

LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

MODUL 2

Array



DISUSUN OLEH:

PRIESTY AMEILIANA MAULIDAH

2311102175

S1 IF-11-E

DOSEN:

Muhammad Afrizal Amrustian, S. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO PURWOKERTO

2024

A.DASAR TEORI

Array adalah struktur data dasar yang perlu dipelajari sebelum mempelajari struktur data lainnya. Ada beberapa jenis array: 1) Array Satu Dimensi, tipe variabel dengan data yang disusun dalam satu baris. 2) Array Dua Dimensi, kumpulan array satu dimensi yang disusun dalam baris dan kolom. 3) Array Multidimensi, yang memiliki kapasitas memori lebih besar dan digunakan untuk merepresentasikan array dengan dimensi lebih dari dua.

B.Guided

Guided 1

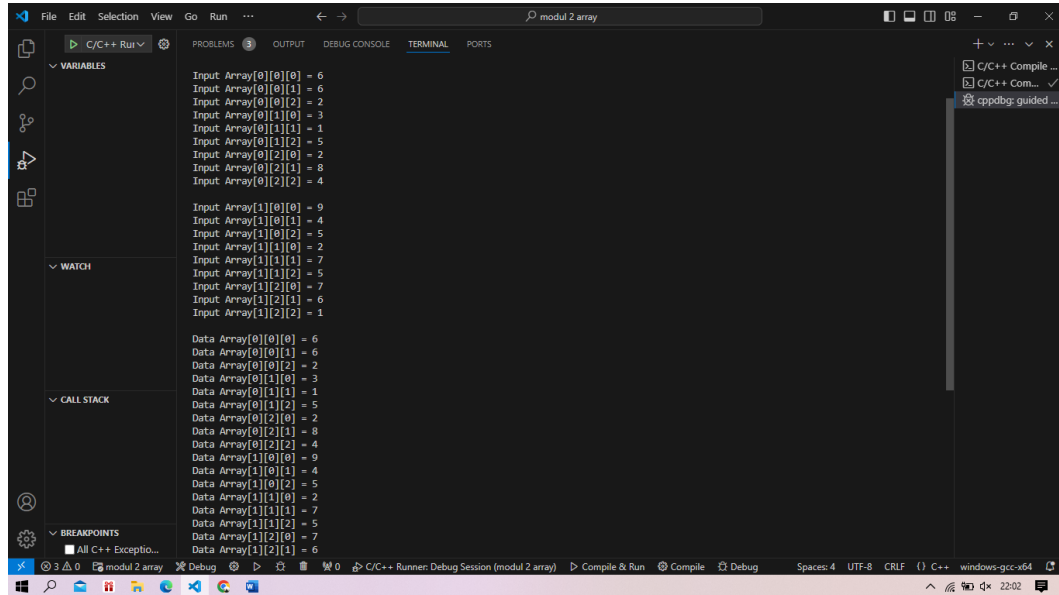
```
// priesty ameiliana maulidah
// 2311102175

#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;
}
}
```

Screenshots output



Deskripsi :

Deklarasi Variabel:

- `int arr[2][3][3]`: Mendeklarasikan array 3 dimensi bertipe integer bernama arr. Dimensi pertama memiliki ukuran 2 (jumlah irisan), dimensi kedua memiliki ukuran 3 (jumlah baris dalam setiap irisan), dan dimensi ketiga memiliki ukuran 3 (jumlah kolom dalam setiap baris).

2. Perulangan Input:

- Tiga loop for bersarang iterasi melalui setiap elemen array, meminta pengguna untuk memasukkan nilai untuk setiap posisi.
 - Loop luar (x) iterasi atas irisan (0 dan 1).

- Loop tengah (y) iterasi atas baris dalam setiap irisan (0, 1, dan 2).
- Loop dalam (z) iterasi atas kolom dalam setiap baris (0, 1, dan 2).
- Di dalam loop dalam, cout menampilkan pesan yang meminta input untuk elemen saat ini di arr[x][y][z].
- cin membaca input pengguna dan menyimpannya di elemen array yang sesuai.

