

### Aturan Praktikum Struktur Data

1. **Akun GitHub:** Setiap praktikan wajib memiliki akun GitHub yang aktif dan digunakan selama praktikum berlangsung.
2. **Invite Collaborator:** Setiap praktikan diwajibkan untuk menambahkan collaborator di setiap repository
  - a. Asisten Praktikum: AndiniNH
  - b. Asisten Praktikum: 4ldiputra
3. **Repository Praktikum:** Setiap praktikan diwajibkan untuk membuat satu repository di GitHub yang akan digunakan untuk seluruh tugas dan laporan praktikum. Repository ini harus diatur dengan rapi dan sesuai dengan instruksi yang akan diberikan di lampiran.
4. **Penamaan Folder:** Penamaan folder dalam repository akan dibahas secara rinci di lampiran. Praktikan wajib mengikuti aturan penamaan yang telah ditentukan.

Nomor	Pertemuan	Penamaan
1	Pengalaman Bahasa C++ Bagian Pertama	01_Pengenalan_CPP_Bagian_1
2	Pengenalan Bahasa C++ Bagian Kedua	02_Pengenalan_CPP_Bagian_2
3	Abstract Data Type	03_Abstract_Data_Type
4	Single Linked List Bagian Pertama	04_Single_Linked_List_Bagian_1
5	Single Linked List Bagian Kedua	05_Single_Linked_List_Bagian_2
6	Double Linked List Bagian Pertama	06_Double_Linked_List_Bagian_1
7	Stack	07_Stack
8	Queue	08_Queue
9	Assessment Bagian Pertama	09_Assessment_Bagian_1
10	Tree Bagian Pertama	10_Tree_Bagian_1

11	Tree Bagian Kedua	11_Tree_Bagian_2
12	Asistensi Tugas Besar	12_Asistensi_Tugas_Besar
13	Multi Linked List	13_Multi_Linked_List
14	Graph	14_Graph
15	Assessment Bagian Kedua	15_Assessment_Bagian_2
16	Tugas Besar	16_Tugas_Besar

## 5. Jam Praktikum:

- Jam masuk praktikum adalah **1 jam lebih lambat** dari jadwal yang tercantum. Sebagai contoh, jika jadwal praktikum adalah pukul 06.30 - 09.30, maka aturan praktikum akan diatur sebagai berikut:
  - **06.30 - 07.30:** Waktu ini digunakan untuk **Tugas Praktikum dan Laporan Praktikum** yang dilakukan di luar laboratorium.
  - **07.30 - 09.30:** Sesi ini mencakup **tutorial, diskusi, dan kasus problem-solving**. Kegiatan ini berlangsung di dalam laboratorium dengan alokasi waktu sebagai berikut:
    - **60 menit pertama:** Tugas terbimbing.
    - **60 menit kedua:** Tugas mandiri.

6. **Pengumpulan Tugas Pendahuluan:** Tugas Pendahuluan (TP) wajib dikumpulkan melalui GitHub sesuai dengan format berikut:

**nama\_repo/nama\_pertemuan/TP\_Pertemuan\_Ke.md**

Sebagai contoh:

**STD\_Yudha\_Islalmi\_Sulistya\_XXXXXXXX/01\_Running\_Modul/TP\_01.md**

7. **Pengecekan Tugas Pendahuluan:** Pengumpulan laporan praktikum akan diperiksa **1 hari sebelum praktikum selanjutnya** dimulai. Pastikan tugas telah diunggah tepat waktu untuk menghindari sanksi.

## **8. Struktur Laporan Praktikum**

### **1. Cover :**

# **LAPORAN PRAKTIKUM**

## **Modul 1**

### **CODE BLOCKS IDE & PENGENALAN BAHASA C++ (BAGIAN PERTAMA)**



**Disusun Oleh:**

**Izzaty Zahara Br Barus – 2311104052**

**Kelas: SE 07-02**

**Dosen :**

**WAHYU ANDI SAPUTRA**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**PURWOKERTO**

**2024**

## **2. Tujuan**

1. Memahami penggunaan pointer dan alamat memori
2. Mengimplementasikan fungsi dan prosedur dalam program

## 2.1 Landasan Teori

Array merupakan kumpulan data dengan nama yang sama dan setiap elemen bertipe data sama. untuk mengakses setiap komponen / elemen array berdasarkan indeks dari setiap elemen.

