

LAPORAN PRAKTIKUM

MODUL 2 ARRAY



**Disusun oleh:
Nofita Fitriyani
NIM: 2311102001**

Dosen Pengampu:
Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2024**

BAB I

TUJUAN PRAKTIKUM

1. Mahasiswa dapat memahami konsep Array.
2. Mahasiswa dapat mengetahui jenis dimensi Array dan cara penulisannya.
3. Mahasiswa dapat mengimplementasikan Array pada kode program yang dibuat.

BAB II

DASAR TEORI

Array merupakan struktur data yang paling dasar. Sebelum mempelajari struktur data lainnya, alangkah baiknya, kita perlu mempelajari Array terlebih dahulu. Berikut ini adalah beberapa jenis array :

1. Array Satu Dimensi

Array satu dimensi adalah tipe variabel yang terdiri dari kumpulan data dengan tipe yang sama yang disusun dalam satu baris atau satu dimensi. Setiap elemen di dalam array memiliki sebuah indeks atau nomor yang digunakan untuk mengakses elemen tersebut. Indeks dimulai dari 0 dan berakhir pada jumlah elemen dikurangi satu.

Contohnya, sebuah array satu dimensi yang berisi bilangan bulat {1, 2, 3, 4, 5} memiliki lima elemen dan indeksinya dimulai dari 0. Indeks 0 merujuk pada elemen pertama, indeks 1 merujuk pada elemen kedua, dan seterusnya hingga indeks 4 merujuk pada elemen kelima.

Contoh :

```
#include <iostream>
using namespace std;
int main() {
    int arr[5] = {9, 3, 5, 2, 1}; //deklarasi array
    cout<< arr[1] << endl;
    cout<< arr[4]; }
```

Ouput :

```
3
1
```

2. Array Dua Dimensi

Array dua dimensi adalah variable yang terdiri dari kumpulan array satu dimensi dengan tipe yang sama yang disusun dalam baris dan kolom. Dalam array dua dimensi, setiap elemen memiliki dua indeks, yaitu indeks baris dan indeks kolom. Indeks baris menunjukkan posisi elemen dalam baris, sementara indeks kolom menunjukkan posisi elemen dalam kolom.

Contoh :

```
#include <iostream>
using namespace std;
int main() {
    int arr[2][2] = {{3, 2}, {2, 5}};
    for (int i=0; i<2; i++) { //baris
        for(int j=0; j<2; j++) { //kolom
            cout<< arr[i][j] << ends;
        };
        cout << endl;
    };
}
```

Output :

```
3 2
2 5
```

3. Array Multidimensi

Array multidimensi memiliki kesamaan dengan array satu dimensi dan dua dimensi, namun memiliki kapasitas memori yang lebih besar. Array ini digunakan untuk merepresentasikan array dengan dimensi lebih dari dua atau array yang memiliki lebih dari dua indeks, seperti array tiga dimensi, array

empat dimensi, array lima dimensi, dan seterusnya.

Contoh :

```
#include <iostream>
using namespace std; int main() {
    int arr[2][2][3] = {{{2, 8, 7}, {6, 5, 1}}, {{8,
5, 2}, {9, 2, 7}}};
    for (int i=0; i<2; i++) {
        for(int j=0; j<2; j++) {
            for(int k=0; k<3; k++) {
                cout<< arr[i][j][k] << ends;
            };
            cout<< endl;
        };
        cout<< endl;
    };
}
```

Output :

```
2 8 7
6 5 1

8 5 2
9 2 7
```

4. Array Empat Dimensi

Contoh :

```
int arr [3] [2] [4] [4];
```

5. Array lima dimensi

Contoh :

```
int arr [2] [4] [4] [3] [3];
```

BAB III

GUIDED

1. Guided 1

Program input Array Tiga Dimensi

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;
}
}
```


Screenshoot program

