

## **BAB IV**

### **ARRAY**

#### **A. TUJUAN**

1. Mahasiswa mampu mengetahui dan memahami array sebagai struktur data linear
2. Mahasiswa mampu memahami array 1 dimensi
3. Mahasiswa mampu memahami array multidimensi
4. Mahasiswa mampu mengimplementasikan array berdasarkan kasus yang diberikan menggunakan bahasa pemrograman C++.

#### **B. ALAT DAN BAHAN**

1. Laptop
2. Mouse
3. IDE C++ (Visual Studio Code)

#### **C. KESEHATAN DAN KESELAMATAN KERJA**

1. Hati-hatilah dalam memakai perangkat elektronik
2. Pastikan kabel listrik terpasang dan dalam kondisi baik
3. Lakukan praktikum dalam posisi duduk yang benar
4. Jauhkan kabel listrik dari sentuhan anda
5. Gunakan alas kaki celana Panjang dan kemeja
6. Gunakan kacamata anti radiasi layar

#### **D. TEORI**

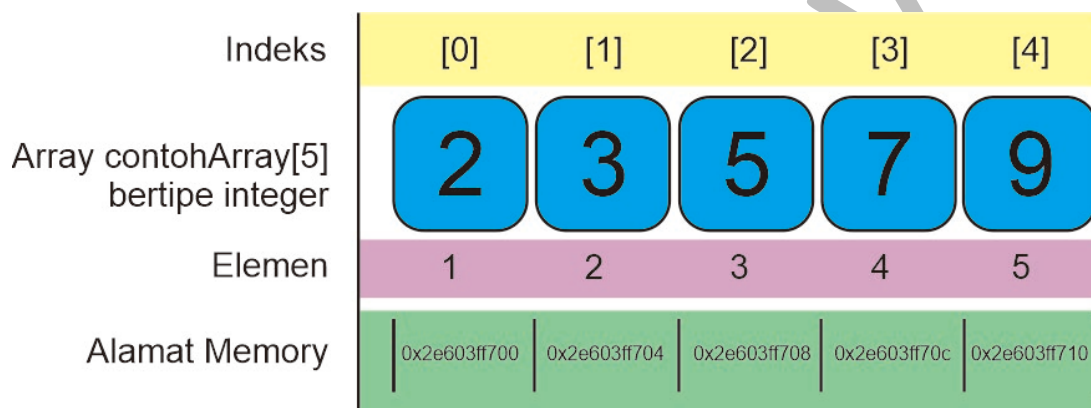
##### **1. Array**

Array adalah serangkaian elemen dengan tipe data yang sama yang ditempatkan di lokasi memori yang berdekatan yang dapat direferensikan secara individual dengan menambahkan indeks ke pengidentifikasi unik. Hal tersebut berarti bahwa, misalnya, lima nilai bertipe integer dapat dideklarasikan sebagai array tanpa perlu mendeklarasikan lima variable yang berbeda (masing-masing dengan nama variable atau identifiernya sendiri). Sebagai gantinya, dengan menggunakan array, kelima nilai integer tersebut disimpan pada lokasi memory yang saling berdekatan, dan kelimanya dapat diakses dengan menggunakan pengenal yang sama menggunakan indeks tertentu. Sebagai contoh dideklarasikan sebuah array sebagai berikut yang gambar ilustrasinya dapat dilihat pada Gambar 4.1.

Contoh:

```
int contohArray[5]={2, 3, 5, 7, 9};
```

Pada contoh diatas dapat diketahui bahwa suatu variable bertipe array dideklarasikan dengan nama “contohArray”. Array “contohArray” dapat menampung 5 elemen atau data bertipe integer. Ilustrasi posisi indeks, elemen, dan alamat memory dapat dilihat pada Gambar 4.1.



Gambar 4.1. Ilustrasi Array 1 dimensi bertipe integer dengan 5 elemen.

Perlu diingat bahwa tipe integer berukuran 4 byte. Sehingga setiap elemen array memiliki ruang 4 byte pada memory. Setiap elemen disimpan pada alamat memory yang berdekatan.

## 2. Array 1 Dimensi

Array 1 dimensi adalah array yang hanya memiliki satu indeks. Artinya bahwa elemen pada array tersusun dalam satu baris sehingga untuk mengakses elemennya hanya diperlukan satu nilai indeks.

