

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

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



**Disusun Oleh :**

**NAMA : LAHRA BUDI SAPUTRA**

**NIM : 103112430054**

**Dosen**

**FAHRUDIN MUKTI WIBOWO**

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

## A. Dasar Teori

Modul ini memperkenalkan konsep Array, yang merupakan kumpulan data sejenis dengan nama yang sama, diakses melalui indeks yang dimulai dari 0, dan disimpan secara berurutan dalam memori. Array dapat berupa satu dimensi, dua dimensi (mirip tabel), atau berdimensi banyak. Konsep penting lainnya adalah Pointer, yaitu variabel yang menyimpan alamat memori variabel lain. Untuk mengetahui alamat suatu variabel digunakan operator `&` dan untuk mengakses nilai yang ditunjuk pointer digunakan operator `*`.

Terdapat hubungan kuat antara array dan pointer, di mana `pa + i` merupakan alamat dari elemen array `a[i]`. String dalam C++ pada dasarnya adalah array dari karakter yang diakhiri dengan `\0`. Pemrograman modular dicapai melalui Fungsi, blok kode yang dapat mengembalikan nilai, dan Prosedur (dikenal sebagai fungsi void) yang tidak mengembalikan nilai. Melewatkan parameter ke fungsi dapat dilakukan dengan tiga cara: Call by Value, di mana nilai disalin sehingga variabel aktual tidak berubah; Call by Pointer, di mana alamat dilewatkan menggunakan operator `&`, sehingga variabel aktual dapat diubah ; dan Call by Reference, yang juga melewatkan alamat tetapi menggunakan operator referensi `&` pada parameter formal untuk kemudahan penulisan.

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

### Guided 1

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

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 *px     = " << y << endl;
    cout << "Nilai y       = " << y << endl;

    return 0;
}
```

## Screenshots Output:

```
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)\\" ; if ($?) { g++ P2guided1.cpp -o P2guided1 } ; if ($?) { .\P2guided1 }
Alamat x      = 0xf4555ffaf0
Isi px        = 0xf4555ffaf0
Isi x         = 87
Nilai *px     = 87
Nilai y       = 87
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)>
```

## Deskripsi:

Program di atas merupakan contoh sederhana penggunaan pointer dalam bahasa C++. Pertama, variabel x diberi nilai 87, kemudian pointer px diisi dengan alamat memori dari x menggunakan operator &. Setelah itu, nilai yang disimpan di alamat tersebut diambil menggunakan px dan disimpan ke variabel y. Program kemudian menampilkan alamat variabel x, isi dari pointer px (yang merupakan alamat x), serta nilai x, px, dan y. Hasilnya menunjukkan bahwa px dan y memiliki nilai yang sama dengan x, karena pointer digunakan untuk mengakses nilai dari alamat memori yang ditunjuknya.

## Guided 2

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

using namespace std;

int main()
{
    int i, j;
    float nilai[MAX];
    static int nilai_tahun[MAX][MAX] = {
        {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}
    };

    cout << "=== INPUT NILAI MAHASISWA ===\n";
    for (i = 0; i < MAX; i++) {
        cout << "Masukkan nilai Mahasiswa ke-" << i + 1 << ": ";
        cin >> nilai[i];
    }

    cout << "\n=== DATA NILAI MAHASISWA ===\n";
    for (i = 0; i < MAX; i++) {
        cout << "Nilai ke-" << i + 1 << " = " << nilai[i] << endl;
    }
}
```

```

    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;
}

```

### Screenshots Output

```

PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)" ; if ($?) { g++ P2guided
2.cpp -o P2guided2 } ; if ($?) { .\P2guided2 }
=== INPUT NILAI MAHASISWA ===
Masukkan nilai Mahasiswa ke-1: 75
Masukkan nilai Mahasiswa ke-2: 80
Masukkan nilai Mahasiswa ke-3: 85
Masukkan nilai Mahasiswa ke-4: 90
Masukkan nilai Mahasiswa ke-5: 100

=== DATA NILAI MAHASISWA ===
Nilai ke-1 = 75
Nilai ke-2 = 80
Nilai ke-3 = 85
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
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)>

