

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL II  
PENGENALAN BAHASA C++**



**Disusun oleh :**

Junadil Muqorobin (103112400281)

**Dosen**

Fahrudin Mukti Wibowo S.Kom., M.Eng

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## **A. Dasar Teori**

Array merupakan struktur data yang digunakan untuk menyimpan sekumpulan data dengan tipe yang sama dalam satu variabel. Setiap elemen array dapat diakses menggunakan indeks, di mana indeks pertama dimulai dari 0. Pada C++, array dapat berbentuk satu dimensi, dua dimensi, maupun multidimensi. Array satu dimensi hanya memiliki satu baris data, sedangkan array dua dimensi berbentuk seperti tabel dengan baris dan kolom. Array multidimensi digunakan untuk menyimpan data dalam beberapa lapisan indeks. Keunggulan array adalah kemampuannya dalam menyimpan data yang jumlahnya banyak dengan akses elemen yang cepat dan terstruktur. Menurut dokumen “Arrays and Pointers” di Stony Brook University, elemen array disimpan secara kontigu dalam memori, sehingga alamat elemen ke-*i* dapat dihitung dari alamat awal plus offset tertentu (Shebuti Rayana, 2017)

Pointer adalah variabel yang menyimpan alamat memori dari variabel atau objek lain. Dengan pointer, program dapat mengakses dan memanipulasi data secara langsung melalui alamat tersebut. Dalam C++, deklarasi pointer menggunakan tanda \*, sedangkan operator & digunakan untuk memperoleh alamat suatu variabel. Presentasi “Pointers in C++” menjelaskan bahwa pointer menyimpan alamat variabel lain dan memungkinkan dereferensi untuk mengakses nilai yang ditunjuk (Tarfa Hamed, 2021).

Fungsi merupakan blok kode terstruktur yang dirancang untuk menjalankan tugas tertentu dalam suatu program. Kelebihan dari penggunaan fungsi yaitu program menjadi lebih terstruktur, mudah dipahami, dan mengurangi pengulangan kode yang sama. Fungsi

dapat menerima parameter input dan dapat menghasilkan return. Selain fungsi yang memberikan return, ada pula prosedur atau fungsi bertipe void. Prosedur digunakan untuk melaksanakan tugas tertentu tanpa memberikan nilai balik, sehingga berguna untuk intruksi yang hanya butuh hasil atau tidak butuh return.

Selain itu, ketika array dilewatkan ke fungsi, terjadi “array decaying”, artinya array secara otomatis diperlakukan sebagai pointer ke elemen pertamanya. Sebagai hasilnya, ukuran array tidak ikut disertakan ke dalam parameter, sehingga sering diperlukan parameter tambahan untuk menyampaikan ukuran array tersebut (misalnya int size). Pendekatan ini dijelaskan dalam artikel “Passing Array Pointer to Function in C++” bahwa array yang dilewatkan ke fungsi akan menjadi pointer dan informasi ukuran harus dipertahankan secara terpisah (GeeksforGeeks, 2025).

## B. Guided

### 1. Implementasi pointer

```
#include <iostream>

using namespace std;

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

    //menampilkan informasi
    cout << "Alamat x      : " << &x << endl;
    cout << "Isi px         : " << px << endl;
    cout << "Isi x           : " << x << endl;
    cout << "Nilai *px        : " << *px << endl;
```

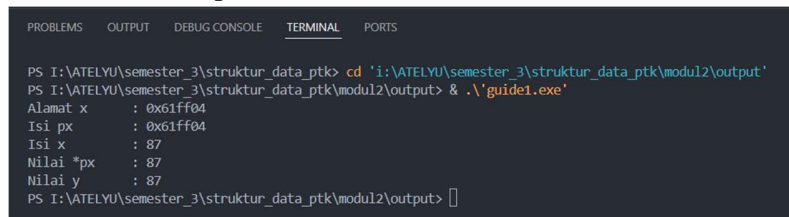
```

    cout << "Nilai y      : " << y << endl;

    return 0;
}