### a. Mendeklarasikan Array

Terdapat beberapa versi sintaks untuk mendeklarasikan sebuah array 1 dimensi. Berikut sintaks tersebut:

Sintak I (Mendeklarasikan array)

```
Tipe_data nama_array[jml_elemen];
```

Sintak II (Mendeklarasikan array dengan inisialisasi jumlah dan nilai elemen)

```
Tipe_data nama_array[jml_elemen]={nilai1,nilai2,...,nilaiN};
```

Sintak III (Mendeklarasikan array dengan inisialisasi nilai tanpa menentukan jumlah elemen)

```
Tipe_data nama_array[ ]= {nilai1,nilai2,...,nilaiN};
```

Keterangan untuk sintaks I - III:

- **Tipe\_data**: merupakan tipe data dari variable array yang dideklarasikan. Array dapat menampung data sesuai tipe data yang diberikan.
- **nama\_array**: merupakan identifier atau nama variable array yang diinginkan.
- **[jml\_elemen]**: jumlah elemen yang diinginkan yang dapat ditampung oleh array diketikkan didalam kurung siku.
- **{nilai1,nilai2,...,nilaiN}**: nilai elemen array diketikkan didalam area kurung siku dimana setiap elemen dipisahkan dengan tanda koma sesuai urutannya.

#### b. Memasukkan nilai ke dalam array indeks tertentu

Adapun sintaks untuk memasukkan nilai ke dalam indeks tertentu adalah sebagai berikut:

Sintak IV (Memasukkan nilai ke indeks tertentu)

```
nama_array[indeks]=nilai;
```

Keterangan untuk sintaks IV:

- **nama\_array**: merupakan identifier atau nama variable array yang akan diakses.
- **[indeks]**: indeks berisi angka indeks pada array yang akan dimasukkan nilai ke dalamnya.
- **nilai**: merupakan nilai yang akan dimasukkan ke dalam array indeks tertentu.

### c. Menampilkan atau mengambil nilai dari array indeks tertentu

Adapun sintaks untuk mengambil/menampilkan nilai dari array indeks tertentu adalah sebagai berikut:

Sintak V (Menampilkan nilai array indeks tertentu ke layar)

```
cout<<nama_array[indeks]<<endl;
```

Sintak VI (Menugaskan nilai array indeks tertentu ke variable lain)

```
Tipe_data nama_variabel = nama_array[indeks];
```

Keterangan untuk sintaks V-VI:

- **nama\_array**: merupakan identifier atau nama variable array yang akan diakses.
- **[indeks]**: indeks berisi angka indeks pada array yang nilainya akan diambil atau ditampilkan.

## 3. Array Multidimensi

Sesuai dengan namanya, array multidimensi adalah array yang tidak hanya memiliki satu dimensi saja, misalnya baris (istilah yang biasa digunakan pada array 1 dimensi). Namun pada array multi dimensi, bisa terdapat dimensi lain. Bisa berdimensi dua atau berdimensi 3 dan seterusnya. Array multidimensi juga dapat dikatakan sebagai “array dari sebuah array”.

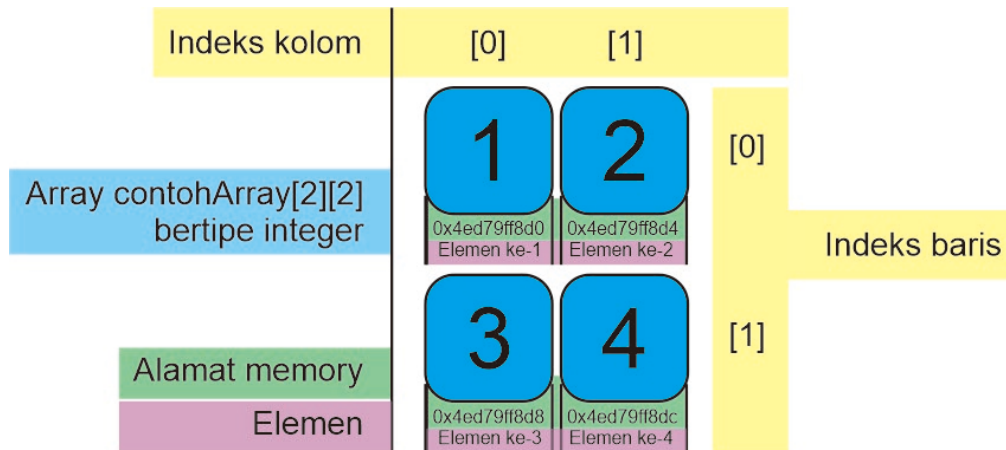
### a. Array 2 Dimensi

Sesuai namanya, array 2 dimensi adalah array yang memiliki 2 dimensi, sebut saja dimensi baris dan dimensi kolom. Sebagai contoh dideklarasikan sebuah array dua dimensi dengan 4 elemen.