```

### Deskripsi:

Program di atas merupakan contoh penggunaan array satu dimensi dan dua dimensi dalam bahasa C++. Pertama, program mendefinisikan konstanta MAX dengan nilai 5 untuk menentukan ukuran array. Array nilai digunakan untuk menyimpan input nilai dari lima mahasiswa, sedangkan array dua dimensi nilai\_tahun berisi data nilai tahunan yang sudah diisi sebelumnya secara statis. Program akan meminta pengguna memasukkan nilai mahasiswa satu per satu, kemudian menampilkan kembali seluruh nilai yang telah dimasukkan. Setelah itu, program juga menampilkan isi dari array dua dimensi nilai\_tahun dalam bentuk tabel 5x5. Tujuan utama program ini adalah memperlihatkan cara penggunaan array satu dan dua dimensi untuk menyimpan serta menampilkan data dalam program C++.

### Guided 3

```

#include <iostream>
using namespace std;

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

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

```

```

    cout << "Memasukan nilai bilangan ke-1 = ";
    cin >> x;
    cout << " Memasukan nilai bilangan ke-2 = ";
    cin >> y;
    cout << " Memasukan 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;
}

```

### Screenshots Output

```

PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)" ; if ($?) { g++ P2guided
3.cpp -o P2guided3 } ; if ($?) { .\P2guided3 }
Memasukan nilai bilangan ke-1 = 10
Memasukan nilai bilangan ke-2 = 25
Memasukan nilai bilangan ke-3 = 7
Nilai maksimumnya adalah = 25
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)>

```

### Deskripsi:

Program di atas merupakan contoh penggunaan fungsi dengan parameter dan nilai balik (return value) dalam bahasa C++. Program ini bertujuan untuk mencari nilai maksimum dari tiga bilangan yang dimasukkan oleh pengguna. Pertama, program meminta input tiga bilangan (x, y, dan z). Setelah itu, fungsi maks3() dipanggil dengan ketiga nilai tersebut sebagai argumen. Di dalam fungsi maks3(), dilakukan proses perbandingan menggunakan struktur if untuk menentukan nilai terbesar di antara ketiganya, yang kemudian disimpan dalam variabel temp\_max dan dikembalikan ke fungsi main(). Hasil akhirnya, program akan menampilkan nilai maksimum dari tiga bilangan yang dimasukkan pengguna.

## Guided 4

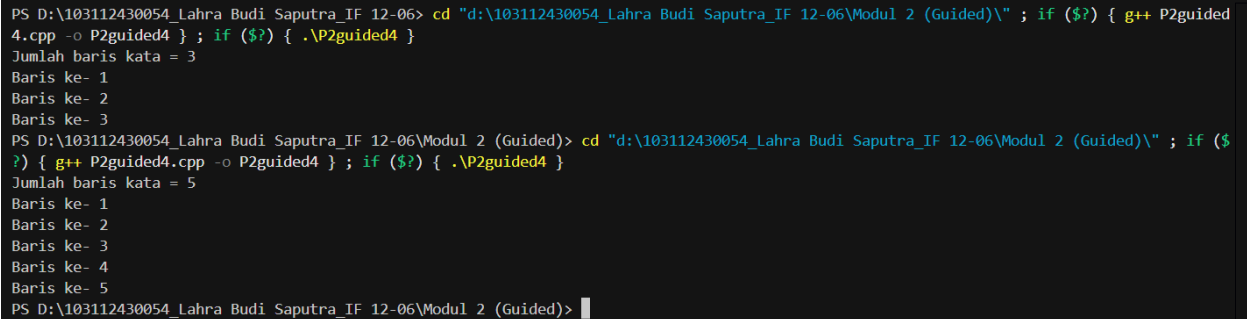
```
#include <iostream>
using namespace std;

//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;
    }
}
```