```

### Screenshoot Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS I:\ATELYU\semester_3\struktur_data_ptk> cd 'i:\ATELYU\semester_3\struktur_data_ptk\modul2\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> & .\guide1.exe
Alamat x      : 0x61ff04
Isi px       : 0x61ff04
Isi x        : 87
Nilai *px    : 87
Nilai y      : 87
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> 

```

### Deskripsi:

Program ini digunakan untuk menampilkan informasi dari beberapa variable menggunakan pointer, dengan langkah kerja sebagai berikut:

- Deklarasi variabel dan pointer  
Variabel x dan y bertipe integer, sedangkan px adalah pointer yang menunjuk ke variabel bertipe integer (int \*px);.
- Inisialisasi Nilai dan Alamat:
  - Variabel x diberi nilai 87.
  - Pointer px diisi dengan alamat dari variabel x menggunakan operator &.
  - Variabel y kemudian diisi dengan nilai yang ditunjuk oleh px menggunakan operator dereferensi \*.
- Menampilkan informasi pointer  
Program akan menampilkan alamat variabel x (&x), isi pointer px yaitu alamat x, nilai variabel x, kemudian nilai yang ditunjuk oleh pointer \*px, serta nilai dari variabel y.

## 2. Implementasi Array multidimensi

```
#include <iostream>
#define MAX 5

using namespace std;

int main() {
    int i, j;
    float nilai[MAX]; // ARRAY 1 DIMENSI
    static int nilai_tahun[MAX][MAX] = {
        {0,2,2,0,0},
        {0,1,1,1,0},
        {0,3,3,3,0},
        {0,4,0,0,4},
        {5,0,0,0,5}
    };

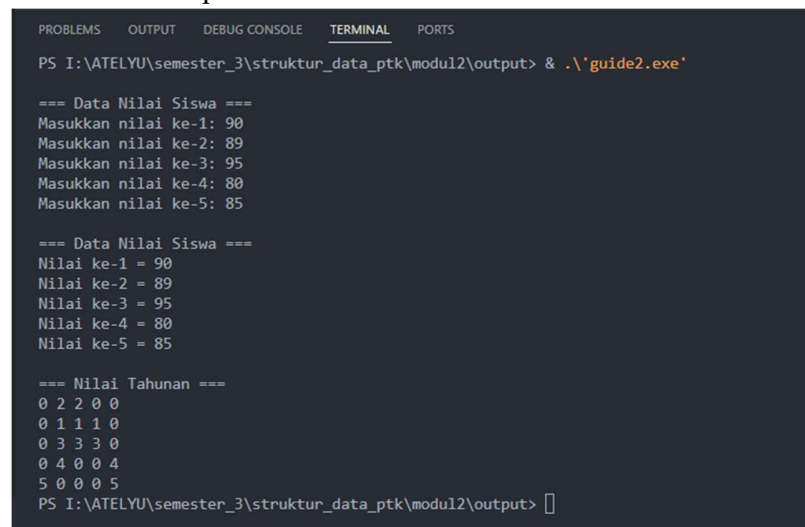
    //INPUT DATA ARRAY 1 DIMENSI
    cout << "\n=== Data Nilai Siswa ===\n";
    for ( i = 0; i < MAX; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << ": ";
        cin >> nilai[i];
    }

    //MENAMPILKAN ISI ARRAY 1 DIMENSI
    cout << "\n=== Data Nilai Siswa ===\n";
    for ( i = 0; i < MAX; i++)
    {
        cout << "Nilai ke-" << i + 1 << " = " << nilai[i] << endl;
    }