3. Perulangan Output (Metode Pertama):

- Kumpulan lain dari loop for bersarang iterasi melalui elemen array, mirip dengan loop input.
 - Di dalam loop dalam, cout menampilkan pesan yang menunjukkan data di arr[x][y][z], diikuti dengan nilai aktual yang disimpan di elemen tersebut.
 - Loop ini mencetak setiap elemen dengan indeks yang sesuai, memberikan tampilan terstruktur dari konten array.

4. Pernyataan return 0; yang Hilang:

- Kode asli menyertakan pernyataan cout << endl; di luar loop apa pun, yang tidak akan berpengaruh dalam konteks ini. Selain itu, pernyataan return 0; hilang di akhir main.

Pernyataan ini penting untuk menunjukkan penghentian program yang berhasil.

5. Perulangan Output (Metode Kedua - Dikomentari)

- Bagian yang dikomentari berisi kumpulan lain dari loop for yang iterasi melalui elemen array.
 - Di dalam loop dalam, cout hanya mencetak nilai `arr[x][y][z]` menggunakan manipulator `ends` untuk menekan karakter baris baru default setelah setiap elemen.
 - Loop ini akan mencetak seluruh isi array dalam satu baris, berpotensi membuatnya lebih sulit untuk dibaca untuk array yang lebih besar.

Guided 2

```
// priesty ameiliana maulidah
// 2311102175

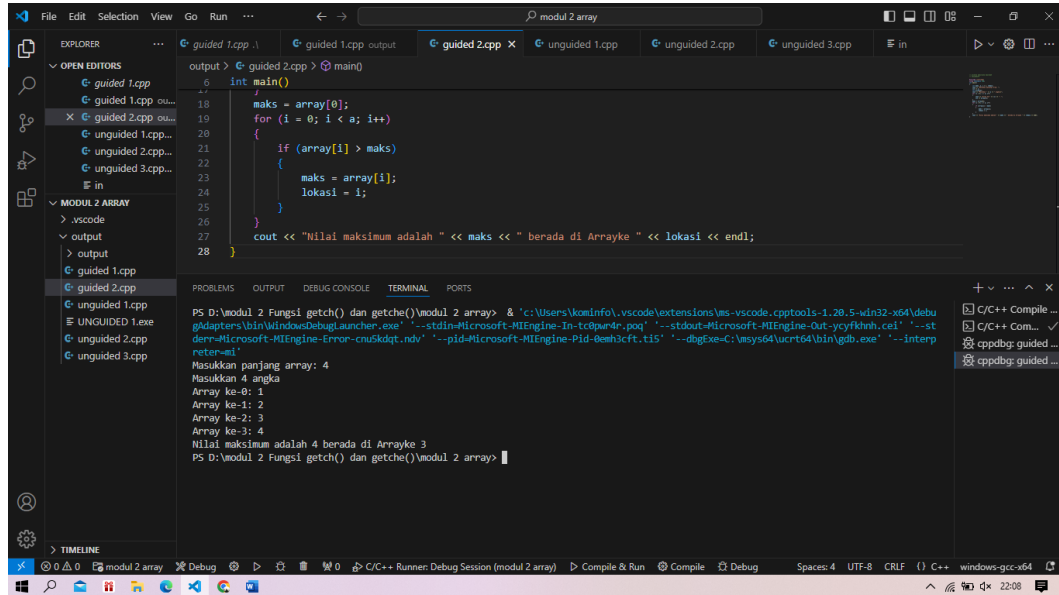
#include <iostream>
using namespace std;
int main()
{
    int maks, a, i = 1, lokasi;

    cout << "Masukkan panjang array: ";
    cin >> a;
    int array[a];
    cout << "Masukkan " << a << " angka\n";
    for (i = 0; i < a; i++)
    {
        cout << "Array ke-" << (i) << ": ";
        cin >> array[i];
    }
    maks = array[0];
    for (i = 0; i < a; i++)
    {
        if (array[i] > maks)
        {
            maks = array[i];
            lokasi = i;
        }
    }
}
```



```
cout << "Nilai maksimum adalah " << maks << " berada di Arrayke " << lokasi << endl;  
}
```

