

ARRAY



Overview

Dalam dunia nyata, struktur data yang dihadapi sangat beragam dan penggunaan variabel dengan tipe data dasar memiliki keterbatasan pada banyaknya nilai yang dapat disimpan. Dengan menggunakan *array* dan tipe data bentukan, dapat dilakukan pemodelan struktur data dengan lebih baik bahkan untuk struktur data yang relatif kompleks.



Tujuan

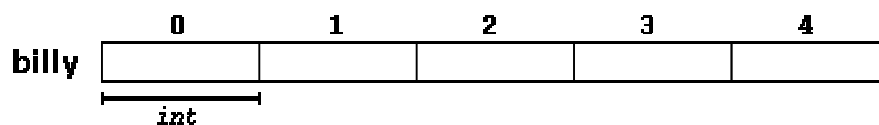
- ☑ Memahami tipe data *array* dan keuntungan yang dapat diberikan
- ☑ Memahami *array* yang memiliki dimensi lebih dari satu
- ☑ Dapat meng-implementasikan tipe data array dalam program
- ☑ Memahami cara menentukan tipe data bentukan dan menggunakannya dalam program

9.1 Array

Tipe data *array* adalah tipe data terstruktur yang merujuk kepada sebuah atau sekumpulan elemen yang mempunyai tipe data yang sama melalui indeks. *Array* biasanya disebut juga sebagai **tabel**, **vektor** atau **larik**.

Elemen dari *array* dapat diakses langsung **jika dan hanya jika** indeks terdefinisi (telah ditentukan nilainya sesuai dengan domain yang didefinisikan untuk indeks tersebut). Struktur data array disimpan dengan urutan yang sesuai dengan definisi indeks **secara kontigu (berurutan) dalam memori** komputer. Karena itu indeks haruslah merupakan suatu tipe data yang memiliki keterurutan (ada suksesor dan predesesor), misal tipe integer dan karakter.

Dengan kata lain Array adalah himpunan elemen (variable) dengan tipe yang sama dan disimpan secara berurutan dalam memory yang ditandai dengan memberikan index pada suatu nama variable. Contohnya, kita dapat menyimpan 5 nilai dengan tipe **int** tanpa harus mendeklarasikan 5 identifier variabel yang berbeda. Perhatikan contoh dibawah ini :



Bagian kosong diatas merepresentasikan *elemen* array, dalam kasus ini adalah nilai integer. Angka **0 - 4** merupakan index dan selalu dimulai dari **0**. Seperti penggunaan variable pada umumnya, array harus dideklarasikan terlebih dahulu, dengan format sbb :

type name [elements];

Maka contoh array diatas dideklarasikan sbb :

```
int billy [5];
```

Dilihat dari dimensinya, *array* dapat dibagi menjadi *Array* Satu Dimensi, *Array* Dua Dimensi dan *Array* Multi-Dimensi

9.1.1 Array Satu Dimensi

Sebelum digunakan, variabel array perlu dideklarasikan terlebih dahulu. Cara mendeklarasikan variabel array sama seperti deklarasi variabel yang lainnya, hanya saja diikuti oleh suatu indeks yang menunjukkan jumlah maksimum data yang disediakan.

Deklarasi Array Bentuk Umum pendeklarasian array :

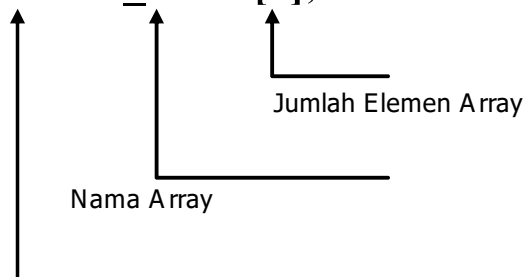
Type-Data Nama_Variabel[Ukuran]

Keterangan :

