

**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 merupakan struktur data yang paling dasar.

Berikut ini adalah beberapa jenis array :

### 1.) Array satu dimensi

Array satu dimensi tipe variabel yang terdiri dari kumpulan data dengan tipe yang sama yang disusun dalam satu barisan 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 indeksnya dimulai dari 0. indeks 0 merujuk pada elemen pertama, indeks 1 merujuk pada elemen kedua, dan seterusnya hingga indeks 4 merujuk pada elemen kelima.

### 2.) Array dua dimensi

Array dua dimensi adalah variabel 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 baris dan indeks kolom. Indeks baris menunjukkan

posisi elemen dalam barisan, sementara indeks kolom menunjukkan posisi elemen dalam kolom.

### 3.) Array multidimensi

Array multidimensi memiliki keamanan 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.

### 4.) Array empat dimensi

### 5.) Array lima dimensi

## B.Guided

### Guided 1

```
#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++)

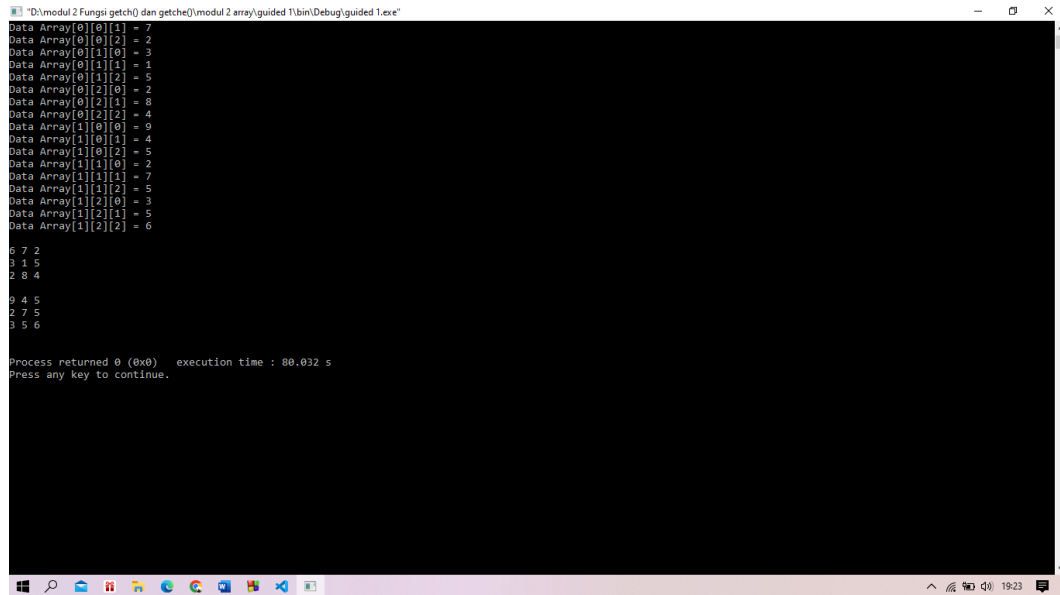
{

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



```
"D:\modul 2 Fungsi getch() dan getchc()\modul 2 array\guided 1\bin\Debug\guided 1.exe"
Data Array[0][0][1] = 7
Data Array[0][0][2] = 2
Data Array[0][1][0] = 3
Data Array[0][1][1] = 1
Data Array[0][1][2] = 5
Data Array[0][2][0] = 2
Data Array[0][2][1] = 8
Data Array[0][2][2] = 4
Data Array[1][0][0] = 9
Data Array[1][0][1] = 4
Data Array[1][0][2] = 5
Data Array[1][1][0] = 2
Data Array[1][1][1] = 7
Data Array[1][1][2] = 5
Data Array[1][2][0] = 3
Data Array[1][2][1] = 5
Data Array[1][2][2] = 6

