

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :

NAMA : Taufik Hafit Zakaria

NIM : 103112430093

Dosen

WAHYU ANDI SAPUTRA

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

A. Dasar Teori

Dasar teori Modul 2 membahas tiga konsep lanjutan dalam bahasa C++ yang menjadi fondasi penting dalam implementasi struktur data, yaitu Array, Pointer, dan Fungsi.

1. Array

Array adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen bertipe data identik dalam satu variabel. Setiap elemen diakses menggunakan indeks numerik yang dimulai dari 0. Array memungkinkan data sejenis dikelola secara terstruktur, baik dalam bentuk satu dimensi (seperti daftar) maupun multi-dimensi (seperti tabel atau matriks).

2. Pointer

Pointer merupakan variabel khusus yang nilainya adalah alamat memori dari variabel lain. Penggunaan pointer memungkinkan manipulasi data secara tidak langsung melalui alamatnya, yang sangat esensial untuk alokasi memori dinamis dan operasi pada struktur data yang kompleks. Dua operator kunci yang digunakan adalah & (address-of) untuk mendapatkan alamat suatu variabel dan * (dereference) untuk mengakses nilai yang tersimpan di alamat yang ditunjuk oleh pointer.

3. Fungsi dan Mekanisme Pelewatan Parameter

Fungsi adalah blok kode yang dirancang untuk melakukan tugas spesifik, bertujuan untuk membuat program lebih modular, terstruktur, dan efisien dengan mengurangi duplikasi kode. Fungsi yang tidak mengembalikan nilai disebut prosedur dan menggunakan tipe void.

Cara sebuah data (parameter) dilewatkan ke dalam fungsi akan memengaruhi apakah variabel asli dapat diubah atau tidak. Terdapat tiga mekanisme utama:

- **Call by Value:** Hanya nilai dari variabel yang disalin ke fungsi. Perubahan di dalam fungsi tidak akan memengaruhi variabel aslinya.
- **Call by Pointer:** Alamat memori dari variabel yang dilewatkan. Fungsi dapat memodifikasi nilai variabel asli secara tidak langsung melalui alamat tersebut.
- **Call by Reference:** Referensi (alias) dari variabel yang dilewatkan, memungkinkan fungsi untuk memodifikasi variabel asli secara langsung dengan sintaksis yang lebih sederhana.

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

Guided 1

```
Struktur Data > LAPRAK2 > C++ guided1.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6 int main(){
7     int x, y;
8     int *px;
9     x = 87;
10    px = &x;
11    y = *px;
12
13    cout << "Alamat x    = " << &x << endl;
14    cout << "Isi px      = " << px << endl;
15    cout << "Isi x        = " << x << endl;
16    cout << "Nilai *px     = " << *px << endl;
17    cout << "Nilai y      = " << y << endl;
18
19    return 0;
20 }
```

Screenshots Output

```
Problems Terminal ... Code - LAPRAK2 + @ @ ... < >
PS C:\programing> cd "c:\programing\Struktur Data\LAPRAK2\" ; if ($?) { g++ guided1.cpp -o guided1 } ; if ($?) { .\guided1 }
Alamat x    = 0x15a97ff890
Isi px      = 0x15a97ff890
Isi x       = 87
Nilai *px   = 87
Nilai y     = 87
PS C:\programing\Struktur Data\LAPRAK2>
```

Deskripsi:

Program ini merupakan demonstrasi fundamental dari penggunaan pointer dalam bahasa C++. Tujuan utamanya adalah untuk mengilustrasikan bagaimana sebuah pointer (px) dapat menyimpan alamat memori dari variabel lain (x) menggunakan operator & (address-of).

