

NAMA MATA KULIAH



**UNIVERSITAS
INABA**

**MODUL - Sesi 2
STRUKTUR DATA**

Penyusun : M THAMRIN BASRI

STRUKTUR DATA ARRAY

PENGERTIAN ARRAY

Array adalah suatu struktur yang terdiri dari sejumlah elemen yang memiliki tipe data yang sama. Elemen-elemen array tersusun secara sekuensial dalam memory computer. Array dapat berupa satu dimensi, dua dimensi, tiga dimensi ataupun banyak dimensi (multi dimensi).

Array atau larik sendiri di definisikan sebagai pemesanan alokasi memory berurutan. Definisi ini kurang tepat, karena terjadi kerancuan antara struktur data dan representasinya. Memang benar array hampir selalu di implementasikan menggunakan memory berurutan tapi tidak selalu demikian. Semua elemem array bertipe sama. Array cocok untuk organisasi kumpulan data homogen yang ukuran atau jumlah elemen maksimumnya telah diketahui dari awal. Homogen adalah bahwa setiap elemen dari sebuah array tertentu haruslah mempunyai tipe data yang sama.

Dari perngertian struktur dan pengertian array di atas, maka dapat ditarik kesimpulan bahwa struktur array adalah kumpulan elemen-elemen data yang digabungkan menjadi suatu kesatuan yang memiliki tipe homogen (sama).

Karakteristik Array :

- **Mempunyai batasan dari pemesanan alokasi memori(bersifat statis).**
- **Mempunyai tipe data sama (bersifat homogen).**
- **Dapat diakses secara acak.**

Deklarasi Array :

Ada tiga hal yang harus diketahui dalam pendeklarasian, yaitu :

- **Type data array.**
- **Nama variable array.**
- **Subkrip / index array.**

ARRAY SATU DIMENSI

Array Satu Dimensi tidak lain adalah kumpulan elemen-elemen identik yang tersusun dalam satu baris. Elemen - elemen tersebut memiliki tipe data yang sama, tetapi isi dari elemen tersebut boleh berbeda.

Bentuk umum dari array:

Nama Array[n] = {elemen0, elemen1, elemen2, elemen3, elemen4, ..., n}

N=jumlah elemen

Array Dua Dimensi

Array Dua Dimensi sering digambarkan sebagai sebuah

matriks, merupakan perluasan dari array satu dimensi. Jika array satu dimensi hanya terdiri dari sebuah baris dan beberapa kolom elemen, maka array dua dimensi terdiri dari beberapa baris dan beberapa kolom elemen yang bertipe sama.

Bentuk umum:

Nama Array [m][n];

Atau

Nama Array [m][n]={ {a,b,...,z},{1,2,....,n-1} };

Contoh:

Double matrik [4][4];

Bool papan [2][2] = { {true,false} };

Pendeklarasian array dua dimensi hampir sama dengan pendeklarasian array satu dimensi, kecuali bahwa array dua dimensi terdapat dua jumlah elemen yang terdapat di dalam kurung siku dan keduanya boleh tidak sama.

Elemen array dua dimensi diakses dengan menuliskan kedua indeks elemennya dalam kurung siku seperti pada contoh berikut:

//papan nama memiliki 2 baris dan 5 kolom

Bool papan [2][5];

Papan[0][0] =true;

Papan[0][4]=false;

Papan[1][2]=true;

Papan[1][4]=false;

Struktur array

Struktur data Array adalah organisasi kumpulan data homogen yang ukuran atau jumlah elemen maksimumnya telah diketahui dari awal. Array umumnya disimpan di memori komputer secara kontigu (berurutan). Deklarasi dari array adalah

sebagai berikut:

int A[5]; artinya variabel A adalah kumpulan data sebanyak 5 bilangan bertipe integer.

