

Array Dua Dimensi

Tujuan Praktikum

1. Memahami struktur Array dua dimensi,
2. Mampu menerapkan Array dua dimensi untuk menyelesaikan berbagai kasus.

Dasar Teori

Array dua dimensi atau array multidimensi pada dasarnya sama dengan array satu dimensi, hanya saja, pada array multidimensi, indeksnya bisa lebih dari 1.

Merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang akan diakses melalui banyak indeks atau subskrip. Array seperti ini biasa digunakan untuk matriks, array 2 dimensi juga termasuk kedalam array multidimensi.

Array dua dimensi biasanya digunakan untuk merepresentasikan nilai dari sebuah tabel. mengidentifikasi tiap elemen array harus dispesifikasikan nilai baris dan kolom. Array multidimensi sebenarnya adalah array dari array. Deklarasi array multidimensi dilakukan dengan adanya lebih dari satu pasangan kurung siku di dalam deklarasi array. Syntax untuk mendeklarasikan array 2 dimensi adalah:

```
type namaVariabel[ukuranBaris][ukuranKolom];
```

`type` adalah tipe data dari array dan `namaVariabel` adalah nama dari array, sedangkan tanda `[ukuranBaris][ukuranKolom]` memberitahu C++ untuk membuat variabel array 2 dimensi.

Contoh:

```
int a[3][4];
```

Pada contoh tersebut artinya membuat sebuah array 2 dimensi dengan nama **a** yang mempunyai tipe data integer. Sebagai ilustrasi, dapat dilihat pada gambar berikut.

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Column subscript

Row subscript

Array name

Praktik (Guided)

1. Pengisian nilai pada array multi-dimensi

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

void printArray(const int[][3]);
const int rows = 2;
const int columns = 3;

int main()
{
    int array1[rows][columns] = {{1, 2, 3}, {4, 5, 6}};
    int array2[rows][columns] = {1, 2, 3, 4, 5};
    int array3[rows][columns] = {{1, 2}, {4}};

    printArray(array1);

    printArray(array2);

    printArray(array3);

    return 0;
}
```

```

void printArray(const int a[rows][columns])
{
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }

    cout << endl;
}

```

2. Penjumlahan Matriks

```

#include <iostream>
using namespace std;

void printArray(const int[][3]);
const int rows = 3;
const int columns = 3;

int main()
{
    int array1[rows][columns] = {{1, 2, 3},
                                   {4, 5, 6},
                                   {7, 8, 9}};

    int array2[rows][columns] = {{1, 2, 3},
                                   {4, 5, 6},
                                   {7, 8, 9}};

    int array3[rows][columns];

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            array3[i][j] = array1[i][j] + array2[i][j];
        }
    }

    printArray(array3);

    return 0;
}

void printArray(const int a[rows][columns])
{

```

```

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }

    cout << endl;
}

```

Latihan (Unguided)

1. Buatlah algoritma dan program dengan struktur data array untuk menambah, mengurangi dan mengalikan 2 buah matriks!

Syarat perkalian matriks:

Jika matriks $A_{m \times n}$ dan matriks $B_{n \times o}$ dikalikan, maka:

- Banyaknya kolom matriks A harus sama dengan banyaknya baris matriks B, sehingga $n = p$
- Matriks hasil perkalian antara A dan B adalah matriks dengan ordo $m \times q$
- Perkalian dilakukan dengan menjumlahkan hasil kali setiap elemen baris matriks A dengan setiap elemen kolom matriks B yang sesuai

Output:

```

2 3 4
1 1 1
1 1 2
1 1 1 1
1 1 1 1
1 1 1 2
3 3 3 4
4 4 4 6

```