### 2.1.1 Array Satu Dimensi

Adalah array yang hanya terdiri dari satu larik data saja . Cara pendeklarasian array satu dimensi:

```
tipe_data nama_var[ukuran]
```

Keterangan:

Tipe\_data → menyatakan jenis elemen array (int, char, float, dll). Ukuran → menyatakan jumlah maksimum array.

Contoh:

```
int nilai[10];
```

Menyatakan bahwa array nilai mengandung 10 elemen dan bertipe integer. Dalam C++ data array disimpan dalam memori pada lokasi yang berurutan. Elemen pertama memiliki indeks 0 dan elemen selanjutnya memiliki indeks 1 dan seterusnya. Jadi jika terdapat array dengan 5 elemen maka elemen pertama memiliki indeks 0 dan elemen terakhir memiliki indeks 4.

nama\_var[indeks]

nilai[5] → elemen ke-5 dari array nilai. Contoh memasukkan data ke dalam array :

```
nilai[4] = 90; /*memasukkan 90 ke dalam array nilai indeks ke-4*/
```

```
cin << nilai[4] /*membaca input-an dari keyboard*/
```

### 2.1.2 Array Dua Dimensi

Bentuk array dua dimensi ini mirip seperti tabel. Jadi array dua dimensi bisa digunakan untuk menyimpan data dalam bentuk tabel. Terbagi menjadi dua bagian, dimensi pertama dan dimensi kedua. Cara akses, deklarasi, inisialisasi, dan menampilkan data sama dengan array satu dimensi, hanya saja indeks yang digunakan ada dua.

Contoh:

```
int data_nilai[4][3];
```

```
nilai[2][0] = 10;
```

	0	1	2
0			
1			
2	10		
3			

Gambar 2-1 Ilustrasi Array Dua Dimensi

### 2.1.3 Array Berdimensi Banyak

Merupakan array yang mempunyai indeks banyak, lebih dari dua. Indeks inilah yang menyatakan dimensi array. Array berdimensi banyak lebih susah dibayangkan, sejalan dengan jumlah dimensi dalam array.

Cara deklarasi:

```
tipe_data nama_var[ukuran1][ukuran2]...[ukuran-N];
```

Contoh:

```
int data_rumit[4][6][6];
```

Array sebenarnya masih banyak pengembangannya untuk penyimpanan berbagai bentuk data, pengembangan array misalnya untuk array tak berukuran.

## 2.2 pointer

Pointer adalah sebuah variabel khusus yang menyimpan alamat memori dari suatu data, bukan menyimpan nilai data itu sendiri. Bayangkan pointer seperti sebuah peta atau penunjuk arah. Kalau biasanya variabel biasa seperti kotak yang menyimpan barang (misalnya angka 5), pointer adalah seperti selembar kertas yang menunjukkan lokasi kotak di mana angka 5 disimpan. Jadi pointer tidak menyimpan angkanya langsung, tapi tahu di mana angka itu berada di memori komputer. Dalam pemrograman, pointer sangat berguna karena memungkinkan kita untuk:

1. Mengakses data secara langsung di memori.
2. Berbagi data tanpa perlu menyalin seluruh isinya, cukup dengan mengirim alamatnya.
3. Mengelola struktur data dinamis seperti linked list, stack, atau tree yang membutuhkan akses langsung ke memori.

## 2.3 Fungsi

Fungsi merupakan blok dari kode yang dirancang untuk melaksanakan tugas khusus dengan tujuan:

1. Program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan.  
Program dibagi menjadi beberapa modul yang kecil.

2. Dapat mengurangi pengulangan kode (duplikasi kode) sehingga menghemat ukuran program.

Pada umumnya fungsi memerlukan masukan yang dinamakan sebagai parameter. Masukan ini selanjutnya diolah oleh fungsi. Hasil akhir fungsi berupa sebuah nilai (nilai balik fungsi).