Operasi terhadap elemen di array dilakukan dengan pengaksesan langsung. Nilai di masing-masing posisi elemen dapat diambil dan nilai dapat disimpan tanpa melewati

posisi-posisi lain. Terdapat dua tipe operasi, yaitu:

- 1. Operasi terhadap satu elemen/posisi dari array**
- 2. Operasi terhadap array sebagai keseluruhan**

Dua operasi paling dasar terhadap satu elemen/posisi adalah

- 1. Penyimpanan nilai elemen ke posisi tertentu di array**
- 2. Pengambilan nilai elemen dari posisi tertentu di array**

Penyimpanan dan Pengambilan Nilai

Biasanya bahasa pemrograman menyediakan sintaks tertentu untuk penyimpanan dan pengambilan nilai elemen pada posisi tertentu di array.

Contoh:

A[10] = 78, berarti penyimpanan nilai 78 ke posisi ke-10 dari array A

C = A[10], berarti pengambilan nilai elemen posisi ke-10 dari array A

Keunggulan dan Kelemahan Array

Keunggulan array adalah sebagai berikut:

- Array sangat cocok untuk pengaksesan acak. Sembarang elemen di array dapat diacu secara langsung tanpa melalui elemen-elemen lain.**

- Jika berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemen-elemen tetangga, baik elemen pendahulu atau elemen penerus
- Jika elemen-elemen array adalah nilai-nilai independen dan seluruhnya harus terjaga, maka penggunaan penyimpanannya sangat efisien

Kelemahan array adalah sebagai berikut:

Array mempunyai fleksibilitas rendah, karena array mempunyai batasan sebagai berikut:

- Array harus bertipe homogen. Kita tidak dapat mempunyai array dimana satu elemen adalah karakter, elemen lain bilangan, dan elemen lain adalah tipe-tipe lain
- Kebanyakan bahasa pemrograman mengimplementasikan array statik yang sulit diubah ukurannya di waktu eksekusi. Bila penambahan dan pengurangan terjadi terus-menerus, maka Representasi statis
 - Tidak efisien dalam penggunaan memori
 - Menyiakan banyak waktu komputasi
 - Pada suatu aplikasi, representasi statis tidak dimungkinkan.

Contoh deklarasi array dalam bahasa C++ adalah sebagai berikut :

int A[10], artinya variabel A adalah kumpulan data sebanyak 10 bilangan bertipe integer.

PENGURUTAN ARRAY

Pada dasarnya penggunaan array sangat luas tidak hanya digunakan pada bahasa pemrograman.

Contoh penggunaan array sebagai berikut :

- Array digunakan untuk suatu database, contoh : tabel.**
- Array digunakan untuk operasi matematika (vektor).**
- Digunakan untuk struktur data lain, contohnya list.**

Pengurutan Array

Pengurutan atau sorting adalah proses yang paling sering dilakukan dalam pengolahan data, pengurutan dibedakan menjadi dua, yaitu :

- Pengurutan internal**

Pengurutan dilakukan terhadap sekumpulan data di media memory internal komputer di mana data dapat diakses elemennya secara langsung.

- Pengurutan eksternal**

Pengurutan data di memory sekunder. Biasanya data bervolume besar sehingga tidak mampu dimuat semuanya di memory utama.

Operasi Dasar Pada Array

Operasi terhadap elemen array dilakukan dengan pengaksesan langsung. Nilai di masing-masing posisi elemen dapat di ambil dan nilai dapat disimpan tanpa melewati posisi-posisi lain.

Terdapat dua operasi, yaitu :

- Operasi terhadap satu elemen/posisi array**
- Operasi terhadap array sebagai keseluruhan**

Dua operasi paling dasar terhadap satu elemen adalah :

- ☐ **Penyimpanan nilai elemen ke posisi tertentu di array**

- ☐ **Pengambilan nilai elemen dari posisi tertentu di array**

Operasi dasar terhadap array secara keseluruhan adalah :

- ☐ **Operasi penciptaan**
- ☐ **Operasi penghancuran**
- ☐ **Operasi pemrosesan transversal**
- ☐ **Operasi pencarian (table look-up)**
- ☐ **Operasi sorting**

- ☐ **Penyimpanan dan Pengambilan Nilai**

Saat penyimpanan dan pengambilan nilai array, biasanya bahasa pemrograman menyediakan sintaks tertentu untuk penyimpanan dan pengambilan nilai elemen pada posisi tertentu di array.