6 7 2
3 1 5
2 8 4

9 4 5
2 7 5
3 5 6

Process returned 0 (0x0)   execution time : 88.032 s
Press any key to continue.
```

## Deskripsi:

### 1. Namespace:

- `#include <iostream>`: Menyertakan pustaka `iostream` yang menyediakan fungsi input/output seperti `cin` (input standar) dan `cout` (output standar).
- `using namespace std;`: Membawa namespace `std` ke dalam cakupan. Hal ini memungkinkan penggunaan objek seperti `cin` dan `cout` tanpa awalan `std::`.

### 2. Fungsi Utama:

- `int main()`: Merupakan titik masuk program di mana eksekusi dimulai. Tipe data `int` pada return menandakan bahwa program akan

mengembalikan nilai integer (biasanya 0 untuk eksekusi yang berhasil).

### **3. Deklarasi Array:**

- `int arr[2][3][3];`: Mendeklarasikan array integer 3 dimensi bernama `arr`. Dimensi pertama memiliki ukuran 2, dimensi kedua memiliki ukuran 3, dan dimensi ketiga juga memiliki ukuran 3. Ini secara efektif membuat kumpulan 2 "irisan" (bayangkan sebagai mini-array), masing-masing berisi matriks 3x3 integer.

### **4. Perulangan Input:**

- Tiga loop for bertingkat iterates melalui semua elemen array. Loop terluar (x) iterates melalui dimensi pertama (0 dan 1). Loop tengah (y) iterates melalui dimensi kedua (0, 1, dan 2). Loop terdalam (z) iterates melalui dimensi ketiga (0, 1, dan 2).
- Di dalam loop, `cout` meminta pengguna untuk memasukkan nilai untuk setiap elemen, dengan menentukan indeks yang sesuai di array (misalnya, `Input Array[0][1][2] =`).
- `cin` membaca input pengguna dan menyimpannya di elemen yang sesuai dari array `arr`.

- `cout << endl;` kosong setelah setiap irisan input menambahkan baris kosong untuk meningkatkan keterbacaan.

## **5. Perulangan Output (Detail):**

- Bagian ini menggunakan loop for bertingkat yang serupa untuk iterates melalui elemen array.
- `cout << "Data Array[" << x << "][" << y << "][" << z << "]" = " << arr[x][y][z] << endl;`; Di dalam loop, baris ini mencetak pesan deskriptif yang menunjukkan indeks elemen saat ini dan nilai yang sesuai yang diambil dari array `arr`. `endl` menambahkan baris baru setelah setiap elemen.

## **6. Tampilan Array (Output Alternatif):**

- Kumpulan loop for bertingkat lainnya iterates melalui array.
- `cout << arr[x][y][z] << ends;`; Baris ini langsung mencetak nilai elemen saat ini (`arr[x][y][z]`) tanpa pesan deskriptif. Manipulator `ends` menekan baris baru otomatis setelah setiap elemen, memungkinkan mereka untuk dicetak pada baris yang sama.



- `cout << endl;`: `endl` ini di akhir loop dalam memastikan baris baru setelah setiap "irisan" (mini-array) dicetak.

## Guided 2

```
#include <iostream>

using namespace std;

int main()
{
    int maks, a, i = 1, lokasi;

    cout << "Masukkan panjang array: ";
    cin >> a;

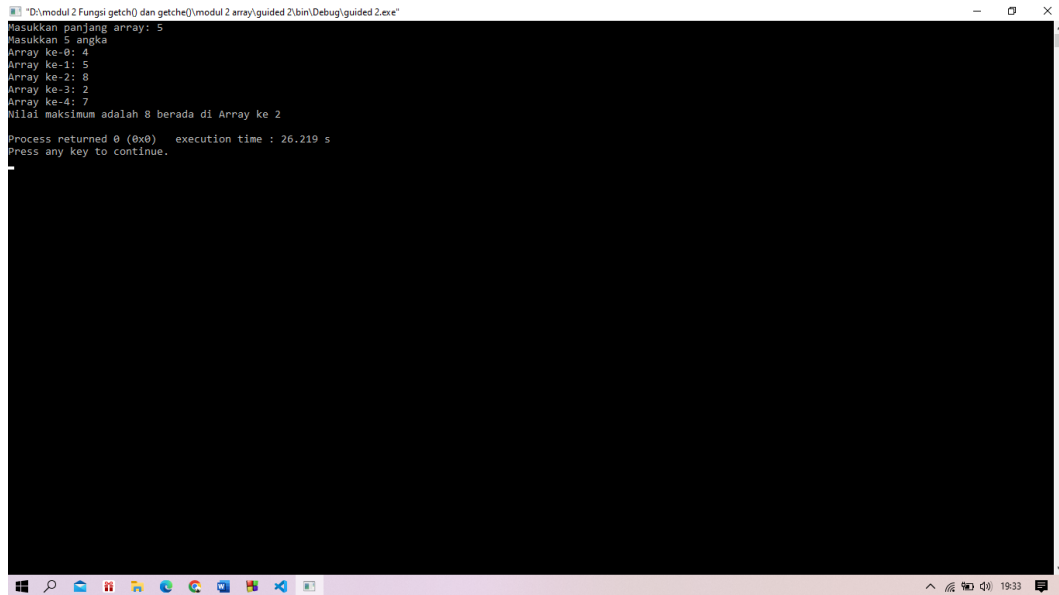
    int array[a];