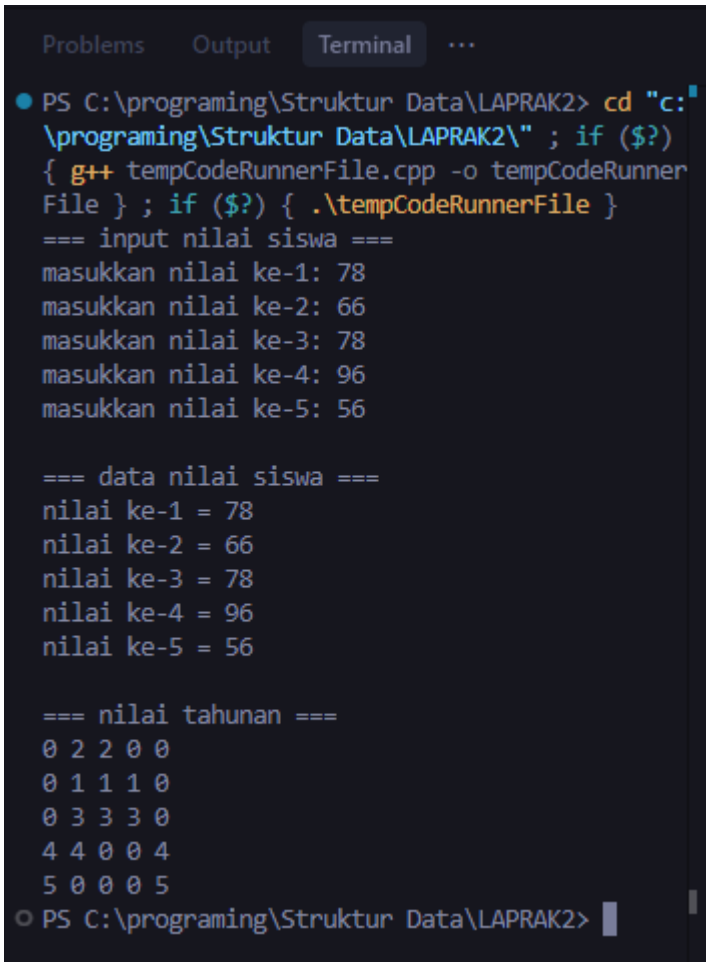
Selanjutnya, program ini menunjukkan cara mengakses nilai yang tersimpan di alamat tersebut secara tidak langsung dengan menggunakan operator * (dereference), di mana

nilai yang ditunjuk oleh px kemudian disalinkan ke variabel lain (y). Output program secara eksplisit menampilkan alamat memori variabel x, isi dari pointer px, serta nilai dari variabel x dan y untuk membuktikan bahwa pointer berhasil menunjuk dan mengakses data dengan benar.

Guided 2

```
Struktur Data > LAPRAK2 > C++ guided2.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #define MAX 5
6 using namespace std;
7
8 int main(){
9     int i, j;
10    float nilai[MAX];
11    static int nilai_tahun[MAX][MAX] = {
12        {0, 2, 2, 0, 0},
13        {0, 1, 1, 1, 0},
14        {0, 3, 3, 3, 0},
15        {4, 4, 0, 0, 4},
16        {5, 0, 0, 0, 5}
17    };
18    cout << "=== input nilai siswa ===\n";
19    for (i = 0; i < MAX; i++){
20        cout << "masukkan nilai ke-" << i + 1 << ": ";
21        cin >> nilai[i];
22    }
23    cout << "\n=== data nilai siswa ===\n";
24    for (i = 0; i < MAX; i++){
25        cout << "nilai ke-" << i + 1 << " = "
26        << nilai[i]
27        << endl;
28    }
29    cout << "\n=== nilai tahunan ===\n";
30    for (i = 0; i < MAX; i++) {
31        for (j = 0; j < MAX; j++){
32            cout << nilai_tahun[i][j] << " ";
33        }
34        cout << endl;
35    }
36    return 0;
37 }
38
```

Screenshots Output

A screenshot of a terminal window showing the execution of a C++ program. The terminal has tabs for 'Problems', 'Output', and 'Terminal'. The command prompt is 'PS C:\programing\Struktur Data\LAPRAK2>'. The program code is displayed in a monospaced font. It starts with a directory change to 'c:\programing\Struktur Data\LAPRAK2\' and a conditional compilation check. The main logic consists of three sections: inputting student scores, displaying them, and displaying a predefined 5x5 matrix of values. The input section shows five scores: 78, 66, 78, 96, and 56. The display section shows these scores as 'nilai ke-1' through 'ke-5'. The matrix section shows a 5x5 grid of values: Row 1: 0, 2, 2, 0, 0; Row 2: 0, 1, 1, 1, 0; Row 3: 0, 3, 3, 3, 0; Row 4: 4, 4, 0, 0, 4; Row 5: 5, 0, 0, 0, 5. The prompt 'PS C:\programing\Struktur Data\LAPRAK2>' is visible at the bottom.

```
PS C:\programing\Struktur Data\LAPRAK2> cd "c:\programing\Struktur Data\LAPRAK2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
=== input nilai siswa ===  
masukkan nilai ke-1: 78  
masukkan nilai ke-2: 66  
masukkan nilai ke-3: 78  
masukkan nilai ke-4: 96  
masukkan nilai ke-5: 56  
  
=== data nilai siswa ===  
nilai ke-1 = 78  
nilai ke-2 = 66  
nilai ke-3 = 78  
nilai ke-4 = 96  
nilai ke-5 = 56  
  
=== 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  
PS C:\programing\Struktur Data\LAPRAK2>
```