Contoh :

A[5] = 78, berarti simpan nilai 78 ke posisi ke-5 dari array

A

C = A[5], berarti ambil nilai elemen posisi ke-5 dari array

A

- ☐ **Penciptaan dan Penghancuran**

Operasi penciptaan biasa disebut inisialisasi. Operasi ini untuk mempersiapkan struktur data untuk operasi-operasi berikutnya. Operasi penghancuran menyatakan ketidakberlakuan struktur data atau membebaskan memory, menyerahkan memory ke manajemen memory agar dapat di pergunakan keperluan lain. Operasi penghancuran penting terutama bila struktur data di implementasikan secara dinamis menggunakan pointer.

- ☐

Pemrosesan Transversal

Operasi pemrosesan transversal adalah pengolahan seluruh elemen secara sistematis.

☐ **Pencarian (Searching) di Array (table look-up)**

Pencarian di array (table look-up) adalah proses pencarian suatu nilai di array.

Klasifikasi pencarian di array adalah :

- **Pencarian sekuen (sequential searching), yaitu :**
Tanpa boolean, terbagi :

☐ **Tanpa sentinen**

☐ **Dengan sentinen**

Menggunakan boolean

- **Pencarian secara biner/dokotom (binary = dochotomy searching).**

Bentuk-Bentuk Array

- **Array Satu Dimensi**

Array satu dimensi yaitu kumpulan elemen-elemen identik yang hanya terdiri dari satu baris atau hanya satu kolom saja alamat penyimpanan data (indeks). Elemen-elemen tersebut memiliki tipe data yang sama, tetapi isi dari elemen tersebut boleh berbeda.

Bentuk umum :

Tipe_data namaArray[n] = {elemen0, elemen1, elemen2,.....,n};

n = jumlah elemen

contoh pada progam : int ukur[5] = {39, 40, 41, 38, 40};

- ☐ **int adalah tipe data yang berupa bilangan bulat.**
- ☐ **Ukur adalah nama variabel array.**
- ☐ **[5] adalah ukuran untuk menyatakan jumlah maksimal elemen array.**
- ☐ **{..} adalah tempat pemberian nilai/elemen array.**

- Array Dua Dimensi

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

Bentuk umum :

Tipe_data namaArray [m][n] = {{a,b,...z},{1,2,...,n-1}};

contoh : int lulus[4][3];

- ☐ **Nilai 4 untuk menyatakan banyaknya baris dan 3 untuk menyatakan banyaknya kolom.**

Pendeklarasian array dua dimensi hampir sama dengan pendeklarasian array satu dimensi, kecuali bahwa array dua dimensi terdapat dua jumlah elemen yang terdapat dikurung kurung siku dan keduanya boleh tidak sama. Elemen array dua dimensi diakses dengan menuliskan kedua indeks elemennya dalam kurung.

- ☐ **Pemetaan (mapping) array dua dimensi ke storage :**

Terbagi dua cara pandang (representasi) yang berbeda :

1) $M[i][j] = M[0][0] + \{(j - 1) * K + (i - 1)\} * L$

Secara kolom per kolom (coloumn major order / CMO)

2) $M[i][j] = M[0][0] + \{(i - 1) * N + (j - 1)\} * L$

Secara baris per baris (row major order / RMO)

Keterangan :

- ☐ **$M[i][j]$ = Posisi array yang di cari.**
- ☐ **$M[0][0]$ = Posisi alamat awal indeks array.**
- ☐ **i = Baris**
- ☐ **j = Kolom**
- ☐ **L = Ukuran memory type data**
- ☐ **K = Banyaknya elemen per kolom**
- ☐ **N = Banyaknya elemen per baris.**

Array MultiDimensi

Array ini seperti array dimensi dua tetapi dapat memiliki ukuran yang lebih besar. Sebenarnya array dimensi banyak ini tidak terlalu sering digunakan, tetapi sewaktu-waktu kalau dimensi yang dibutuhkan banyak, maka array ini sangat memegang peranan yang penting.