    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 Array ke " << lokasi << endl;
}
```

## Screenshots output



```
"D:\modul 2 Fungsi getch() dan getchc()\modul 2 array\guided 2\bin\Debug\guided 2.exe"
Masukkan panjang array: 5
Masukkan 5 angka
Array ke-0: 4
Array ke-1: 5
Array ke-2: 8
Array ke-3: 2
Array ke-4: 7
Nilai maksimum adalah 8 berada di Array ke 2
Process returned 0 (0x0)   execution time : 26.219 s
Press any key to continue.
```

Deskripsi :

### 1. Header dan Namespace:

- `#include <iostream>`: Baris ini menyertakan pustaka `iostream` yang menyediakan fungsi penting untuk input (menggunakan `cin`) dan output (menggunakan `cout`) dalam C++.

### 2. Fungsi Utama:

- `int main()`: Ini adalah titik masuk program tempat eksekusi dimulai. Tipe data `int` pada return menunjukkan program akan mengembalikan nilai integer, biasanya 0 untuk eksekusi yang berhasil.

### 3. Deklarasi Variabel:

- `int maks, a, i = 1, lokasi;`: Baris ini mendeklarasikan empat variabel integer:

- maks: Variabel ini akan menyimpan nilai maksimum yang ditemukan dalam array.
- a: Variabel ini akan menampung panjang array yang akan dibuat, seperti yang dimasukkan oleh pengguna.
- i: Variabel ini bertindak sebagai penghitung loop, diinisialisasi menjadi 1.
- lokasi: Variabel ini akan melacak indeks (posisi) nilai maksimum dalam array.

#### **4. Input Pengguna untuk Panjang Array:**

- `cout << "Masukkan panjang array: ";` Baris ini menampilkan prompt yang meminta pengguna untuk memasukkan panjang array yang diinginkan.
- `cin >> a;` Baris ini membaca input pengguna untuk panjang array dan menyimpannya di variabel `a`.

#### **5. Deklarasi Array:**

- `int array[a];` Baris ini mendeklarasikan array integer bernama `array` dengan ukuran yang ditentukan oleh input pengguna (`a`). Ini membuat array yang dapat menampung `a` nilai integer.

#### **6. Input Pengguna untuk Elemen Array:**

- `cout << "Masukkan " << a << " angka\n";`; Baris ini mencetak pesan yang menginstruksikan pengguna untuk memasukkan jumlah nilai yang ditentukan (*a*) untuk elemen array.
- Loop for (`for (i = 0; i < a; i++)`) berulang *a* kali, memungkinkan pengguna untuk memasukkan nilai untuk setiap elemen array.
  - Di dalam loop:
    - `cout << "Array ke-" << (i) << ": ";`; Baris ini menampilkan prompt yang menunjukkan indeks elemen saat ini (mulai dari 0).
    - `cin >> array[i];`; Baris ini membaca input pengguna untuk nilai elemen saat ini dan menyimpannya di posisi yang sesuai (*i*) dari array.

## **7.Menemukan Nilai Maksimum:**

- `maks = array[0];`; Baris ini menginisialisasi variabel *maks* dengan nilai elemen pertama (`array[0]`). Ini berfungsi sebagai titik awal untuk perbandingan nilai maksimum.
- Loop for kedua (`for (i = 0; i < a; i++)`) iterasi melalui array lagi:
  - Di dalam loop:

- `if (array[i] > maks)`: Kondisi ini memeriksa apakah elemen saat ini (`array[i]`) lebih besar dari nilai maksimum saat ini (`maks`).
- Jika kondisi benar:
  - `maks = array[i];`  
Variabel `maks` diperbarui dengan nilai maksimum baru yang ditemukan di elemen saat ini.
  - `lokasi = i;`  
Variabel `lokasi` menyimpan indeks elemen yang berisi nilai maksimum saat ini.

## **8.Mencetak Nilai Maksimum dan Indeksnya:**

- `cout << "Nilai maksimum adalah " << maks << " berada di Array ke " << lokasi << endl;`; Baris ini mencetak hasil akhir, menampilkan nilai maksimum (`maks`) dan indeks yang sesuai (`lokasi`) dalam array.

## c. unguided/tugas

### unguided 1

```
#include <iostream>

using namespace std;

int main(){
    int n;
    cout << "masukkan jumlah data: ";
    cin >> n;

    int data [n];
    cout << "masukkan data: ";
    for (int i = 0; i < n; i++){
        cin >> data[i];
    }