Screenshots output



The screenshot shows a C++ program in Visual Studio Code. The source code in `guided 2.cpp` is as follows:

```
1 //  
2 #include <iostream>  
3 using namespace std;  
4  
5 int main()  
6 {  
7     int maks = array[0];  
8     for (i = 0; i < a; i++)  
9     {  
10        if (array[i] > maks)  
11        {  
12            maks = array[i];  
13            lokasi = i;  
14        }  
15    }  
16    cout << "Nilai maksimum adalah " << maks << " berada di Arrayke " << lokasi << endl;  
17 }  
18
```

The output of the program is shown in the terminal:

```
PS D:\modul 2 Fungsi getch() dan getchey\modul 2 array> .\modul 2 array.exe  
Masukkan panjang array: 4  
Masukkan 4 angka  
Array ke-0: 1  
Array ke-1: 2  
Array ke-2: 3  
Array ke-3: 4  
Nilai maksimum adalah 4 berada di Arrayke 3  
PS D:\modul 2 Fungsi getch() dan getchey\modul 2 array>
```

Deskripsi :

Deklarasi Variabel:

- maks: Sebuah variabel integer untuk menyimpan nilai maksimum yang ditemukan dalam array.
- a: Sebuah variabel integer untuk menyimpan panjang array yang dimasukkan oleh pengguna.
- i: Sebuah variabel penghitung loop yang diinisialisasi dengan 1 (umumnya digunakan sebagai penghitung indeks dalam C++).
- lokasi: Sebuah variabel integer untuk menyimpan indeks elemen yang berisi nilai maksimum.

2. Input Pengguna:

- Menampilkan pesan kepada pengguna untuk memasukkan panjang array menggunakan cout.
- Membaca nilai input dari pengguna menggunakan cin dan menyimpannya di a.
- Mendeklarasikan sebuah array integer bernama array dengan ukuran a untuk menampung panjang yang ditentukan pengguna.
- Menampilkan pesan kepada pengguna untuk memasukkan a angka menggunakan cout.
- Menggunakan loop for untuk mengulangi a kali:
 - Di dalam loop, menampilkan pesan kepada pengguna untuk memasukkan nilai elemen saat ini pada indeks i menggunakan cout.
 - Membaca nilai yang dimasukkan dari pengguna menggunakan cin dan menyimpannya di elemen array[i] yang sesuai dalam array.

3. Menemukan Nilai Maksimum:

- Menginisialisasi maks dengan nilai elemen pertama array[0]. Hal ini mengasumsikan bahwa elemen pertama berpotensi menjadi yang maksimum.
- Loop for lain mengulangi a kali:

- Di dalam loop, memeriksa apakah elemen saat ini `array[i]` lebih besar dari nilai `maks` saat ini.
 - Jika `array[i]` lebih besar:
 - Memperbarui `maks` ke nilai `array[i]`.
 - Memperbarui lokasi ke indeks `i` saat ini. Hal ini menangkap indeks elemen dengan nilai maksimum.

4. Output:

- Mencetak pesan ke konsol menggunakan `cout`, menampilkan nilai maksimum yang ditemukan (`maks`) dan indeks yang sesuai (lokasi) dalam array.

c. unguided/tugas

unguided 1

```
// priesty ameiliana maulidah
// 2311102175

#include <iostream>
using namespace std;

int main() {
    int x, y, z;

    // Meminta pengguna untuk memasukkan ukuran array tiga dimensi
    cout << "Masukkan ukuran dimensi x: ";
    cin >> x;
    cout << "Masukkan ukuran dimensi y: ";
    cin >> y;
    cout << "Masukkan ukuran dimensi z: ";
    cin >> z;

    // Deklarasi array tiga dimensi
    int ***array3D = new int**[x];
    for (int i = 0; i < x; ++i) {
        array3D[i] = new int*[y];
        for (int j = 0; j < y; ++j) {
            array3D[i][j] = new int[z];
        }
    }
}
```

```

// Memasukkan data ke dalam array

int data[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int index = 0;

cout << "Mengisi array dengan data:" << endl;

for (int i = 0; i < x && index < 10; ++i) {
    for (int j = 0; j < y && index < 10; ++j) {
        for (int k = 0; k < z && index < 10; ++k) {
            array3D[i][j][k] = data[index++];
        }
    }
}

// Menampilkan nilai yang ada di dalam array

cout << "Nilai yang ada di dalam array:" << endl;

for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        for (int k = 0; k < z; ++k) {
            cout << "array3D[" << i << "][" << j << "][" << k << "] = " << array3D[i][j][k] <<
endl;
        }
    }
}