## 2.4 Prosedur

Dalam C sebenarnya tidak ada prosedur, semua berupa fungsi, termasuk main() pun adalah sebuah fungsi. Jadi prosedur dalam C merupakan fungsi yang tidak mengembalikan nilai, biasa diawali dengan kata kunci void di depan nama prosedur. Bentuk umum sebuah prosedur:

```
void nama_prosedur (daftar_parameter) { blok  
pernyataan prosedur ;  
}
```

## 2.5 Parameter Fungsi

Parameter formal adalah variabel yang ada pada daftar parameter ketika mendefinisikan fungsi. Pada fungsi maks3() contoh diatas, a, b dan merupakan parameter formal.

```
float perkalian (float x, float y) { return (x*y);  
}
```

Pada contoh di atas x dan y adalah parameter formal.

Adapun parameter aktual adalah parameter (tidak selamanya menyatakan variabel) yang dipakai untuk memanggil fungsi.

```
x = perkalian(a, b);  
y = perkalian(10,30);
```

Dari pernyataan diatas a dan b merupakan parameter aktual, begitu pula 10 dan 30. parameter aktual tidak harus berupa variabel, melainkan bisa berupa konstanta atau ungkapan.

## 3. GUIDE

### 3.1 Array Satu Dimensi

code

```
#include <iostream>
```



```
#include <fstream>
#include <conio.h>

using namespace std;

int main(){
    int nilai [5] = {1, 2, 3, 4, 5};
    cout << nilai [0];
    cout << nilai [1];
    cout << nilai [2];
    cout << nilai [3];
    cout << nilai [4];

    int nilai [5] = {1,2,3,4,5};

    for(int i=0; i<5; i++){
        cout << nilai[i] << endl;
    }
}
```

### Output

```
PS D:\STD.02\Modul.2> cd 'd:\STD.02\Modul.2\guide\output'
PS D:\STD.02\Modul.2\guide\output> & .\'modul2.exe'
123451
2
3
4
5
PS D:\STD.02\Modul.2\guide\output> █
```

### Penjelasan

- **#include <iostream>**: Untuk menampilkan output ke layar.
- **using namespace std;**: Agar bisa pakai fungsi seperti cout tanpa menulis std::.
- **int nilai[5] = {1, 2, 3, 4, 5};**: Deklarasi array nilai dengan 5 elemen (1 sampai 5).

- `cout << nilai[0];` hingga `cout << nilai[4];`: Menampilkan setiap elemen array secara manual.
- **Loop for**: Cara efisien untuk menampilkan seluruh elemen array menggunakan perulangan.

### 3.2 Array Dua Dimensi

Code

```
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  //array 2 dimensi
7  int main(){
8      int nilai[3][4] = {
9          {1,2,3,4},
10         {5,6,7,8},
11         {9,10,11,12}
12     };
13
14     for(int i=0; i<3; i++){
15         for(int j=0; j<4; j++){
16             cout << nilai[i][j] << " ";
17         }
18         cout << endl;
19     }
20 }
```

Output

```
PS D:\STD.02\Modul.2> cd 'd:\STD.02\Modul.2\guide\output'
PS D:\STD.02\Modul.2\guide\output> & .\'modul2.exe'
1 2 3 4
5 6 7 8
9 10 11 12
PS D:\STD.02\Modul.2\guide\output> █
```

Penjelasan

- **int** **nilai[3][4]**  
Ini adalah deklarasi array **2 dimensi** bernama nilai dengan ukuran 3 baris dan 4 kolom. Array ini menyimpan 12 angka yang diorganisasikan dalam bentuk tabel 3x4.
- **for(int i = 0; i < 3; i++)**  
Ini adalah **loop luar** yang akan berjalan 3 kali, sekali untuk setiap **baris** array nilai.