## Screenshots Output



```
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)\\" ; if ($?) { g++ P2guided
4.cpp -o P2guided4 } ; if ($?) { .\P2guided4 }
Jumlah baris kata = 3
Baris ke- 1
Baris ke- 2
Baris ke- 3
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)\\" ; if ($
?) { g++ P2guided4.cpp -o P2guided4 } ; if ($?) { .\P2guided4 }
Jumlah baris kata = 5
Baris ke- 1
Baris ke- 2
Baris ke- 3
Baris ke- 4
Baris ke- 5
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Guided)> |
```

## Deskripsi:

Program di atas merupakan contoh penggunaan prosedur (fungsi tanpa nilai balik) dalam bahasa C++. Program ini meminta pengguna memasukkan jumlah baris yang ingin ditampilkan melalui variabel `jum`. Nilai tersebut kemudian dikirim sebagai argumen ke prosedur `tulis()`. Di dalam prosedur `tulis()`, terdapat perulangan `for` yang akan menampilkan teks “Baris ke-” diikuti dengan nomor urut baris sebanyak jumlah yang dimasukkan oleh pengguna. Program ini menunjukkan cara penggunaan prosedur untuk menjalankan tugas tertentu tanpa harus mengembalikan nilai ke fungsi utama (`main`).

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

**Unguided 1**

```
/*Nama Lengkap : Lahra budi Saputra
NIM : 103112430054
Kelas : IF 12-06*/
#include <iostream>
#include <string>

using namespace std;

// Konstanta ukuran array
const int JUMLAH_MAHASISWA = 5;
const int JUMLAH_MATA_KULIAH = 3;

// Fungsi untuk menampilkan garis pemisah
void print_separator() {
    cout << "-----\n";
}

int main() {
    double nilai[JUMLAH_MAHASISWA][JUMLAH_MATA_KULIAH];
    string nama[JUMLAH_MAHASISWA];
    double rata2[JUMLAH_MAHASISWA];

    // Variabel untuk melacak mahasiswa terbaik
    double max_rata2 = -1.0;
    int best_idx = -1;

    for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
        cout << "\n==> Input Data Mahasiswa ke-" << i + 1 << " <==\n";

        cout << "Masukkan Nama: ";
        getline(cin >> ws, nama[i]);

        double total = 0;

        // Input nilai 3 mata kuliah
        for (int j = 0; j < JUMLAH_MATA_KULIAH; j++) {
            cout << "Nilai Mata Kuliah ke-" << j + 1 << ": ";
            cin >> nilai[i][j];
            total += nilai[i][j];
        }

        // Hitung rata-rata
        rata2[i] = total / JUMLAH_MATA_KULIAH;
```

```

        // Tentukan terbaik
        if (rata2[i] > max_rata2) {
            max_rata2 = rata2[i];
            best_idx = i;
        }
    }

    cout << "\n\n=== TABEL HASIL NILAI AKHIR ===\n";
    print_separator();
    cout << "NO\tNAMA\tMK 1\tMK 2\tMK 3\tRATA-RATA\tKETERANGAN\n";

    // Mengatur presisi untuk tampilan
    cout.precision(4);

    for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
        cout << i + 1 << "\t" << nama[i] << "\t";

        if (nama[i].length() < 12) {
            cout << "\t";
        }

        for (int j = 0; j < JUMLAH_MATA_KULIAH; j++) {
            cout << nilai[i][j] << "\t";
        }

        cout << rata2[i] << "\t\t";

        if (i == best_idx) {
            cout << "=== TERBAIK ===";
        } else {
            cout << "-";
        }
        cout << "\n";
    }

    print_separator();
    cout << "\nMahasiswa dengan Rata-rata Tertinggi: " << nama[best_idx]
        << " (Rata-rata: " << max_rata2 << ")\n";

    return 0;
}