```
PS C:\Praktikum Struktur Data\Modul 2> cd "c:\Praktikum Struktur Data\Modul 2\" ; if ($?) { g++ Guided1.cpp -o Guided1 } ; if ($?) { .\Guided1 }
Input Array[0][0] = 3
Input Array[0][0][1] = 4
Input Array[0][0][2] = 5
Input Array[0][1][0] = 7
Input Array[0][1][1] = 8
Input Array[0][1][2] = 2
Input Array[0][2][0] = 1
Input Array[0][2][1] = 4
Input Array[0][2][2] = 3

Input Array[1][0][0] = 9
Input Array[1][0][1] = 0
Input Array[1][0][2] = 7
Input Array[1][1][0] = 6
Input Array[1][1][1] = 4
Input Array[1][1][2] = 3
Input Array[1][2][0] = 2
Input Array[1][2][1] = 2
Input Array[1][2][2] = 2

Data Array[0][0][0] = 3
Data Array[0][0][1] = 4
Data Array[0][0][2] = 5
Data Array[0][1][0] = 7
Data Array[0][1][1] = 8
Data Array[0][1][2] = 2
Data Array[0][2][0] = 1
Data Array[0][2][1] = 4
Data Array[0][2][2] = 3
Data Array[1][0][0] = 9
Data Array[1][0][1] = 0

Data Array[1][0][2] = 7
Data Array[1][1][0] = 6
Data Array[1][1][1] = 4
Data Array[1][1][2] = 3
Data Array[1][2][0] = 2
Data Array[1][2][1] = 2
Data Array[1][2][2] = 2

345
782
143

907
643
222

PS C:\Praktikum Struktur Data\Modul 2>
```

Deskripsi program

Pertama, program mendeklarasikan array arr dengan dimensi 2x3x3. Kemudian, program menggunakan tiga perulangan bersarang untuk meminta input dari pengguna untuk setiap elemen array. Perulangan pertama digunakan untuk mengakses indeks pertama array, perulangan kedua untuk mengakses indeks kedua, dan perulangan ketiga untuk mengakses indeks ketiga.

Setelah pengguna selesai memasukkan data, program akan menampilkan isi array tersebut dengan menggunakan tiga perulangan bersarang lagi. Perulangan yang pertama, kedua, dan ketiga digunakan untuk mengakses setiap elemen array dan menampilkannya satu per satu dengan menyertakan indeksinya. Terakhir, program

juga menampilkan tampilan array dalam format yang lebih ringkas, yaitu dengan hanya menampilkan nilai elemen array tanpa menyertakan indeksinya.

2. Guided 2

Program mencari nilai maksimal pada array

Source code

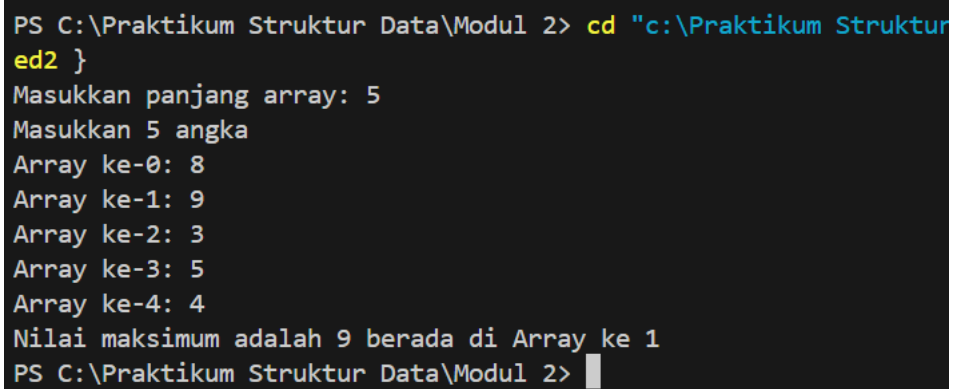
```
#include <iostream>
using namespace std;
int main()
{
    int maks, a, i = 1, lokasi;
    cout << "Masukkan panjang array: ";
    cin >> a;
    int array[a];
    cout << "Masukkan " << a << " angka\n";
    //input array
    for (i = 0; i < a; i++)
    {
        cout << "Array ke-" << (i) << ": ";
        cin >> array[i];
    }
    maks = array[0];
    //cek nilai maksimum
    for (i = 0; i < a; i++)
    {
        if (array[i] > maks)
        {
            maks = array[i];
            lokasi = i;
        }
    }
}
```