```

```
// Memisahkan dan menampilkan elemen genap dan ganjil
```

```
cout << "Nomor genap: ";  
for (int i = 0; i < x; ++i) {  
    for (int j = 0; j < y; ++j) {  
        for (int k = 0; k < z; ++k) {  
            if (array3D[i][j][k] % 2 == 0) {  
                cout << array3D[i][j][k] << " ";  
            }  
        }  
    }  
}  
cout << endl;
```

```
cout << "Nomor ganjil: ";  
for (int i = 0; i < x; ++i) {  
    for (int j = 0; j < y; ++j) {  
        for (int k = 0; k < z; ++k) {  
            if (array3D[i][j][k] % 2 != 0) {  
                cout << array3D[i][j][k] << " ";  
            }  
        }  
    }  
}  
cout << endl;
```

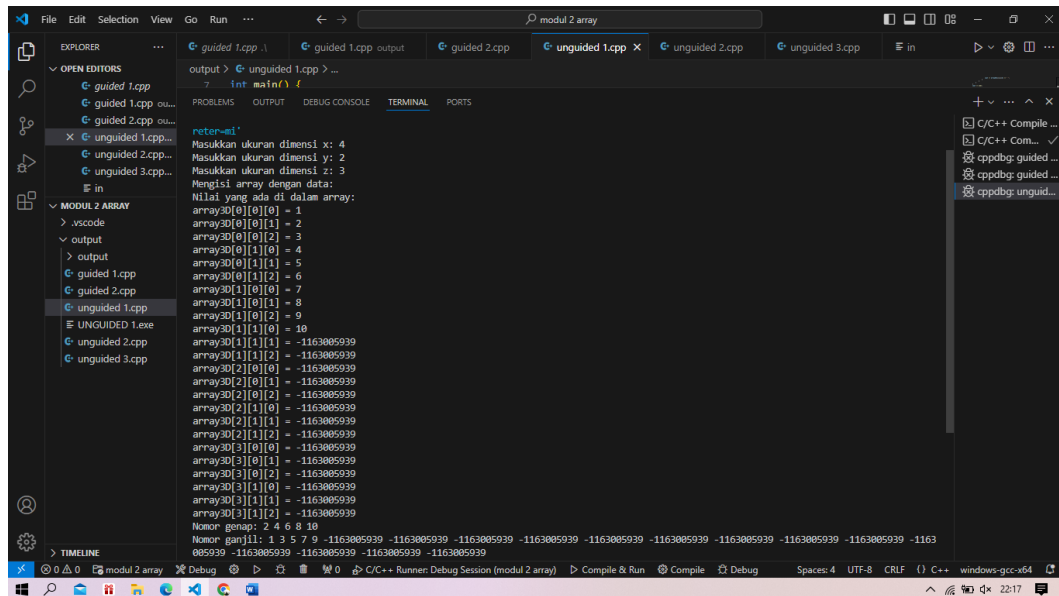
```
// Dealokasi memori yang digunakan oleh array tiga dimensi

for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        delete[] array3D[i][j];
    }
    delete[] array3D[i];
}

delete[] array3D;

return 0;
}
```


Screenshot output



```
int main() {
    int x, y, z;
    cout << "Masukkan ukuran dimensi x: ";
    cin >> x;
    cout << "Masukkan ukuran dimensi y: ";
    cin >> y;
    cout << "Masukkan ukuran dimensi z: ";
    cin >> z;

    cout << "Mengisi array dengan data:";
    int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                array3D[i][j][k] = data[i * y * z + j * z + k];
            }
        }
    }

    cout << "Nomor ganjil: ";
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                if (array3D[i][j][k] % 2 == 1) {
                    cout << array3D[i][j][k] << " ";
                }
            }
        }
    }
    cout << endl;

    cout << "Nomor genap: ";
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                if (array3D[i][j][k] % 2 == 0) {
                    cout << array3D[i][j][k] << " ";
                }
            }
        }
    }
    cout << endl;
}
```