Deskripsi:

Program ini bertujuan untuk mendemonstrasikan implementasi dan manipulasi array satu dimensi dan array dua dimensi dalam bahasa C++.

Program ini mengilustrasikan dua skenario penggunaan array:

1. Array Satu Dimensi (nilai): Digunakan untuk menyimpan serangkaian data (nilai siswa) yang dimasukkan oleh pengguna. Proses pengisian dan penampilan kembali data ini dilakukan menggunakan perulangan for.
2. Array Dua Dimensi (nilai_tahun): Diinisialisasi dengan nilai statis (pre-defined) untuk merepresentasikan data dalam format tabel atau matriks. Program menampilkannya menggunakan perulangan bersarang (nested loop) untuk mengakses setiap elemen berdasarkan indeks baris dan kolomnya.

Secara keseluruhan, kode ini menunjukkan cara mendeklarasikan, mengisi (baik melalui input pengguna maupun inisialisasi langsung), dan mengakses elemen dari kedua jenis array tersebut.

Guided 3

```
Struktur Data > LAPRAK2 > C++ guided3.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6 int maks3(int a, int b, int c);
7
8 int main(){
9     int x,y,z;
10    cout<<"masukkan nilai bilangan ke-1 = ";
11    cin>>x;
12    cout<<"masukkan nilai bilangan ke-2 = ";
13    cin>>y;
14    cout<<"masukkan nilai bilangan ke-3 = ";
15    cin>>z;
16    cout<<"nilai maksimumnya adalah = "
17    <<maks3(x,y,z);
18    return 0;
19 }
20
21 int maks3(int a, int b, int c){
22
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

```
Problems  Output  Terminal  ...
● PS C:\programing\Struktur Data\LAPRAK2> cd "c:\programing\Struktur Data\LAPRAK2\" ; if ($?) { g++ guided3.cpp -o guided3 } ; if ($?) { .\guided3 }
masukkan nilai bilangan ke-1 = 46
masukkan nilai bilangan ke-2 = 69
masukkan nilai bilangan ke-3 = 55
nilai maksimumnya adalah = 69
○ PS C:\programing\Struktur Data\LAPRAK2>
```

Deskripsi:

Program ini merupakan contoh implementasi **fungsi** dalam bahasa C++ untuk menyelesaikan tugas spesifik, yaitu mencari nilai maksimum dari tiga bilangan.

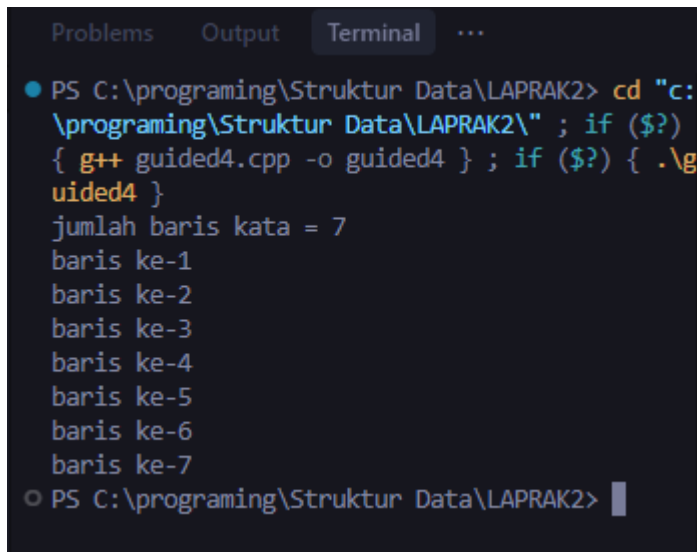
Struktur program ini dibagi menjadi dua bagian utama: fungsi main yang bertugas untuk menangani interaksi dengan pengguna (input dan output), dan fungsi maks3 yang berisi logika untuk membandingkan tiga nilai integer. Penggunaan **prototipe fungsi** di awal program memungkinkan main untuk memanggil maks3 meskipun definisinya berada setelahnya.

Data dari pengguna dilewatkan ke fungsi maks3 menggunakan mekanisme **call by value**, dan hasilnya dikembalikan (*return value*) ke main untuk ditampilkan. Secara keseluruhan, program ini mendemonstrasikan prinsip dasar modularisasi kode, di mana sebuah tugas dipecah menjadi fungsi yang lebih kecil dan dapat digunakan kembali (*reusable*).

Guided 4