```

    }
}
cout << "Nilai maksimum adalah " << maks << " berada di Array ke "
<< lokasi << endl;
}

```

Screenshot Program



```

PS C:\Praktikum Struktur Data\Modul 2> cd "c:\Praktikum Struktur
Data\Modul 2"
Masukkan panjang array: 5
Masukkan 5 angka
Array ke-0: 8
Array ke-1: 9
Array ke-2: 3
Array ke-3: 5
Array ke-4: 4
Nilai maksimum adalah 9 berada di Array ke 1
PS C:\Praktikum Struktur Data\Modul 2>

```

Deskripsi Program

Pertama, program meminta pengguna untuk memasukkan panjang array dan menyimpan nilai tersebut dalam variabel `a`. Selanjutnya, program mendeklarasikan array dengan panjang sesuai dengan nilai yang dimasukkan pengguna sebelumnya. Setelah itu, program menggunakan perulangan `for` untuk meminta pengguna memasukkan nilai-nilai untuk setiap elemen array.

Setelah semua nilai dimasukkan, program menginisialisasi variabel `maks` dengan nilai dari elemen array pertama (`array[0]`) dan variabel `lokasi` untuk menyimpan posisi nilai maksimum dalam array. Program kemudian menggunakan perulangan `for` kembali untuk memeriksa setiap elemen array. Jika nilai elemen array yang sedang diperiksa lebih besar dari nilai maksimum yang telah ditentukan sebelumnya, maka nilai maksimum diupdate dan posisinya disimpan dalam variabel `lokasi`.

Terakhir, program menampilkan nilai maksimum yang ditemukan beserta posisinya dalam array menggunakan output ke layar.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Program nomor ganjil dan genap

Source code

```
#include <iostream>
using namespace std;
//2311102001_NofitaFitriyani
int main() {
    const int data = 10;
    int array[data];

    cout << "Masukkan " << data << " angka:\n";
    for (int i = 0; i < data; i++) {
        cout << "Angka ke-" << (i + 1) << ": ";
        cin >> array[i];
    }

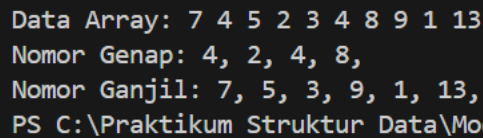
    cout << "\nData Array: ";
    for (int i = 0; i < data; i++) {
        cout << array[i] << " ";
    }

    cout << "\nNomor Genap: ";
    for (int i = 0; i < data; i++) {
        if (array[i] % 2 == 0) {
            cout << array[i] << ", ";
        }
    }

    cout << "\nNomor Ganjil: ";
    for (int i = 0; i < data; i++) {
        if (array[i] % 2 != 0) {
```

```
        cout << array[i] << ", ";  
    }  
}  
cout << endl;  
  
return 0;  
}
```

Screenshoot program



```
Data Array: 7 4 5 2 3 4 8 9 1 13  
Nomor Genap: 4, 2, 4, 8,  
Nomor Ganjil: 7, 5, 3, 9, 1, 13,  
PS C:\Praktikum Struktur Data\Modul 2>
```

Deskripsi program

Pertama, program mendeklarasikan konstanta data dengan nilai 10 untuk menentukan jumlah angka yang akan dimasukkan pengguna. Kemudian, program mendeklarasikan array dengan panjang sesuai dengan nilai data.

Selanjutnya, program menggunakan perulangan for untuk meminta pengguna memasukkan nilai-nilai untuk setiap elemen array. Setelah semua nilai dimasukkan, program menampilkan isi array menggunakan perulangan for untuk menampilkan angka-angka yang telah dimasukkan pengguna.

Kemudian, program menggunakan perulangan for lagi untuk memeriksa setiap elemen array. Jika nilai elemen array tersebut adalah angka genap (habis dibagi 2), maka nilai tersebut ditampilkan dalam kategori "Nomor Genap". Begitu juga dengan angka ganjil (tidak habis dibagi 2), nilai tersebut ditampilkan dalam kategori "Nomor Ganjil".

Terakhir, program menampilkan hasil pemisahan angka genap dan angka ganjil sesuai dengan kategori masing-masing ke layar.

2. Unguided 2

Buatlah program Input array tiga dimensi (seperti pada guided) tetapi jumlah atau ukuran elemennya diinputkan oleh user!

Source code