Deskripsi :

- **Membuat dan Mengalokasikan Array 3 Dimensi:**
 - Pengguna diminta memasukkan ukuran untuk setiap dimensi (x, y, dan z) array.
 - Memori dialokasikan secara dinamis menggunakan operator `new` bertingkat untuk membangun struktur array 3 dimensi.
 - Struktur ini terdiri dari array pointer 2 dimensi yang masing-masing berisi array integer 1 dimensi.
- **Memasukkan Data ke dalam Array:**
 - Array data `data` berisi 10 nilai contoh untuk digunakan dalam pengisian array 3 dimensi.
 - Loop `for` bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
 - Nilai dari array `data` disalin ke elemen array 3 dimensi yang sesuai.
- **Menampilkan Isi Array:**
 - Loop `for` bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
 - Nilai setiap elemen dicetak ke konsol menggunakan `cout`.
- **Memisahkan dan Menampilkan Elemen Genap dan Ganjil:**

- Elemen genap dicetak dengan judul "Nomor genap".
- Loop `for` bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
- Jika elemen adalah bilangan genap (hasil bagi sisa pembagian dengan 2 sama dengan 0), elemen dicetak ke konsol.
- Elemen ganjil dicetak dengan judul "Nomor ganjil".
- Loop `for` bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
- Jika elemen adalah bilangan ganjil (hasil bagi sisa pembagian dengan 2 tidak sama dengan 0), elemen dicetak ke konsol.

- **Dealokasi Memori yang Dialokasikan Secara Dinamis:**

- Operator `delete` bertingkat digunakan untuk membebaskan memori yang dialokasikan secara dinamis untuk array 3 dimensi.
- Ini mencegah kebocoran memori dan memastikan sumber daya dibebaskan dengan benar setelah program selesai.

- **Penghentian Program:**

- Program mengembalikan nilai 0 untuk menunjukkan penghentian program yang berhasil.

Unguided2

```
// priesty ameiliana maulidah
// 2311102175

#include <iostream>
using namespace std;

int main() {
    int x, y, z;

    // Meminta pengguna untuk memasukkan ukuran array tiga dimensi
    cout << "Masukkan ukuran dimensi x: ";
    cin >> x;
    cout << "Masukkan ukuran dimensi y: ";
    cin >> y;
    cout << "Masukkan ukuran dimensi z: ";
    cin >> z;

    // Deklarasi array tiga dimensi
    int ***array3D = new int**[x];
    for (int i = 0; i < x; ++i) {
        array3D[i] = new int*[y];
        for (int j = 0; j < y; ++j) {
            array3D[i][j] = new int[z];
        }
    }
}
```

```

// Memasukkan data ke dalam array

cout << "Masukkan elemen-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 << "Array[" << i << "][" << j << "][" << k << "] = ";
            cin >> array3D[i][j][k];
        }
    }
}

// Menampilkan nilai yang ada di dalam array

cout << "Nilai yang ada di dalam array:" << endl;

for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        for (int k = 0; k < z; ++k) {
            cout << "Array[" << i << "][" << j << "][" << k << "] = " << array3D[i][j][k] <<
endl;
        }
    }
}

```

```
// Memisahkan dan menampilkan elemen genap dan ganjil
```

```
cout << "Nomor genap: ";  
  
for (int i = 0; i < x; ++i) {  
    for (int j = 0; j < y; ++j) {  
        for (int k = 0; k < z; ++k) {  
            if (array3D[i][j][k] % 2 == 0) {  
                cout << array3D[i][j][k] << " ";  
            }  
        }  
    }  
}  
  
cout << endl;
```