Bentuk umum pendeklarasian array multidimensi :

Tipe_data namaArray[ukuran1][ukuran2]...[ukuranN];

Sebagai contoh :

int data_huruf[2][8][8];

- **contoh di atas merupakan pendeklarasian array data_huruf sebagai array berdimensi tiga.**

Keunggulan Array

Keunggulan array adalah sebagai berikut :

- Array sangat cocok untuk pengaksesan acak. Sembarang elemen di array dapat diacu secara langsung tanpa melalui elemen-elemen lain.**
- Jika berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemen-elemen tetangga, baik elemen pendahulu atau elemen peberus.**
- Jika elemen-elemen array adalah nilai-nilai independen dan seluruhnya harus terjaga, maka penggunaan penyimpanannya sangat efisien.**

Kelemahan Array

Kelemahan array adalah sebagai berikut :

Array mempunyai fleksibilitas rendah, sehingga tidak cocok untuk berbagai aplikasi karena array mempunyai batasan sebagai berikut :

- **Array harus bertipe homogen, kita tidak dapat mempunyai array dimana satu elemen adalah karakter, elemen yang lain adalah bilangan atau tipe lain.**
- **Kebanyakan bahasa pemrograman menggunakan array statik yang sulit diubah ukurannya di waktu eksekusi. Bila penambahan dan pengurangan terjadi terus-menerus, maka representasi statis :**

- ☐ **Tidak efisien dalam penggunaan memory**
- ☐ **Menyiakan banyak waktu komputasi**
- ☐ **Pada suatu aplikasi, representasi statis tidak dimungkinkan**

Bila penambahan dan pengurangan terjadi terus-menerus, maka representasi statis (array) :

- ☐ **Tidak efisien dalam penggunaan memory**
- ☐ **Menyiakan banyak waktu komputasi**
- ☐ **Pada suatu aplikasi, representasi statis tidak dimungkinkan**

Kesimpulan

Struktur array adalah kumpulan elemen-elemen data yang digabungkan menjadi suatu kesatuan yang memiliki tipe homogen (sama). Array merupakan bagian dari struktur data yang dapat didefinisikan sebagai pemesanan alokasi memori sementara pada komputer. Apabila kita membuat program dengan data yang sudah kita ketahui batasnya, maka kita bisa menggunakan array (tipe data statis), namun apabila datanya belum kita ketahui batasnya maka gunakan pointer (tipe data dinamis). Elemen-elemen array tersusun

secara sekuensial dalam memori komputer. Array dapat berupa satu dimensi, dua dimensi, ataupun multidimensi.

Array

Pengertian Array

Array / Larik : Struktur Data Sederhana yang dapat didefinisikan sebagai pemesanan alokasi memory sementara pada komputer. Array dapat didefinisikan sebagai suatu himpunan hingga elemen yang teratur dan homogen. *Terurut* : Dapat diartikan bahwa elemen tersebut dapat diidentifikasi sebagai elemen pertama, elemen kedua dan seterusnya sampai elemen ke-n.

***Homogen* : Adalah bahwa setiap elemen dari sebuah.**

Array tertentu haruslah mempunyai type data yang sama. Sebuah Array dapat mempunyai elemen yang seluruhnya berupa integer atau character atau String bahkan dapat pula terjadi suatu Array mempunyai

elemen berupa Array.

Karakteristik Array

- **Mempunyai batasan dari pemesanan alokasi memory (Bersifat Statis)**
- **Mempunyai Type Data Sama (Bersifat Homogen)**
- **Dapat Diakses Secara Acak**

Tiga hal yang harus diketahui dalam mendeklarasikan array :

- **Type data array**
- **Nama variabel array**
- **Subskrip / index array**

Jenis Array (yang akan dipelajari) adalah :

- **Array Dimensi Satu (One Dimensional Array)**

Deklarasi Array Dimensi Satu

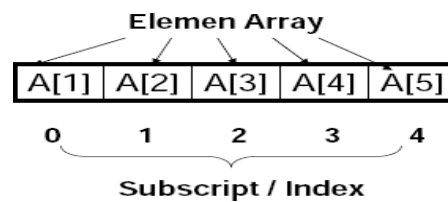
Dapat disebut juga dengan istilah vektor yang menggambarkan data dalam suatu urutan.