    //MENAMPILKAN ISI ARRAY 2 DIMENSI
    cout << "\n=== Nilai Tahunan ===\n";
    for ( i = 0; i < MAX; i++){
        for (j = 0; j < MAX; j++){
            cout << nilai_tahun[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
    return 0;
}
```

### Screenshot Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> & .\'guide2.exe'

=== Data Nilai Siswa ===
Masukkan nilai ke-1: 90
Masukkan nilai ke-2: 89
Masukkan nilai ke-3: 95
Masukkan nilai ke-4: 80
Masukkan nilai ke-5: 85

=== Data Nilai Siswa ===
Nilai ke-1 = 90
Nilai ke-2 = 89
Nilai ke-3 = 95
Nilai ke-4 = 80
Nilai ke-5 = 85

=== Nilai Tahunan ===
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
0 4 0 0 4
5 0 0 0 5
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> 
```

### Deskripsi:

Program tersebut berfungsi untuk input kedalam array dan menampilkannya menggunakan array, dengan langkah kerja sebagai berikut:

- Array satu dimensi  
Array nilai digunakan untuk menyimpan data bertipe float sebanyak 5 elemen (MAX = 5). Program meminta pengguna memasukkan nilai ke dalam array melalui perulangan for.
- Array dua dimensi  
Array nilai\_tahun merupakan array dua dimensi dengan ukuran  $5 \times 5$  yang sudah diinisialisasi secara statis. Array ini menggambarkan data berbentuk tabel atau matriks.
- Menampilkan data  
Data dari array satu dimensi ditampilkan kembali satu per satu. Setelah itu, array dua dimensi nilai\_tahun ditampilkan dalam bentuk baris dan kolom dengan dua perulangan bersarang (for).

### 3. Implementasi Fungsi

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

// deklarasi prototype fungsi
int maks3(int a, int b, int c);

int main(){
    int x, y, z;
    cout << "Masukkan nilai bilangan ke-1 : ";
    cin >> x;
    cout << "Masukkan nilai bilangan ke-2 : ";
    cin >> y;
    cout << "Masukkan nilai bilangan ke-3 : ";
    cin >> z;

    cout << "Nilai maksimumnya adalah = " << maks3(x, y, z);
    return 0;
}

//badan fungsi
int maks3(int a, int b, int c){
    int temp_max = a;
    if (b > temp_max)
    {
        temp_max = b;
    }
    if (c > temp_max)
    {
        temp_max = c;
    }
    return temp_max;
}
```

#### Screenshoot Output:



The screenshot shows a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS I:\ATELYU\semester_3\struktur_data_ptk> cd 'i:\ATELYU\semester_3\struktur_data_ptk\modul2\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> & .\guide3.exe
Masukkan nilai bilangan ke-1 : 90
Masukkan nilai bilangan ke-2 : 95
Masukkan nilai bilangan ke-3 : 80
Nilai maksimumnya adalah = 95
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> 
```

#### Deskripsi:

Program ini berfungsi untuk mencari nilai maximal dari tiga input bilangan, dan menggunakan fungsi dengan parameter dan nilai return, dengan langkah kerja sebagai berikut:

- Deklarasi fungsi (prototype)  
Fungsi maks3(int a, int b, int c) dideklarasikan di awal program dan berfungsi mengembalikan nilai bertipe int.
- Input data  
Pengguna memasukkan tiga bilangan melalui cin dan masuk ke variable x, y, z.
- Pemanggilan fungsi  
Nilai-nilai input (x, y, z) dikirim sebagai parameter aktual ke fungsi maks3().
- Isi fungsi
  - Variabel sementara temp\_max diisi dengan nilai a.
  - Fungsi membandingkan nilai b dan c untuk menentukan nilai terbesar menggunakan pernyataan if.
  - Nilai maksimum dikembalikan melalui perintah return temp\_max;.
- Menampilkan hasil  
Setelah fungsi dipanggil dan parameter masuk kedalam fungsi, kemudian dilakukan pengkondisian. Fungsi akan mengembalikan nilai



yang kemudian menjadi output “Nilai maksimumnya adalah = ...”.