```
#include <iostream>
using namespace std;
//2311102001_NofitaFitriyani
int main() {
    int inX, inY, inZ;

    // Meminta pengguna memasukkan ukuran array
    cout << "Masukkan ukuran array 3 dimensi:\n";
    cout << "Jumlah elemen x: ";
    cin >> inX;
    cout << "Jumlah elemen y: ";
    cin >> inY;
    cout << "Jumlah elemen z: ";
    cin >> inZ;
    cout << endl;

    // Deklarasi array sesuai dengan ukuran yang dimasukkan
    pengguna
    int arr[inX][inY][inZ];

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

```
}

// Tampilan array
cout << "\nTampilan Array:\n";
for (int x = 0; x < inX; x++) {
    for (int y = 0; y < inY; y++) {
        for (int z = 0; z < inZ; z++) {
            cout << arr[x][y][z] << " ";
        }
        cout << endl;
    }
    cout << endl;
}

return 0;
}
```


Screenshoot program

```
Unguided2 }
Masukkan ukuran array 3 dimensi:
Jumlah elemen x: 2
Jumlah elemen y: 2
Jumlah elemen z: 2

Input Array[0][0][0] = 23
Input Array[0][0][1] = 24
Input Array[0][1][0] = 25
Input Array[0][1][1] = 26

Input Array[1][0][0] = 21
Input Array[1][0][1] = 43
Input Array[1][1][0] = 56
Input Array[1][1][1] = 78

Tampilan Array:
23 24
25 26

21 43
56 78

PS C:\Praktikum Struktur Data\Modul 2>
```

Deskripsi Program

Pertama, program meminta pengguna untuk memasukkan ukuran array tiga dimensi, yaitu jumlah elemen x, y, dan z. Setelah pengguna memasukkan ukuran array, program mendeklarasikan array arr dengan ukuran sesuai dengan input yang diberikan pengguna.

Selanjutnya, program menggunakan tiga perulangan bersarang (untuk x, y, dan z) untuk meminta pengguna memasukkan nilai untuk setiap elemen array. Perulangan pertama digunakan untuk mengakses indeks x, perulangan kedua untuk indeks y, dan perulangan ketiga untuk indeks z. Program akan mencetak pesan "Input Array[x][y][z] = " untuk setiap elemen yang diminta pengguna.

Setelah semua elemen array dimasukkan, program akan menampilkan tampilan array ke layar. Program menggunakan tiga perulangan bersarang lagi untuk mengakses setiap elemen array dan mencetak nilainya ke layar.

3. Unguided 3

Source code

```
#include <iostream>
#include<iomanip>
#include <climits>
using namespace std;
int main() {
    int pilihan_2311102001;
    int bilangan[10];
    int jumlahBilangan = 0;

    while (true) {
        cout << setw(13)<< "MENU\n";
        cout << "1. Input Array \n";
        cout << "2. Tampil Array\n";
        cout << "3. Cari Nilai Minimum\n";
        cout << "4. Cari Nilai Maksimum\n";
        cout << "5. Hitung Rata-Rata\n";
        cout << "0. Keluar\n";
        cout << "Masukan Pilihan : ";
        cin >> pilihan_2311102001;
        cout<<endl;

        switch (pilihan_2311102001) {
            case 0:
                return 0;

            case 1:
                cout << "Masukkan 10 bilangan bulat : \n";
```

```

        for (int i = 0; i < 10; i++) {
            cin >> bilangan[i];
        }
        jumlahBilangan = 10;
        cout<<endl;
        break;

case 2:
    cout << "Isi array = \n";
    for (int i = 0; i < jumlahBilangan; i++) {
        cout << bilangan[i] << " ";
    }
    cout << endl;
    cout<<endl;
    break;

case 3:
    if (jumlahBilangan > 0) {
        int minimum = INT_MAX;
        for (int i = 0; i < jumlahBilangan; i++) {
            if (bilangan[i] < minimum) {
                minimum = bilangan[i];
            }
        }
        cout << "Nilai minimum = " << minimum << endl;
    } else {
        cout << "Array kosong. Silakan inputkan
bilangan terlebih dahulu.\n";
    }
    cout<<endl;
    break;

case 4:
    if (jumlahBilangan > 0) {
        int maksimum = INT_MIN;