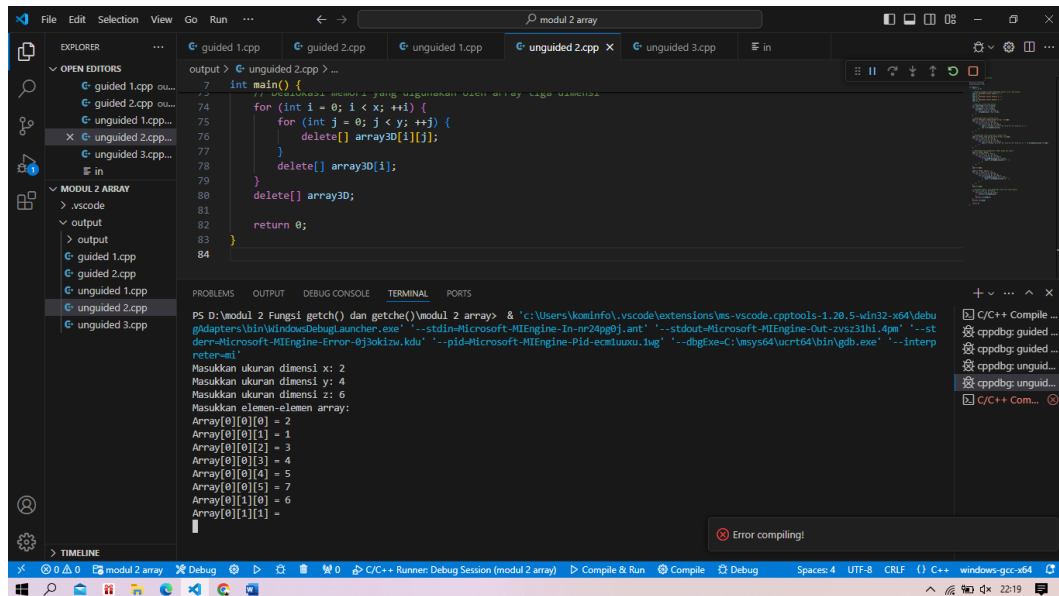
```
cout << "Nomor ganjil: ";  
  
for (int i = 0; i < x; ++i) {  
    for (int j = 0; j < y; ++j) {  
        for (int k = 0; k < z; ++k) {  
            if (array3D[i][j][k] % 2 != 0) {  
                cout << array3D[i][j][k] << " ";  
            }  
        }  
    }  
}  
  
}
```

```
cout << endl;

// Dealokasi memori yang digunakan oleh array tiga dimensi
for (int i = 0; i < x; ++i) {
    for (int j = 0; j < y; ++j) {
        delete[] array3D[i][j];
    }
    delete[] array3D[i];
}
delete[] array3D;

return 0;
}
```

Screenshots output:



Deskripsi :

1. Membuat dan Mengalokasikan Array 3 Dimensi:

- Pengguna diminta memasukkan ukuran untuk setiap dimensi (x, y, dan z) array.
- Memori dialokasikan secara dinamis menggunakan operator new bertingkat untuk membangun struktur array 3 dimensi.
- Struktur ini terdiri dari array pointer 2 dimensi yang masing-masing berisi array integer 1 dimensi.

2. Memasukkan Data ke dalam Array:

- Pengguna diminta memasukkan nilai untuk setiap elemen array 3 dimensi secara manual.

- Loop for bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
- Nilai yang dimasukkan pengguna disimpan di elemen array 3 dimensi yang sesuai.

3. Menampilkan Isi Array:

- Loop for bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
- Nilai setiap elemen dicetak ke konsol menggunakan cout.

4. Memisahkan dan Menampilkan Elemen Genap dan Ganjil:

- Elemen genap dicetak dengan judul "Nomor genap".
- Loop for bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.
- Jika elemen adalah bilangan genap (hasil bagi sisa pembagian dengan 2 sama dengan 0), elemen dicetak ke konsol.
- Elemen ganjil dicetak dengan judul "Nomor ganjil".
- Loop for bersarang digunakan untuk mengakses setiap elemen array 3 dimensi.

- Jika elemen adalah bilangan ganjil (hasil bagi sisa pembagian dengan 2 tidak sama dengan 0), elemen dicetak ke konsol.