    //menampilkan data array
    cout << "data array: ";
    for (int i = 0; i < n; i++) {
        cout << data[i] << " ";
    }
    cout << endl;
```

```
// menampilkan nomor genap
```

```
cout << "nomor genap: ";
```

```
for (int i= 0; i< n; i++){
```

```
    if (data[i] % 2 == 0){
```

```
        cout << data [i] << " ";
```

```
    }
```

```
}
```

```
cout << endl;
```

```
// menampilkan nomor ganjil
```

```
cout << "nomor ganjil: ";
```

```
for (int i=0; i< n; i++) {
```

```
    if (data[i] % 2 ==1) {
```

```
        cout << data[i] << " ";
```

```
    }
```

```
}
```

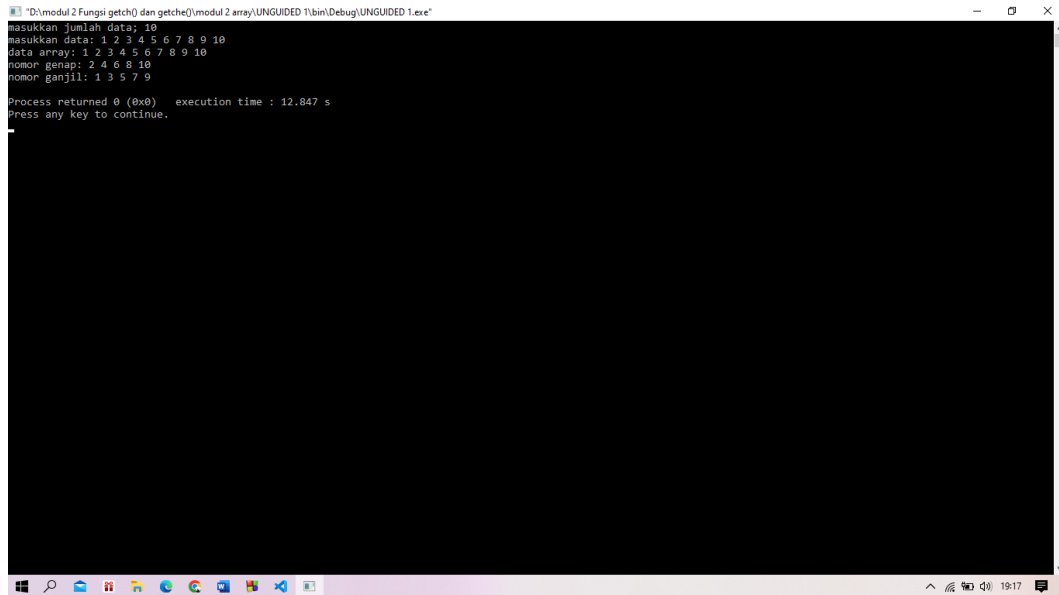
```
cout << endl;
```

```
    return 0;
```

```
}
```



## Screenshots output



```
"D:\modul 2 Fungsi getch() dan getchc()\modul 2 array\UNGUIDED 1\bin\Debug\UNGUIDED 1.exe"
masukkan jumlah data: 10
masukkan data: 1 2 3 4 5 6 7 8 9 10
data array: 1 2 3 4 5 6 7 8 9 10
nomor genap: 2 4 6 8 10
nomor ganjil: 1 3 5 7 9
Process returned 0 (0x0)   execution time : 12.847 s
Press any key to continue.
```

Deskripsi :

### 1.Header dan Namespace:

- `#include <iostream>`: Baris ini menyertakan pustaka `iostream` yang menyediakan fungsi penting untuk input (menggunakan `cin`) dan output (menggunakan `cout`) dalam C++.
- `using namespace std;`: Pernyataan ini membawa namespace `std` ke dalam cakupan, memungkinkan penggunaan objek seperti `cin` dan `cout` tanpa awalan `std::`.

### 2.Fungsi Utama:

- `int main()`: Ini adalah titik masuk program tempat eksekusi dimulai. Tipe data `int` pada `return`

menunjukkan program akan mengembalikan nilai integer, biasanya 0 untuk eksekusi yang berhasil.

### **3.Input Pengguna untuk Ukuran Array:**

- `int n;` Baris ini mendeklarasikan variabel integer bernama `n` untuk menyimpan jumlah elemen yang akan dimasukkan pengguna untuk array.
- `cout << "masukkan jumlah data; ";` Baris ini menampilkan pesan yang meminta pengguna untuk memasukkan jumlah elemen (data) yang diinginkan untuk array.
- `cin >> n;` Baris ini membaca input pengguna untuk jumlah elemen dan menyimpannya di variabel `n`.

### **4.Deklarasi Array:**

- `int data[n];` Baris ini mendeklarasikan array integer bernama `data` dengan ukuran yang ditentukan oleh input pengguna (`n`). Ini membuat array yang mampu menampung `n` nilai integer.

### **5.Input Pengguna untuk Elemen Array:**

- `cout << "masukkan data: ";` Baris ini menampilkan pesan yang menginstruksikan pengguna untuk memasukkan nilai data aktual untuk elemen array.