Contoh:

```
int contohArray[2][2]={{1, 2},{3, 4}};
```

Pada contoh diatas dapat diketahui bahwa suatu variable bertipe array dideklarasikan dengan nama “contohArray”. Array “contohArray” berjenis array dua dimensi dan dapat menampung 4 elemen atau data bertipe integer. Hal tersebut ditandai dengan adanya 2 kurung siku yang didalamnya terdapat angka masing-masing 2, artinya terdapat 2 baris dan 2 kolom. Ilustrasi posisi indeks, elemen, dan alamat memory dapat dilihat pada Gambar 4.2.



Gambar 4.2. Array dua dimensi

Perlu diingat bahwa tipe integer berukuran 4 byte. Sehingga setiap elemen array memiliki ruang 4 byte pada memory. Setiap elemen disimpan pada alamat memory yang berdekatan. Nilai 1 menempati “contohArray[0][0]”, nilai 4 menempati indeks array “contohArray[1][1]”.

Adapun sintaks untuk mendeklarasikan array dua dimensi adalah sebagai berikut:

Sintak I (Mendeklarasikan array 2 dimensi)

```
Tipe_data nama_array[jml_baris][jml_kolom];
```

Sintak II (Mendeklarasikan array dengan inisialisasi jumlah dan nilai elemen)

```
Tipe_data nama_array[jml_baris][jml_kolom]={ {nilai_baris  
1}, {nilai_barisN}};
```

Keterangan untuk sintakas I - II:

- **Tipe\_data**: merupakan tipe data dari variable array yang dideklarasikan. Array dapat menampung data sesuai tipe data yang diberikan.
- **nama\_array**: merupakan identifier atau nama variable array yang diinginkan.
- **[jml\_baris]**: jumlah elemen baris yang diinginkan yang dapat ditampung oleh array diketikkan didalam kurung siku.
- **[jml\_kolom]**: jumlah elemen kolom yang diinginkan yang dapat ditampung oleh array diketikkan didalam kurung siku.
- **{nilai\_barisN}**: nilai elemen array pada baris ke N. Didalam nilai baris, nilai-nilai kolom dipisahkan dengan koma.

Adapun sintaks untuk memasukkan nilai ke dalam indeks tertentu adalah sebagai berikut:

Sintak III (Memasukkan nilai ke indeks tertentu)

```
nama_array[indeks_baris][indeks_kolom]=nilai;
```

Keterangan untuk sintakas III:

- **nama\_array**: merupakan identifier atau nama variable array yang akan diakses.
- **[indeks\_baris]**: indeks berisi angka indeks baris pada array yang akan dimasukkan nilai ke dalamnya.
- **[indeks\_kolom]**: indeks berisi angka indeks kolom pada array yang akan dimasukkan nilai ke dalamnya.
- **nilai**: merupakan nilai yang akan dimasukkan ke dalam array indeks baris dan kolom tertentu.

Adapun sintaks untuk mengambil/menampilkan nilai dari array indeks tertentu adalah sebagai berikut:

Sintak IV (Menampilkan nilai array indeks tertentu ke layar)

```
cout<<nama_array[indeks_baris][indeks_kolom]<<endl;
```

Sintak V (Menugaskan nilai array indeks tertentu ke variable lain)

```
Tipe nama_variabel =
    nama_array[indeks_baris][indeks_kolom];
```

Keterangan untuk sintakas IV-V:

- **nama\_array**: merupakan identifier atau nama variable array yang akan diakses.
- **[indeks\_baris]**: indeks berisi angka indeks baris pada array yang akan diambil atau ditampilkan nilainya.
- **[indeks\_kolom]**: indeks berisi angka indeks kolom pada array yang akan diambil atau ditampilkan nilainya.