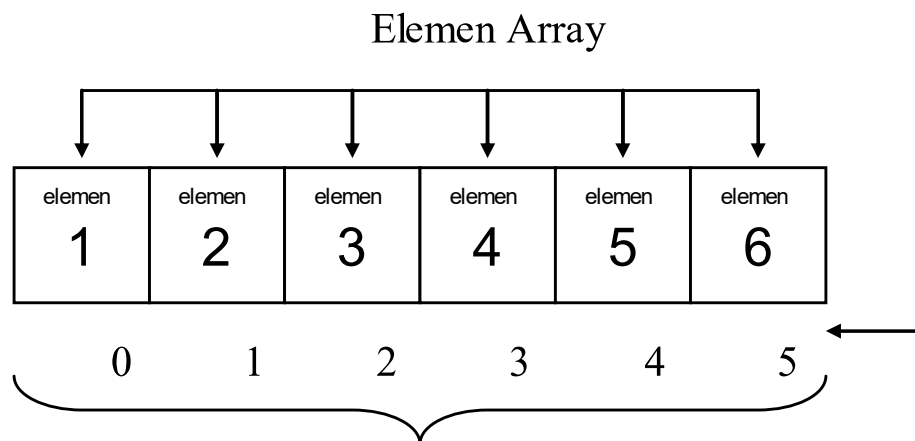
- ↳ Type Data: Untuk menyatakan type data yang digunakan.
- ↳ Ukuran : Untuk menyatakan jumlah maksimum elemen array.

Contoh Pendeklarasian Array

float Nil_Akhir[6];



Suatu array dapat digambarkan sebagai kotak panjang yang berisi kotak-kotak kecil didalam kotak panjang tersebut.



Suatu array, dapat diakses dengan menggunakan subscript atau index nya. Bentuk umum pengaksesan adalah :

Nama_Array[Subscript/Index]

Contoh

```

Nil_Akhir[3];
Nil_Akhir[1];
Nil_Akhir[0];
    
```

Contoh 28	Hasil
<pre> /*-----* /* Contoh 28 : program menggunakan * /* array * /*-----* #include <iostream.h> #include <conio.h> void main() { float suhu[5]; // Array dengan elemen 5 bertipe float clrscr(); // Membaca data dari key board dan meletakkan array cout << "Masukkan 5 buah data suhu" << endl; for (int i = 0; i < 5; i++) { cout << i + 1 << " . "; cin >> suhu[i]; } // Menampilkan isi array ke layar cout << "Data suhu yang anda masukkan : " << endl; for (i = 0; i < 5; i++) cout << suhu[i] << endl; } </pre>	<pre> Masukkan 5 buah data suhu 1 : 27.5 ↵ 2 : 28 ↵ 3 : 27.5 ↵ 4 : 30.5 ↵ 5 : 27 ↵ Data suhu yang anda masukkan 27.5 28 27.5 30.5 27 </pre>

Tampak diatas terdapat pernyataan :

- **float suhu[5];** menyatakan array **suhu** dapat menyimpan 5 (lima) buah data bertipe **float**.
- **suhu[i]** menyatakan elemen suhu dengan *subscript* sama dengan **i**.
- **cin >> suhu[i];** : membaca data dari keyboard dan meletakkan ke elemen nomor **i** pada array **suhu**.
- **cout << suhu[i];** : akan menampilkan elemen bernomor **i** pada array **suhu**.

Apa output dari program berikut ?

Contoh 29

```
/* ----- */
/* Program Array Satu Dimensi */
/* ----- */
#include<conio.h>
#include<stdio.h>
#include<iostream.h>
#include<iomanip.h>

main()
{
    int i;
    char nama[5][20];
    float nilai1[5];
    float nilai2[5];
    float hasil[5];

    clrscr();
    for(i=1;i<=2;i++)
    {
        cout<<"Data Ke - "<<i<<endl;
        cout<<"Nama Siswa : "; gets(nama[i]);
        cout<<"Nilai Teori : "; cin>>nilai1[i];
        cout<<"Nilai Praktek : "; cin>>nilai2[i];
        hasil[i] = (nilai1[i] * 0.40)+ (nilai2[i] * 0.60);
        cout<<endl;
    }

    cout<<"-----";
    cout<<"-----"<<endl;
    cout<<"No. Nama Siswa Nilai Nilai ";
    cout<<"Hasil"<<endl;
    cout<<" Mid Tes FInal ";
    cout<<"Ujian"<<endl;
    cout<<"-----";
    cout<<"-----"<<endl;

    for(i=1;i<=2;i++)
    {
        cout<<setiosflags(ios::left)<<setw(4)<<i;

        cout<<setiosflags(ios::left)<<setw(20)<<nama[i];
        cout<<setprecision(2)<<" "<<nilai1[i];
        cout<<setprecision(2)<<" "<<nilai2[i];
        cout<<setprecision(2)<<" "<<hasil[i]<<endl;
    }

    cout<<"-----";
    cout<<"-----"<<endl;
    getch();
}
```