```



## Screenshots Output:

```
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)\"; if ($?) { g++ Soal1.
cpp -o Soal1 } ; if ($?) { .\Soal1 }

===> Input Data Mahasiswa ke-1 <===
Masukkan Nama: jokowi
Nilai Mata Kuliah ke-1: 90
Nilai Mata Kuliah ke-2: 80
Nilai Mata Kuliah ke-3: 80

===> Input Data Mahasiswa ke-2 <===
Masukkan Nama: anton
Nilai Mata Kuliah ke-1: 90
Nilai Mata Kuliah ke-2: 89
Nilai Mata Kuliah ke-3: 76

===> Input Data Mahasiswa ke-3 <===
Masukkan Nama: jojo
Nilai Mata Kuliah ke-1: 67
Nilai Mata Kuliah ke-2: 89
Nilai Mata Kuliah ke-3: 90

===> Input Data Mahasiswa ke-4 <===
Masukkan Nama: tami
Nilai Mata Kuliah ke-1: 90
Nilai Mata Kuliah ke-2: 80
Nilai Mata Kuliah ke-3: 90

===> Input Data Mahasiswa ke-5 <===
Masukkan Nama: toto
Nilai Mata Kuliah ke-1: 78
Nilai Mata Kuliah ke-2: 98
Nilai Mata Kuliah ke-3: 70

=== TABEL HASIL NILAI AKHIR ===
-----
NO      NAMA      MK 1   MK 2   MK 3   RATA-RATA   KETERANGAN
1       jokowi    90     80     80     83.33      -
2       anton     90     89     76     85         -
3       jojo      67     89     90     82         -
4       tami      90     80     90     86.67     == TERBAIK ==
5       toto      78     98     70     82         -
-----

Mahasiswa dengan Rata-rata Tertinggi: tami (Rata-rata: 86.67)
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)>
```

## Deskripsi:

Program di atas merupakan aplikasi sederhana untuk mengelola dan menampilkan data nilai mahasiswa menggunakan array dua dimensi dan konsep dasar perulangan serta seleksi. Program ini dimulai dengan mendeklarasikan konstanta jumlah mahasiswa dan jumlah mata kuliah. Kemudian, terdapat beberapa array: `nama[ ]` untuk menyimpan nama mahasiswa, `nilai[ ][ ]` untuk menyimpan nilai tiap mata kuliah, serta `rata2[ ]` untuk menyimpan nilai rata-rata masing-masing mahasiswa.

Pada bagian input, pengguna diminta untuk memasukkan nama dan tiga nilai mata kuliah untuk setiap mahasiswa. Nilai rata-rata kemudian dihitung dan disimpan dalam array `rata2`. Program juga secara otomatis mencari mahasiswa dengan nilai rata-rata tertinggi menggunakan variabel `max_rata2` dan `best_idx`.

Setelah semua data dimasukkan, program menampilkan tabel hasil akhir yang berisi nama, nilai tiap mata kuliah, rata-rata, serta keterangan `=== TERBAIK ===` bagi mahasiswa dengan nilai tertinggi. Fungsi `print_separator()` digunakan untuk memperindah tampilan dengan garis pemisah.

## Unguided 2

```
/*Nama Lengkap : Lahra budi Saputra
NIM : 103112430054
Kelas : IF 12-06*/
#include <iostream>

using namespace std;

int main() {
    int N;

    cout << "Masukkan jumlah bilangan (N): ";
    cin >> N;

    int *arr = new int[N];
    int *p = arr;

    cout << "Masukkan " << N << " bilangan:\n";
    for (; p < arr + N; ++p) {
        cout << " Bilangan ke-" << (p - arr + 1) << ": ";
        cin >> *p;
    }

    p = arr;

    long long jumlah = 0;
    int maks = *arr;
    int min = *arr;

    for (; p < arr + N; ++p) {
        jumlah += *p;
        if (*p > maks) maks = *p;
        if (*p < min) min = *p;
    }

    // Menampilkan Hasil
    cout << "\nHasil Perhitungan:\n";
    cout << "Jumlah = " << jumlah << endl;
    cout << "Nilai Maks = " << maks << endl;
    cout << "Nilai Min = " << min << endl;

    // Bersihkan memori
    delete[] arr;

    return 0;
}
```