#### 4. Implementasi Prosedur

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

// PROTOTYPE PROSEDUR
void tulis(int x);

int main(){
    int jum;
    cout << "Jumlah baris kata : " ;
    cin >> jum;
    tulis(jum);
    return 0;
}

// BADAN PROSEDUR
void tulis(int x){
    for (int i = 0; i < x; i++)
    {
        cout << "Baris ke-" << i + 1 << endl;
    }
}
```

Screenshot Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS I:\ATELYU\semester_3\struktur_data_ptk> cd 'i:\ATELYU\semester_3\struktur_data_ptk\modul2\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> & .\guide4.exe
Jumlah baris kata : 5
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-4
Baris ke-5
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\output> 
```

Deskripsi:

Program ini berfungsi untuk menampilkan baris kata dengan jumlah sesuai yang dimasukkan oleh

pengguna. Dengan memanfaatkan prosedur. Berikut adalah langkah kerjanya:

- Deklarasi prosedur  
Prosedur tulis(int x) dideklarasikan sebagai fungsi void, artinya tidak memiliki nilai balik.
- Input data  
pengguna melakukan input jumlah baris dengan menggunakan cin dan masuk ke dalam variable jum.
- Pemanggilan prosedur  
Nilai jum dikirim ke prosedur tulis() sebagai parameter
- Isi prosedur  
Terdapat perulangan for untuk mencetak teks cout "Baris ke-" sebanyak x kali. Setiap iterasi menampilkan nomor baris sesuai urutannya.
- Selesai dengan program menampilkan jumlah baris ke-x sejumlah baris yang dimasukkan pengguna.

## C. Unguided

### 1. Implementasi array dua dimensi

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

int main() {
    const int MHS = 5;
    const int MK = 3;

    string nama[MHS];
    float nilai[MHS][MK];
    float rata[MHS];

    for (int i = 0; i < MHS; i++) {
        cout << "Nama mahasiswa ke-" << i+1 << ": ";
        cin >> nama[i];
        float total = 0;
        for (int j = 0; j < MK; j++) {
            cout << "  Nilai MK" << j+1 << ": ";
            cin >> nilai[i][j];
            total += nilai[i][j];
        }
        rata[i] = total / MK;
        cout << endl;
    }

    int terbaik = 0;
    for (int i = 1; i < MHS; i++) {
        if (rata[i] > rata[terbaik]) {
            terbaik = i;
        }
    }

    cout << "====="
<< endl;
    cout << "\n          Data Nilai Mahasiswa          \n";
    cout << "====="
<< endl;
    for (int i = 0; i < MHS; i++) {
```

```

        cout << nama[i] << " | ";
        for (int j = 0; j < MK; j++) {
            cout << nilai[i][j] << " ";
        }
        cout << "| Rata-rata: " << rata[i];
        if (i == terbaik) cout << " <-- Terbaik";
        cout << endl;
    }
    cout << "=====
<< endl;
    cout << "\nMahasiswa terbaik : " << nama[terbaik] << ", dengan
rata-rata " << rata[terbaik] << endl;

    return 0;
}

```

### Screenshot Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS I:\ATELYU\semester_3\struktur_data_ptk> cd 'i:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> & .\soall.exe
Nama mahasiswa ke-1: Azka
  Nilai MK1: 90
  Nilai MK2: 100
  Nilai MK3: 75

Nama mahasiswa ke-2: Zami
  Nilai MK1: 85
  Nilai MK2: 90
  Nilai MK3: 80

Nama mahasiswa ke-3: Zala
  Nilai MK1: 80
  Nilai MK2: 85
  Nilai MK3: 75

Nama mahasiswa ke-4: Zaki
  Nilai MK1: 90
  Nilai MK2: 85
  Nilai MK3: 95

Nama mahasiswa ke-5: Kiki
  Nilai MK1: 80
  Nilai MK2: 85
  Nilai MK3: 100

=====
Data Nilai Mahasiswa
=====
Azka | 90 100 75 | Rata-rata: 88.3333
Zami | 85 90 80 | Rata-rata: 85
Zala | 80 85 75 | Rata-rata: 80
Zaki | 90 85 95 | Rata-rata: 90 <-- Terbaik
Kiki | 80 85 100 | Rata-rata: 88.3333
=====

Mahasiswa terbaik : Zaki, dengan rata-rata 90
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output>

