

**MODUL 6****ARRAY(LARIK)****A. TUJUAN**

Setelah menyelesaikan modul ini diharapkan mahasiswa akan mampu:

1. Mahasiswa mengenal dan memahami penggunaan array.
2. Mahasiswa mampu membuat program sederhana dengan menerapkan konsep array.

**B. PETUNJUK**

1. Awali setiap aktivitas anda dengan doa, agar anda lancar dalam belajar
2. Kerjakan tugas-tugas praktikum dengan baik, jujur, dan sabar
3. Tanyakan kepada instruktur apabila ada hal-hal yang kurang jelas

**C. DASAR TEORI****1. Array**

Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai-nilai data di suatu array disebut dengan elemen-elemen array. Letak urutan dari elemen-elemen array ditunjukkan oleh suatu subscript atau indeks.

**2. Array satu demenisi**

- a. Setiap elemen array diakses melalui index
- b. Index array dimulai dari 0 bukan 1

Deklarasi array secara umum :

Tipe\_data\_array nama\_array[ukuran]

Contoh : float nilai[4]

Nilai [0]	Nilai [1]	Nilai [2]	Nilai [3]
<b>70</b>	<b>80</b>	<b>82</b>	<b>60</b>

Cara mengisi nilai ada 2 yaitu :

1. secara langsung  
float nilai[4] = {70, 80, 82, 60};
2. secara tak langsung  
nilai[0] = 70;  
nilai[1] = 80;  
nilai[2] = 82;  
nilai[3] = 60;

### 3. Array 2 dimensi

Array dua dimensi merupakan array yang terdiri dari m buah baris dan n buah kolom. Bentuknya dapat berupa matriks atau tabel.

Bentuk Umum:

Tipe\_data nama\_array[baris][kolom]

	0	1	2	← indeks kedua (tahun)
0	80	540	1032	
1	15	83	301	
2	8	12	15	
3	10	129	257	

indeks pertama  
(program kursus) ↑

`int data_lulus[4][3];`

Cara aksesnya misal kita ingin mendapatkan nilai 301 maka :

```
cout << data_lulus[2][1];
```

Cara mengisi nilainya adalah :

```
int huruf_A[8][8] = { { 0, 1, 1, 1, 1, 1, 0, 0 } ,
                      { 0, 1, 0, 0, 0, 1, 0, 0 } ,
                      { 0, 1, 0, 0, 0, 1, 0, 0 } ,
                      { 1, 1, 1, 1, 1, 1, 1, 0 } ,
                      { 1, 1, 0, 0, 0, 0, 1, 0 } ,
                      { 1, 1, 0, 0, 0, 0, 1, 0 } ,
                      { 1, 1, 0, 0, 0, 0, 1, 0 } ,
                      { 0, 0, 0, 0, 0, 0, 0, 0 } };
```

### 4. Array multidimensi

Array multi-dimensi merupakan array yang mempunyai ukuran lebih dari dua.

Bentuk pendeklarasian array sama saja dengan array dimensi satu maupun

array dimensi dua. Bentuk Umum:

Tipe\_data nama\_array[ukuran1][ukuran2][ukuran3].....[ukuran N]

Array juga dapat digunakan sebagai parameter dalam fungsi atau prosedur. Contoh :

```
void data_masuk(int nilai[],int total){ ... }
*jika menjadi parameter panjang array tidak perlu di tuliskan
```

Array memiliki kelemahan yaitu ukurannya statis sehingga tidak dapat di ubah. Hati - hati dalam mengakses index array Karena jika mengakses index yang tidak ada dalam array tersebut akan terjadi error. Misalnya kita menginialisasi array dengan panjang 5. Jika kita mengakses index ke 6 maka akan terjadi error

#### D. LATIHAN

1. Latihan 1 (mencari rata – rata dari suatu daftar nilai)  
Perbaiki error pada syntax di bawah ini

```
#include <iostream>
using namespace std;

int main(){
    int nilai[6] = {80, 70, 82, 90, 77}; //inisialisasi
    array float average; int jumlah = 0;

    cout << "daftar nilai : ";
    for(int i = 0; i < 5; i++){
        cout << nilai[6] << " ";
    }

    cout << endl;

    for(int i = 1; i < 5; i++){
        jumlah += nilai[i];
    }

    average = jumlah / 5.0;

    cout << "rata - rata dari nilai tersebut adalah : " << average << endl;
}
```