- `for(int j = 0; j < 4; j++)`

Di dalam setiap baris, ada **loop dalam** yang berjalan 4 kali, sekali untuk setiap **kolom** di baris tersebut. Jadi ini akan mencetak semua elemen dalam satu baris.

- `cout << nilai[i][j] << " "`

Ini menampilkan setiap elemen array nilai dari baris ke-i dan kolom ke-j di layar, dengan spasi di antara elemen-elemen dalam baris yang sama.

- `cout << endl;`

Setelah satu baris array selesai ditampilkan, ini membuat **baris baru** sehingga elemen di baris berikutnya akan ditampilkan di bawahnya.

### 3.3 Pointer

Code

```
int main(){
    int x,y;
    int *px;
    x= 87;
    px = &x;
    y = *px;

    cout << "alamat x= " << &x << endl;
    cout << "isi px= " << px << endl;
    cout << "isi x= " << x << endl;
    cout << "Nilai yang ditunjuk px= " << *px << endl;
    cout << "nilai y = " << y << endl;
    getch();
}

int penjumlahan (int a, int b){
    return a+b;
}
```

Output

```
PS D:\STD.02\Modul.2\guide\output> cd
PS D:\STD.02\Modul.2\guide\output> & .
alamat x= 0x61ff04
isi px= 0x61ff04
isi x= 87
Nilai yang ditunjuk px= 87
nilai y = 87
```

Penjelasan

#### 1. Deklarasi variabel:

- **int x, y;** Deklarasi dua variabel x dan y bertipe integer.
  - **int \*px;** Deklarasi sebuah **pointer** px, yaitu variabel yang akan menyimpan alamat memori dari variabel lain.
2. **x = 87;** Nilai x diisi dengan angka 87.
3. **px = &x;**

Pointer px diisi dengan **alamat memori** dari variabel x. Tanda & digunakan untuk mengambil alamat memori dari variabel.

4. **y = \*px;**

Nilai yang disimpan di alamat yang ditunjuk oleh px (yaitu nilai dari x, 87) disalin ke variabel y. Tanda \* digunakan untuk mengakses **nilai yang ditunjuk** oleh pointer.

5. **cout <<**

Baris-baris ini menampilkan informasi ke layar:

- **alamat x=** : Menampilkan alamat memori di mana x disimpan (&x).
- **isi px=** : Menampilkan alamat memori yang disimpan dalam pointer px.
- **isi x=** : Menampilkan nilai dari x, yaitu 87.
- **Nilai yang ditunjuk px=** : Menampilkan nilai yang ditunjuk oleh px, yaitu nilai x (87).
- **nilai y =** : Menampilkan nilai y, yang juga 87 (karena disalin dari x melalui pointer).

6. **getch();**

Ini berfungsi untuk menunggu input dari pengguna sebelum menutup program (biasanya hanya digunakan di lingkungan Windows dengan conio.h).

#### 7. Fungsi penjumlahan

Ini adalah fungsi sederhana yang:

Menerima dua parameter a dan b bertipe integer.

Mengembalikan hasil penjumlahan dari a + b.

## 4. UNGUIDED

### 4.1 Taks 1

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

```
int main() {  
    int n;  
  
    cout << "Masukkan jumlah elemen array: ";  
    cin >> n;  
  
    int arr[n];  
  
    cout << "Masukkan elemen array: ";  
    for (int i = 0; i < n; i++) {  
        cin >> arr[i];  
    }  
    cout << "Data Array : ";  
    for (int i = 0; i < n; i++) {  
        cout << arr[i] << " ";  
    }  
    cout << endl;  
    cout << "Nomor Genap : ";  
    for (int i = 0; i < n; i++) {  
        if (arr[i] % 2 == 0) {  
            cout << arr[i] << ", ";  
        }  
    }  
    cout << endl;  
  
    cout << "Nomor Ganjil : ";  
    for (int i = 0; i < n; i++) {  
        if (arr[i] % 2 != 0) {  
            cout << arr[i] << ", ";  
        }  
    }  
}
```

```
cout << endl;

return 0;
}
```