```

### Deskripsi:

Program array dua dimensi diatas digunakan untuk menghitung nilai rata-rata dari 5 mahasiswa yang kemudian juga dapat menentukan siapa yang memiliki nilai terbaik. Program berjalan dengan Langkah kerja sebagai berikut:

- Deklarasi konstanta dan variable  
`const int MHS = 5;` menentukan jumlah mahasiswa. `const int MK = 3;` menentukan jumlah mata kuliah. Array `nama[MHS]` menyimpan nama mahasiswa. Array dua dimensi `nilai[MHS][MK]` menyimpan nilai setiap mahasiswa untuk tiap mata kuliah. Array `rata[MHS]` menyimpan hasil rata-rata setiap mahasiswa.
- Input data mahasiswa  
 Program menggunakan perulangan `for` untuk meminta input nama dan nilai dari setiap mahasiswa. Di dalam perulangan, nilai setiap mata kuliah dijumlahkan untuk kemudian dihitung rata-ratanya dengan rumus:  

$$\text{rata}[i] = \text{total} / \text{MK};$$
 Proses ini dilakukan untuk semua mahasiswa yaitu pada iterasi  $i = 0$  sampai  $i < \text{MHS}$
- Menentukan mahasiswa terbaik  
 Variabel terbaik digunakan untuk menyimpan indeks mahasiswa dengan nilai rata-rata tertinggi. Dengan perulangan `for`, setiap nilai rata-rata dibandingkan menggunakan:  

```
if (rata[i] > rata[terbaik]) {
    terbaik = i;
}
```

 Yang kemudian mahasiswa dengan rata-rata tertinggi akan disimpan sebagai terbaik.
- Menampilkan data nilai  
 Setelah perhitungan selesai dilakukan, program menampilkan semua data yang sudah dimasukkan dan dihitung dengan beberapa `cout` yang berbentuk tabel, terdapat nama mahasiswa, nilai tiap mata kuliah, serta nilai rata-rata. Selain itu diberikan tanda “← Terbaik” untuk mahasiswa dengan rata-rata tertinggi. Setelah tabel juga ada output nama mahasiswa dengan nilai tertingginya.

## 2. implementasi pointer

```
#include <iostream>
using namespace std;
int main() {
    int N;

    cout << "Masukkan jumlah elemen array (maks 100): ";
    cin >> N;

    int arr[100];
    int *ptr = arr;

    cout << "\nMasukkan " << N << " bilangan bulat:\n";
    for (int i = 0; i < N; i++) {
        cout << "Elemen ke-" << i + 1 << ": ";
        cin >> *(ptr + i);
    }

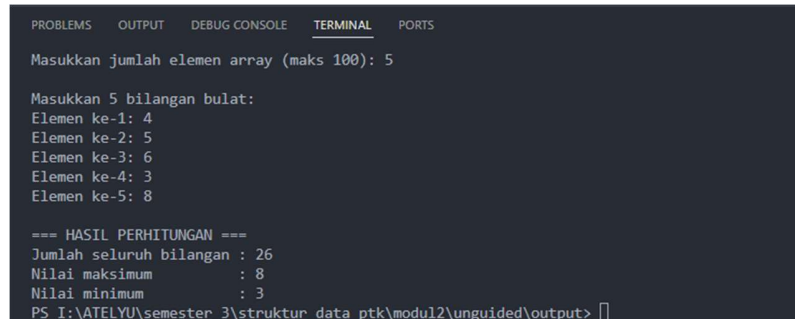
    int jumlah = 0;
    int maksimum = *ptr;
    int minimum = *ptr;

    for (int i = 0; i < N; i++) {
        int nilai = *(ptr + i);
        jumlah += nilai;
        if (nilai > maksimum) maksimum = nilai;
        if (nilai < minimum) minimum = nilai;
    }

    cout << "\n=== HASIL PERHITUNGAN ===\n";
    cout << "Jumlah seluruh bilangan : " << jumlah << endl;
    cout << "Nilai maksimum          : " << maksimum << endl;
    cout << "Nilai minimum           : " << minimum << endl;

    return 0;
}
```