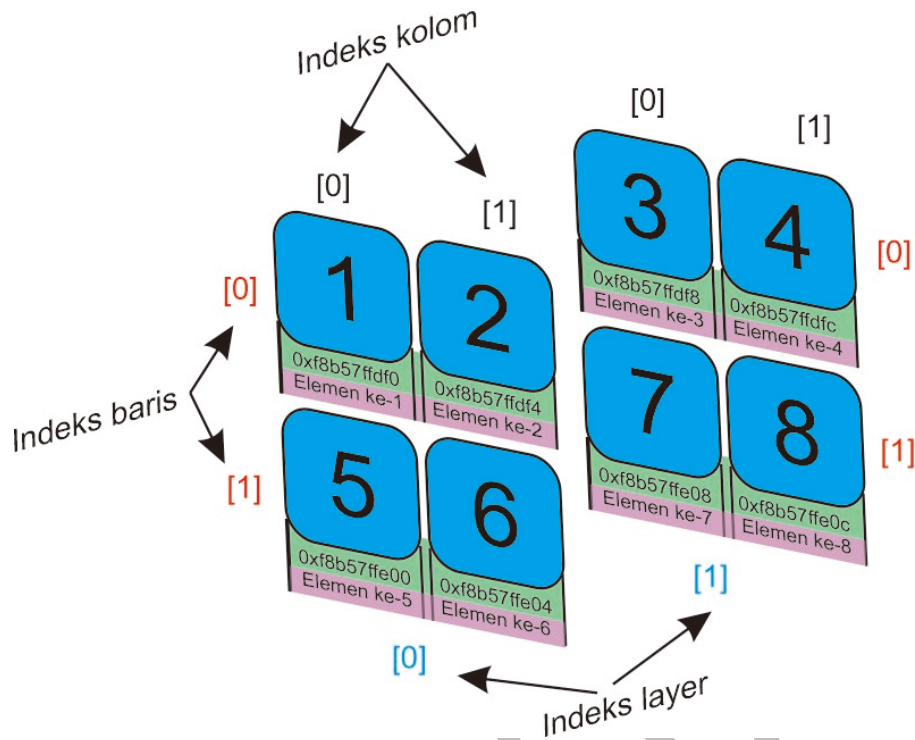
### b. Array 3 Dimensi

Sesuai Namanya array 3 dimensi adalah array yang memiliki 3 dimensi. Tidak lagi hanya sekedar memiliki indeks baris dan kolom tetapi array 3 dimensi memiliki 1 dimensi tambahan. Dalam hal ini anda boleh memberikan nama apapun untuk mewakili tiap dimensi pada array.

Contoh:

```
int contohArray[2][2][2] = {
    {{1,2},{3,4}},
    {{5,6},{7,8}}
};
```

Jika kode program di atas diilustrasikan, maka kurang lebih akan tersusun seperti Nampak pada Gambar 4.3. Sekali lagi jika diperhatikan alamat memory elemen array selalu berdekatan. Pada array yang dicontohkan setiap elemen menempati 4 byte memory karena array bertipe integer.



Gambar 4.2. Array tiga dimensi

## E. PRAKTIKUM

### Praktikum 4.1. Array 1 dimensi

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

int main() {
    int angka[5] = {6, 5, 6, 12, 90};

    cout << "Nilai di dalam array adalah: ";

    //menggunakan range-based for loop
    for (const int &n : angka) {
        cout << n << " ";
    }

    return 0;
}
```

### Praktikum 4.2. Menginput dan menampilkan nilai array 1 dimensi disertai class

1. Ketikkan kode berikut pada Main.cpp



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

int main() {
    int banyakElemen;
    cout<<"===Contoh Program Array==="<<endl;
    cout<<"Berapa jumlah elemen array yang anda inginkan?"<<endl;
    cin>>banyakElemen;
    Array array1;
    int *array = array1.buatArrayku(banyakElemen);
    array1.tampilArrayku(array,banyakElemen);
    return 0;
}
```

2. Buat file baru bernama “Array.cpp” dan “Array.h” pada folder src
3. Pada file Array.h, ketikkan kode berikut.

```
#include <iostream>

class Array{
public:
    int *buatArrayku(int banyakElemen);
    void tampilArrayku(int arraytampil[], int banyakElemen);
};
```

