

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN
ALGORITMA**

**MODUL II
ARRAY**



Disusun Oleh :

Raka Andriy Shevchenko
2311102054

Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

A. Dasar Teori

Array adalah tipe data yang terdiri dari elemen-elemen yang memiliki tipe data yang sama dan diberi nama tertentu. Setiap elemen dalam *array* diakses menggunakan indeks, yang memudahkan pengguna untuk mendapatkan nilai spesifik dengan cepat. C++ menyediakan perulangan (*loop*) yang memudahkan pengguna untuk mengakses semua elemen dalam *array* dengan mudah.^[1] Manipulasi data mungkin dilakukan, seperti mengubah nilai elemen, mengurutkan data, mencari elemen tertentu, dan melakukan berbagai operasi lain dengan mudah. Berikut adalah beberapa jenis *array*:

a. *Array* satu dimensi

Cara mendeklarasikan *array* 1 dimensi dalam C++ adalah dengan menggunakan tipe data yang diinginkan, nama variabel *array*, dan kurung siku []. Terdapat beberapa cara mendeklarasikan *array* 1 dimensi dalam C++, seperti dengan ukuran langsung ditentukan, inisialisasi awal, dan ukuran ditentukan pada *runtime*.^[1] Setiap elemen di *array* 1 dimensi diidentifikasi oleh indeksnya yang dimulai dari 0.

Untuk mengakses elemen *array* 1 dimensi, pengguna dapat menggunakan indeks dalam kurung siku []. Misalnya, untuk mengakses elemen ke-2 dalam *array* 1 dimensi, pengguna dapat menulis `nama_array`. Indeks *array* 1 dimensi selalu dimulai dari 0.

Untuk mengubah nilai elemen *array* 1 dimensi, pengguna dapat menggunakan indeks dan sisipkan nilai baru pada elemen tersebut. Misalnya, untuk mengubah nilai elemen ke-2 dalam *array* 1 dimensi, pengguna dapat menulis `nama_array = new_value`.^[1]

b. *Array* dimensi dua

Cara mendeklarasikan *array* dimensi 2 dalam C++ adalah dengan menggunakan tipe data yang diinginkan, nama variabel *array*, dan kurung siku []. Terdapat beberapa cara mendeklarasikan *array* dimensi 2 dalam C++, seperti dengan ukuran langsung ditentukan, inisialisasi awal, dan ukuran ditentukan pada *runtime*.^[1] Setiap elemen di *array* dimensi 2 diidentifikasi

oleh indeksnya yang dimulai dari 0.

Untuk mengakses elemen *array* dimensi 2, pengguna dapat menggunakan indeks dalam kurung siku []. Misalnya, untuk mengakses elemen ke-2 dalam *array* dimensi 2, pengguna dapat menulis `nama_array`. Indeks *array* dimensi 2 selalu dimulai dari 0.

Untuk mengubah nilai elemen *array* dimensi 2, pengguna dapat menggunakan indeks dan sisipkan nilai baru pada elemen tersebut. Misalnya, untuk mengubah nilai elemen ke-2 dalam *array* dimensi 2, pengguna dapat menulis `nama_array = new_value.`^[1]

c. *Array* multidimensi

Cara mendeklarasikan *array* multidimensi dalam C++ adalah dengan menggunakan tipe data yang diinginkan, nama variabel *array*, dan kurung siku []. Terdapat beberapa cara mendeklarasikan *array* multidimensi dalam C++, seperti dengan ukuran langsung ditentukan, inisialisasi awal, dan ukuran ditentukan pada *runtime*.^[1] Setiap elemen di *array* multidimensi diidentifikasi oleh indeksnya yang dimulai dari 0.

Untuk mengakses elemen *array* multidimensi, pengguna dapat menggunakan indeks dalam kurung siku []. Misalnya, untuk mengakses elemen ke-2 dalam *array* multidimensi, pengguna dapat menulis `nama_array`. Indeks *array* multidimensi selalu dimulai dari 0.