5. Dealokasi Memori yang Dialokasikan Secara Dinamis:

- Operator delete bertingkat digunakan untuk membebaskan memori yang dialokasikan secara dinamis untuk array 3 dimensi.
- Ini mencegah kebocoran memori dan memastikan sumber daya dibebaskan dengan benar setelah program selesai.

6. Penghentian Program:

- Program mengembalikan nilai 0 untuk menunjukkan penghentian program yang berhasil.

Perbedaan Utama dengan Versi Sebelumnya:

- Versi ini meminta pengguna untuk memasukkan nilai untuk setiap elemen array 3 dimensi secara manual, sedangkan versi sebelumnya menggunakan array data yang telah ditentukan.
- Hal ini memberikan kontrol lebih kepada pengguna atas data yang dimasukkan ke dalam array.

Unguided 3

```
// priesty ameiliana maulidah
// 2311102175

#include <iostream>
using namespace std;

// Fungsi untuk mencari nilai maksimum
int cariMaksimum(int arr[], int n) {
    int maks = arr[0];
    for (int i = 1; i < n; ++i) {
        if (arr[i] > maks) {
            maks = arr[i];
        }
    }
    return maks;
}

// Fungsi untuk mencari nilai minimum
int cariMinimum(int arr[], int n) {
    int min = arr[0];
    for (int i = 1; i < n; ++i) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    return min;
}
```

```
// Fungsi untuk mencari nilai rata-rata
double cariRataRata(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        sum += arr[i];
    }
    return static_cast<double>(sum) / n;
}

int main() {
    int n;

    // Meminta pengguna untuk memasukkan jumlah elemen dalam array
    cout << "Masukkan jumlah elemen dalam array: ";
    cin >> n;

    // Deklarasi array dengan ukuran n
    int *arr = new int[n];

    // Memasukkan elemen-elemen ke dalam array
    cout << "Masukkan elemen-elemen array:" << endl;
    for (int i = 0; i < n; ++i) {
        cout << "Elemen ke-" << i + 1 << ": ";
        cin >> arr[i];
    }
```

```
int pilihan;

do {
    // Menampilkan menu
    cout << "\nMenu:" << endl;
    cout << "1. Cari Nilai Maksimum" << endl;
    cout << "2. Cari Nilai Minimum" << endl;
    cout << "3. Cari Nilai Rata-Rata" << endl;
    cout << "4. Keluar" << endl;
    cout << "Masukkan pilihan: ";
    cin >> pilihan;

    switch (pilihan) {
        case 1:
            cout << "Nilai Maksimum: " << cariMaksimum(arr, n) << endl;
            break;
        case 2:
            cout << "Nilai Minimum: " << cariMinimum(arr, n) << endl;
            break;
        case 3:
            cout << "Nilai Rata-Rata: " << cariRataRata(arr, n) << endl;
            break;
```

case 4:

```
    cout << "Keluar dari program." << endl;
```

```
    break;
```

default:

```
    cout << "Pilihan tidak valid. Silakan coba lagi." << endl;
```

```
    }
```

```
} while (pilihan != 4);
```