```
Struktur Data > LAPRAK2 > C++ guided4.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6
7 void tulis(int x);
8 int main()
9 {
10     int jum;
11     cout << "jumlah baris kata = ";
12     cin >> jum;
13     tulis(jum);
14     return 0;
15 }
16
17 void tulis(int x){
18     for (int i=0;i<x;i++)
19         cout<<"baris ke-"<<i+1<<endl;
20 }
```

Screenshots Output



```
PS C:\programming\Struktur Data\LAPRAK2> cd "C:\programming\Struktur Data\LAPRAK2\" ; if ($?) { g++ guided4.cpp -o guided4 } ; if ($?) { .\guided4 }
jumlah baris kata = 7
baris ke-1
baris ke-2
baris ke-3
baris ke-4
baris ke-5
baris ke-6
baris ke-7
PS C:\programming\Struktur Data\LAPRAK2>
```

Deskripsi:

Program ini mendemonstrasikan penggunaan **prosedur**, yaitu sebuah **fungsi yang tidak mengembalikan nilai (void)**, dalam bahasa C++. Tujuannya adalah untuk menjalankan sebuah aksi tanpa perlu menghasilkan nilai kembalian.

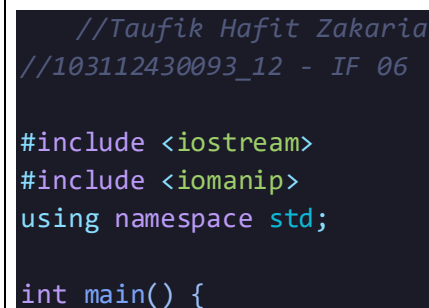
Struktur program dibagi menjadi dua:

1. **Fungsi main:** Bertindak sebagai titik awal dan antarmuka pengguna, di mana program meminta input sebuah bilangan untuk menentukan jumlah baris yang akan dicetak.
2. **Prosedur tulis:** Menerima input dari main sebagai parameter. Fungsi ini berisi logika utama berupa perulangan for untuk mencetak teks ke layar berulang kali.

Mekanisme pelewatan parameter yang digunakan adalah **call by value**, di mana nilai dari variabel `jum` disalin ke parameter `x` di dalam prosedur tulis. Program ini secara efektif menunjukkan bagaimana memisahkan tugas input/output di main dari tugas pemrosesan atau aksi di fungsi lain untuk menciptakan kode yang lebih modular.

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

Unguided 1



```
//Taufik Hafit Zakaria
//103112430093_12 - IF 06

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

int main() {
```



```

const int JUMLAH_MAHASISWA = 5;
const int JUMLAH_MATA_KULIAH = 3;

string namaMahasiswa[JUMLAH_MAHASISWA];
float nilaiMahasiswa[JUMLAH_MAHASISWA][JUMLAH_MATA_KULIAH];
float rataRata[JUMLAH_MAHASISWA];

cout << "=== INPUT DATA NILAI MAHASISWA ===\n";
for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
    cout << "\nMahasiswa ke-" << (i + 1) << ":\n";
    cout << "Nama: ";

    getline(cin >> ws, namaMahasiswa[i]);
    cout << "Masukkan nilai 3 mata kuliah:\n";
    float totalNilai = 0;
    for (int j = 0; j < JUMLAH_MATA_KULIAH; j++) {
        cout << "    Mata kuliah " << (j + 1) << ": ";
        cin >> nilaiMahasiswa[i][j];
        totalNilai += nilaiMahasiswa[i][j];
    }

    rataRata[i] = totalNilai / JUMLAH_MATA_KULIAH;
}

int indeksTerbaik = 0;
for (int i = 1; i < JUMLAH_MAHASISWA; i++) {
    if (rataRata[i] > rataRata[indeksTerbaik]) {
        indeksTerbaik = i;
    }
}

