

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 2  
PENGENALAN BAHASA C++  
(BAGIAN KEDUA)**



**Disusun Oleh :**

NAMA : Azzahra Farelika Esti Ning Tyas

NIM : 103112430023

**Dosen**

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

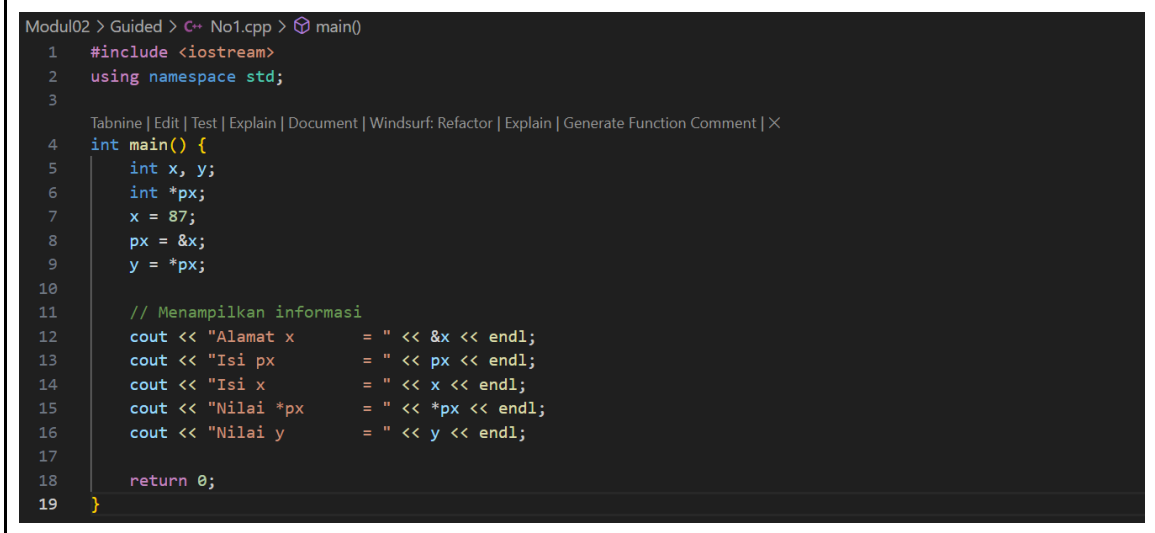
## A. Dasar Teori

Array dalam C++ adalah kumpulan elemen dengan tipe data yang sama yang disimpan secara berurutan di memori dan diakses menggunakan indeks. Dengan array, kita bisa menyimpan banyak data sejenis tanpa membuat banyak variabel. Misalnya, `int angka[5];` dapat menyimpan lima bilangan bertipe `int` yang diakses melalui indeks 0 sampai 4. Array satu dimensi merupakan bentuk paling sederhana dari array, berupa deretan data linear seperti daftar, contohnya `int nilai[5] = {1, 2, 3, 4, 5};` di mana setiap elemen diakses dengan indeks, seperti `nilai[2]` yang bernilai 3. Jenis array ini sering digunakan untuk menyimpan data sederhana seperti daftar nilai atau harga barang. Sementara itu, array dua dimensi memiliki dua indeks dan dapat dibayangkan seperti tabel dengan baris dan kolom. Contohnya `int matriks[2][3] = {{1, 2, 3}, {4, 5, 6}};` yang menyimpan dua baris dan tiga kolom, dan setiap elemen diakses dengan dua indeks, misalnya `matriks[1][2]` bernilai 6. Array dua dimensi biasanya digunakan untuk menyimpan data berbentuk tabel atau matriks angka.

Sedangkan pointer adalah variabel khusus yang digunakan untuk menyimpan alamat memori dari variabel lain. Dengan pointer, kita dapat mengakses atau mengubah nilai suatu variabel melalui alamatnya, bukan langsung lewat namanya. Misalnya pada kode `int x = 10; int *px = &x;`, variabel `px` menyimpan alamat dari `x`. Saat kita menulis `*px`, artinya kita mengambil nilai yang disimpan di alamat tersebut, yaitu 10. Pointer sering digunakan untuk menghemat memori, membuat program lebih efisien, dan mendukung struktur data dinamis seperti array dan linked list.