- Loop for (for (int i = 0; i < n; i++)) iterasi n kali, memungkinkan pengguna untuk memasukkan nilai untuk setiap elemen array.
  - Di dalam loop:
    - `cin >> data[i];` Baris ini membaca input pengguna untuk nilai elemen saat ini dan menyimpannya di posisi yang sesuai (i) dari array data.

## **6.Menampilkan Array Data:**

- `cout << "data array: ";` Baris ini mencetak label yang menunjukkan awal dari output array data.
- Loop for kedua (for (int i = 0; i < n; i++)) iterasi melalui elemen array untuk mencetaknya.
  - Di dalam loop:
    - `cout << data[i] << " ";` Baris ini mencetak nilai elemen saat ini (`data[i]`) diikuti oleh spasi, efektif menampilkan semua elemen dalam satu baris.

## **7.Menampilkan Angka Genap:**

- `cout << "nomor genap: ";` Baris ini mencetak label yang menunjukkan awal dari output angka genap.

- Loop for ketiga (`for (int i = 0; i < n; i++)`) iterasi melalui elemen array lagi.
  - Di dalam loop:
    - `if (data[i] % 2 == 0)`: Kondisi ini memeriksa apakah elemen saat ini (`data[i]`) habis dibagi 2 (bilangan genap).
      - Jika kondisi benar:
        - `cout << data[i] << " ";` Baris ini mencetak nilai elemen genap saat ini diikuti oleh spasi.

## **8.Menampilkan Angka Ganjil:**

- `cout << "nomor ganjil: ";` Baris ini mencetak label yang menunjukkan awal dari output angka ganjil.
- Loop for keempat (`for (int i = 0; i < n; i++)`) iterasi melalui elemen array untuk terakhir kalinya.
  - Di dalam loop:
    - `if (data[i] % 2 == 1)`: Kondisi ini memeriksa apakah elemen saat ini (`data[i]`) memiliki sisa 1 ketika dibagi 2 (bilangan ganjil).
      - Jika kondisi benar:

- `cout << data[i] << " ";` Baris ini mencetak nilai elemen ganjil saat ini diikuti oleh spasi.

## **9. Penghentian Program:**

- `return 0;` Baris ini menunjukkan eksekusi program yang berhasil dengan mengembalikan nilai 0.

## Unguided 2

```
#include <iostream>

using namespace std;

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

    //input dimensi array
    cout << "masukkan dimensi x: ";
    cin >> x;

    cout << "masukkan dimensi y: ";
    cin >> y;

    cout << "masukkan dimensi z: ";
    cin >> z;

    //deklarasi array tiga dimensi
    int arr [x][y][z];
```

```

//input elemen array

for (int i = 0; i < x; i++){
    for (int j = 0; j < y; j++){
        for (int k = 0; k < z; k++){
            cout << "masukkan elemen arr[" << i << "][" << j << "][" << k << "]: ";
            cin >> arr [i][j][k];
        }
    }
}

//output array

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;
}

```

Screenshots output