Deklarasi : Type_Data

Nama_Variabel [index]

Misalnya : int A[5];

Penggambaran secara Logika :




Contoh :

```
void main()
{ int bil
[5];
clrscr;
cout<<"Masukkan 5 bilangan
genap : "<<endl;for (int i = 0;
i < 5; i++)
{
cout<<"Bilan
gan ";cout<<
i + 1 <<" : ";
cin>> bil[i];
cout<<endl;
}
cout<<endl;
cout<<"5 bilangan genap yang
dimasukkan " <<endl;for (int i =
0; i < 5; i++)
cout<<"
```



```
"<<bil[i];
getch();
}
```



```
D:\ACPP\array1.exe
Masukkan 5 bilangan genap :
Bilangan 1 : 45
Bilangan 2 : 50
Bilangan 3 : 100
Bilangan 4 : 75
Bilangan 5 : 30

5 bilangan genap yang dimasukkan :
45 50 100 75 30
```

Gambar 1.1 Tampilan Program array dimensi 1

- **Rumus untuk menentukan jumlah elemen dalam Array :**

$$\prod_{i=1}^n \pi \text{ (Elemen Array)}$$

**Π = Perkalian dari elemen sebelumnya
(untuk array dimensi dua & tiga)**

Contoh :

**Suatu Array A dideklarasikan sbb : int
A[10]; maka jumlah elemen Array dimensi
satu tersebut adalah = 10**

Pemetaan Array Dimensi 1

$$\text{Rumus : } @A[i] = B + (i - 1) * L$$

Dimana :

@A[i] : Posisi Array yg dicari

B : Posisi awal index di

**memory komputer I : Subkrip
atau indeks array yg dicari**

L : Ukuran / Besar memory

suatu type data Contoh :

**Suatu Array A dideklarasikan sebagai
berikut :**

**int A[5]; dengan alamat awal index
berada di 0011 (H) dan ukuran memory
type data integer = 2. Tentukan berapa
alamat array A[3] ?**

Rumus : $@A[i] = B + (i - 1) * L$

Diketahui :
@A[i] = A[3]
B = 0011 (H)
i = 3
L = 2

Penyelesaian :
 $A[3] = 0011(H) + (3 - 1) * 2$
 $= 0011(H) + 4 (D)$
 $= 0011(H) + 4 (H)$
 $= 0015(H)$

4 Desimal = 4 Hexa

0	1	2	3	4
A[1]	A[2]	A[3]	A[4]	A[5]
0011	0013	0015	0017	0019

→

0	1	2	3	4	5	6	7	indeks
								value
21d2	21d4	21d6	21d8	21da	21dc	21de	21e0	alamat

```
#include <stdio.h>

void main(){
    int a[8];
    for (int i=0; i<8; i++){
        printf("%x\n", &a[i]); %x adalah hexadesimal
    }
}
```

- Array Dimensi Dua (Two Dimensional Array)

Deklarasi array Dimensi Dua Sering digunakan dalam matriks pada pemrograman.

Deklarasi : Type_Data Nama_Variabel

[Index1] [index2]; Misal : int A[3][2];

Penggambaran secara Logika :

	0	1
0		
1		
2		

Menentukan jumlah elemen dalam Array dimensi dua:

$$\sum_{i=1}^n \pi \text{ (Elemen array)}$$

Π = Perkalian dari elemen sebelumnya (untuk array dimensi dua & tiga)

Contoh :

Suatu Array X dideklarasikan sbb : int X[4][3], maka jumlah elemen Array dimensi dua tersebut adalah : $(4) * (3) = 12$ Terbagi Dua cara pandang (representasi) yang berbeda :

Secara Kolom Per Kolom (Coloumn Major Order/CMO)

$$@M[i][j] = M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

Secara Baris Per Baris (Row Major Order / RMO)

$$@M[i][j] = M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

Keterangan :