## B. Guided (berisi screenshot source code & output program disertai penjelasannya)

### Guided 1



```
Modul02 > Guided > C++ No1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x, y;
6      int *px;
7      x = 87;
8      px = &x;
9      y = *px;
10
11     // Menampilkan informasi
12     cout << "Alamat x      = " << &x << endl;
13     cout << "Isi px       = " << px << endl;
14     cout << "Isi x        = " << x << endl;
15     cout << "Nilai *px    = " << *px << endl;
16     cout << "Nilai y      = " << y << endl;
17
18     return 0;
19 }
```

## Screenshots Output

```
PS D:\StrukturData> cd "d:\StrukturData\Modul02\Guided\" ; if ($?) { g++ No1.cpp -o No1 } ; if ($?) { .\No1 }  
Alamat x      = 0xbffd24  
Isi px        = 0xbffd24  
Isi x         = 87  
Nilai *px     = 87  
Nilai y       = 87
```

## Deskripsi:

Program ini menunjukkan bagaimana pointer bekerja di C++. Pertama, variabel x diberi nilai 87. Lalu pointer px digunakan untuk menyimpan alamat dari variabel x menggunakan `px = &x;`. Artinya, px “menunjuk” ke x. Setelah itu, `y = *px;` berarti kita mengambil nilai yang disimpan di alamat yang ditunjuk px, yaitu nilai x, sehingga y juga menjadi 87. Saat program dijalankan, hasil yang ditampilkan menunjukkan bahwa px berisi alamat x, sedangkan \*px dan x memiliki nilai yang sama. Intinya, pointer adalah cara untuk menyimpan dan mengakses nilai suatu variabel lewat alamatnya di memori komputer.

## Guided 2

```
Modul02 > Guided > C++ No2.cpp > main()
1  #include <iostream>
2  #define MAX 5 // Ukuran array
3  using namespace std;
4
5  int main () {
6      int i, j;
7      float nilai[MAX]; // Array satu dimensi
8      static int nilai_tahun[MAX][MAX] = { // Array dua dimensi
9          {0, 2, 2, 0, 0},
10         {0, 1, 1, 1, 0},
11         {0, 3, 3, 3, 0},
12         {4, 4, 0, 0, 4},
13         {5, 0, 0, 0, 5}
14     };
15
16     // Input data array satu dimensi
17     cout << "=== Input Nilai Siswa ===\n";
18     for (i = 0; i < MAX; i++) {
19         cout << "Masukan nilai ke-" << i + 1
20             << ": ";
21         cin >> nilai[i];
22     }
23
24     //Menampilkan isi array satu dimensi
25     cout << "\n=== Data Nilai Siswa ===\n";
26     for (i = 0; i < MAX; i++) {
27         cout << "Nilai ke-" << i + 1 << " = "
28             << nilai[i]
29             << endl;
30     }
31
32     // Menampilkan isi array dua dimensi
33     cout << "\n=== Nilai Tahunan ===\n";
34     for (i = 0; i < MAX; i++) {
35         for (j = 0; j < MAX; j++) {
36             cout << nilai_tahun[i][j] << " ";
37         }
38         cout << endl;
39     }
40
41     return 0;
42 }
```

## Screenshots Output

```
PS D:\StrukturData\Modul02\Guided> cd "d:\StrukturData\Modul02\Guided\" ; if ($?) { g++ No2.cpp -o No2 } ; if ($?) { .\No2 }
=== Input Nilai Siswa ===
Masukan nilai ke-1: 60
Masukan nilai ke-2: 70
Masukan nilai ke-3: 80
Masukan nilai ke-4: 90
Masukan nilai ke-5: 100

=== Data Nilai Siswa ===
Nilai ke-1 = 60
Nilai ke-2 = 70
Nilai ke-3 = 80
Nilai ke-4 = 90
Nilai ke-5 = 100

=== Nilai Tahunan ===
0 2 2 0 0
0 1 1 1 0
0 3 3 3 0
4 4 0 0 4
5 0 0 0 5
```

## Deskripsi:

Program di atas menunjukkan penggunaan array satu dimensi dan dua dimensi dalam C++. Pertama, program mendefinisikan konstanta MAX bernilai 5 yang digunakan sebagai ukuran array. Variabel nilai merupakan array satu dimensi yang menyimpan 5 nilai bertipe float, sedangkan nilai\_tahun adalah array dua dimensi berukuran 5×5 yang telah diisi dengan nilai tetap (static). Program meminta pengguna untuk memasukkan lima nilai siswa ke dalam array nilai, kemudian menampilkan kembali semua nilai tersebut satu per satu. Setelah itu, program mencetak isi dari array nilai\_tahun dalam bentuk tabel 5 baris dan 5 kolom. Secara sederhana, program ini memperlihatkan bagaimana array satu dimensi digunakan untuk menyimpan data linear, dan array dua dimensi digunakan untuk menyimpan data berbentuk matriks atau tabel.