Untuk mengubah nilai elemen *array* multidimensi, pengguna dapat menggunakan indeks dan sisipkan nilai baru pada elemen tersebut. Misalnya, untuk mengubah nilai elemen ke-2 dalam *array* multidimensi, pengguna dapat menulis `nama_array = new_value.`^[1]

B. Guided

a. Guided 1

Source Code:

```
#include <iostream>
using namespace std;
// PROGRAM INPUT ARRAY 3 DIMENSI
int main()
{
    // Deklarasi array
    int arr[2][3][3];

    // Input elemen
    for (int x = 0; x < 2; x++)
    {
        for (int y = 0; y < 3; y++)
        {
            for (int z = 0; z < 3; z++)
            {
                cout << "Input Array[" << x << "][" << y <<
                "]" << z << "] = ";
                cin >> arr[x][y][z];
            }
        }
        cout << endl;
    }

    // Output Array
    for (int x = 0; x < 2; x++)
    {
        for (int y = 0; y < 3; y++)
        {
            for (int z = 0; z < 3; z++)
            {
                cout << "Data Array[" << x << "][" << y <<
                "]" << z << "] = " << arr[x][y][z] << endl;
            }
        }
    }
    cout << endl;

    // Tampilan array
```

```
for (int x = 0; x < 2; x++)
{
    for (int y = 0; y < 3; y++)
    {
        for (int z = 0; z < 3; z++)
        {
            cout << arr[x][y][z] << ends;
        }
        cout << endl;
    }
    cout << endl;
}
```

Output:

```
Input Array[0][0][0] = 1
Input Array[0][0][1] = 2
Input Array[0][0][2] = 3
Input Array[0][1][0] = 1
Input Array[0][1][1] = 2
Input Array[0][1][2] = 3
Input Array[0][2][0] = 1
Input Array[0][2][1] = 2
Input Array[0][2][2] = 3
```

```
Input Array[1][0][0] = 1
Input Array[1][0][1] = 2
Input Array[1][0][2] = 3
Input Array[1][1][0] = 1
Input Array[1][1][1] = 2
Input Array[1][1][2] = 3
Input Array[1][2][0] = 1
Input Array[1][2][1] = 2
Input Array[1][2][2] = 3
```

```
Data Array[0][0][0] = 1
Data Array[0][0][1] = 2
Data Array[0][0][2] = 3
Data Array[0][1][0] = 1
Data Array[0][1][1] = 2
Data Array[0][1][2] = 3
Data Array[0][2][0] = 1
Data Array[0][2][1] = 2
Data Array[0][2][2] = 3
Data Array[1][0][0] = 1
Data Array[1][0][1] = 2
Data Array[1][0][2] = 3
Data Array[1][1][0] = 1
Data Array[1][1][1] = 2
Data Array[1][1][2] = 3
Data Array[1][2][0] = 1
Data Array[1][2][1] = 2
Data Array[1][2][2] = 3
```

```
123
123
123
```

```
123
123
123
```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko

NIM: 2311102054

Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk pembelajaran *array* multidimensi, disini *array* berukuran 2x3x3. Pertama program melakukan deklarasi *array* dan ukurannya, setelah itu program melakukan *input* elemen (diinputkan oleh *user*). Selanjutnya program mengoutputkan data *array* per indeksnya, lalu menampilkan isi *array*.

b. Guided 2

Source Code:

```
#include <iostream>
using namespace std;
int main()
{
    //Deklarasi variabel
    int maks, a, i = 1, lokasi;
    //Memasukkan panjang array
    cout << "Masukkan panjang array: ";
    cin >> a;

    //Input array
    int array[a];
    cout << "Masukkan " << a << " angka\n";

    for (i = 0; i < a; i++)
    {
        cout << "Array ke-" << (i) << ": ";
        cin >> array[i];
    }

    //Deklarasi variabel nilai maksimum
    maks = array[0];

    //Mengecek berapa dan dimana nilai maksimum
    for (i = 0; i < a; i++)
    {
        if (array[i] > maks)
        {
            maks = array[i];
            lokasi = i;
        }
    }
}
```

```

    }
    //Mengoutputkan nilai maksimum dan posisinya
    cout << "Nilai maksimum adalah " << maks << " berada di
Array ke " << lokasi << endl;
}

```

Output:

```

Masukkan panjang array: 10
Masukkan 10 angka
Array ke-0: 1
Array ke-1: 2
Array ke-2: 3
Array ke-3: 4
Array ke-4: 5
Array ke-5: 6
Array ke-6: 7
Array ke-7: 8
Array ke-8: 9
Array ke-9: 10
Nilai maksimum adalah 10 berada di Array ke 9