- Latihan 2 (mencari nilai maksimal pada suatu array)  
Perbaiki eror pada syntax di bawah ini

```
#include <iostream>
using namespace std;

int main(){
    int n,s,maks;

    cout << "masukkan jumlah data : ";
    cin >> n; cout << endl;
    int data[n];

    for(int i = 0; i < 3; i++){
        cout << "masukkan data ke-" << i+1 << " : ";
        cin >> s;
        data[i] = s;
    }

    for(int i = 0; i < 3; i++){
        if(data[i] > maks){
            maks = data[i];
        }
    }

    cout << "data terbesar adalah : " << maks << endl;
}
```

3. Latihan 3 (operasi penjumlahan 2 matrik dengan ordo  $n \times n$ )  
Perbaiki eror pada syntax di bawah ini

```
#include <iostream>
using namespace std;

int main(){
    int n;

    cout << "masukkan ukuran ordo matrik pertama dan kedua : ";
    cin >> n;

    int matrik_1[n][n], matrik_2[n][n];

    cout << endl;
    cout << "masukkan nilai matrik pertama : " << endl;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < 2; j++){
            cout << "baris " << i << " kolom " << j << " : ";
            cin >> matrik_1[i][j];
        }
    }

    cout << endl;

    cout << "masukkan nilai matrik kedua : " << endl;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < 2; j++){
            cout << "baris " << i << " kolom " << j << " : ";
            cin >> matrik_1[i][j];
        }
    }

    cout << endl;
    cout << "hasil penjumlahan matrik pertama dan matrik kedua" << endl;

    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            cout << matrik_1[i][j] + matrik_2[i][j] << " ";
        }
        cout << endl;
    }
}
```

**E. TUGAS PRAKTIKUM**

1. Buatlah program untuk menampilkan banyaknya kemunculan suatu angka pada suatu baris angka. Jumlah baris angka di inputkan oleh user. batas angka pada suatu baris angka tidak lebih dari 100. Jika data yang di masukkan lebih besar dari 100 munculkan peringatan dan ulangi inputan

```

D:\projeK\coba\bin\Debug\coba.exe
masukkan jumlah angka : 5
masukkan angka ke-1 : 4
masukkan angka ke-2 : 101
data yang di masukkan lebih dari 100, silahkan masukkan ulang
masukkan angka ke-2 : 900
data yang di masukkan lebih dari 100, silahkan masukkan ulang
masukkan angka ke-2 : 2
masukkan angka ke-3 : 3
masukkan angka ke-4 : 2
masukkan angka ke-5 : 1

frekuensi kemunculan angka :
1 : 1 kali
2 : 2 kali
3 : 1 kali
4 : 1 kali

Process returned 0 (0x0)   execution time : 42.791 s
Press any key to continue.

```

2. Buatlah program perkalian matriks , lalu transpose hasil perkalian matriks tersebut

```

=====
program perkalian dan tranpose matriks 3x3
=====

input data matriks A[0][1] : 1
input data matriks A[1][2] : 1
input data matriks A[2][3] : 1
input data matriks A[1][1] : 4
input data matriks A[2][2] : 4
input data matriks A[3][3] : 4
input data matriks A[2][1] : 6
input data matriks A[3][2] : 6
input data matriks A[4][3] : 6

input data matriks B[0][1] : 2
input data matriks B[1][2] : 2
input data matriks B[2][3] : 4
input data matriks B[1][1] : 54
input data matriks B[2][2] : 5
input data matriks B[3][3] : 6
input data matriks B[2][1] : 7
input data matriks B[3][2] : 8
input data matriks B[4][3] : 1

matriks A
  1   1   1
  4   4   4
  6   6   6

matriks B
  2   2   4
  54  5   6
  7   8   1

matriks AxB
  63  15  11
  252 60  44
  378 90  66

=====
Hasil tranpose matriks AxB
=====
63 252 378
15 60 90
11 44 66
press enter to exit!

```