

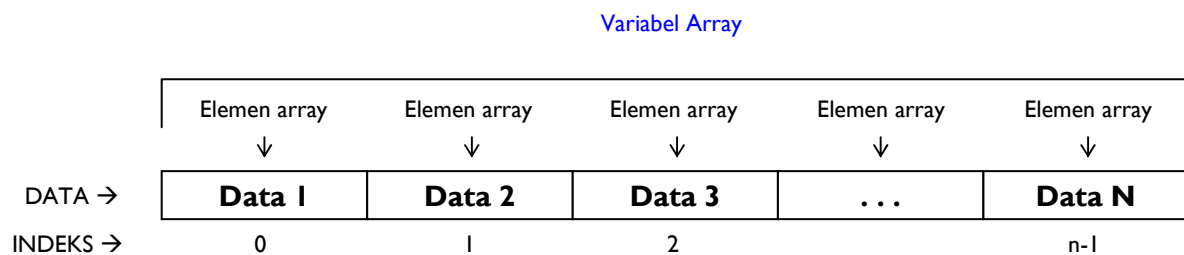
ARRAY

Materi Praktikum	Pemograman Bahasa C++ dengan menggunakan variabel Array
Durasi	180 menit
TIU/TIK	<ol style="list-style-type: none">1. Pendahuluan2. Deklarasi Variabel Array3. Array Berdimensi Satu4. Array Berdimensi Dua5. Operasi Matriks
Referensi	<ol style="list-style-type: none">1. Diktat dan Literatur2. Schildt, C++: <i>The Complete Reference Third Edition</i>, McGraw-Hill, 1998

I. PENDAHULUAN

Pada materi-materi sebelumnya kita telah membahas struktur dasar dari program, meliputi pengenalan bahasa dan variabel, tipe dan konstanta, ekspresi, proses percabangan, dan proses perulangan. Hampir seluruh program sederhana mampu dikerjakan oleh kita. Namun, untuk mengerjakan program yang lebih kompleks, dimana umumnya dibutuhkan variabel dalam jumlah yang besar, kita membutuhkan sesuatu yang dapat memudahkan kita dalam mengerjakannya. Memang, kita dapat membuat variabel tersebut satu demi satu. Bagaimana bila kita harus membuat 100 variabel atau lebih dalam suatu program? Tentu akan memakan waktu yang lama untuk membuatnya dan memungkinkan program tidak berjalan secara efisien. Untuk itulah dalam program C++ diperkenalkan konsep *array*.

Array pada dasarnya adalah sebuah variabel yang memiliki suatu nama tertentu, namun didalamnya terbagi menjadi variabel-variabel yang dibedakan berdasarkan nomor indeks. Untuk memudahkan pemahaman mengenai konsep array, berikut adalah ilustrasinya :



Dengan memperhatikan ilustrasi di atas, kita dapat melihat bahwa kita bisa memasukkan banyak data dalam satu variabel array. Saat kita membuat variabel array, kita dapat membagi variabel tersebut menjadi *n* buah elemen array. Setiap elemen ditandai dengan suatu nomor indeks yang selalu dimulai dari angka 0 (nol). Sehingga bila kita membuat variabel array bernama *num* dan membaginya menjadi 5 buah elemen, maka variabel yang terbentuk adalah : **num[0], num[1], num[2], num[3], dan num[4]**.

Untuk memanggil nilai dari elemen tersebut, kita hanya perlu memanggil nama variabelnya dan nomor indeksnya saja. Selanjutnya kita akan membahas tentang bagaimana mendeklarasikan variabel array tersebut dalam program.