cout << "\n\n=== HASIL NILAI MAHASISWA ===\n";
cout <<
"===== \n";
cout << left << setw(12) << "Nama"
    << setw(8) << "MK1"
    << setw(8) << "MK2"
    << setw(8) << "MK3"
    << setw(12) << "Rata-rata"
    << "Keterangan\n";
cout <<
"===== \n";

for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
    cout << left << setw(12) << namaMahasiswa[i];

    for (int j = 0; j < JUMLAH_MATA_KULIAH; j++) {

```

```
        cout << setw(8) << fixed << setprecision(1) <<
nilaiMahasiswa[i][j];
    }

    cout << setw(12) << fixed << setprecision(2) << rataRata[i];

    if (i == indeksTerbaik) {
        cout << "TERBAIK";
    }
    cout << endl;
}

cout << "\n=== MAHASISWA TERBAIK ===\n";
cout << "Nama: " << namaMahasiswa[indeksTerbaik] << endl;
cout << "Rata-rata nilai: " << fixed << setprecision(2) <<
rataRata[indeksTerbaik] << endl;

    return 0;
}
```

Screenshots Output

```
Mahasiswa ke-1:
Nama: Taufik H
Masukkan nilai 3 mata kuliah:
  Mata kuliah 1: 67
  Mata kuliah 2: 98
  Mata kuliah 3: 57
```

```
Mahasiswa ke-2:
Nama: Devita A
Masukkan nilai 3 mata kuliah:
  Mata kuliah 1: 98
  Mata kuliah 2: 78
  Mata kuliah 3: 86
```

```
Mahasiswa ke-3:
Nama: Vincent PG
Masukkan nilai 3 mata kuliah:
  Mata kuliah 1: 76
  Mata kuliah 2: 58
  Mata kuliah 3: 87
```

```
Mahasiswa ke-4:
Nama: Deo A
Masukkan nilai 3 mata kuliah:
  Mata kuliah 1: 67
  Mata kuliah 2: 93
  Mata kuliah 3: 64
```

```
Mahasiswa ke-5:
Nama: Rio
Masukkan nilai 3 mata kuliah:
  Mata kuliah 1: 78
  Mata kuliah 2: 56
  Mata kuliah 3: 34
```

```
=== HASIL NILAI MAHASISWA ===
```

```
=====
Nama      MK1      MK2      MK3      Rata-rata  Keterangan
=====
Taufik H  67.0      98.0      57.0      74.00
Devita A  98.0      78.0      86.0      87.33      TERBAIK
Vincent PG 76.0      58.0      87.0      73.67
Deo A     67.0      93.0      64.0      74.67
Rio       78.0      56.0      34.0      56.00
```

```
=== MAHASISWA TERBAIK ===
```

```
Nama: Devita A
Rata-rata nilai: 87.33
```

```
PS C:\programing\Struktur Data\LAPRAK2> █
```

Deskripsi:

Program ini merupakan implementasi untuk manajemen dan analisis data nilai mahasiswa yang secara efektif memanfaatkan array satu dimensi untuk menyimpan nama dan rata-rata nilai, serta array dua dimensi untuk mengelola nilai dari tiga mata kuliah untuk lima orang mahasiswa. Prosesnya dimulai dengan tahap input data interaktif, di mana program menggunakan `getline` untuk menangkap nama lengkap dan secara langsung menghitung rata-rata setiap mahasiswa. Selanjutnya, sebuah algoritma dijalankan untuk menentukan mahasiswa dengan rata-rata tertinggi dengan cara menyimpan indeksinya. Hasil analisis kemudian disajikan dalam sebuah tabel yang terstruktur dan rapi menggunakan pustaka `<iomanip>`, yang secara visual menandai mahasiswa "TERBAIK" dan diakhiri dengan ringkasan khusus untuk menyoroti pencapaiannya.

Unguided 2