```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko

NIM: 2311102054

Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk mencari nilai maksimum dalam *array* 1 dimensi, serta menemukan posisi dari nilai maksimum tersebut. Pertama kita deklarasikan dahulu variabel yang akan digunakan, setelah itu inputkan panjang *array*. Selanjutnya adalah membuat *user* memasukkan isi *array* menggunakan *looping*, setelah *array* jadi maka dilakukan pengecekan untuk nilai maksimum, dan posisi dari nilai maksimum tersebut. Setelah selesai maka outputkan hasilnya.

C. Unguided

a. Unguided 1

Source Code:

```
#include <iostream>

using namespace std;

int main(int argc, char const *argv[])
{
    //Deklarasi variabel yang digunakan
    int n, i;
    //User memasukkan panjang array
    cout << "Masukkan panjang array: ";
    cin >> n;

    //Membuat array dinamis
    int* arr = new int [n];
    //User memasukkan isi array menggunakan loop
    cout << "Masukkan isi array: \n";
    for (i = 0; i < n; i++) {
        cout << "Array ke-" << i+1 << ": ";
        cin >> arr[i];
    }
    //Membuat variabel untuk menampung angka ganjil dan genap
    int ganjil[n], genap[n];
    int count_ganjil = 0, count_genap = 0;

    //Mengecek angka ganjil dan genap
    for (i = 0; i < n; i++) {
        //Melakukan pengecekan, jika modulus 2 = 0, maka
        //genap. Jika sebaliknya maka ganjil
        if (arr[i] % 2 == 0) {
            genap[count_genap] = arr[i];
            count_genap++;
        } else {
            ganjil[count_ganjil] = arr[i];
            count_ganjil++;
        }
    }

    //Mengoutputkan hasil
```

```
    cout << "Data Array: ";
    for (i = 0; i < n; i++) {
        cout << arr[i];
        //Agar rapih, menambahkan koma (,) di setiap iterasi.
        //Kecuali untuk iterasi terakhir
        if (i != n - 1) {
            cout << ", ";
        }
    }
    cout << '\n';

    cout << "Angka Genap: ";
    for (i = 0; i < count_genap; i++) {
        cout << genap[i];
        if (i != count_genap - 1) {
            cout << ", ";
        }
    }
    cout << '\n';

    cout << "Angka Ganjil: ";
    for (i = 0; i < count_ganjil; i++) {
        cout << ganjil[i];
        if (i != count_ganjil - 1) {
            cout << ", ";
        }
    }
    cout << '\n';

    //Menghapus array
    delete[] arr;

    return 0;
}
```

Output:

```
Masukkan panjang array: 10
Masukkan isi array:
Array ke-1: 1
Array ke-2: 2
Array ke-3: 3
Array ke-4: 4
Array ke-5: 5
Array ke-6: 6
Array ke-7: 7
Array ke-8: 8
Array ke-9: 9
Array ke-10: 10
Data Array: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Angka Genap: 2, 4, 6, 8, 10
Angka Ganjil: 1, 3, 5, 7, 9
```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk mencari nilai genap dan ganjil pada *array* 1 dimensi. Karena penjelasan detailnya sudah ada di *source code*, maka saya akan menjelaskan alur program. Pertama deklarasikan variabel yang akan digunakan, setelah itu *user* memasukkan panjang *array*. Karena saya menggunakan *array* dinamis maka setelah itu saya membuat *pointer* untuk *size array* nya. Setelah itu buat *looping* untuk isi *array* (*user input*). Buat variabel penampung angka ganjil dan genap, lalu lakukan pengecekan pada *array* menggunakan *looping*. Terakhir, outputkan semua hasil lalu hapus isi *array*.