```
Select "D:\modul 2 Fungsi getch() dan getche()\modul 2 array\UNGUIDED 2\bin\Debug\UNGUIDED 2.exe"
masukkan dimensi x: 2
masukkan dimensi y: 2
masukkan dimensi z: 2
masukkan elemen arr[0][0]: 6
masukkan elemen arr[0][1]: 7
masukkan elemen arr[1][0]: 2
masukkan elemen arr[1][1]: 3
masukkan elemen arr[0][0]: 1
masukkan elemen arr[1][1]: 5
masukkan elemen arr[1][0]: 2
masukkan elemen arr[1][1]: 8
6 7
2 3
1 5
2 8
Process returned 0 (0x0)   execution time : 92.023 s
Press any key to continue.
```

Deskripsi :

## 1. Namespace:

- `#include <iostream>`: Baris ini menyertakan pustaka `iostream` yang menyediakan fungsi penting untuk input (menggunakan `cin`) dan output (menggunakan `cout`) dalam C++.
- `using namespace std;`: Pernyataan ini membawa namespace `std` ke dalam cakupan, memungkinkan penggunaan objek seperti `cin` dan `cout` tanpa awalan `std::`.

## 2. Fungsi Utama:

- `int main()`: Ini adalah titik masuk program tempat eksekusi dimulai. Tipe data `int` pada return menunjukkan program akan mengembalikan nilai integer, biasanya 0 untuk eksekusi yang berhasil.



### **3.Input Pengguna untuk Dimensi Array:**

- `int x, y, z;` Baris ini mendeklarasikan tiga variabel integer: `x`, `y`, dan `z` untuk menyimpan dimensi (ukuran) array tiga dimensi yang akan dibuat.
- Berikutnya, prompt dan baris input memungkinkan pengguna memasukkan nilai yang diinginkan untuk setiap dimensi:
  - `cout << "masukkan dimensi x: ";` Meminta pengguna untuk memasukkan nilai dimensi `x`.
  - `cin >> x;` Membaca input pengguna untuk dimensi `x` dan menyimpannya di variabel `x`.
  - Baris prompt dan input serupa mengikuti untuk dimensi `y` dan `z`.

### **4.Deklarasi Array:**

- `int arr[x][y][z];` Baris ini mendeklarasikan array tiga dimensi integer bernama `arr`. Ukuran setiap dimensi ditentukan oleh input pengguna (`x`, `y`, dan `z`) dari langkah sebelumnya. Ini membuat array yang dapat menampung `x` baris, `y` kolom dalam setiap baris, dan `z` elemen dalam setiap kolom, dengan total  $x * y * z$  nilai integer.

### **5.Input Pengguna untuk Elemen Array:**

- Loop for (bertingkat) ini berulang melalui setiap elemen array, meminta pengguna untuk memasukkan nilai untuk setiap elemen:
  - for (int i = 0; i < x; i++): Loop terluar berulang x kali, sekali untuk setiap baris.
    - for (int j = 0; j < y; j++): Loop tengah berulang y kali dalam setiap iterasi loop terluar, sekali untuk setiap kolom dalam baris.
      - for (int k = 0; k < z; k++): Loop terdalam berulang z kali dalam setiap iterasi loop tengah, sekali untuk setiap elemen dalam kolom.
        - cout << "masukkan elemen arr[" << i << "][" << j << "][" << k << "]: "; Meminta pengguna untuk memasukkan nilai elemen pada indeks i, j, dan k saat ini.
        - cin >> arr[i][j][k]; Membaca input pengguna dan menyimpannya di elemen yang sesuai dari array arr.

## 6.Mencetak Array:

- Loop for (bertingkat) ini berulang melalui array lagi, kali ini untuk mencetak elemen-elemennya:
  - for (int i = 0; i < x; i++): Loop terluar berulang x kali.
    - for (int j = 0; j < y; j++): Loop tengah berulang y kali dalam setiap iterasi loop terluar.
      - for (int k = 0; k < z; k++): Loop terdalam berulang z kali dalam setiap iterasi loop tengah.
        - cout << arr[i][j][k] << " ";  
Mencetak nilai elemen pada indeks i, j, dan k, diikuti oleh spasi.
      - cout << endl; Setelah semua elemen dalam baris dicetak, karakter baris baru disisipkan untuk pindah ke baris berikutnya untuk output.
- Baris cout << endl; terakhir menambahkan baris baru tambahan setelah semua baris dicetak.

## **7. Penghentian Program:**

- return 0; Baris ini menunjukkan eksekusi program yang berhasil dengan mengembalikan nilai 0.

## Unguided 3

```
#include <iostream>

#include <algorithm>

using namespace std;

int main() {

    int n;

    double sum = 0;

    // Input jumlah elemen array
    cout << "Masukkan jumlah elemen array: ";

    cin >> n;

    // Deklarasi array
    double arr[n];

