

PERTEMUAN 13 ARRAY DUA DIMENSI

A. Tujuan Pembelajaran

Setelah mempelajari materi ini, mahasiswa mampu :

1. Memahami penerapan dan konsep Array 2 Dimensi
2. Memahami Penerapan tipe data pada Array 2 Dimensi
3. membuat membuat array 2 dimensi dalam pemrograman

1. Uraian Materi

1.1. Kelebihan & Kekurangan Array

Kelebihan Array :

1. Array sangat cocok untuk pengaksesan acak, sembarang komponen di array dapat di acu secara langsung tanpa melalui komponen- komponen lain.
2. Jika berada di suatu lokasi komponen, maka sangat mudah menerusuri ke komponen- komponen tetangga, baik komponen pendahulu atau komponen penerus 3.
3. Jika komponen - komponen array adalah nilai-nilai independen dan seluruhnya harus terjaga, maka penggunaan dan penyimpanannya sangat efisien.

Kekurangan Array :

1. Array harus bertipe homogen, kita tidak dapat mempunyai array dimana satu komponen adalah karakter, komponen lain bilangan, dan komponen lain adalah tipe-tipe lain.
2. Kebanyakan Bahasa pemrograman mengimplementasikan array static yang sulit diubah ukurannya di waktu eksekusi. Bila penambahan dan pengurangan terjadi terus-menerus, maka representasi statis
 - a. Tidak efisien dalam penggunaan memori
 - b. Menyimpan banyak waktu komputasi
 - c. Pada suatu aplikasi, representasi statis tidak dimungkinkan

1.2 Definisi

Jika sebelumnya telah di jelaskan perihal array 1 dimensi, yakni bahwa array 1 deimensi merupakan metode yang di gunakan untuk menampung sejumlah data dalam bentuk numerik maupun non numerik,

Array dua dimensi biasa disebut sebagai array yang mempunyai dua subskrip yaitu baris dan kolom, array dua dimensi adalah array yang terdiri dari n buah baris dan m buah kolom. Dimana elemen pertama menunjukkan baris dan elemen kedua menunjukkan kolom.

Array dua dimensi yang sering digambarkan sebagai sebuah matriks, merupakan perluasan dari sebuah array satu dimensi. Jika array satu dimensi hanya terdiri dari sebuah baris dengan beberapa kolom elemen maka array dua dimensi terdiri dari beberapa baris dan beberapa kolom elemen yang bertipe sama sehingga dapat di gambarkan seperti ini :

int B[9][5];

	[0]	[1]	[2]	[3]	[4]
[0]	0,0	0,1	0,2	0,3	0,4
[1]	1,0	1,1	1,2	1,3	1,4
[2]	2,0	2,1	2,2	2,3	2,4
[3]	3,0	3,1	3,2	3,3	3,4
[4]	4,0	4,1	4,2	4,3	4,4
[5]	5,0	5,1	5,2	5,3	5,4
[6]	6,0	6,1	6,2	6,3	6,4
[7]	7,0	7,1	7,2	7,3	7,4
[8]	8,0	8,1	8,2	8,3	8,4

Array dua dimensi ini, misalnya mengilustrasikan sebuah bangunan terdiri dari 9 lantai dan masing-masing terdiri dari 5 ruangan. disebut dua dimensi karena menunjukkan sebuah ruangan diperlukan dua penunjuk yaitu nomor lantai dan nomor ruang.

Bentuk umum array 2 dimensi :

Nama Matriks : array [1. .MaxBaris, 1. . MaxKolom] of TipeData

Tipe_Data : menyatakan jenis elemen matriks (int, float, char, unsigned, dan lain-lain), tidak boleh jenis void.

Nama_Matriks : adalah nama matriks, harus memenuhi ketentuan pengenalan.

Baris : menyatakan jumlah maksimal elemen baris matriks,

Kolom : menyatakan jumlah maksimal elemen kolom matriks.