II. DEKLARASI VARIABEL ARRAY

Bentuk umum dalam mendeklarasikan suatu variabel array adalah sebagai berikut :

```
tipe data namaVariabel [jumlah_elemen]
```

Untuk mendeklarasikan suatu variabel array, komponen yang dibutuhkan adalah :

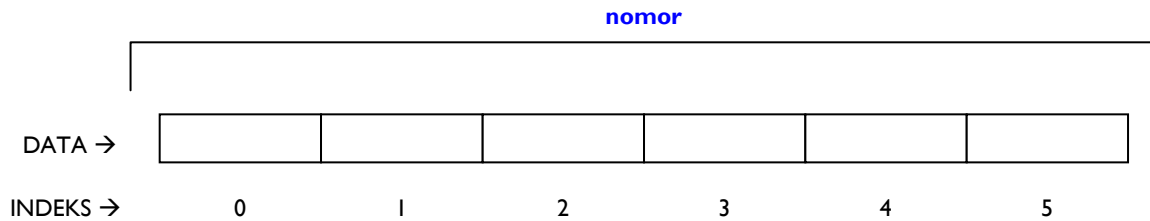
- **Tipe data elemen array** : Tipe data apakah yang akan disimpan dalam variabel array ini
- **Nama array** : Merupakan nama dari variabel array
- **Jumlah elemen array** : Jumlah elemen yang dibutuhkan dalam array tersebut

Terdapat dua cara umum dalam mendeklarasikan variabel array :

- a. Mendeklarasikan variabel array, **tanpa** memasukkan nilai ke dalam variabel tersebut

int nomor [6] ;

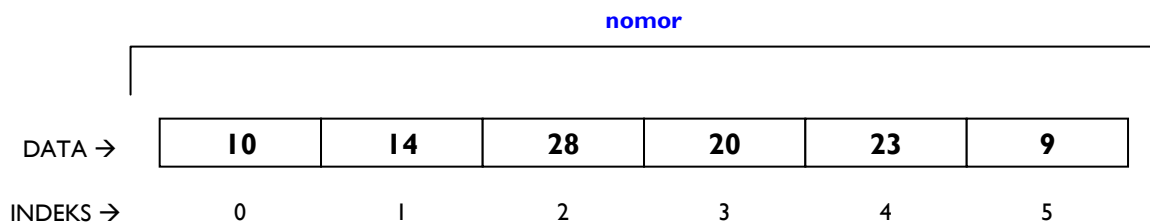
Instruksi tersebut menyatakan bahwa suatu variabel array bernama 'nomor' bertipe integer dideklarasikan dan memesan elemen array berjumlah 6 buah. Ilustrasi yang terbentuk dari instruksi ini adalah sebagai berikut:



- b. Mendeklarasikan variabel array, **dengan** memasukkan nilainya sekaligus ke dalam variabel tersebut

int nomor [6] = { 10, 14, 28, 20, 23, 9 } ;

Instruksi tersebut menyatakan bahwa suatu variabel array bernama 'nomor' bertipe integer dideklarasikan dan memesan elemen array berjumlah 6 buah. Kemudian, nilai untuk setiap elemen array disimpan sesuatu urutan pada instruksi tersebut. Ilustrasi yang terbentuk dari instruksi ini adalah sebagai berikut:



III. ARRAY BERDIMENSI SATU

Array berdimensi satu adalah data-data akan disimpan dalam satu baris array, sehingga hanya dibutuhkan satu penomoran indeks. Berikut adalah input program untuk array berdimensi satu :

INPUT DATA #1:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int nilai[5];
    int i;

