(this.nBaris);
126	for (int i = 0; i < this.nBaris; i++) {
127	double itemKolom = this.getItem(i, idKolom);
128	l.isiItem(i, itemKolom);
129	}
130	return l;
131	}
132	
133	/**
134	* procedure cetak
135	*
136	* @param kom
137	*/
138	public void cetak(String kom) {
139	System.out.println(kom);
140	for (int i = 0; i < this.nBaris; i++) {
141	for (int j = 0; j < this.nKolom; j++) {
142	System.out.printf("%.2f ",
143	this.itemDt[i][j]);
144	}
145	System.out.println();
146	}

Lengkapi program di atas sehingga dengan potongan program berikut akan menghasilkan keluaran sebagaimana disamping

Matrik A,B,C;	A
double [][]data1 = {{1,2,3},{3,4,7}};	1.00 2.00 3.00
double [][]data2 = {{4,5,1},{6,1,9}};	3.00 4.00 7.00
A = new Matrik(data1);	B
B = new Matrik(data2);	4.00 5.00 1.00
A.cetak("A");	6.00 1.00 9.00
B.cetak("B");	C
C = Matrik.tambah(A,B);	5.00 7.00 4.00
C.cetak("C");	9.00 5.00 16.00
System.out.println("Determinan C :	Determinan C : Bukan matriks persegi
")+C.determinan());	Tranpos
Matrik CT = Matrik.tranpos(C);	5.00 9.00
CT.cetak("Tranpos");	7.00 5.00
Larik l1 = C.getBaris(1);	4.00 16.00
l1.cetak("Baris ke 1 dari C");	Baris ke 1 dari C
Larik l2 =	9.00 5.00 16.00
Matrik.VektorKaliMatrik(l1,CT);	144.0
l2.cetak("Hasil kali C.L1");	362.0
	Hasil kali C.L1
	144.00 362.00

B. SOURCE CODE

Matrik2.java	
1	package ADT_Larik;
2	

```

3 public class Matrik2 {
4
5     // Data
6     private int nBaris, nKolom;
7     private double[][] itemDt;
8
9     public Matrik2(int nBrs, int nKlm) {
10
11         nBaris = nBrs;
12         nKolom = nKlm;
13         itemDt = new double[nBaris][nKolom];
14
15     }
16
17     public Matrik2(double[][] dt) {
18
19         nBaris = dt.length;
20         nKolom = dt[0].length;
21         this.itemDt = new double[nBaris][nKolom];
22         for (int i = 0; i < nBaris; i++) {
23             for (int j = 0; j < nKolom; j++) {
24                 this.setItem(i, j, dt[i][j]);
25             }
26         }
27     }
28
29     /*
30      * Fungsi untuk mendapatkan jumlah baris      *
31      * @return jumlah baris      *
32      */
33     public int getNBaris() {
34         return nBaris;
35     }
36
37     public int getNKolom() {
38         return nKolom;
39     }
40
41     public double getItem(int idB, int idK) {
42         return this.itemDt[idB][idK];
43     }
44
45     public void setItem(int idB, int idK, double dt) {
46         this.itemDt[idB][idK] = dt;
47     }
48
49     public Matrik2 tambah(Matrik2 x) {
50         Matrik2 y = null;
51         if ((this.nBaris == x.getNBaris()) &&
52             (this.nKolom == x.getNKolom())) {
53             y = new Matrik2(x.getNBaris(),
54                             x.getNKolom());
55             for (int i = 0; i < this.nBaris; i++) {
56                 for (int j = 0; j < this.nKolom; j++) {
57                     y.setItem(i, j, this.itemDt[i][j] +
58                                 x.getItem(i, j));
59                 }
60             }
61         }
62     }
63 }

```

```

59         return y;
60     }
61
62     public void cetak(String kom) {
63
64         System.out.println(kom);
65         for (int i = 0; i < this.nBaris; i++) {
66             for (int j = 0; j < this.nKolom; j++) {
67                 System.out.print(this.itemDt[i][j] + "
68             ");
69             }
70             System.out.println();
71         }
72     }
73
74     public Larik getBaris(int indexBaris) {
75         double[] baru = new double[this.nKolom];
76         for (int i = 0; i < this.nKolom; i++) {
77             baru[i] = this.getItem(indexBaris - 1, i);
78         }
79         Larik lBaru = new Larik(baru);
80         return lBaru;
81     }
82
83     public Larik getKolom(int indexKolom) {
84         double[] baru = new double[this.nKolom];
85         for (int i = 0; i < this.nKolom; i++) {
86             baru[i] = this.getItem(i, indexKolom - 1);
87         }
88         Larik lBaru = new Larik(baru);
89         return lBaru;
90     }
91
92     public Matrik2 kali(Matrik2 m) {
93         Matrik2 hasil = null;
94         double[][] temp = new
double[nBaris][m.getNKolom()];
95         int idK = 0;
96         if (nKolom == m.getNBaris()) {
97             hasil = new Matrik2(nBaris, m.getNKolom());
98             for (int i = 0; i < nBaris; i++) {
99                 for (int j = 0; j < m.getNKolom(); j++)
100                 {
101                     for (int k = 0; k < nKolom; k++) {
102                         temp[i][j] += this.itemDt[i][k]
103                     * m.getItem(k, j);
104                     }
105                     hasil.setItem(i, j, temp[i][j]);
106                 }
107             }
108             return hasil;
109         }
110
111         public Matrik2 getTranspose() {
112             double[][] transpose = new
double[getNKolom()][getNBaris()];
113             for (int j = 0; j < getNKolom(); j++) {