Contoh :

Sebuah matrik X berukuran 2x3 dapat dideklarasikan sebagai berikut :

Algoritma :

a : array [1..2,1..3] of integer

a1= 11 b1 = 12

a2 = 7 b2 = 3

a3 = 4 b3 = 9

C++ :

Int X [2] [3] = {{11, 7, 4} , {12, 3, 9}};

Yang akan menempati lokasi memori dengan susunan berikut :

	0	1	2
0	11	7	4
1	12	3	9

Dan definisi variabel untuk setiap elemen tersebut adalah :

	0	1	2
0	a1	a2	a3
1	b1	b2	b3

Contoh Program 1 :

```
#include<iostream.h>
main ( )
{
    int x [3] [3] = { { 1, 2, 3 } , { 4, 5, 6 } , { 7, 8, 9 } };
    int i , j ;
```

```

    for (i=0; i < 3; i++)
    {
        for ( j=0; j < 3; j++)
            count << x [ i ] [ j ] << "  ";
        count << endl;
    }
}

```

Output :

1	2	3
4	5	6
7	8	9

Contoh Program 2 :

```

#include<stdio.h>

void printArray(int[][3]);
main() {
    int matrikA [2][3] = { {2, 3, 4}, {5, 4, 3}};
    int matrikB [2][3] = { {1, 1, 1}, {2, 2}};
    int matrikC [2][3] = { {4, 3}, {2} };
    printArray(matrik1);
    printArray(matrik2);
    printArray(matrik3);
    return 0;
}

void printArray(int a[][3]) {
    int i, j;
    for (i = 0; i <= 1; i++) {
        for (j = 0; j <= 2; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
}

```

Output :

2	3	4
5	4	3
1	1	1

2	2	0
4	3	0
2	0	0

Penjelasan :

Dari source code dan table diatas untuk matriks A, inputannya adalah 234 543, sedangkan matriks B inputannya adalah 111 220. 0 yang mempunyai arti tempat yang ada untuk kolom ketiga dari baris kedua tidak diisi, dan pada matriks C disini kita bisa melihat bahwa pada baris pertama kolom ketiga data tidak diisi dan dianggap 0 dan pada baris kedua kolom kedua dan ketiga juga tidak diisi juga diisi 0.

Dalam source code tersebut, juga digunakan fungsi untuk menampung hasil penjumlahan matriks.

Perhatikan contoh lain:

Int datasiswa [4] [3] ;

Deklarasi diatas digunakan untuk mendeklarasikan suatu data produksi kayu yang berbentuk demikian:

No	Bagian	Jumlah Karyawan		
		2016	2017	2018
1	Kayu Jati	20	22	25
2	Kayu Mahoni	80	52	77
3	Kayu Pinus	70	74	97
4	Kayu Sungkai	78	70	77

Dari deklarasi diatas maka angka [4] menyatakan jumlah kelas, dan angka [3] menyatakan tahun.

Data produksi [0][3] adalah kayu jati dan jumlah siswa tahun 2018 yaitu 25. Atau jumlah kayu jati pada tahun 2018 adalah 25.

Bentuk data siswa dapat juga digambarkan sebagai berikut:

	1	2	3
1	20	22	25
2	80	52	77

3	70	74	97
4	78	70	77

Array ini dapat pula diberi nilai tetap dengan *static* seperti pada array dimensi satu. Deklarasinya adalah sebagai berikut:

```
Static int jumlah [4] ;[ 3] =
{
    20, 22, 25,
    80, 52, 77,
    70, 74, 97,
    78, 70, 77
};
```

1.3 Alur Kerja Array 2 Dimensi

Pemrograman dasar dari array 2 dimensi adalah bagaimana cara melakukan input data matrik dan cara mencetak atau menampilkan hasilnya di layar komputer. Dalam contoh ini, diberikan 2 buah matriks A dengan ukuran $m \times n$ dan matriks B dengan ukuran $p \times q$.

Algoritma :