```
Struktur Data > LAPRAK2 > C++ nilai.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6
7 int main() {
8     int N;
9     cout << "Masukkan jumlah bilangan: ";
10    cin >> N;
11
12    int *ptr = new int[N];
13    for (int i = 0; i < N; i++) {
14        cout << "Masukkan bilangan ke-" << (i+1) << ": ";
15        cin >> ptr[i];
16    }
17
18    int jumlah = 0;
19    int maksimum = ptr[0];
20    int minimum = ptr[0];
21
22    for (int i = 1; i < N; i++) {
23        jumlah += ptr[i];
24        if (ptr[i] > maksimum) {
25            maksimum = ptr[i];
26        }
27        if (ptr[i] < minimum) {
28            minimum = ptr[i];
29        }
30    }
31
32    cout << "\nHasil:" << endl;
33    cout << "Jumlah: " << jumlah << endl;
34    cout << "Nilai Maksimum: " << maksimum << endl;
35    cout << "Nilai Minimum: " << minimum << endl;
36
37    delete[] ptr;
38    return 0;
39 }
```

Screenshots Output

```
PS C:\programing\Struktur Data\LAPRAK2> cd "c:\programing\Str
if ($?) { g++ nilai.cpp -o nilai } ; if ($?) { .\nilai }
Masukkan jumlah bilangan: 4
Masukkan bilangan ke-1: 8
Masukkan bilangan ke-2: 7
Masukkan bilangan ke-3: 5
Masukkan bilangan ke-4: 9

Hasil:
Jumlah: 21
Nilai Maksimum: 9
Nilai Minimum: 5
PS C:\programing\Struktur Data\LAPRAK2> █
```

Deskripsi:

Program ini secara efektif mendemonstrasikan konsep alokasi memori dinamis dan manajemen array menggunakan pointer dalam C++. Ukuran array tidak ditentukan secara statis, melainkan berdasarkan input N dari pengguna saat program berjalan (runtime). Memori untuk array dialokasikan menggunakan operator new, dan alamat awalnya disimpan dalam sebuah pointer int *ptr. Program kemudian mengisi array ini dengan nilai dari pengguna, mengakses setiap elemen menggunakan notasi ptr[i]. Untuk proses kalkulasi, program menginisialisasi nilai maksimum dan minimum dengan elemen pertama (ptr[0]), lalu melakukan iterasi mulai dari elemen kedua (i = 1) untuk mencari nilai ekstrem serta mengakumulasi jumlah. Perlu dicatat bahwa karena perulangan ini, elemen pertama tidak disertakan dalam perhitungan jumlah akhir. Sebagai penutup dan praktik manajemen memori yang baik, program secara eksplisit melepaskan memori yang telah dialokasikan dengan delete[] ptr untuk mencegah kebocoran memori (memory leak), sebelum akhirnya menampilkan hasil perhitungan kepada pengguna.

Unguided 3

```
Struktur Data > LAPRAK2 > C++ ujian.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6
7 float hitungRata(float nilai[], int n) {
8     float total = 0;
9     for (int i = 0; i < n; i++) {
10         total += nilai[i];
11     }
12     return total / n;
13 }
14
15 void cariNilai(float nilai[], int n, float &maks, float &min) {
16     maks = nilai[0];
17     min = nilai[0];
18     for (int i = 1; i < n; i++) {
19         if (nilai[i] > maks) maks = nilai[i];
20         if (nilai[i] < min) min = nilai[i];
21     }
22 }
23
24 int main() {
25     int n;
26     cout << "Masukkan jumlah siswa: ";
27     cin >> n;
28
29     float nilai[n];
30     for (int i = 0; i < n; i++) {
31         cout << "Nilai siswa ke-" << (i+1) << ": ";
32         cin >> nilai[i];
33     }
34
35     float avg = hitungRata(nilai, n);
36     float maks, min;
37     cariNilai(nilai, n, maks, min);
38
39     cout << "\nRata-rata: " << avg << endl;
40     cout << "Nilai tertinggi: " << maks << endl;
41     cout << "Nilai terendah: " << min << endl;
42
43     return 0;
44 }
```

Screenshots Output

```
PS C:\programing\Struktur Data\LAPRAK2> cd "c:\programing\Struktur
if ($?) { g++ ujian.cpp -o ujian } ; if ($?) { .\ujian }
Masukkan jumlah siswa: 4
Nilai siswa ke-1: 98
Nilai siswa ke-2: 78
Nilai siswa ke-3: 89
Nilai siswa ke-4: 87