    cout << "Masukkan nilai ujian mahasiswa [5 mahasiswa] :\n";
    for (i = 0; i < 5; i++)
    {
        cout << "Mahasiswa " << (i+1) << " : ";
        cin >> nilai[i]; // Memasukkan nilai ke dalam elemen array
    }
    cout << "\nNilai mahasiswa yang telah Anda masukkan adalah : ";
    for (i = 0; i < 5; i++)
    {
        cout << "\nMahasiswa " << (i+1) << " : " << nilai[i];
    }
    cout << "\n\nTerima kasih.";
    getch();
    return 0;
}
```

OUTPUT DATA #1:

Masukkan nilai ujian mahasiswa [5 mahasiswa] :

Mahasiswa 1 : 23
Mahasiswa 2 : 67
Mahasiswa 3 : 99
Mahasiswa 4 : 54
Mahasiswa 5 : 77

Nilai mahasiswa yang telah Anda masukkan adalah :

Mahasiswa 1 : 23
Mahasiswa 2 : 67
Mahasiswa 3 : 99
Mahasiswa 4 : 54
Mahasiswa 5 : 77

Terima kasih.

KETERANGAN

Hal utama yang diperlukan dalam melakukan proses input-output pada variabel array adalah penggunaan fungsi *looping* (dalam hal ini adalah perintah **for**). Dengan melakukan proses perulangan, kita akan lebih mudah dalam memasukkan data ke dalam array dan mengeluarkannya kembali. Apabila kita memerlukan 5 buah data untuk dimasukkan ke dalam masing-masing elemen, maka kita perlu melakukan proses perulangan sebanyak 5 kali juga agar seluruh elemen dapat diisi oleh data.

Data-data yang terdapat di dalam array dapat diolah lebih lanjut untuk menghasilkan sesuatu yang lain. Dalam kasus kali ini, kita akan mencoba mencari nilai rata-rata dari 5 mahasiswa :

INPUT DATA #2:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    float nilai[5];
    float total = 0;
    int i;

    // proses memasukkan data ke array
    cout << "Masukkan nilai ujian mahasiswa [5 mahasiswa] :\n";
    for (i = 0; i < 5; i++)
    {
        cout << "Mahasiswa " << (i+1) << " : ";
        cin >> nilai[i];
    }

    // proses menampilkan data dari array
    cout << "\nNilai mahasiswa yang telah Anda masukkan adalah : ";
    for (i = 0; i < 5; i++)
    {
        cout << "\nMahasiswa " << (i+1) << " : " << nilai[i];
    }

    // proses menghitung nilai rata-rata
    for (i = 0; i < 5; i++)
    {
        total += nilai[i];          // sama halnya dengan : total = total + nilai[i]
    }

    cout << "\n\nNilai rata-rata mahasiswa adalah : "<< (total/5);

    getch();
    return 0;
}
```

OUTPUT DATA #2:

Masukkan nilai ujian mahasiswa [5 mahasiswa] :

Mahasiswa 1 : 13
Mahasiswa 2 : 56
Mahasiswa 3 : 39
Mahasiswa 4 : 58
Mahasiswa 5 : 30

Nilai mahasiswa yang telah Anda masukkan adalah :

Mahasiswa 1 : 13
Mahasiswa 2 : 56
Mahasiswa 3 : 39
Mahasiswa 4 : 58
Mahasiswa 5 : 30

Nilai rata-rata mahasiswa adalah : 39.2

KETERANGAN

Pada contoh kedua ini kita menambahkan proses untuk menghitung rata-rata. Setiap nilai pada elemen dijumlahkan dengan cara menjumlahkannya dengan variabel **total**. Selanjutnya nilai akhir dari **total** dibagi dengan jumlah data, yakni 5. Sehingga secara matematis kita dapat menerjemahkannya sebagai berikut :

$$\text{total} = 13 + 56 + 39 + 58 + 30$$

$$\text{rata-rata} = \text{total}/5$$

IV. ARRAY BERDIMENSI DUA

Array berdimensi dua memberikan kita kesempatan untuk menyimpan data baik dalam bentuk baris maupun dalam bentuk kolom. Karena kita akan menyimpan data ke dalam baris dan kolom, maka dibutuhkan dua buah nilai indeks. Bentuk umum dalam mendeklarasikan array berdimensi dua adalah sebagai berikut :

tipe data namaVariabel [jumlah_baris] [jumlah_kolom]

Dalam contoh berikut misalkan kita akan menyimpan data-data berikut ke dalam program :