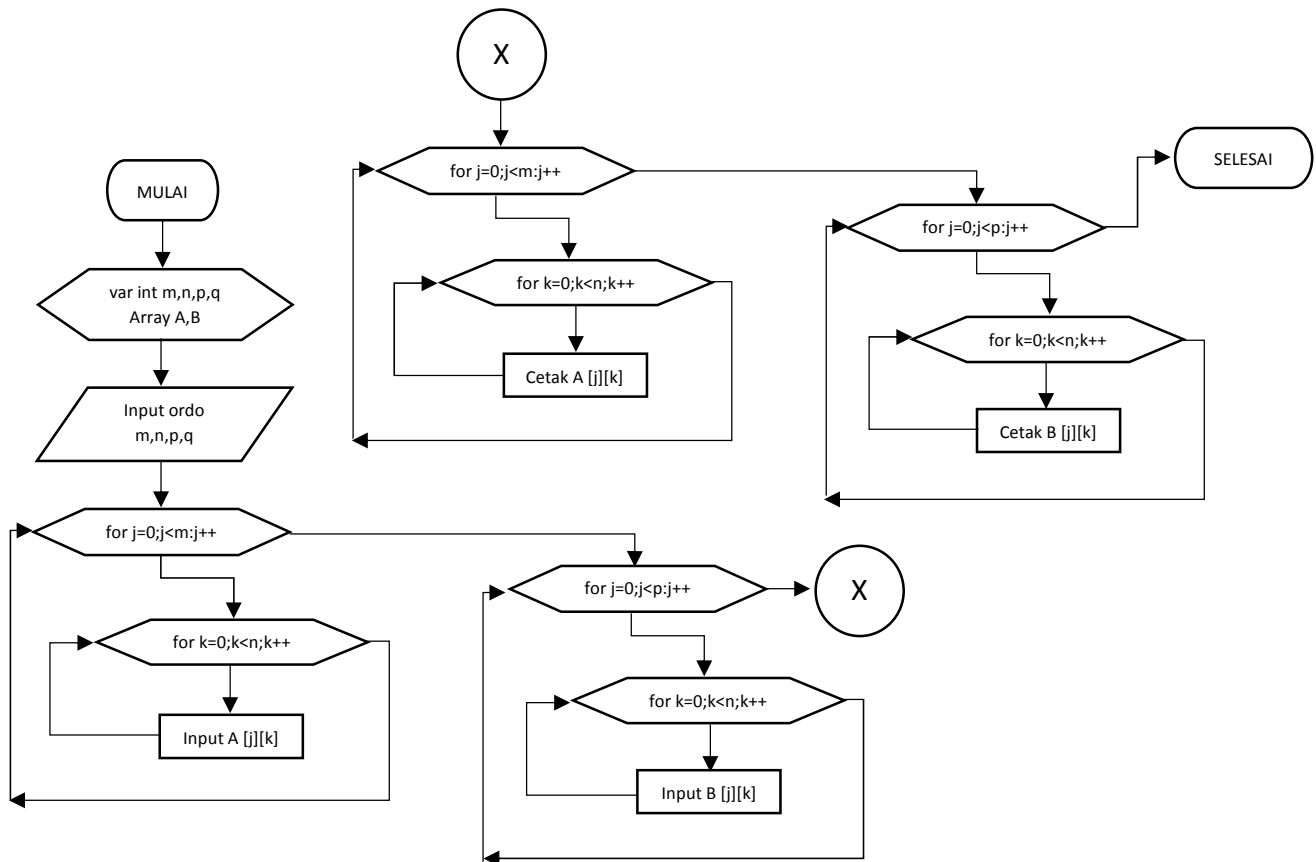
1. Mulai
2. Deklarasikan variabel baris dan kolom, array matrik A dan B
3. Input nilai baris dan kolom tiap matriks
4. Input data matriks A
 - a. Buat perulangan baris j mulai dari $j=0$ hingga j

jika tidak, lanjutkan ke langkah 5
 - b. Buat perulangan kolom k mulai dari $k=0$ hingga k

jika tidak, ulangi langkah 4a.
5. Input data matriks B

- a. Buat perulangan baris j mulai dari $j=0$ hingga j
jika tidak, lanjutkan ke langkah 6
 - b. Buat perulangan kolom k mulai dari $k=0$ hingga k
jika tidak, ulangi langkah 5a.
6. Cetak data Catriks A
- a. Buat perulangan baris j mulai dari $j=0$ hingga j
jika tidak, lanjutkan ke langkah 7
 - b. Buat perulangan kolom k mulai dari $k=0$ hingga k
jika tidak, ulangi langkah 6a
7. Cetak data matriks B
- a. Buat perulangan baris j mulai dari $j=0$ hingga j
jika tidak, lanjutkan ke langkah 8
 - b. Buat perulangan kolom k mulai dari $k=0$ hingga k
jika tidak, ulangi langkah 7a
8. selesai

1.5 Flowchart



1.6 Pejumlahan Array

Sudah ada dua buah array 2 dimensi yang di buat dengan int A [3][2] dan B [3][2] dengan ilustrasi seperti ini :

a	b		g	h		a+g	b+h
c	d		i	j		c+i	d+j
e	f		k	l		e+k	f+l

Syarat dua buah matriks (array dua dimensi) dapat di jumlahkan, adalah kedua matriks harus sebangun, jumlah baris dan kolom matriks A harus sama dengan jumlah baris dan kolom matriks B.