Rata-rata: 88
Nilai tertinggi: 98
Nilai terendah: 78
PS C:\programing\Struktur Data\LAPRAK2>
```

Deskripsi:

Program ini mendemonstrasikan prinsip **pemrograman modular** dalam C++ dengan memisahkan logika ke dalam beberapa fungsi spesifik untuk analisis data nilai siswa. Fungsi main bertindak sebagai pusat kendali, di mana ia pertama-tama meminta jumlah siswa (n) untuk menentukan ukuran array nilai, kemudian mengisinya dengan input pengguna. Selanjutnya, main memanggil dua fungsi helper: pertama, `hitungRata`, yang menerima array dan ukurannya untuk menghitung dan **mengembalikan (return)** satu nilai rata-rata. Kedua, main memanggil fungsi void `cariNilai` untuk menemukan nilai tertinggi dan terendah; fungsi ini secara efisien "mengembalikan" dua hasil (maks dan min) dengan memodifikasi variabel asli di main melalui mekanisme **pass-by-reference** (ditandai dengan `&`). Pada akhirnya, semua hasil yang dikumpulkan dari pemanggilan fungsi ini ditampilkan secara terorganisir kepada pengguna.

Unguided 4

```
Struktur Data > LAPRAK2 > C++ pola.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 using namespace std;
6
7 int main() {
8     int n;
9     cout << "Masukkan panjang segitiga: ";
10    cin >> n;
11    for (int i = 1; i <= n; i++) {
12        for (int j = 1; j <= i; j++) {
13            cout << j << " ";
14        }
15        cout << endl;
16    }
17
18    return 0;
19 }
```

Screenshots Output

```
● PS C:\programing\Struktur Data\LAPRAK2> cd "c:\programing\Struktur
if ($?) { g++ pola.cpp -o pola } ; if ($?) { .\pola }
Masukkan panjang segitiga: 9
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
○ PS C:\programing\Struktur Data\LAPRAK2> █
```

Deskripsi:

Program ini bertujuan untuk menghasilkan pola segitiga siku-siku numerik yang ukurannya ditentukan oleh input (n) dari pengguna. Struktur utama program ini adalah penggunaan perulangan bersarang (nested loop). Perulangan luar (for dengan variabel i) berfungsi untuk mengontrol jumlah baris pada segitiga, berjalan dari 1 hingga nilai n yang dimasukkan. Sementara itu, perulangan dalam (for dengan variabel j) bertanggung jawab

untuk mencetak urutan angka pada setiap baris. Jumlah iterasi perulangan dalam ini bergantung secara dinamis pada nilai *i* dari perulangan luar, di mana ia akan mencetak angka dari 1 hingga *i*. Setelah setiap baris selesai dicetak, perintah `cout << endl;` digunakan untuk berpindah ke baris baru, sehingga menyusun pola segitiga secara vertikal.

D. Kesimpulan

Praktikum Modul II ini berhasil memberikan pemahaman yang mendalam mengenai konsep-konsep lanjutan dalam bahasa C++ yang menjadi pilar fundamental untuk struktur data. Melalui implementasi serangkaian program, baik dari bagian guided maupun unguided, mahasiswa telah menguasai tiga area utama: pengelolaan data menggunakan array, manipulasi memori secara langsung dengan pointer, dan penerapan pemrograman modular melalui fungsi. Penggunaan array satu dan dua dimensi terbukti efektif untuk mengelola kumpulan data statis maupun dinamis. Konsep pointer, khususnya dalam alokasi memori dinamis (`new` dan `delete[]`), memberikan kemampuan untuk membuat program yang lebih fleksibel dan efisien dalam penggunaan memori. Selain itu, pemahaman tentang berbagai mekanisme pelewatan parameter—seperti `call by value` dan `call by reference`—terbukti krusial dalam merancang fungsi yang efisien untuk mengolah data dan mengembalikan hasil. Secara keseluruhan, praktikum ini tidak hanya memperkuat pemahaman teoretis tetapi juga mengembangkan kemampuan logika pemrograman yang esensial untuk merancang dan mengimplementasikan struktur data yang lebih kompleks di masa depan.

E. Referensi

1. Ma'arif, A. (2022). Modul Praktikum Dasar Pemrograman dengan Code Blocks. Universitas Ahmad Dahlan. Diakses dari <http://eprints.uad.ac.id/id/eprint/32511>
2. Schildt, H. (2017). C++: The Complete Reference (5th ed.). McGraw-Hill Education.
3. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley Professional.
4. Saputra, W. A. (2025). Modul II: Pengenalan Bahasa C++ (Bagian Kedua). Purwokerto: Program Studi Struktur Data, Fakultas Informatika, Telkom University Purwokerto