### Screenshot Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukkan jumlah elemen array (maks 100): 5

Masukkan 5 bilangan bulat:
Elemen ke-1: 4
Elemen ke-2: 5
Elemen ke-3: 6
Elemen ke-4: 3
Elemen ke-5: 8

=== HASIL PERHITUNGAN ===
Jumlah seluruh bilangan : 26
Nilai maksimum          : 8
Nilai minimum           : 3
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output>
```

### Deskripsi:

Program diatas adalah perhitungan 5 bilangan bulat dengan menerapkan array. Program dapat menjumlahkan seluruh bilangan bulat, mencari nilai maksimum dan mencari nilai minimum. Bilangan bulat dapat dimasukkan langsung oleh pengguna. Langkah kerja program diatas sebagai berikut:

- deklarasi variabel dan pointer □  
int arr[100]; membuat array statis dengan kapasitas maksimum 100 elemen. int \*ptr = arr; men - deklarasikan pointer ptr dan menunjuk ke elemen pertama dari array arr. Penggunaan pointer ini memungkinkan pengaksesan elemen array menggunakan bentuk \*(ptr + i).
- input elemen array melalui pointer  
Program meminta pengguna memasukkan jumlah elemen array (N) terlebih dahulu. Dengan perulangan for, setiap input disimpan menggunakan pointer aritmatika:  
cin >> \*(ptr + i);  
pointer tersebut berfungsi untuk menyimpan data di Alamat arr[i] tanpa menyebutkan indeks secara langsung.
- inisialisasi pointer untuk perhitungan  
Nilai awal maksimum dan minimum diambil dari elemen pertama, dengan pointer berikut:  
int maksimum = \*ptr;  
int minimum = \*ptr;

- perhitungan jumlah, maks, dan min  
Perulangan kedua digunakan untuk membaca kembali setiap elemen array melalui pointer (\*(ptr + i)). Program menjumlahkan semua nilai (jumlah += nilai) serta membandingkan untuk mencari nilai tertinggi dan terendah:  
if (nilai > maksimum) maksimum = nilai;  
if (nilai < minimum) minimum = nilai;
- menampilkan hasil perhitungan  
Setelah semua elemen diproses, program menampilkan hasil akhir berupa jumlah seluruh bilangan, nilai maksimum, dan nilai minimum.

### 3. implementasi fungsi

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

float hitungRataRata(int nilai[], int n) {
    int total = 0;
    for (int i = 0; i < n; i++) {
        total += nilai[i];
    }
    return (float) total / n;
}

void cariNilaiTertinggiTerendah(int nilai[], int n, int &maks, int &min) {
    maks = nilai[0];
    min = nilai[0];
    for (int i = 1; i < n; i++) {
        if (nilai[i] > maks) maks = nilai[i];
        if (nilai[i] < min) min = nilai[i];
    }
}

int main() {
    int N;
```



```

    cout << "Masukkan jumlah siswa: ";
    cin >> N;

    int nilai[N];

    cout << "\nMasukkan nilai ujian untuk " << N << " siswa:\n";
    for (int i = 0; i < N; i++) {
        cout << "Nilai siswa ke-" << i + 1 << ": ";
        cin >> nilai[i];
    }

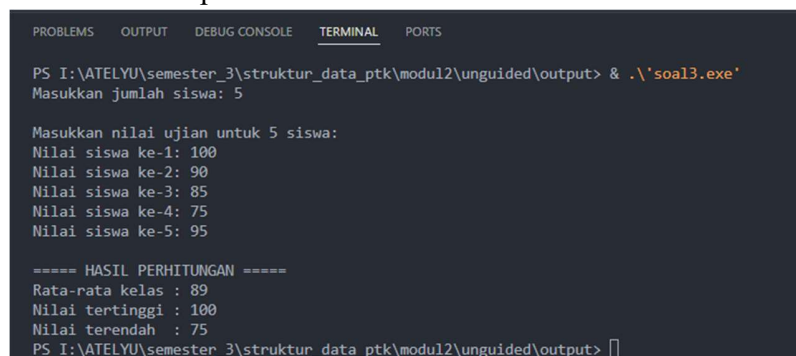
    float rataRata = hitungRataRata(nilai, N);
    int maks, min;
    cariNilaiTertinggiTerendah(nilai, N, maks, min);

    cout << "\n===== HASIL PERHITUNGAN =====\n";
    cout << "Rata-rata kelas : " << rataRata << endl;
    cout << "Nilai tertinggi : " << maks << endl;
    cout << "Nilai terendah : " << min << endl;

    return 0;
}