#### Output

```
Masukkan jumlah elemen array: 10
Masukkan elemen array: 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,
PS D:\STD.02\Modul.2\Unguided\output> |
```

#### Penjelasan

- **int n;**  
Menyimpan jumlah elemen array yang akan dimasukkan oleh pengguna.
- **cin >> n;**  
Meminta pengguna untuk memasukkan jumlah elemen array.
- **int arr[n];**  
Deklarasi array arr dengan ukuran sesuai input pengguna (berisi n elemen).
- **Mengisi array:**

```
for (int i = 0; i < n; i++) {
    cin >> arr[i];
}
```

Mengisi elemen array dengan nilai yang dimasukkan oleh pengguna.

- **Menampilkan isi array:**

```
for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";
}
```

Menampilkan semua elemen array.

- **Menampilkan elemen genap:**

```
for (int i = 0; i < n; i++) {
    if (arr[i] % 2 == 0) {
        cout << arr[i] << ", ";
    }
}
```

```
}  
  
}
```

Menampilkan elemen yang bernilai genap dari array.

- Menampilkan elemen ganjil:

```
for (int i = 0; i < n; i++) {  
  
    if (arr[i] % 2 == 0) {  
        cout << arr[i] << ", ";  
    }  
}
```

- Menampilkan elemen yang bernilai ganjil dari array.

## 4.2 Taks 2

Code

```
1  #include <iostream>  
2  
3  using namespace std;  
4  
5  int main() {  
6      int dim1, dim2, dim3;  
7  
8      cout << "Masukkan ukuran dimensi pertama: ";  
9      cin >> dim1;  
10     cout << "Masukkan ukuran dimensi kedua: ";  
11     cin >> dim2;  
12     cout << "Masukkan ukuran dimensi ketiga: ";  
13     cin >> dim3;  
14  
15     int array_3d[dim1][dim2][dim3];  
16  
17     cout << "Masukkan Array 3D : " << endl;  
18     for (int i = 0; i < dim1; i++) {  
19         for (int j = 0; j < dim2; j++) {  
20             for (int k = 0; k < dim3; k++) {  
21                 cout << "Array 3D : [" << i << "]"[" << j << "]"[" << k << "] : ";  
22                 cin >> array_3d[i][j][k];  
23             }  
24         }  
25     }  
26  
27     cout << " Array 3D Adalah : " << endl;  
28     for (int i = 0; i < dim1; i++) {  
29         for (int j = 0; j < dim2; j++) {  
30             for (int k = 0; k < dim3; k++) {  
31                 cout << "Array 3D : [" << i << "]"[" << j << "]"[" << k << "] - " << array_3d[i][j][k] << endl;  
32             }  
33             cout << endl;  
34         }  
35         cout << endl;  
36     }  
37  
38     return 0;  
39 }
```

Output

```
PS D:\STD.02\Modul.2\Unguided\output> & .\'taks2.  
Masukkan ukuran dimensi pertama: 1  
Masukkan ukuran dimensi kedua: 2  
Masukkan ukuran dimensi ketiga: 3  
Masukkan Array 3D :  
Array 3D : [0][0][0] : 1  
Array 3D : [0][0][1] : 2  
Array 3D : [0][0][2] : 3  
Array 3D : [0][1][0] : 4  
Array 3D : [0][1][1] : 5  
Array 3D : [0][1][2] : 6  
Array 3D Adalah :  
Array 3D : [0][0][0] - 1  
Array 3D : [0][0][1] - 2  
Array 3D : [0][0][2] - 3  
  
Array 3D : [0][1][0] - 4  
Array 3D : [0][1][1] - 5  
Array 3D : [0][1][2] - 6
```

### Penjelasan

1. Input Ukuran Array 3D:
  - Program meminta pengguna memasukkan ukuran untuk tiga dimensi (panjang, lebar, tinggi) array.
2. Deklarasi Array 3D:
  - Array 3D dibuat berdasarkan ukuran yang telah dimasukkan oleh pengguna.
  - Contoh: Jika pengguna memasukkan dimensi 2x2x2, array akan memiliki 2 blok dengan 2 baris dan 2 kolom di setiap bloknya.
3. Pengisian Nilai Array 3D:
  - Program menggunakan tiga perulangan for bertingkat untuk meminta input dari pengguna dan mengisi setiap elemen dari array 3D berdasarkan indeks (i, j, k).
4. Menampilkan Nilai Array 3D:
  - Setelah semua nilai diisi, program kembali menggunakan tiga perulangan for bertingkat untuk menampilkan nilai dari setiap elemen array 3D dengan format [i][j][k].