## Guided 3

```
Modul02 > Guided > C++ No3.cpp > maks3(int, int, int)
1  #include <iostream>
2  using namespace std;
3
4  // Mendeklarasikan prototype fungsi
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X
5  int maks3(int a, int b, int c);
6
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate Function Comment | X
7  int main() {
8      int x, y, z;
9      cout << "Masukan nilai bilangan ke-1    = ";
10     cin >> x;
11     cout << "Masukan nilai bilangan ke-2    = ";
12     cin >> y;
13     cout << "Masukan nilai bilangan ke-3    = ";
14     cin >> z;
15
16     cout << "\nNilai maksimumnya adalah      = "
17     << maks3(x, y, z);
18     return 0;
19 }
20
21 // Badan fungsi
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X
22 int maks3(int a, int b, int c) {
23     int temp_max = a;
24     if (b > temp_max)
25         temp_max = b;
26     if (c > temp_max)
27         temp_max = c;
28     return temp_max;
29 }
```

## Screenshots Output

```
PS D:\StrukturData> cd "d:\StrukturData\Modul02\Guided\" ; if ($?) { g++ No3.cpp -o No3 } ; if ($?) { .\No3 }
• Masukan nilai bilangan ke-1    = 3
Masukan nilai bilangan ke-2    = 2
Masukan nilai bilangan ke-3    = 1

Nilai maksimumnya adalah      = 3
```

Deskripsi:

Program di atas digunakan untuk menentukan nilai terbesar dari tiga bilangan yang dimasukkan oleh pengguna. Pertama, program meminta input tiga angka dan menyimpannya dalam variabel x, y, dan z. Kemudian, program memanggil fungsi maks3(x, y, z) untuk mencari nilai maksimumnya. Fungsi maks3 membandingkan ketiga bilangan tersebut dengan menggunakan struktur if, dan menyimpan nilai terbesar dalam variabel temp\_max. Setelah semua perbandingan selesai, fungsi mengembalikan nilai terbesar tersebut ke fungsi main(), lalu hasilnya ditampilkan di layar. Program ini menunjukkan bagaimana fungsi digunakan untuk memisahkan logika perhitungan dari program utama, sehingga kode menjadi lebih rapi dan mudah dipahami.

#### Guided 4

```
Modul02 > Guided > C++ No4.cpp > tulis(int)
1  #include <iostream>
2  using namespace std;
3
4  // Prototype prosdur
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X
5  void tulis(int x);
6
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate Function Comment | X
7  int main() {
8      int jum;
9      cout << "Jumlah baris kata = ";
10     cin >> jum;
11     tulis(jum);
12     return 0;
13 }
14
15 // Badan prosedur
   Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X
16 void tulis(int x) {
17     for (int i = 0; i < x; i++) {
18         cout << "Baris ke-" << i + 1 << endl;
19     }
20 }
```

#### Screenshots Output

```
PS D:\StrukturData\Modul02\Guided> cd "d:\StrukturData\Modul02\Guided"; if ($?) { g++ No4.cpp -o No4 }; if ($?) { .\No4 }
Jumlah baris kata = 3
Baris ke-1
Baris ke-2
Baris ke-3
```

Deskripsi:

Program di atas berfungsi untuk menampilkan sejumlah baris teks ke layar sesuai dengan angka yang dimasukkan oleh pengguna. Pertama, pengguna diminta mengisi nilai jum, yaitu jumlah baris yang ingin ditampilkan. Nilai tersebut kemudian dikirim ke prosedur tulis(int x). Di dalam prosedur ini, digunakan perulangan for yang akan menampilkan tulisan "Baris ke-1", "Baris ke-2", dan seterusnya hingga mencapai jumlah yang diminta. Program ini menunjukkan cara menggunakan prosedur (fungsi tanpa nilai balik) untuk menjalankan tugas tertentu, yaitu menampilkan teks berulang berdasarkan input.