Nama	Minggu 1	Minggu 2	Minggu 3
Nobita	5	3	2
Suneo	72	88	60
Shizuka	90	100	85
Giant	55	76	46

Untuk membuat program, kita perlu mendeklarasikan variabel array terlebih dahulu :

```
int Nilai [4] [3] ;
```

Untuk program selengkapnya, perhatikan input data berikut :

INPUT DATA #3:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int Nilai[4][3];
    int pilih, i;

    // memasukkan data ke dalam array berdimensi dua
    Nilai[0][0] = 5;    // nilai Nobita minggu ke 1
    Nilai[0][1] = 3;    // nilai Nobita minggu ke 2
    Nilai[0][2] = 2;    // nilai Nobita minggu ke 3
    Nilai[1][0] = 72;   // nilai Suneo minggu ke 1
    Nilai[1][1] = 88;   // nilai Suneo minggu ke 2
    Nilai[1][2] = 60;   // nilai Suneo minggu ke 3
    Nilai[2][0] = 90;   // nilai Shizuka minggu ke 1
    Nilai[2][1] = 100;  // nilai Shizuka minggu ke 2
    Nilai[2][2] = 85;   // nilai Shizuka minggu ke 3
    Nilai[3][0] = 55;   // nilai Giant minggu ke 1
    Nilai[3][1] = 76;   // nilai Giant minggu ke 2
    Nilai[3][2] = 46;   // nilai Giant minggu ke 3

    // menampilkan nilai
    while (1)
    {
        cout << "(0 = Nobita, 1 = Suneo, 2 = Shizuka, 3 = Giant)\n";
        cout << "Nama Siswa (masukkan kode angka) : "; cin >> pilih;

        if ((pilih == 0) || (pilih == 1) || (pilih == 2) || (pilih == 3))
            break; //keluar dari loop-while
    }

    cout << "\nNama Siswa : ";
    if (pilih == 0) cout << "Nobita\n";
    if (pilih == 1) cout << "Suneo\n";
    if (pilih == 2) cout << "Shizuka\n";
    if (pilih == 3) cout << "Giant\n";