b. Unguided 2

Source Code:

```
#include <iostream>

using namespace std;

int main() {
    // Deklarasi variabel untuk ukuran array
    int x, y, z;

    // Meminta user untuk memasukkan ukuran array
    cout << "Masukkan ukuran array x: ";
```

```

    cin >> x;
    cout << "Masukkan ukuran array y: ";
    cin >> y;
    cout << "Masukkan ukuran array z: ";
    cin >> z;

    // Membuat array tiga dimensi dengan ukuran yang
    // diinputkan user
    int arr[x][y][z];

    // Meminta user untuk memasukkan nilai untuk setiap
    // elemen array
    cout << "Masukkan nilai untuk setiap elemen array:" <<
endl;
    for (int i = 0; i < x; ++i) {
        for (int j = 0; j < y; ++j) {
            for (int k = 0; k < z; ++k) {
                cout << "arr[" << i << "][" << j << "][" <<
k << "]: ";
                cin >> arr[i][j][k];
            }
        }
    }

    // Mencetak isi array tiga dimensi
    cout << "Isi array tiga dimensi:" << endl;
    for (int i = 0; i < x; ++i) {
        for (int j = 0; j < y; ++j) {
            for (int k = 0; k < z; ++k) {
                cout << arr[i][j][k] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }

    return 0;
}

```

Output:

```
Masukkan ukuran array x: 2
Masukkan ukuran array y: 3
Masukkan ukuran array z: 3
Masukkan nilai untuk setiap elemen array:
arr[0][0][0]: 1
arr[0][0][1]: 2
arr[0][0][2]: 3
arr[0][1][0]: 1
arr[0][1][1]: 2
arr[0][1][2]: 3
arr[0][2][0]: 1
arr[0][2][1]: 2
arr[0][2][2]: 3
arr[1][0][0]: 1
arr[1][0][1]: 2
arr[1][0][2]: 3
arr[1][1][0]: 1
arr[1][1][1]: 2
arr[1][1][2]: 3
arr[1][2][0]: 1
arr[1][2][1]: 2
arr[1][2][2]: 3
Isi array tiga dimensi:
1 2 3
1 2 3
1 2 3

1 2 3
1 2 3
1 2 3
```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko

NIM: 2311102054

Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk membuat *array* 3 dimensi, tetapi ukuran *array* ditentukan oleh *user*. Karena penjelasan detailnya sudah ada di *source code*, maka saya akan menjelaskan alur program. Pertama deklarasikan variabel untuk ukuran *array*. Setelah itu *user* diminta untuk memasukkan ukuran *array*, sebagai contoh di atas saya memasukkan ukuran 2x3x3. Setelah itu buat *array* dengan ukuran yang sudah ditentukan *user*. Lalu gunakan *loop* untuk *input* isi *array* (*user input*). Lalu terakhir adalah cetak

isi *array*.

c. Unguided 3

Source Code:

```
#include <iostream>

using namespace std;

int main(int argc, char const *argv[])
{
    //Deklarasi variabel yang digunakan
    int n, i;
    //Buat variabel posisi dengan awalnya berada pada indeks
    ke 0
    int posisi = 0, posisi2 = 0;
    //Buat variabel total untuk mean yang awalnya 0
    double total = 0;
    //User memasukkan panjang array
    cout << "Masukkan panjang array: ";
    cin >> n;

    //Membuat array dinamis
    int* arr = new int [n];
    //User memasukkan isi array menggunakan loop
    cout << "Masukkan isi array: \n";
    for (i = 0; i < n; i++) {
        cout << "Array ke-" << i+1 << ": ";
        cin >> arr[i];
        //Tambahkan total sesuai dengan ukuran array
        total += arr[i];
    }

    //Membuat variabel untuk nilai maksimum, minimum, dan
    mean.
    int maks = arr[0];
    int minn = arr[0];
    double mean = total / n;

    //Melakukan pengecekan untuk nilai maksimum dan minimum.
    for (i = 0; i < n; i++) {
        if (arr[i] > maks) {
```

```

        maks = arr[i];
        posisi = i;
    } else if (arr[i] < minn) {
        minn = arr[i];
        posisi2 = i;
    }
}

//Mengoutputkan hasil
cout << "Data Array: ";
for (i = 0; i < n; i++) {
    cout << arr[i];
    //Agar rapih, menambahkan koma (,) di setiap iterasi.
    //Kecuali untuk iterasi terakhir
    if (i != n - 1) {
        cout << ", ";
    }
}
cout << '\n';

cout << "Nilai maksimum adalah : " << maks << " dan berada
pada posisi ke-" << posisi + 1 << '\n';
cout << "Nilai minimum : " << minn << " dan berada pada
posisi ke-" << posisi2 + 1 << '\n';
cout << "Nilai rata-rata : " << mean << '\n';

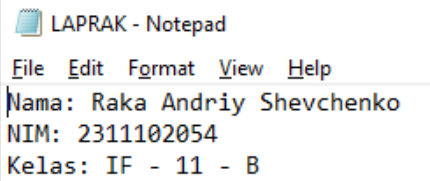
//Menghapus array
delete[] arr;

return 0;
}