pengguna.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```
Modul02 > Unguided > C- Soal1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      string nama[5];
6      float nilai[5][3];
7      float rata[5];
8      int terbaik = 0;
9
10     cout << "-----Input Data Mahasiswa-----" << endl;
11
12     // Input nama dan nilai
13     for (int i = 0; i < 5; i++) {
14         cout << "\nNama mahasiswa ke-" << i + 1 << ": ";
15         cin >> nama[i];
16
17         float total = 0;
18         for (int j = 0; j < 3; j++) {
19             cout << "    Masukkan nilai mata kuliah ke-" << j + 1 << ": ";
20             cin >> nilai[i][j];
21             total += nilai[i][j];
22         }
23         rata[i] = total / 3;
24     }
25
26     // Cari rata-rata tertinggi
27     for (int i = 1; i < 5; i++) {
28         if (rata[i] > rata[terbaik]) {
29             terbaik = i;
30         }
31     }
32
33     // Hasil akhir
34     cout << "\n----- Daftar Nilai Mahasiswa -----" << endl;
35     cout << "Nama\tNilai1\tNilai2\tNilai3\tRata-rata" << endl;
36
37     for (int i = 0; i < 5; i++) {
38         cout << nama[i] << "\t";
39         for (int j = 0; j < 3; j++) {
40             cout << nilai[i][j] << "\t";
41         }
42         cout << rata[i];
43         if (i == terbaik)
44             cout << " <-- Terbaik";
45         cout << endl;
46     }
47
48     cout << "\nMahasiswa dengan rata-rata tertinggi adalah: "
49         << nama[terbaik] << " (" << rata[terbaik] << ")" << endl;
50
51     return 0;
52 }
```

## Screenshots Output

```
-----Input Data Mahasiswa-----
Nama mahasiswa ke-1:
PS D:\StrukturData\Modul02\Unguided> cd "d:\StrukturData\Modul02\Unguided\" ; if ($?) { g++ Soal1.cpp -o Soal1 } ; if ($?) { .\Soal1 }
-----Input Data Mahasiswa-----

Nama mahasiswa ke-1: Azzahra
Masukkan nilai mata kuliah ke-1: 100
Masukkan nilai mata kuliah ke-2: 99
Masukkan nilai mata kuliah ke-3: 98

Nama mahasiswa ke-2: Fafa
Masukkan nilai mata kuliah ke-1: 97
Masukkan nilai mata kuliah ke-2: 96
Masukkan nilai mata kuliah ke-3: 95

Nama mahasiswa ke-3: Fara
Masukkan nilai mata kuliah ke-1: 94
Masukkan nilai mata kuliah ke-2: 93
Masukkan nilai mata kuliah ke-3: 92

Nama mahasiswa ke-4: Farelika
Masukkan nilai mata kuliah ke-1: 91
Masukkan nilai mata kuliah ke-2: 90
Masukkan nilai mata kuliah ke-3: 89

Nama mahasiswa ke-5: Tyas
Masukkan nilai mata kuliah ke-1: 88
Masukkan nilai mata kuliah ke-2: 87
Masukkan nilai mata kuliah ke-3: 86

----- Daftar Nilai Mahasiswa -----
Nama  Nilai1  Nilai2  Nilai3  Rata-rata
Azzahra 100    99      98      99 <-- Terbaik
Fafa    97      96      95      96
Fara    94      93      92      93
Farelika 91      90      89      90
Tyas    88      87      86      87

Mahasiswa dengan rata-rata tertinggi adalah: Azzahra (99)
```

## Deskripsi:

Program di atas digunakan untuk mengolah dan menampilkan data nilai mahasiswa menggunakan array dua dimensi. Pertama, program meminta input nama lima mahasiswa dan tiga nilai mata kuliah untuk masing-masing mahasiswa. Setiap kali nilai dimasukkan, program menghitung rata-rata dari tiga mata kuliah tersebut dan menyimpannya dalam array rata. Setelah semua data dimasukkan, program mencari mahasiswa dengan rata-rata tertinggi menggunakan perulangan dan menyimpan posisinya dalam variabel terbaik. Terakhir, program menampilkan tabel berisi nama, nilai, dan rata-rata setiap mahasiswa, serta menandai mahasiswa dengan rata-rata tertinggi dengan tulisan "<-- Terbaik". Program ini membantu memahami konsep array satu dan dua dimensi, perulangan bersarang, dan logika pencarian nilai maksimum dalam C++.



## Unguided 2