Contoh Program 1:

```

#include <iostream>
using namespace std;
int main(){
    // Dengan jumlah elemen baris = 3
    // dan jumlah elemen kolom = 2
    int matrika [3][2];
  
```



```
int matrikb [3][2];
int matrikc [3][2];

// Mendeklarasi variabel untuk
// Indeks perulangan
int i,j;

cout<<"\n\tPenjumlahan Array 2 Dimensi (Matrik)\n";

// Mengisi nilai kedalam
// Elemen-elemen array matrika
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<"matrik A ["<<i<<" ["<<j<<" = ";
        cin>>matrika[i][j];
        cout<<"matrik B ["<<i<<" ["<<j<<" = ";
        cin>>matrikb[i][j];
    }
}
cout<<endl;

// Melakukan penjumlahan array matrik A dan matrik B
// Dan menyimpan hasilnya ke array matrik C;
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        matrikc[i][j]=matrika[i][j] + matrikb[i][j];
    }
}

// Menampilkan matrik A
cout<<"\nMatrik A\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrika[i][j]<<" ";
    }
    cout<<endl;
}

// Menampilkan matrik B
cout<<"\nMatrik B\n";
for(i=0;i<3;i++){
    for(j=0;j<2;j++){
        cout<<matrikb[i][j]<<" ";
    }
    cout<<endl;
}

// Menampilkan hasil perhitungan / matrik C
// (dalam bentuk matrik dengan ordo 3x2)
cout<<"\nMatrik C (hasil)\n";
for(i=0;i<3;i++){
```

```

        for(j=0;j<2;j++){
            cout<<matrikc[i][j]<<" ";
        }
        cout<<endl;
    }
    return 0;
}

```

Hasil :

```
C:\Users\AMARTAI7\Documents\coba1.exe

Penjumlahan Array 2 Dimensi <Matrik>
matrik A [0] [0] = 1
matrik B [0] [0] = 9
matrik A [0] [1] = 2
matrik B [0] [1] = 8
matrik A [1] [0] = 3
matrik B [1] [0] = 7
matrik A [1] [1] = 4
matrik B [1] [1] = 6
matrik A [2] [0] = 5
matrik B [2] [0] = 5
matrik A [2] [1] = 6
matrik B [2] [1] = 4

Matrik A
1 2
3 4
5 6

Matrik B
9 8
7 6
5 4

Matrik C <hasil>
10 10
10 10
10 10

=====
Process exited after 24.59 seconds with return value 0
Press any key to continue . . .
```

1.7 Perkalian Array 2 Dimensi

Sudah ada dua buah array 2 dimensi yang di buat dengan int A [2][3] dan B [3][2] dengan ilustrasi seperti ini :

a	b	c
d	e	f

X

g	h
i	j
k	l

$$=$$

M	N
O	P

$$M = a \times g + b \times i + c \times k$$

$$N = a \times h + b \times j + c \times l$$

$$O = d \mathbf{x}_g + e \mathbf{x}_i + f \mathbf{x}_k$$

$$N = d \times h + e \times j + f \times l$$

Syarat dua buah matriks (array dua dimensi) dapat di kalikan adalah jumlah kolom matriks A harus sama dengan jumlah baris matriks B.