@M[i][j]: Posisi Array yg

dicari, M[0][0] : Posisi

alamat awal index array,

i : Baris

j : kolom

L : Ukuran memory type data

K : Banyaknya elemen

per kolom N : Banyaknya

elemen per baris

Penggambaran secara logika

Misal : `int M[3][2];` (Array dengan 3 Baris & 2 Kolom)

	0	1
0		
1		
2		

Berdasarkan Cara pandang :

1. Kolom Per Baris (Row Major Order / RMO)

M[0,0]	M[0,1]	M[1,0]	M[1,1]	M[2,0]	M[2,1]
--------	--------	--------	--------	--------	--------

Jumlah elemen per baris = 2

2. Baris Per Kolom (Coloumn Major Order / CM|O)

M[0,0]	M[1,0]	M[2,0]	M[0,1]	M[1,1]	M[2,1]
--------	--------	--------	--------	--------	--------

Jumlah elemen per kolom = 3

Pemetaan Array Dimensi

Dua Contoh Pemetaan :

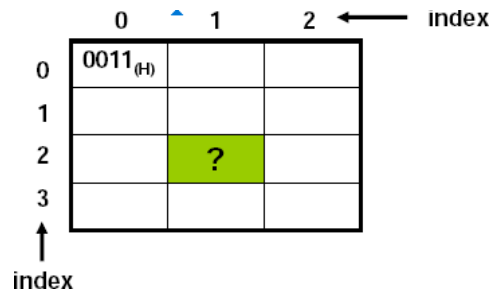
Suatu Array **X** dideklarasikan sebagai berikut :

Float **X[4][3]**, dengan alamat index **X[0][0]** berada

di **0011(H)** dan ukuran type data float = Tentukan

berapa alamat array **X[3][2]** berdasarkan cara

pandang baris dan kolom ?



Gambar 1.2 Contoh Pemetaan array dimensi dua

Secara Baris Per Baris (Row Major Oder / RMO)

$$@M[i][j] = @M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

$$\begin{aligned} X[3][2] &= 0011_{(H)} + \{(3 - 1) * 3 + (2 - 1)\} * 4 \\ &= 0011_{(H)} + 28_{(D)} \quad 1C_{(H)} \\ &= 0011_{(H)} + 1C_{(H)} \end{aligned}$$

$$= 002D_{(H)}$$

Secara Kolom Per Kolom (Coloumn Major Oder / CMO)

$$@M[i][j] = @M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

$$\begin{aligned} X[3][2] &= 0011_{(H)} + \{(2 - 1) * 4 + (3 - 1)\} * 4 \\ &= 0011_{(H)} + 24_{(D)} \quad 18_{(H)} \\ &= 0011_{(H)} + 18_{(H)} \end{aligned}$$

$$= 0029_{(H)}$$

Contoh Program Array Dimensi Dua :

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a[3][5];
    for (int i=0;i<3;i++)
    {
        for (int j=0;j<5;j++)
        {
            printf("%x ",&a[j][i]);
        }
        printf("\n");
    }
}
```

```
}  
getch();  
}
```



A screenshot of a program output displaying a 3x5 grid of hexadecimal values. The values are arranged in three rows and five columns, separated by spaces. The first row contains 12ff50, 12ff64, 12ff78, 12ff8c, and 12ffa0. The second row contains 12ff54, 12ff68, 12ff7c, 12ff90, and 12ffa4. The third row contains 12ff58, 12ff6c, 12ff80, 12ff94, and 12ffa8.

12ff50	12ff64	12ff78	12ff8c	12ffa0
12ff54	12ff68	12ff7c	12ff90	12ffa4
12ff58	12ff6c	12ff80	12ff94	12ffa8

Gambar 1.3 Tampilan Program Array Dimensi Dua

Daftar Pustaka

<http://dyananyun.blogspot.com/2010/02/struktur-data-array.html>
<http://elqieen.blogspot.com/2010/12/makalah-struktur-data-program-c.html>
<http://www.nusinau.com/pengenalan-array-dan-string/>
<http://www.nusinau.com/pengertian-struktur-data-2/>
<http://kintung05.wordpress.com/2008/11/23/pengertian-struktur-data/>
modul Praktikum Konsep Pemrograman : Array