9.1.2 Array Dua Dimensi

Array dua dimensi merupakan *array* yang terdiri dari *m* buah baris (*row*) dan *n* buah kolom (*column*). Bentuk *array* semacam ini menggunakan 2 (dua) buah kelompok indeks yang masing-masing direpresentasikan sebagai indeks baris dan kolom. Jika ingin memasukkan atau membaca sebuah nilai pada matriks maka, harus diketahui terlebih dahulu indeks baris dan kolomnya.

Array ini dapat digunakan untuk berbagai keperluan. Sebagai gambaran, data kelulusan dari jurusan Teknik Informatika, Manajemen Informatika dan Teknik Komputer pada sekolah tinggi Komputer dari tahun 1992 hingga 1995 dapat dinyatakan dengan array berdimensi dua.

Sebelum membahas cara pendefinisian array berdimensi dua, perhatikan tabel berikut :

Jurusan	1992	1993	1994	1995
1. Teknik Informatika	35	45	80	120
2. Manajemen Informatika	100	110	70	101
3. Teknik Komputer	10	15	20	17

Bentuk seperti tabel diatas dapat dituangkan kedalam array berdimensi dua. Pendefinisiannya :

```
int data_lulus[3][4];
```

pada pendefiniasian diatas :

3 menyatakan jumlah baris (mewakili jurusan)

4 menyatakan jumlah kolom (mewakili tahun kelulusan)

Array berdimensi dua dapat diakses dengan bentuk :

```
nama_array[subscript_baris, subscript_kolom]
```

Representasi *array* dua dimensi:

Diagram illustrating a 2D array structure (Matrix A) with dimensions 3 rows (0-2) and 5 columns (0-4). The array is labeled $A[0,0]$ at the top-left corner. The columns are indexed 0 to 4, and the rows are indexed 0 to 2. The array is divided into three sections: $A[1,2]$ (row 1, column 2), $A[2,4]$ (row 2, column 4), and a section labeled "Ada 3 baris (0-2)" (rows 0-2, column 0).

Gambar di atas merepresentasikan *array* yang terdiri dari 3 baris dan 5 kolom, dan jumlah elemennya = $3 \times 5 = 15$ elemen. Karena terdiri dari baris (*row*) dan kolom (*column*), maka *array* dua dimensi sering juga disebut **matrix**.

Mengakses data *array* dua dimensi:

Seperti *array* satu dimensi, *array* dua dimensi juga diakses melalui indeksnya. Contoh: A[1,2], menunjuk pada posisi nilai *array* pada baris 1, kolom 2.

Untuk menyimpan nilai dalam *array* dua dimensi, dapat dilakukan dengan cara sebagai berikut:

$A[0,0] \leftarrow 2$ /*simpan 2 pada array A baris 0, kolom 0*/

$A[0,1] \leftarrow 4$ /*simpan 3 pada array \mathcal{A} baris 0, kolom 1*/

A[1,2] \leftarrow 8 /*simpan 5 pada array A baris 1, kolom 2*/

$A[2,2] \leftarrow A[0,0] + A[1,2]$ /*tambahkan nilai pada array \mathcal{A} baris 0, kolom 0 dengan nilai pada array \mathcal{A}

baris 1, kolom 2 dan simpan hasilnya
pada array A baris 2, kolom 2 */

Sehingga gambaran array A menjadi seperti berikut ini:

Ada 5 kolom (0-4)

	0	1	2	3	4
0	2	4			

A	1			8		
	2			10		
	⋮					
Ada 3 baris (0-2)				A[1,2]		A[2,4]