    for (i = 0; i < 3 ; i++)
    {
        cout << "Minggu " << i+1 << " : " << Nilai[pilih][i] << "\n";
    }
    getch();
    return 0;
}
```

OUTPUT DATA #3:

(0 = Nobita, 1 = Suneo, 2 = Shizuka, 3 = Giant)

Nama Siswa (masukkan kode angka) : 2

Nama Siswa : Shizuka

Minggu 1 : 90

Minggu 2 : 100

Minggu 3 : 85

KETERANGAN

Pada program ini input data dilakukan di awal program secara otomatis. Selanjutnya user memasukkan kode angka untuk memilih siswa. Harap diperhatikan, pada proses pemilihan siswa ini terdapat instruksi **while**. Instruksi ini berfungsi agar user tidak memasukkan kode lain selain 0-3. Andaikata user memasukkan angka lain, misal angka 6, maka proses perulangan akan terjadi dan user akan ditanyakan kembali hingga nilai yang dimasukkan benar.

Selanjutnya proses menampilkan data dieksekusi dimana nilai baris pada array ditentukan berdasarkan siswa yang kita pilih. Nilai setiap minggu ditampilkan karena menggunakan proses perulangan **for**.

V. OPERASI MATRIKS

Salah satu proses yang dilakukan oleh array adalah operasi matriks. Seperti yang telah diketahui, baris dan kolom adalah elemen utama dalam membentuk suatu matrik, dan hal tersebut tersedia pada array berdimensi dua. Dengan kata lain, array berdimensi dua akan membantu kita dalam melakukan berbagai operasi matriks, seperti penjumlahan, pengurangan, perkalian, transpose, inverse, dan lain sebagainya.

PENJUMLAHAN MATRIKS

Untuk menjumlahkan kedua matriks, dibutuhkan matriks yang memiliki ordo yang sama. Ordo adalah ukuran baris dan kolom. Apabila kita memiliki suatu matriks berukuran 2×3 , maka kita memerlukan matriks lain yang berukuran 2×3 juga untuk dapat dijumlahkan. Proses yang terjadi adalah setiap elemen pada matriks pertama dijumlahkan dengan elemen pada matriks kedua sesuai dengan alamat baris dan kolomnya.

Misalkan, **Matriks A** dijumlahkan dengan **Matriks B** dengan ordo $m \times n$. Makanya hasil adalah **Matriks C** dimana ukuran ordonya adalah $m \times n$.

Perhatikan contoh berikut ini :

INPUT DATA #4:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int matrikA[3][4] = {7,6,9,3,5,7,4,2,7,9,2,2};
    int matrikB[3][4] = {{3,6,8,6},{7,3,9,1},{8,5,9,4}};
    int matrikC[3][4];
    int i, j;
    //penjumlahan matriks
    for (i=0; i<3; i++)
    {
        for (j=0; j<4; j++)
        {
            matrikC[i][j] = matrikA[i][j] + matrikB[i][j];
        }
    }

    //menampilkan data masing-masing matriks
    cout << "\nData Matriks A : \n";
    cout << "===== \n";
    for (i=0; i<3; i++)
    {
        for (j=0; j<4; j++)
            cout << matrikA[i][j] << " ";
        cout << "\n";
    }

    cout << "\nData Matriks B : \n";
    cout << "===== \n";
    for (i=0; i<3; i++)
    {
        for (j=0; j<4; j++)
            cout << matrikB[i][j] << " ";
        cout << "\n";
    }

    cout << "\nData Matriks C = Matriks A + Matriks B : \n";
    cout << "===== \n";
    for (i=0; i<3; i++)
    {
        for (j=0; j<4; j++)
            cout << matrikC[i][j] << " ";
        cout << "\n";
    }
    getch();
    return 0;
}
```

OUTPUT DATA #4:

```
Data Matriks A :
=====
7 6 9 3
5 7 4 2
7 9 2 2

Data Matriks B :
=====
3 6 8 6
7 3 9 1
8 5 9 4

Data Matriks C = Matriks A + Matriks B :
=====
10 12 17 9
12 10 13 3
15 14 11 6
```

KETERANGAN

Pada program ini data dimasukkan ke dalam array pada awal program. Ada beberapa cara dalam memasukkan data langsung (seperti yang ditunjukkan dalam program ini). Selanjutnya program akan menjumlahkan masing-masing elemen (sesuai baris dan kolomnya) dan hasilnya disimpan ke dalam MatrikC. Pada bagian akhir program, seluruh matriks ditampilkan pada layar.

MEMPEROLEH BILANGAN TERBESAR

Pada program ini, data akan dimasukkan ke dalam array, kemudian masing-masing data diperiksa hingga pada akhirnya didapatkan nilai terbesar.

INPUT DATA #5:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int data[10] = {3,23,10,44,87,8,29,37,26,21};
    int max, i;

    // menampilkan data pada array
    cout << "Data pada array : \n";
    for (i=0; i<10; i++)
    {
        cout << data[i] << "\n";
    }

    // memperoleh bilangan terbesar
    max = data[0];
    for (i=1; i<10; i++)
    {
        if (data[i] > max)
            max = data[i];
    }

    cout << "\nBilangan terbesar yang diperoleh adalah : " << max;