```

```

113         for (int i = 0; i < getNBaris(); i++) {
114             transpose[i][j] = getItem(j, i);
115         }
116     }
117     return new Matrik2(transpose);
118 }
119
120 public double determinan() {
121     double hasilTambah = 0;
122     double[] hasilKali = new double[3];
123     double[][] matriksBantuan = new double[3][5];
124     for (int i = 0; i < getNBaris(); i++) {
125         for (int j = 0; j < getNKolom(); j++) {
126             matriksBantuan[i][j] = getItem(i, j);
127         }
128     }
129     for (int i = 0; i < getNBaris(); i++) {
130         for (int j = 3; j < 5; j++) {
131             matriksBantuan[i][j] = getItem(i, (j -
132 3));
133         }
134         for (int i = 0; i < getNBaris(); i++) {
135             hasilKali[i] = 1;
136             for (int j = 0; j < 3; j++) {
137                 for (int k = i + j; k < i + j + 1; k++)
138 {
139                     hasilKali[i] *=
140 matriksBantuan[j][k];
141                 }
142             }
143             hasilTambah += hasilKali[i];
144         }
145         for (int i = 0; i < getNBaris(); i++) {
146             hasilKali[i] = 1;
147             for (int j = 2; j > -1; j--) {
148                 for (int k = i - j + 2; k < i - j + 3;
149 k++) {
150                     hasilKali[i] *=
151 matriksBantuan[j][k];
152                 }
153             }
154             hasilTambah -= hasilKali[i];
155         }
156         return hasilTambah;
157     }
158 }
159
160 public static void main(String[] args) {
161     Matrik2 A, B, C;
162
163     double[][] X = {{1, 2, 3}, {2, 14, 5}, {16, 8,
164 13}};
165     double[][] Y = {{10, 12, 0}, {5, 1, 5}, {3, 1,
166 10}};
167
168     A = new Matrik2(X);
169     B = new Matrik2(Y);
170
171     A.cetak("Matrik A");

```

165	B.cetak("Matrik B");
166	System.out.println("Determinan dari Matrik A adalah : " + A.determinan());
167	System.out.println("Determinan dari Matrik B adalah : " + B.determinan());
168	C = A.tambah(B);
169	C.cetak("Matrik C = A + B");
170	C = A.kali(B);
171	C.cetak("Matrik C = A * B");
172	Larik lb, lk;
173	lb = C.getBaris(1);
174	lb.cetak("Larik baris ke-1");
175	lk = C.getKolom(1);
176	lk.cetak("Larik kolom ke-1");
177	
178	C = A.getTranspose();
179	C.cetak("Matrik D = Hasil Transpose matrik A");
180	}
181	
182	}

C. PEMBAHASAN

Matrik2.java	
1	Deklarasi package dengan nama ADT_Larik
3	Membuat class bernama Matrik2
9	Membuat Konstruktor Matrik2 dengan 2 parameter int untuk memberikan jumlah baris dan kolom
17	Pembuatan Konstruktor Matrik2 dengan 1 parameter array double yang berguna untuk mengisi Matrik 2
33	Pembuatan method accessor getNBaris untuk mengisi nilai variable NBaris
37	Pembuatan method accessor getNKolom untuk mengisi nilai variable NKolom
41	Pembuatan method getItem dengan parameter int idB dan int idK untuk mengisi nilai Untuk mengisi nilai variable itemDt
45	Pembuatan method mutator setItem dengan parameter int idB, idK dan double dt untuk mengubah nilai variable itemDt dengan nilai variable dt
49	Pembuatan method matrik2 tambah untuk menambahkan 2 matrik2 dengan menggunakan perulangan for
53	Melakukan perulangan for dengan batas I = 0 dan I < nilai variable nBaris
54	Melakukan perulangan nested for dengan batas j = 0 da j < nKolom
62	Pembuatan method void cetak untuk mencetak matrix
65	Melakukan perulangan for dengan batas I = 0 dan I < nilai nBaris
66	Melakukan nested for dengan batas j = 0 dan J < nilai nKolom
74	Pembuatan method larik getBaris untuk mendapatkan data sesuai baris yang telah diinputkan oleh user dengan melakukan perulangan for
76	Melakukan perulangan for dengan batas I = 0 dan i < nKolom
83	Pembuatan method larik getKolom utnuk mendapatkan data sesuai kolom yang telah diinputkan oleh user dengan menggunakan perulangan for
85	Melakukan perulangan for I = 0 dan I < nilai nKolom
92	Pembuatan method kali matrix dengan melakukan perulangan for untuk mengkalikan 2 matrix untuk menghasilkan matrix baru dengan melakukan seleksi kondisi dan perulangan for