Contoh 30	Hasil
<pre> /*-----* /* Contoh 30 : Pemakaian array berdimensi * /* dua * /*-----* #include <iostream.h> #include <conio.h> void main() { int data_lulus[3][4]; // Array berdimensi dua int tahun, jurusan; clrscr(); // Memberikan data ke elemen array data_lulus data_lulus[0][0] = 35; // data TI - 1992 data_lulus[0][1] = 45; // data TI - 1993 data_lulus[0][2] = 90; // data TI - 1994 data_lulus[0][3] = 120; // data TI - 1995 data_lulus[1][0] = 100; // data MI - 1992 data_lulus[1][1] = 110; // data MI - 1993 data_lulus[1][2] = 70; // data MI - 1994 data_lulus[1][3] = 101; // data MI - 1995 data_lulus[2][0] = 10; // data TK - 1992 data_lulus[2][1] = 15; // data TK - 1993 data_lulus[2][2] = 20; // data TK - 1994 data_lulus[2][3] = 17; // data TK - 1995 // Proses untuk memperoleh informasi kelulusan while (1) { cout << "Jurusan (0 = TI, 1 = MI, 2 = TK) : " ; cin >> jurusan; if ((jurusan == 0) (jurusan == 1) (jurusan == 3)) break; // Keluar dari while } while (1) { cout << "Tahun (1992 - 1995) : " ; cin >> tahun; if ((tahun >= 1992) && (tahun <= 1995) { tahun -= 1992; // Konversi ke 0, 1, 2 atau 3 break; // Keluar dari while } } } </pre>	<p>Jurusan (0 = TI, 1 = MI, 2 = TK) : 1 ↵ Tahun (1992 - 1995) : 1992 ↵ Jumlah yang lulus = 100</p>


```

}
}
cout << "Jumlah yang lulus = "
<< data_lulus[jurusan][tahun] << endl;
}

```

Mula-mula program mengisi data ke array data_lulus. Kemudian program meminta data jurusan dan tahun dari keyboard. Kedua data masukkan dipastikan tidak akan berada diluar jangkauan kedua subscript-nya.

Contoh 31

```

/* ----- */
/* Array Dimensi 2 */
/* ----- */
#include<conio.h>
#include<stdio.h>
#include<iostream.h>
#include<iomanip.h>

main()
{
    int i, j;
    int data_jual[4][4];

    clrscr();

    for(i=1;i<=3;i++)
    {
        for(j=1;j<=3;j++)
        {
            cout<<"Data Ke - "<<i<<" "<<j<<endl;
            cout<<"Jumlah Penjualan : ";

            cin>>data_jual[i][j];
        }
    }

    cout<<"Data Penjualan Pertahun"<<endl;
    cout<<"-----"<<endl;
    cout<<"NO 2001 2002 2003"<<endl;
    cout<<"-----"<<endl;

    for(i=1;i<=3;i++)
    {
        cout<<setiosflags(ios::left)<<setw(5)<<i;
        for(j=1;j<=3;j++)
        {
            cout<<setiosflags(ios::right)<<setw(4);
            cout<<data_jual[i][j];
            cout<<" ";

```

```
        }  
        cout<<endl;  
    }  
  
    cout<<"-----"<<endl;  
    getch();  
}
```

Apa output dari program berikut ?



Rangkuman

- ☑ Tipe data *array* digunakan untuk menampung/menyimpan banyak nilai pada satu variabel.
- ☑ Setiap elemen pada tipe data *array* ditandai dengan indeks.
- ☑ Indeks penanda elemen pada *array* menggunakan tipe data yang memiliki keterurutan.
- ☑ Tipe data *array* memiliki dimensi minimal satu hingga n-dimensi.
- ☑ Pada tipe data *array* satu dimensi memiliki satu indeks, kemudian pada *array* dua dimensi memiliki dua indeks, demikian seterusnya dimana jumlah indeks mengikuti banyaknya dimensi *array* yang dibentuk.
- ☑ Tipe data bentukan adalah tipe data yang dirancang/dibentuk (dan diberi nama) dari beberapa elemen bertipe tertentu.
- ☑ Tipe data bentukan dapat disimpan dalam variabel bertipe *array*.
- ☑ Elemen dalam tipe data bentukan dapat menggunakan variabel bertipe *array*.
- ☑ Tipe data bentukan yang di dalamnya terdapat elemen bertipe *array*, dapat disimpan dalam variabel bertipe *array*.