```

```

        for (int i = 0; i < jumlahBilangan; i++) {
            if (bilangan[i] > maksimum) {
                maksimum = bilangan[i];
            }
        }
        cout << "Nilai maksimum = " << maksimum <<
endl;

        } else {
            cout << "Array kosong. Silakan inputkan
bilangan terlebih dahulu.\n";
        }
        cout<<endl;
        break;

    case 5:
        if (jumlahBilangan > 0) {
            int total = 0;
            for (int i = 0; i < jumlahBilangan; i++) {
                total += bilangan[i];
            }
            double rataRata = static_cast<double>(total) /
jumlahBilangan;
            cout << "Rata-rata = " << rataRata <<endl;
        } else {
            cout << "Array kosong. Silakan inputkan
bilangan terlebih dahulu.\n";
        }
        cout<<endl;
        break;

    default:
        cout << "Pilihan menu tidak valid. Silakan pilih menu
yang benar.\n";
        break;
    }
}

```

```
}  
  
return 0;  
}
```

Screenshoot program

```
Isi array =  
21 22 23 24 25 26 27 28 29 20  
  
MENU  
1. Input Array  
2. Tampil Array  
3. Cari Nilai Minimum  
4. Cari Nilai Maksimum  
5. Hitung Rata-Rata  
0. Keluar  
Masukan Pilihan : 5  
  
Rata-rata = 24.5  
  
MENU  
1. Input Array  
2. Tampil Array  
3. Cari Nilai Minimum  
4. Cari Nilai Maksimum  
5. Hitung Rata-Rata  
0. Keluar  
Masukan Pilihan : 0
```

Deskripsi Program

Setelah pengguna memilih salah satu opsi, program akan menjalankan operasi yang sesuai dengan pilihan pengguna menggunakan perulangan switch case. Misalnya, jika pengguna memilih opsi "1. Input Array", maka program akan meminta pengguna memasukkan 10 bilangan bulat ke dalam array. Begitu juga dengan opsi lainnya, seperti menampilkan array, mencari nilai minimum, maksimum, dan menghitung nilai rata-rata.

Jika pengguna memilih opsi "0. Keluar", program akan keluar dari perulangan while dan program selesai dijalankan.

BAB IV

KESIMPULAN

Penggunaan array memungkinkan penyimpanan sejumlah data dalam struktur yang terorganisir, memfasilitasi operasi-operasi seperti input, manipulasi, dan analisis data dengan lebih efisien. Misalnya, dalam program yang meminta pengguna untuk memasukkan sejumlah bilangan ke dalam array, array digunakan untuk menyimpan dan mengelola data-data tersebut, memudahkan pengolahan data berulang dengan perulangan for.

Selain itu, program-program tersebut juga menunjukkan penggunaan array dalam konteks multidimensi, seperti array dua dimensi untuk representasi matriks dan array tiga dimensi untuk struktur data yang lebih kompleks. Penggunaan array multidimensi memungkinkan penyimpanan data dalam bentuk yang lebih terstruktur dan mempermudah operasi-operasi khusus seperti pencarian nilai minimum, maksimum, dan perhitungan rata-rata dari elemen-elemen array tersebut. Dengan demikian, penggunaan array dalam program-program tersebut tidak hanya memungkinkan penyimpanan data secara terstruktur, tetapi juga memfasilitasi pengolahan data yang lebih kompleks dan variatif sesuai dengan kebutuhan program yang dibuat.