4. Pada file Array.cpp, ketikkan kode berikut.

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

int * Array::buatArrayku(int banyakElemen){
    int *arrayku = new int[banyakElemen];
    for (int i=0; i<banyakElemen; i++){
        cout<<"Input elemen ke-"<<i<<endl;
        cin>>arrayku[i];
    }
    return arrayku;
}

void Array::tampilArrayku(int arraytampil[], int banyakElemen){
    for (int i=0; i<banyakElemen; i++){
        cout<<"Elemen ke-"<<i+1<<" ";
        cout<<arraytampil[i]<<endl;
    }
}
```

}

5. Jalankan kode program (Ctrl+Shift+B)

#### Praktikum 4.3. Array 2 dimensi dengan class dan pointer array

1. Ketikkan kode berikut pada Main.cpp

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

class Arraysaya{
public:
    void tampilArray(int *arr, int baris, int kolom){
        for (int i=0; i<baris; i++){
            for (int j=0; j<kolom; j++){
                cout<<*((arr+i*kolom)+j)<<" ";
            }
            cout<<"\n";
        }
    }
};

int main() {
    int baris = 2, kolom = 2;
    int arr[baris][kolom] = {{1, 2}, {3, 4}};
    Arraysaya arrayku;
    arrayku.tampilArray((int *)arr, baris, kolom);
}
```

2. Jalankan kode program (Ctrl+Shift+B)

#### Praktikum 4.4. Operasi aritmatika pada array 3 dimensi

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

int main() {
    int m = 2, n = 2, o = 2;
    int arr1[m][m][o] = {
        {{1, 2}, {3, 4}},
        {{5, 6}, {7, 8}}
    }
```

```
};

int arr2[m][m][o] = {
    {{10, 20}, {30, 40}},
    {{50, 60}, {70, 80}}
};

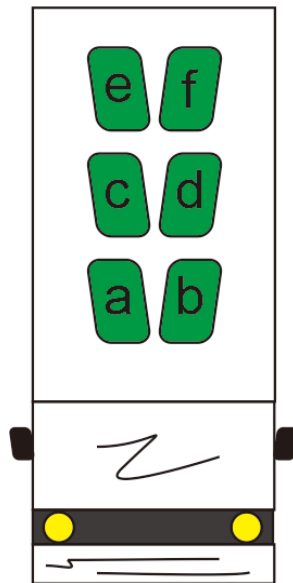
int arr3[m][m][o] = {
    {{10, 20}, {30, 40}},
    {{50, 60}, {70, 80}}
};

for(int i = 0; i<m; i++){
    for(int j = 0; j<n; j++){
        for(int k = 0; k<o; k++){
            arr3[i][j][k]=arr1[i][j][k]+arr2[i][j][k];
        }
    }
}

for(int i = 0; i<m; i++){
    for(int j = 0; j<n; j++){
        for(int k = 0; k<o; k++){
            cout<<"arr3 indeks "<<i<<" "<< j<<" "<<k<<endl;
            cout<<"berisi nilai: "<<arr3[i][j][k]<<endl;
        }
    }
}
}
```

## F. LATIHAN

1. Perhatikan gambar skema minibus berikut



Sebuah perusahaan travel memiliki sebuah minibus yang dapat menampung 6 penumpang. Kursi di dalam bus telah diberi kode untuk memudahkan calon penumpang memesan tiket kursi. Kodenya yaitu a, b, c, d, e, dan f. Pemilik perusahaan bus ingin dibuatkan sebuah system untuk mencatat kursi yang telah dipesan. Bantulah si pemilik bus! Buatlah sebuah program sederhana pemesanan tiket kursi bus.

Adapun syarat programnya sebagai berikut:

- Kursi dipesan berdasarkan kode kursinya
- Nama pemesan tercatat pada kursi yang telah dipesan.
- Kursi yang telah dipesan tidak bisa dipesan kembali.
- Jika kursi yang telah dipesan dipesan kembali, maka akan muncul pemberitahuan “kursi tidak tersedia”
- Program dapat menerima pesanan selama masih ada kursi yang kosong.
- List kursi yang telah dipesan berikut nama pemilik kursi ditampilkan setiap pemesanan selesai dilakukan
- Data pemesanan kursi berupa nama calon penumpang dan kursi yang dipilih diinput melalui keyboard.
- Kode program yang dibangun wajib memanfaatkan array

**Kunci:** manfaatkan percabangan, perulangan, dan array.

CATATAN

DO NOT COPY