

PRAKTIKUM 4

Larik/Array

Tujuan Praktikum :

1. Memperkenalkan penggunaan larik Dimensi Satu dan Dua
2. Mempraktekkan pemakaian larik dimensi satu dan dua untuk elemen yang diinputkan
3. Mempraktekkan pemakaian larik dimensi satu dan dua untuk elemen yang didefinisikan

Variabel Larik atau lebih dikenal dengan **Array** adalah adalah Tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Suatu **Array** mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indek untuk membedakan variabel yang satu dengan variabel yang lainnya.

Deklarasi Array

Sama seperti variabel lainnya, array juga harus dideklarasikan dulu sebelum digunakan,

```
tipe_data nama_variabel_array []
```

Berdasarkan bentuk umum diatas, bahwa array terdiri dari dua bagian **tipe_data** yang ada di dalam bahasa pemrograman c++ seperti **int**, **float**, **dll**.

nama_variabel_array[] yang dibuat berdasarkan keperluan dan di dalam tanda siku dapat diisi dengan nilai **int** berapa jumlah maksimum elemen array yang akan dibuat.

Inisiasi Array

Jika nilai suatu variabel array dapat di inisialisasi secara langsung pada saat deklarasi, sebagai contoh :

```
int nilai [5] = {1,2,3,4,5}
```

Maka, penyimpanan di dalam array dapat digambarkan sebagai berikut :

nilai	1	2	3	4	5
	0	1	2	3	4

Mengakses Nilai Array

Untuk mengakses nilai yang terdapat dalam array, ditulis dalam bentuk, sebagai berikut :

```
Nama [index];
```

pada contoh diatas, variabel nilai memiliki 5 buah elemen yang masing – masing berisi data. Maka, pengaksesan tiap – tiap elemen data adalah :

	nilai[0]	nilai[1]	nilai[2]	nilai[3]	nilai[4]
nilai					



Percobaan 4.1: Inisiasi Array

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

main() {
    int data[5] = {11, 22, 33, 44, 55};

    for(int i=0;i<5;i++) {
        cout<<"Data "<<i+1<<" = "<<data[i] << endl;
    }
}
```

Percobaan 4.2: Deklarasi Array dengan Input

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

main() {
    int n;
    cout<<"Banyak data : "; cin>>n;
    int data[n];

    for (int i=0; i<n; i++) {
        cout<<"Data "<<i+1<<" : "; cin>>data[i];
    }
    cout<<endl;

    for(int i=0; i<n; i++) {
        cout<<"Data"<<i+1<<" = "<<data[i]<<endl;
    }
}
```



Percobaan 4.3: Operasi Input String Sebagai *array of character*

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

main () {
    char kalimat [100];
    cout << "Siapa nama anda? ";
    cin.getline (kalimat,100);
    cout << "Halo " << kalimat << ".\n";
    cout << "Apa makanan favorit anda? ";
    cin.getline (kalimat,100);
    cout << "Saya suka " << kalimat << " juga.\n";
}
```

Percobaan 4.4: Inisiasi Array 2 dimensi

Array dimensi dua tersusun dalam bentuk baris dan kolom, dimana indeks pertama menunjukkan **baris** dan indeks kedua menunjukan **kolom**. **Array dimensi dua** dapat digunakan seperti pendapatan penjualan, pendataan nilai dan lain sebagainya.

Bentuk umum penulisan array dimensi dua :

tipe_data nama_variabel [index-1] [index-2]

Keterangan :

Tipe data : untuk menyatakan tipe data yang digunakan

Index-1 : untuk menyatakan jumlah baris

Index-2 : untuk menyatakan jumlah kolom

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

main() {
    int i, j;
    int data[2][3] = {{2, 4, 1}, {5, 3, 7}}; //inisialisasi
    for(i=0;i<2;i++) {
        for(j=0;j<3;j++) {
            cout<<data[i][j];
            cout<<" ";
        }
        cout<<endl;
    }
}
```



Percobaan 4.5: Deklarasi Array 2 Dimensi dengan Input

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

main() {
    int nBaris, nKolom;
    cout << "Banyak baris : "; cin >> nBaris; cout<<endl;
    cout << "Banyak kolom : "; cin >> nKolom; cout<<endl;
    int matriks[nBaris][nKolom];

    for (int i=0; i < nBaris; i++) {
        for (int j=0; j < nKolom; j++) {
            cout<<"Data ke-["<<i+1<<","<<j+1 <<"] : ";
            cin >> matriks[i][j];
        }
    }
    cout << "\nPencetakan Matriks :"<<endl;
    for (int i=0; i < nBaris; i++) {
        for (int j=0; j < nKolom; j++) {
            cout << matriks[i][j] << " ";
        }
        cout << endl;
    }
}
```

Sumber/Referensi:

- (1) Munir, R. (2012). Algoritma dan Pemrograman. Jilid 1 Bandung: Penerbit Informatika.
- (2) Wirth, N. (1990). Algorithms + Data Structures = Programs. India: Prentice-Hall Of India Pvt. Limited.
- (3) Hubbard, J.R. (1996). Programming With C++, Schaum's outlines Series. USA: McGraw Hill
- (4) Suarga, M.Math., (2012). Algoritma Dan Pemrograman (Edisi 2). Yogyakarta: Penerbit Andi.
- (5) Shalahuddin, M dan AS. Rosa. (2007) Belajar Bahasa Pemrograman dengan C++ dan Java: Penerbit Informatika.
- (6) Joel Adams-Sanford Leestma-Larry Nyhoof, (1995). C++ An Introduction To Computing: Prentice-Hall, Inc
- (7) James P. Cohoon-Jack W.Davidson, McGraw-Hill, (1997). C++ Programming Design.
- (8) Budi Raharjo, (2004). Mengungkap Rahasia Pemrograman Dalam C++: Penerbit Informatika.
- (9) Abdul Kadir, (2003). Pemrograman C++, Andi, Yogyakarta
- (10) Yosua Onesimus Suheru, (2004). Trik Memecahkan Masalah Dengan Tiga bahasa Pemrograman – C++, Pascal dan Visual Basic, Gava Media, Yogyakarta