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

```
// Mencari nilai maksimum

double max = arr[0];
for (int i = 1; i < n; i++) {
    if (arr[i] > max) {
        max = arr[i];
    }
}

// Mencari nilai minimum

double min = arr[0];
for (int i = 1; i < n; i++) {
    if (arr[i] < min) {
        min = arr[i];
    }
}

// Mencari nilai rata-rata

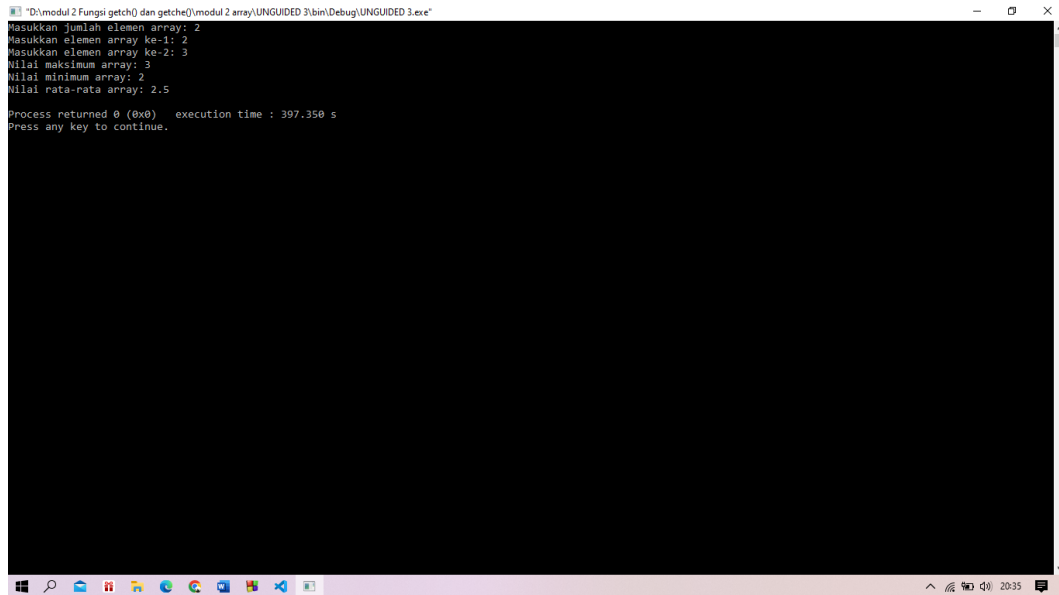
double avg = sum / n;

// Tampilkan hasil

cout << "Nilai maksimum array: " << max << endl;
cout << "Nilai minimum array: " << min << endl;
cout << "Nilai rata-rata array: " << avg << endl;

return 0;
}
```

## Screenshots output

A screenshot of a Windows console window titled "D:\modul 2 Fungsi getch() dan getch() \modul 2 array\UNGUIDED 3\bin\Debug\UNGUIDED 3.exe". The console displays the following text: "Masukkan jumlah elemen array: 2", "Masukkan elemen array ke-1: 2", "Masukkan elemen array ke-2: 3", "Nilai maksimum array: 3", "Nilai minimum array: 2", and "Nilai rata-rata array: 2.5". Below this, it shows "Process returned 0 (0x0) execution time : 397.350 s" and "Press any key to continue.". The Windows taskbar is visible at the bottom with various application icons and a system clock showing 20:35.

Deskripsi :

### 1. Header yang disertakan:

- `#include <iostream>`: Baris ini menyertakan pustaka `iostream` yang menyediakan fungsi penting untuk input (menggunakan `cin`) dan output (menggunakan `cout`) dalam C++.
- `#include <algorithm>`: Baris ini menyertakan pustaka `algorithm`, yang menawarkan berbagai algoritme untuk bekerja dengan struktur data seperti array. Namun, program khusus ini tidak menggunakan algoritme apa pun dari `<algorithm>` dalam bentuknya saat ini.

### 2. Ruang Nama:

- `using namespace std;`: Pernyataan ini membawa namespace `std` ke dalam cakupan. Ini memungkinkan Anda untuk menggunakan objek seperti `cin`, `cout`, `max`, dan `min` (jika disertakan dari `<algorithm>`) tanpa awalan `std::`.

### **3. Fungsi Utama:**

- `int main()`: Ini adalah titik masuk program tempat eksekusi dimulai. Tipe data `int` pada `return` menunjukkan program akan mengembalikan nilai integer, biasanya 0 untuk eksekusi yang berhasil.

### **4. Input Pengguna untuk Ukuran Array:**

- `int n;`: Baris ini mendeklarasikan variabel integer `n` untuk menyimpan jumlah elemen (ukuran) yang akan dimasukkan pengguna untuk array.
- `cout << "Masukkan jumlah elemen array: ";`  
Meminta pengguna untuk memasukkan jumlah elemen yang diinginkan (dalam Bahasa Indonesia).
- `cin >> n;`: Membaca input pengguna dan menyimpannya di variabel `n`.

### **5. Deklarasi Array:**

- `double arr[n];`: Baris ini mendeklarasikan array bilangan real desimal ganda (double-precision

floating-point) bernama arr dengan ukuran yang ditentukan oleh input pengguna (n). Ini membuat array yang dapat menampung n nilai double.