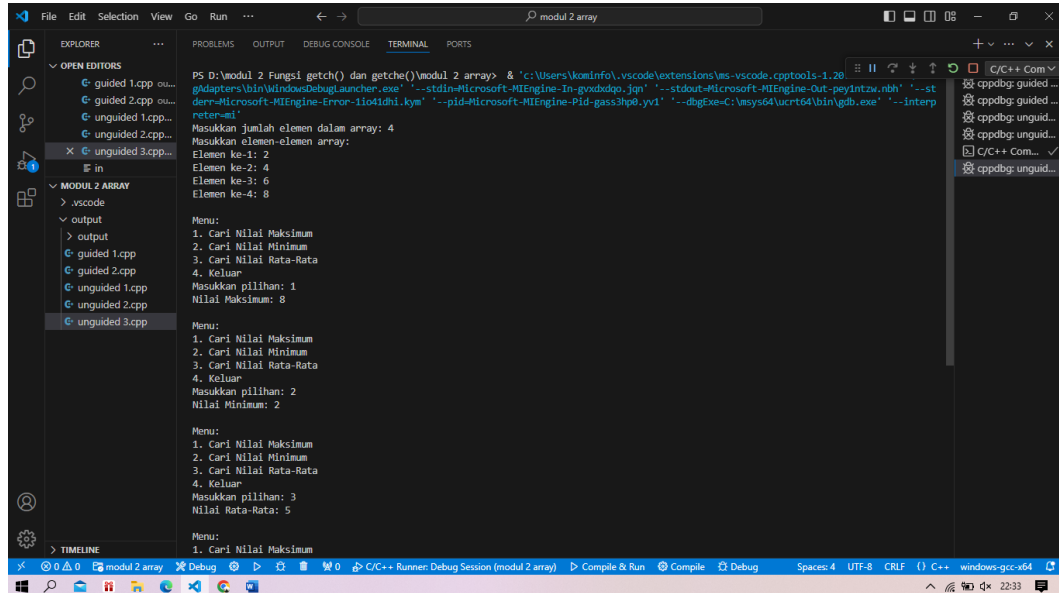
```
// Dealokasi memori yang digunakan oleh array
```

```
delete[] arr;
```

```
return 0;
```

```
}
```

Screenshot output



Deskripsi :

1. Fungsi cariMaksimum:

- Fungsi ini menerima array integer arr dan ukurannya n sebagai parameter.
- Menginisialisasi variabel maks dengan nilai elemen pertama array.
- Melakukan iterasi melalui elemen array arr dimulai dari elemen kedua ($i = 1$).
- Membandingkan nilai elemen array saat ini ($arr[i]$) dengan nilai maks.
 - Jika $arr[i]$ lebih besar dari maks, maka nilai maks diperbarui dengan $arr[i]$.

- Mengembalikan nilai maksimum yang ditemukan dalam array.

2. Fungsi cariMinimum:

- Mirip dengan cariMaksimum, fungsi ini menerima array integer arr dan ukurannya n sebagai parameter.
- Menginisialisasi variabel min dengan nilai elemen pertama array.
- Melakukan iterasi melalui elemen array arr dimulai dari elemen kedua ($i = 1$).
- Membandingkan nilai elemen array saat ini ($arr[i]$) dengan nilai min.
 - Jika $arr[i]$ lebih kecil dari min, maka nilai min diperbarui dengan $arr[i]$.
- Mengembalikan nilai minimum yang ditemukan dalam array.

3. Fungsi cariRataRata:

- Fungsi ini menerima array integer arr dan ukurannya n sebagai parameter.
- Menginisialisasi variabel sum dengan nilai 0.
- Melakukan iterasi melalui elemen array arr dan menambahkan nilainya ke sum.

- Menghitung rata-rata dengan membagi sum dengan n.
- Mengembalikan nilai rata-rata array sebagai tipe data double.

4. Fungsi Utama (main):

- Mendeklarasikan variabel integer n untuk menyimpan jumlah elemen array.
- Meminta pengguna untuk memasukkan nilai n (jumlah elemen) menggunakan cout dan cin.
- Deklarasikan pointer array integer arr dengan ukuran n menggunakan operator new.
- Meminta pengguna untuk memasukkan nilai-nilai elemen array satu per satu menggunakan cout dan cin.
- Menyajikan menu interaktif menggunakan loop do-while dan variabel pilihan untuk mengontrol pilihan pengguna:
 - Pilihan 1: Memanggil fungsi cariMaksimum dan menampilkan nilai maksimum array.
 - Pilihan 2: Memanggil fungsi cariMinimum dan menampilkan nilai minimum array.
 - Pilihan 3: Memanggil fungsi cariRataRata dan menampilkan nilai rata-rata array.

- Pilihan 4: Mengakhiri program dan menampilkan pesan keluar.
- Membebaskan memori yang dialokasikan untuk array arr menggunakan operator delete[].
- Mengembalikan nilai 0 untuk menunjukkan eksekusi program yang berhasil.

E. Referensi

<https://www.scholarhat.com/tutorial/datastructures/arrays-in-data-structures>