    getch();
    return 0;
}
```


OUTPUT DATA #5:

Data pada array :

3
23
10
44
87
8
29
37
26
21

Bilangan terbesar yang diperoleh adalah : 87

KETERANGAN

Langkah pertama dalam program ini adalah memasukkan 10 data ke dalam array. Selanjutnya variabel max dimasukkan data sesuai data pertama dalam array. Inti proses dari program ini adalah sebagai berikut :

- Setiap data pada array dicek satu persatu dan dibandingkan dengan nilai yang terdapat pada variabel max.
- Apabila data pada array lebih besar dari data max sekarang, maka data array tersebut disimpan ke dalam variabel max, menimpa nilai yang lebih kecil tadi.
- Namun apabila data pada array lebih kecil, maka proses perulangan for dilanjutkan, dimana nilai pada variabel max tetap dibiarkan (tidak mengalami perubahan).

MENCARI SUATU DATA PADA ARRAY

Prinsip utama dalam program ini adalah mengecek satu per satu data yang ada untuk disesuaikan dengan nilai yang dimasukkan oleh user. Apabila ditemukan nilai yang sama, maka program akan berakhir dengan menampilkan kesimpulan dari pencarian tadi. Berikut adalah contoh sederhana dari program tersebut :

INPUT DATA #6:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int i,x,found;
    int data[] = {3, 12, 84, 111, 48, 28, 39, 100, 31, 1, 99, 18, 48};
    cout << "Data yang Anda cari : ";
    cin >> x;

    found=0;
    for (i=0; i<sizeof(data)/sizeof(int); i++)
    {
        if (data[i] == x)
        {
            found = 1;    // ubah kondisi menjadi true
            break;        // keluar dari perulangan for
        }
    }

    if (found)
        cout << "Data tersebut ada pada posisi ke-" << (i+1);
    else
        cout << "Data tersebut tidak ada";
    getch();
    return 0;
}
```

OUTPUT DATA #6:

Data yang Anda cari : 31
Data tersebut ada pada posisi ke-9

KETERANGAN

Variabel **found** dalam hal ini berfungsi sebagai kontrol status. Apabila data ditemukan, maka program akan memberikan kesimpulan bahwa data telah ditemukan dan memberikan informasi lokasinya. Namun jika tidak ditemukan, maka program akan memberikan kesimpulan bahwa data tidak terdapat di dalam array yang dicari.

Hal lain yang menarik disini adalah kita menemukan suatu perhitungan :

```
sizeof(data)/sizeof(int)
```

Kita mengetahui ukuran dari tipe integer dan telah diketahui bahwa ukuran dari suatu variabel adalah jumlah total dari nilai-nilai yang ada didalamnya. Sehingga dengan membagi ukuran data dengan ukuran dari tipe integer, maka akan didapatkan jumlah angka yang terdapat di dalam array tersebut. Jumlah inilah yang menentukan berapa banyak proses perulangan harus dilakukan.

VI. HASIL CAPAIAN

1. Mahasiswa mengetahui dan memahami pengertian array di C++ untuk digunakan dalam pemrograman
2. Mahasiswa memahami cara mendeklarasikan array dan jenis-jenis dari array yang dapat dibentuk
3. Mahasiswa mampu membuat program yang melibatkan penggunaan variabel array untuk mencapai hasil yang diinginkan

VII. ANALISA DAN TUGAS

Perhatikan kedua matriks di bawah ini :

$$A = \begin{vmatrix} 2 & 3 & 8 \\ 5 & 2 & 1 \\ 8 & 4 & 7 \\ 4 & 1 & 6 \end{vmatrix}$$

$$B = \begin{vmatrix} 7 & 4 & 1 & 2 \\ 8 & 7 & 5 & 6 \\ 2 & 8 & 2 & 5 \end{vmatrix}$$

Tugas Anda adalah :

Mencari nilai dari **Matriks C**, dimana **Matriks C** adalah hasil perkalian **Matriks A** dengan **Matriks B**

$$[C] = [A] \times [B]$$

Nilai Tambah :

Nilai tambah akan diberikan apabila Anda dapat memperoleh nilai dari **Matriks D**, yaitu matriks yang menyatakan hasil penjumlahan dari **Matriks A Tranpose** dengan **Matriks B**

$$[D] = [A]^T + [B]$$

Tugas dikumpulkan kepada asisten.

(format file *.cpp. Berikan Nama dan NIM anda pada bagian atas program).