## **6. Input Pengguna untuk Elemen Array:**

- Loop for (bertingkat) ini berulang melalui setiap elemen array, meminta pengguna untuk memasukkan nilai untuk setiap elemen:
  - for (int i = 0; i < n; i++): Loop terluar berulang n kali, sekali untuk setiap elemen.
    - cout << "Masukkan elemen array ke-" << i + 1 << ": "; Meminta pengguna untuk memasukkan nilai elemen saat ini, menunjukkan posisinya (indeks) sebagai i + 1 (pengindeksan yang mudah dibaca manusia dimulai dari 1).
    - cin >> arr[i]; Membaca input pengguna dan menyimpannya di elemen yang sesuai dari array arr.
    - sum += arr[i]; Di dalam loop, baris ini mengakumulasi nilai semua elemen saat dimasukkan, menyimpan total yang sedang berjalan dalam variabel sum. Ini akan digunakan nanti untuk menghitung rata-rata.



## 7. Menghitung Maksimum dan Minimum:

- Bagian-bagian ini menemukan nilai maksimum dan minimum di dalam array:
  - **Maksimum:**
    - `double max = arr[0];`: Menginisialisasi variabel `double max` dan menetapkan nilai elemen pertama (`arr[0]`) ke dalamnya, dengan asumsi elemen tersebut awalnya terbesar.
    - `for (int i = 1; i < n; i++)`: Loop berulang dari elemen kedua (indeks 1) ke elemen terakhir (`n - 1`).
      - `if (arr[i] > max)`: Jika elemen saat ini (`arr[i]`) lebih besar dari nilai `max` saat ini, maka `max` diperbarui untuk menyimpan nilai yang lebih besar.
  - **Minimum:** (Logika serupa)
    - `double min = arr[0];`: Menginisialisasi variabel `double min` dan menetapkan nilai elemen pertama ke dalamnya, dengan asumsi elemen tersebut awalnya terkecil.

- `for (int i = 1; i < n; i++)`: Loop berulang dari elemen kedua ke terakhir.
- `if (arr[i] < min)`: Jika elemen saat ini lebih kecil dari nilai min saat ini, maka min diperbarui untuk menyimpan nilai yang lebih kecil.

## 8. Menghitung Rata-rata:

- `double avg = sum / n;` Baris ini menghitung nilai rata-rata elemen dalam array. Ini membagi total sum dengan jumlah elemen (`n`).

## 9. Mencetak Hasil:

- `cout << "Nilai maksimum array: " << max << endl;` Mencetak nilai maksimum yang dihitung dengan pesan deskriptif (dalam Bahasa Indonesia).
- `cout << "Nilai minimum array: " << min << endl;` Mencetak nilai minimum yang dihitung.
- `cout << "Nilai rata-rata array: " << avg << endl;` Mencetak nilai rata-rata yang dihitung.

#### D. kesimpulan

#### **Kesimpulan Materi Array**

**Array** adalah struktur data yang digunakan untuk menyimpan kumpulan data dengan tipe data yang sama. Array memungkinkan Anda untuk mengakses data secara efisien dengan menggunakan indeks.

#### **Berikut beberapa poin penting mengenai array:**

- **Tipe data:** Array dapat menyimpan berbagai jenis data, seperti integer, floating-point, karakter, dan string.
- **Ukuran:** Ukuran array ditentukan saat deklarasi dan tidak dapat diubah setelahnya.
- **Indeks:** Elemen array diakses dengan menggunakan indeks, yang dimulai dari 0.
- **Operasi:** Anda dapat melakukan berbagai operasi pada array, seperti menambahkan, menghapus, dan mencari elemen.

#### **Berikut beberapa keuntungan menggunakan array:**

- **Efisiensi:** Array memungkinkan Anda untuk mengakses data secara efisien dengan menggunakan indeks.
- **Kemudahan penggunaan:** Array mudah digunakan dan dipahami.

- **Fleksibilitas:** Array dapat digunakan untuk menyimpan berbagai jenis data.

**Berikut beberapa kekurangan menggunakan array:**

- **Ukuran:** Ukuran array tidak dapat diubah setelah deklarasi.
- **Ketidakefisienan:** Array dapat menjadi tidak efisien jika Anda perlu menambahkan atau menghapus elemen secara sering.

**Berikut beberapa aplikasi array:**

- Menyimpan daftar data
- Menyimpan tabel data
- Menerapkan algoritma
- Mengimplementasikan struktur data lain

## E. Referensi

<http://download.garuda.kemdikbud.go.id/article.php?article=1719115&val=10928&title=Algoritma%20Metode%20Pengurutan%20Bubble%20Sort%20dan%20Quick%20Sort%20Dalam%20Bahasa%20Pemrograman%20C>