```
Modul02 > Unguided > C++ Soal2.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate Function Comment | X
5  int main() {
6      int N;
7      cout << "Masukkan jumlah elemen array (N): ";
8      cin >> N;
9
10     // Alokasi array secara dinamis
11     int *arr = new int[N];
12
13     // Input nilai melalui pointer
14     cout << "\nMasukkan " << N << " bilangan bulat:\n";
15     for (int i = 0; i < N; i++) {
16         cout << "Elemen ke-" << i + 1 << ": ";
17         cin >> *(arr + i);
18     }
19
20     // Inisialisasi nilai awal
21     int jumlah = 0;
22     int maks = *arr;
23     int min = *arr;
24
25     // Menghitung jumlah, maksimum, dan minimum dengan pointer
26     for (int *p = arr; p < arr + N; p++) {
27         jumlah += *p;
28         if (*p > maks)
29             maks = *p;
30         if (*p < min)
31             min = *p;
32     }
33
34     // Tampilkan hasil
35     cout << "\n----- Hasil Perhitungan -----";
36     cout << "Jumlah seluruh bilangan : " << jumlah << endl;
37     cout << "Nilai maksimum : " << maks << endl;
38     cout << "Nilai minimum : " << min << endl;
39
40     // Hapus memori yang dialokasikan
41     delete[] arr;
42
43     return 0;
44 }
```

## Screenshots Output

```
• Masukkan jumlah elemen array (N): 3
Masukkan 3 bilangan bulat:
Elemen ke-1: 11
Elemen ke-2: 22
Elemen ke-3: 33

----- Hasil Perhitungan -----
Jumlah seluruh bilangan : 66
Nilai maksimum : 33
Nilai minimum : 11
```

## Deskripsi:

Program di atas digunakan untuk mengolah sejumlah bilangan bulat menggunakan pointer dan array dinamis. Pertama, program meminta pengguna memasukkan banyaknya elemen (N), lalu mengalokasikan memori secara dinamis menggunakan `new int[N]`. Setiap elemen array diisi melalui pointer aritmatika (`*(arr + i)`), bukan dengan cara indeks biasa. Setelah itu, program menghitung jumlah seluruh bilangan, serta mencari nilai maksimum dan minimum dengan menggunakan perulangan pointer (`for (int *p = arr; p < arr + N; p++)`). Hasil perhitungan kemudian ditampilkan ke layar, dan memori yang telah digunakan dibebaskan dengan `delete[] arr`. Program ini menunjukkan cara kerja pointer aritmatika dan manajemen memori dinamis dalam C++.

## Unguided 3

```
Modul02 > Unguided > C++ Soal3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Menghitung rata-rata nilai
5  float hitungRata(int nilai[], int n) {
6      int total = 0;
7      for (int i = 0; i < n; i++) {
8          total += nilai[i];
9      }
10     return static_cast<float>(total) / n;
11 }
12
13 // Mencari nilai maksimal dan minimal
14 void cariMinMax(int nilai[], int n, int &maks, int &min) {
15     maks = nilai[0];
16     min = nilai[0];
17     for (int i = 1; i < n; i++) {
18         if (nilai[i] > maks)
19             maks = nilai[i];
20         if (nilai[i] < min)
21             min = nilai[i];
22     }
23 }
24
25 int main() {
26     int N;
27     cout << "Masukkan jumlah siswa: ";
28     cin >> N;
29
30     int nilai[N];
31     cout << "\nMasukkan nilai ujian untuk " << N << " siswa:\n";
32     for (int i = 0; i < N; i++) {
33         cout << "Nilai siswa ke-" << i + 1 << ": ";
34         cin >> nilai[i];
35     }
36
37     // Panggil fungsi
38     float rata = hitungRata(nilai, N);
39     int maks, min;
40     cariMinMax(nilai, N, maks, min);
41
42     // Tampilkan hasil
43     cout << "\n----- Hasil Perhitungan ----- \n";
44     cout << "Rata-rata kelas : " << rata << endl;
45     cout << "Nilai tertinggi : " << maks << endl;
46     cout << "Nilai terendah : " << min << endl;
47
48     return 0;
49 }
```

## Screenshots Output

```
PS D:\StrukturData> cd "d:\StrukturData\Modul02\Unguided\" ; if ($?) { g++ Soal3.cpp -o Soal3 } ; if ($?) { .\Soal3 }
Masukkan jumlah siswa: 3