96	Melakukan seleksi kondisi if dengan syarat nKolom = nilai dari getNbaris
98	Melakukan perulangan for untuk mengkalikan antara kolom dengan baris antar matrix
110	Pembuatan method matrix getTranspose dengan menggunakan array untuk
111	Mendapatkan nilai transpose dari matrix yang telah ada sebelumnya kemudian ditranspose
112	Melakukan perulangan for untuk megubah antara baris menajdi kolom pada matix yang ingin diubah
120	Pembuatan method double determinan untuk menentukan nilai determinan suatu matrix dengan menggunkan array dan perulangan for
155	Pembuatan method main untuk implementasi method method yang telah dibuat sebelumnya
158	Inisiasi array multidimensi dengan nama X
159	Inisiasi array multidimensi dengan nama Y
161	Instansiasi objek A dan B
164	Pemanggilan method cetak untuk mencetak objek A
165	Pemanggilan method cetak untuk mencetak objek B
166	Pencetakan hasil determinan dari matrik A
167	Pencetakan hasil determinan dari matrik B
169	Pemanggilan method tambah untuk menambahkan objek A dan B menjadi objek C
170	Pemanggilan method cetak untuk mencetak objek C
171	Pemanggilan method kali untuk mengkalikan objek A dan objek B
173	Pemanggilan method cetak untuk mencetak objek C
174	Pemanggilan method getbaris untuk mendapatkan data sesuai baris yang diinputkan user
175	Pemanggilan method cetak untuk mencetak objek C
176	Pemanggilan method getKolom untuk mendapatkan data sesuai kolom yang diinputkan user
178	Pemanggilan method transpose untuk mentarnspose objek A
179	Pemanggilan method cetak untuk mencetak objek C

D. SCREENSHOT PROGRAM

```

Output - Praktikum_array_1dimensi (run) *
run:
Matrik A
1.0 2.0 3.0
2.0 14.0 5.0
16.0 8.0 13.0
Matrik B
10.0 12.0 0.0
5.0 1.0 5.0
3.0 1.0 10.0
Determinan dari Matrik A adalah : -374.0
Determinan dari Matrik B adalah : -370.0
Matrik C = A + B
11.0 14.0 3.0
7.0 15.0 10.0
19.0 9.0 23.0
Matrik C = A * B
29.0 17.0 40.0
105.0 43.0 120.0
239.0 213.0 170.0
Larik baris ke-1
29.00 17.00 40.00
Larik kolom ke-1
29.00 105.00 239.00
Matrik D = Hasil Transpose matrik A
1.0 2.0 16.0
2.0 14.0 8.0
3.0 5.0 13.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

E. KESIMPULAN

Array 2 dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe data sama dan elemen yang akan diakses melalui 2 indeks atau subskrip yaitu indeks baris dan indeks kolom.

Deklarasi array 2 dimensi:

```
tipeData[][] namaVariabel = new tipeData[];
```

Mengakses array 2 dimensi:

```
namaArray.[indexBaris][indexKolom];
```

Contoh:

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

Array 2 Dimensi digunakan apabila suatu data yang berbentuk struktur list atau daftar dimana data tersebut memiliki tipe data yang sama, dan sering digunakan untuk membuat tabel dan matriks dalam matematika.

Penerapan Array 2Dimensi dalam kehidupan sehari-hari dapat ditemukan pada; Program mengambil nilai seperti Statistik, Program menghitung matriks ber ordo, dll.

Contoh:

```
double[][] matriks = {{192,289 ,245 }, {667, 324, 892},  
{287, 768, 459}};  
  
for (int i = 0; i < matriks.length; i++) {  
    for (int j = 0; j < matriks[0].length; j++) {  
        System.out.print(matriks[i][j] + " ");  
    }  
    System.out.println();  
}
```