Contoh Program 1:

```
#include <iostream>
#include <iomanip>
#include <conio.h>
using namespace std;
main(){
int a1[2],b1[2],temp=-1; // Source code pengambil jumlah kolom dan baris
char pil; // Source code membuat pilihan mengulang atau berkhir
ulang: // Source code pilihan mengulang
// AlurMemasukkan jumlah baris dan kolom matrik
cout<<"-----\n";
for (char u='A'; u<='B'; u++){
    temp+=1;

    cout<<"Masukan baris matrik "<<u<<" : "; cin>>a1[temp];
    cout<<"Masukan kolom Matrik "<<u<<" : "; cin>>b1[temp];
    cout<<endl;
}
if(b1[0]==a1[1]){
    cout<<"-----\n";
    int AA [a1[0]][b1[0]]; //Input jumlah kolom dan barismatrik A
    int BB [a1[1]][b1[1]]; // Input jumlah kolom dan baris Matrik B
    int hasil[a1[0]][b1[1]]; // Hasil Perkalian Matrik
    //Memasukkan angka Array A
    cout<<"\nMasukkan angka array A\n";
    for(int i=0; i<a1[0]; i++){
        for (int c=0;c<b1[0];c++){
            cout<<"A ["<<i+1<<"]["<<c+1<<"] = ";
            cin>>AA[i][c]; }
        cout<<endl;
    }
    // Memasukkan angka Array B
    cout<<"-----\n";
    cout<<"\nMasukkan angka array B\n";
    for(int i=0; i<a1[1]; i++){
        for (int c=0;c<b1[1];c++){
            cout<<"B ["<<i+1<<"]["<<c+1<<"] = ";
            cin>>BB[i][c];
        }
        cout<<endl;
    }
    //Proses Perhitungan
    for(int i=0; i<a1[0]; i++){
        for (int c=0;c<b1[1];c++){
            hasil [i][c] = (AA[i][0] * BB[0][c] )+( AA[i][1] * BB[1][c]); }
    }
```

```

cout<<endl;
}
cout<<"-----\n";
cout<<"Hasil Perkalian Matrik A dan B adalah : \n";
cout<<"-----\n";
//Memampilkan hasil Perkalian Matrik
for(int i=0; i<a1[0]; i++){
    cout<<endl;
    for (int c=0;c<b1[1];c++){
        cout<<"["<<hasil[i][c]<<" ] "; }
    }
cout<<"\n\nATAU\n";
for(int i=0; i<a1[0]; i++){
    cout<<endl;
    for (int c=0;c<b1[1];c++){
        cout<<"["<<i+1<<"]["<<c+1<<" ] = "<<hasil[i][c]<<endl;
    }
}
}
else {
    //Menanyakan Pilihan
    cout<<"Kolom Matrik A harus sama dengan Baris Matrik B";
    cout<<"\nUlangi..? (Y/N) : "; cin>>pil;
    if (pil=='Y' || pil=='y'){
        goto ulang;
    }
}
getch();
}

```

2. Soal Latihan / Tugas

1. Array dua dimensi int A [3] [5] (tipe interger numerik) belum ada isinya dengan ilustrasi gambar seperti berikut :

	0	1	2	3	4
0					
1					
2					

Gambarkan kembali array tersebut beserta dengan isinya bila diisi dengan intruksi

<p>a.</p> <pre>for (J=0; J <= 4; J++) { for (I=0; I<=2; I++) A [I] [J] = I ; }</pre>	<p>b.</p> <pre>for (I=0; I <= 2; I++) { for (J=0; J<=4; J+=2) A [I] [J] = J ; }</pre>
<p>c.</p> <pre>N = 1; for (I=0; I <=2 ; I++) { for (J=0; J <= 4 - I ; J++) { A [I] [J] = N; N++; } }</pre>	<p>d.</p> <pre>N = 1; for (I=0; I <=2 ; I++) { for (J=0; J <= I + 2 ; J++) { A [I] [J] = N; N++; } }</pre>

2. Array dua dimensi yang sudah di buat dengan int A [5][5] & int B [3][3], susunlah program untuk mengisi array sehingga isinya menjadi :

a.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

b.

4	5	3
4	5	5
9	8	7

3. Buatlah 2 buah table, matriks A [3][2] dan matriks B [2] [3] kalikan isi table tersebut sehingga menghasilkan matriks seperti di bawah ini:

9	9	9
8	8	8
7	7	7

Referensi

- A.S, R. (2018). *LOGIKA ALGORITMA dan PEMROGRAMAN DASAR*. BANDUNG: MODULA.
- Charibaldi, N. (2004). *Modul Kuliah Algoritma Pemrograman II Edisi Kedua*. Yogyakarta.
- Davis, S. R. (2014). *C++ For Dummies* (7th ed.). John Wiley & Sons, Inc.

Deitel, P., & Deitel, H. (2014). *C++ How To Program* (9th ed.). United State of America: Pearson.

Munir, R. (2005). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.

Sjukani, M. (2014). *Algoritma dan Struktur Data 1 dengan C, C++ dan Java* (Edisi 9 ed.). Jakarta: Mitra Wacana Media.