Masukkan nilai ujian untuk 3 siswa:
Nilai siswa ke-1: 99
Nilai siswa ke-2: 88
Nilai siswa ke-3: 77

----- Hasil Perhitungan -----
Rata-rata kelas : 88
Nilai tertinggi : 99
Nilai terendah : 77
```

## Deskripsi:

Program di atas digunakan untuk menghitung rata-rata, nilai tertinggi, dan nilai terendah dari sejumlah nilai siswa menggunakan fungsi. Pertama, program meminta pengguna memasukkan jumlah siswa (N) dan nilai ujian mereka ke dalam array nilai[]. Fungsi hitungRata() menjumlahkan semua elemen array lalu membaginya dengan jumlah siswa untuk mendapatkan rata-rata kelas. Fungsi cariMinMax() mencari nilai tertinggi dan terendah dengan membandingkan setiap elemen dalam array, dan hasilnya dikirim kembali ke program utama melalui parameter referensi. Terakhir, program menampilkan

rata-rata, nilai maksimum, dan nilai minimum ke layar. Program ini memperlihatkan penggunaan fungsi dengan parameter array untuk memisahkan logika perhitungan agar lebih terstruktur dan mudah dipahami.

## Unguided 4

```
Modul02 > Unguided > C++ Soal4.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Prosedur
5  void tampilSegitiga(int n) {
6      for (int i = 1; i <= n; i++) {
7          for (int j = 1; j <= i; j++) {
8              cout << j << " ";
9          }
10         cout << endl;
11     }
12 }
13
14 int main() {
15     int n;
16     cout << "Masukkan jumlah baris (n): ";
17     cin >> n;
18
19     cout << "\n----- Pola Segitiga Angka ----- \n";
20     tampilSegitiga(n);
21
22     return 0;
23 }
```

## Screenshots Output

```
PS D:\StrukturData\Modul02\Unguided> cd "d:\StrukturData\Modul02\Unguided\"; if ($?) { g++ Soal4.cpp -o Soal4 }; if ($?) { .\Soal4 }
Masukkan jumlah baris (n): 5

----- Pola Segitiga Angka -----
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

## Deskripsi:

Program di atas digunakan untuk menampilkan pola segitiga angka sesuai dengan jumlah baris yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai  $n$ , program memanggil prosedur `tampilSegitiga(int n)`. Di dalam prosedur tersebut, terdapat dua perulangan bersarang perulangan luar menentukan jumlah baris, sedangkan perulangan dalam mencetak angka dari 1 hingga nilai  $i$  pada setiap baris. Hasilnya, terbentuk pola angka seperti segitiga, misalnya jika  $n = 5$ , maka akan ditampilkan barisan angka dari 1 hingga 1 2 3 4 5. Program ini menunjukkan penggunaan prosedur (fungsi tanpa nilai balik) dan perulangan bersarang untuk menampilkan pola sederhana di C++.

#### D. Kesimpulan

Berdasarkan praktikum dan pembahasan program di atas, dapat disimpulkan bahwa array, pointer, fungsi, dan prosedur merupakan konsep dasar penting dalam pemrograman C++. Array memungkinkan penyimpanan banyak data dengan tipe yang sama secara terstruktur, baik dalam bentuk satu dimensi maupun dua dimensi. Pointer digunakan untuk mengakses dan memanipulasi data melalui alamat memori, sehingga memberikan fleksibilitas dalam pengelolaan data dan memori secara dinamis. Sementara itu, fungsi dan prosedur membantu memecah program menjadi bagian-bagian kecil yang lebih mudah dipahami, dipelihara, dan digunakan kembali. Melalui percobaan yang dilakukan, mahasiswa dapat memahami bagaimana array dan pointer bekerja dalam menyimpan serta mengolah data, dan bagaimana fungsi serta prosedur digunakan untuk memisahkan logika program agar lebih terstruktur. Dengan memahami konsep ini, pemrogram dapat membuat program yang lebih efisien, rapi, dan modular.

#### E. Referensi

<https://www.geeksforgeeks.org/cpp/cpp-arrays/>

<https://www.programiz.com/cpp-programming/arrays>

[https://www.w3schools.com/cpp/cpp\\_arrays\\_multi.asp?utm\\_source=chatgpt.com](https://www.w3schools.com/cpp/cpp_arrays_multi.asp?utm_source=chatgpt.com)

<https://www.programiz.com/cpp-programming/pointers>

<https://www.tutorialspoint.com/cplusplus>