```

Output:

```
Masukkan panjang array: 10
Masukkan isi array:
Array ke-1: 1
Array ke-2: 2
Array ke-3: 3
Array ke-4: 4
Array ke-5: 5
Array ke-6: 6
Array ke-7: 7
Array ke-8: 8
Array ke-9: 9
Array ke-10: 10
Data Array: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Nilaimaksimum adalah : 10 dan berada pada posisi ke-10
Nilai minimum : 1 dan berada pada posisi ke-1
Nilai rata-rata : 5.5
```



Deskripsi:

Program ini dibuat untuk mencari nilai maksimum, nilai minimum, dan nilai rata-rata pada *array* 1 dimensi. Karena penjelasan detailnya sudah ada di *source code*, maka saya akan menjelaskan alur program. Pertama deklarasikan semua variabel beserta nilai awalnya (bila dibutuhkan) yang akan digunakan pada program. Setelah itu minta user memasukkan ukuran *array*, lalu buat *pointer* karena di sini saya menggunakan *array* dinamis. Setelah itu buat *loop* untuk *user* menginputkan isi *array*. Lalu buat variabel penampung nilai maksimum, minimum, dan mean, setelah itu lakukan pengecekan pada *array* menggunakan *loop*. Terakhir outputkan isi *array*, nilai maksimum, minimum, dan meannya, dan juga hapus isi *array*. Sebenarnya program ini sangat mirip dengan program pertama (*Unguided 1*) saya juga menggunakan banyak potongan kode dari program pertama.

D. Kesimpulan

Materi *array* dalam C++ mengenai tipe data yang terdiri dari elemen-elemen yang memiliki tipe data yang sama dan diberi nama tertentu. Setiap elemen dalam *array* diakses menggunakan indeks, yang memudahkan pengguna untuk mendapatkan nilai spesifik dengan cepat. Terdapat beberapa perbedaan di antara *array*-*array* yang telah dijelaskan di atas:

1. *Array* satu dimensi:
 - a. Jenis elemen *array* yang hanya terdiri dari satu kolom atau baris.
 - b. Hanya bisa diakses dengan program yang memiliki indeks tertentu.
 - c. Dapat dideklarasikan dengan bentuk tipe_data nama_array[jumlah_eleman];.
 - d. Elemen-elemen *array* dapat diakses oleh program menggunakan suatu indeks tertentu.
2. *Array* dimensi dua:
 - a. Jenis elemen *array* yang terdiri dari 2 bentuk index *array* sehingga memiliki dua atau lebih kolom dengan banyak baris.
 - b. Dapat dideklarasikan dengan bentuk tipe_data nama_array[jumlah_eleman_baris][jumlah_eleman_kolom];.
 - c. Jenis elemen *array* yang hanya terdiri dari satu kolom atau baris.
 - d. Elemen-elemen *array* dapat diakses oleh program menggunakan suatu indeks tertentu.
3. *Array* multidimensi:
 - a. *Array* yang mempunyai ukuran lebih dari dua.
 - b. Bentuk umumnya yaitu tipe_data nama_array[ukuran1][ukuran2]...[ukuranN];.
 - c. *Array* multidimensi merupakan array yang mempunyai ukuran lebih dari dua.
 - d. Elemen-elemen *array* dapat diakses menggunakan banyak indeks atau subskrip.

E. Referensi

[1] Sneh. Two Dimensional Array in C++. Diakses dari

<https://www.digitalocean.com/community/tutorials/two-dimensional-array-in-c-plus-plus>

[2] Universitas Negeri Yogyakarta. Bab 8 Array. Diakses dari

https://staffnew.uny.ac.id/upload/132310817/penelitian/Bab8.Array_.pdf