Contoh alur program:

1. Pengguna memasukkan ukuran 2x2x2 untuk array.
2. Program meminta input untuk setiap elemen array (total 8 elemen).
3. Program menampilkan setiap nilai yang sudah dimasukkan, sesuai dengan indeks posisi dalam array 3D.



### 4.3 Taks 3

Code

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int n;
7
8      cout << "Masukkan ukuran array: ";
9      cin >> n;
10
11     int arr[n];
12
13     for (int i = 0; i < n; i++) {
14         cout << "Masukkan nilai array ke-" << i + 1 << ": ";
15         cin >> arr[i];
16     }
17
18     int max = arr[0];
19     int min = arr[0];
20     double sum = 0;
21
22     for (int i = 0; i < n; i++) {
23         if (arr[i] > max) {
24             max = arr[i];
25         }
26         if (arr[i] < min) {
27             min = arr[i];
28         }
29         sum += arr[i];
30     }
31
32     double avg = sum / n;
33
34     cout << "Nilai Maksimum: " << max << endl;
35     cout << "Nilai Minimum: " << min << endl;
36     cout << "Nilai Rata-rata: " << avg << endl;
37
38     return 0;
39 }
```

## Output

```
PS D:\STD.02\Modul.2\Unguided\output> & .\'taks3.exe'  
Masukkan ukuran array: 10  
Masukkan nilai array ke-1: 2  
Masukkan nilai array ke-2: 1  
Masukkan nilai array ke-3: 4  
Masukkan nilai array ke-4: 5  
Masukkan nilai array ke-5: 6  
Masukkan nilai array ke-6: 7  
Masukkan nilai array ke-7: 2  
Masukkan nilai array ke-8: 5  
Masukkan nilai array ke-9: 7  
Masukkan nilai array ke-10: 9  
Nilai Maksimum: 9  
Nilai Minimum: 1  
Nilai Rata-rata: 4.8  
PS D:\STD.02\Modul.2\Unguided\output> █
```

## Pejelasan

1. Program pertama-tama meminta pengguna memasukkan ukuran array (n).
2. Array arr dengan ukuran n kemudian dibuat.
3. Pengguna diminta memasukkan nilai-nilai untuk setiap elemen array.
4. Program memulai variabel max dan min dengan nilai dari elemen pertama array.
5. Program lalu memproses array untuk:
  - Mencari nilai maksimum dengan membandingkan setiap elemen.
  - Mencari nilai minimum dengan membandingkan setiap elemen.
  - Menjumlahkan seluruh elemen untuk menghitung rata-rata.
6. Setelah semua elemen diproses, program menampilkan nilai maksimum, minimum, dan rata-rata.

## 5. Kesimpulan

Praktikum ini fokus pada pengenalan dasar-dasar bahasa pemrograman C++, termasuk pemahaman tentang array satu dimensi, dua dimensi, dan penggunaan pointer. Tujuan utama dari praktikum adalah memberikan pemahaman mengenai penggunaan pointer, alamat memori, fungsi, dan prosedur dalam C++, serta implementasinya dalam program sederhana. Praktikan diharapkan bisa menerapkan konsep-konsep ini untuk memecahkan masalah pemrograman melalui tugas yang



melibatkan berbagai struktur data seperti linked list, stack, dan queue. Praktikum juga melibatkan penggunaan GitHub untuk mengelola tugas dan laporan secara kolaboratif serta mematuhi aturan penamaan dan pengumpulan yang telah ditetapkan.