```

### Screenshot Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> & .\'soal3.exe'
Masukkan jumlah siswa: 5

Masukkan nilai ujian untuk 5 siswa:
Nilai siswa ke-1: 100
Nilai siswa ke-2: 90
Nilai siswa ke-3: 85
Nilai siswa ke-4: 75
Nilai siswa ke-5: 95

===== HASIL PERHITUNGAN =====
Rata-rata kelas : 89
Nilai tertinggi : 100
Nilai terendah : 75
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> 

```

### Deskripsi:

Program di atas berfungsi untuk menghitung rata-rata nilai tertinggi dari jumlah siswa yang dimasukkan pengguna. Pengguna dapat menentukan banyaknya siswa yang ingin di hitung nilainya, kemudian memasukkan nilai masing-masing siswa. Langkah kerja program sebagai berikut:

- Deklarasi variable dan penggunaan array  
Array nilai[] dikirim ke fungsi sebagai parameter formal agar dapat diproses di dalam fungsi tanpa perlu mendeklarasikan ulang di main(). Selain array, ukuran array (int n) juga dikirim agar fungsi tahu jumlah elemen yang harus diolah.
- Fungsi hitungRataRata()  
Fungsi ini menerima dua parameter: array nilai[] dan jumlah data n. Di dalam fungsi, seluruh elemen array dijumlahkan menggunakan perulangan for, kemudian hasilnya dibagi dengan n untuk memperoleh rata-rata:  
return (float) total / n;  
fungsi ini mengembalikan nilai bertipe float, karena hasil perhitungan rata-rata nilai bisa berupa bilangan decimal.
- Fungsi cariNilaiTertinggiTerendah()  
Fungsi ini digunakan untuk mencari nilai maksimum dan minimum dari data array. Parameter formal yang digunakan adalah int nilai[], int n, serta dua parameter referensi int &maks dan int &min. Parameter referensi (&) memungkinkan fungsi mengubah langsung nilai variabel maks dan min di fungsi main(). Di dalam fungsi, perulangan for digunakan untuk membandingkan setiap elemen array:  
if (nilai[i] > maks) maks = nilai[i];  
if (nilai[i] < min) min = nilai[i];
- Output program  
Pada program utama yaitu main(), pengguna memasukkan jumlah siswa N, dan nilai ujian setiap siswa yang disimpan dalam array nilai[N]. setelah input, kemudian program memanggil fungsi hitungRataRata() dan cariNilaiTertinggiTerendah() untuk melakukan perhitungan. Kemudian hasil perhitungan ditampilkan menggunakan cout.

#### 4. Impelementasi prosedur

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

void tampilSegitiga(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
}

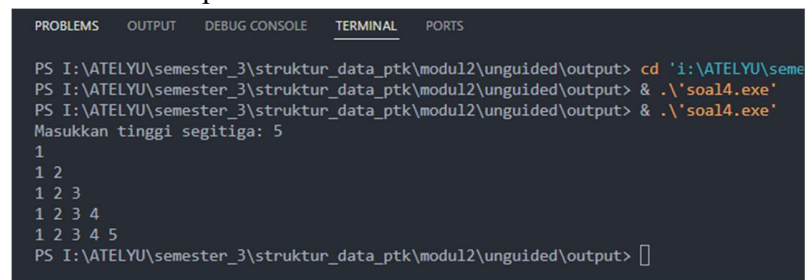
int main() {
    int n;

    cout << "Masukkan tinggi segitiga: ";
    cin >> n;

    tampilSegitiga(n);

    return 0;
}
```

Screenshot Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> cd 'i:\ATELYU\seme
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> & .\'soal4.exe'
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> & .\'soal4.exe'
Masukkan tinggi segitiga: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
PS I:\ATELYU\semester_3\struktur_data_ptk\modul2\unguided\output> []
```

Deskripsi:

Program tersebut digunakan untuk membuat pola segitiga dari bilangan yang dimasukkan pengguna. Jika pengguna memasukkan angka 5 maka tinggi segitiga tersebut adalah 5 baris. Langkah kerja program sebagai berikut:

- Deklarasi prosedur  
Prosedur tampilSegitiga(int n) dibuat dengan tipe void, artinya tidak mengembalikan nilai. Prosedur hanya menjalankan perintah untuk menampilkan pola angka di layar.
- Logika pencetakan segitiga  
Menggunakan perulangan bersarang yaitu:  
Perulangan luar berfungsi untuk mengatur baris dari 1 sampai baris ke-n  
Perulangan dalam berfungsi untuk menampilkan angka dari 1 hingga i setiap barisnya  
Setelah tiap baris selesai, kemudian jalankan cout << endl; untuk berpinda baris berikutnya.
- Input pengguna  
Pengguna diminta memasukkan sebuah bilangan bulat n yang mewakili tinggi segitiga angka.
- Pemanggilan prosedur  
Pada fungsi main(), prosedur tampilSegitiga(n) dipanggil setelah input diterima. Prosedur tersebut akan menghasilkan tampilan pola angka ke layar.

#### **D. Kesimpulan**

Dari praktikum yang telah dilakukan, dapat disimpulkan bahwa pemahaman dasar pemrograman C++ sangat penting sebelum mempelajari materi struktur data yang lebih mendalam. Melalui latihan *guided*, saya belajar bagaimana menggunakan perulangan, percabangan, operator, serta tipe data dasar dan struct secara bertahap. Sedangkan pada latihan *unguided*, saya dapat menggabungkan konsep-konsep tersebut ke dalam kasus nyata, seperti membuat operasi sederhana, mengonversi angka ke bentuk tulisan, dan mencetak pola mirror dengan perulangan bersarang. Praktikum ini menunjukkan bahwa setiap konsep memiliki perannya masing-masing. Perulangan membantu mengurangi pengulangan kode, percabangan memungkinkan program mengambil keputusan, operator digunakan dalam berbagai operasi logika maupun matematika, dan struct mempermudah pengelompokan data yang berhubungan.

Secara keseluruhan, praktikum ini menjadi dasar yang kokoh untuk melangkah ke materi berikutnya. Pemahaman input-output, percabangan, perulangan, operator, dan struktur data sederhana akan sangat membantu ketika mempelajari algoritma serta implementasi struktur data yang lebih rumit di pertemuan selanjutnya.

## E. Referensi

GeeksforGeeks. (2025). *Passing array pointer to function in C++*. GeeksforGeeks. <https://www.geeksforgeeks.org/cpp/pass-array-pointer-as-function-argument-in-cpp/>

Hamed, T. (2021, February). *Pointers in C++*. Department of Computer Science, College of Computer Science and Mathematics, University of Mosul. <https://doi.org/10.13140/RG.2.2.31168.25604>

Rayana, S. (2017, Fall). *Arrays and pointers* [Lecture slides]. CSE 230 Intermediate Programming in C and C++, Stony Brook University. <http://www3.cs.stonybrook.edu/~cse230/>