## Screenshots Output:

```
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)\"; if ($?) { g++ Soal2.cpp -o Soal2 } ; if ($?) { .\Soal2 }
Masukkan jumlah bilangan (N): 5
Masukkan 5 bilangan:
  Bilangan ke-1: 10
  Bilangan ke-2: 25
  Bilangan ke-3: 7
  Bilangan ke-4: 30
  Bilangan ke-5: 15

Hasil Perhitungan:
Jumlah      = 87
Nilai Maks  = 30
Nilai Min   = 7
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)> █
```

## Deskripsi:

Program di atas merupakan contoh penggunaan pointer dan alokasi memori dinamis dalam bahasa C++. Program ini meminta pengguna untuk memasukkan jumlah bilangan (N), lalu mengalokasikan memori sebesar N elemen menggunakan `new`. Pointer `arr` digunakan untuk menyimpan alamat awal array yang dialokasikan, sedangkan pointer `p` digunakan untuk menelusuri elemen-elemen array tersebut.

Pengguna kemudian diminta menginput N buah bilangan, yang disimpan di dalam array menggunakan pointer aritmatika (`p < arr + N`). Setelah semua data dimasukkan, program menghitung jumlah total, nilai maksimum, dan nilai minimum dengan cara mengakses setiap elemen melalui pointer.

Hasil perhitungan kemudian ditampilkan ke layar. Sebelum program selesai, memori yang telah dialokasikan dibebaskan menggunakan `delete [ ] arr` untuk mencegah kebocoran memori. Secara keseluruhan, program ini memperlihatkan cara menggunakan pointer untuk mengakses array secara langsung, sekaligus menerapkan konsep manajemen memori dinamis di C++.

## Unguided 3

```
/*Nama Lengkap : Lahra budi Saputra
NIM           : 103112430054
Kelas        : IF 12-06*/
#include <iostream>

using namespace std;

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

void minmax(const float a[], int n, float &min, float &max) {
    min = max = a[0];
    for (int i = 1; i < n; i++) {
```

```

        if (a[i] > max) max = a[i];
        if (a[i] < min) min = a[i];
    }
}

int main() {
    int n;
    cout << "Jumlah siswa: ";
    cin >> n;

    if (n <= 0) {
        cout << "Jumlah siswa harus > 0." << endl;
        return 1;
    }

    float nilai[n];

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

    float min_val, max_val;

    // Memanggil fungsi untuk perhitungan
    float avg = rata(nilai, n);
    minmax(nilai, n, min_val, max_val);

    // Menampilkan hasil perhitungan
    cout << "\n--- Hasil Analisis ---\n";
    cout << "Rata-rata: " << avg << endl;
    cout << "Tertinggi: " << max_val << endl;
    cout << "Terendah : " << min_val << endl;

    return 0;
}

```

## Screenshots Output

```

PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)\"; if ($?) { g++ Soal3.
cpp -o Soal3 } ; if ($?) { .\Soal3 }
Jumlah siswa: 4
Nilai ke-1: 80
Nilai ke-2: 70
Nilai ke-3: 90
Nilai ke-4: 85

===> Hasil Analisis <===
Rata-rata: 81.25
Tertinggi: 90
Terendah : 70
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)>

```

Deskripsi:

Program ini digunakan untuk mengolah nilai ujian beberapa siswa dengan menggunakan konsep parameter formal berupa array dan ukuran array pada bahasa C++. Program meminta pengguna untuk memasukkan jumlah siswa dan nilai ujian masing-masing, yang disimpan dalam array satu dimensi. Kemudian, program memanggil dua fungsi, yaitu rata() untuk menghitung rata-rata nilai kelas dan minmax() untuk mencari nilai tertinggi serta terendah. Fungsi rata() menggunakan parameter const agar data dalam array tidak berubah, sedangkan fungsi minmax() memakai parameter referensi (&) agar nilai minimum dan maksimum dapat dikembalikan langsung ke program utama. Setelah perhitungan selesai, hasil berupa nilai rata-rata, nilai tertinggi, dan nilai terendah ditampilkan ke layar. Program ini menunjukkan penerapan konsep modular, efisien, dan aman dalam pengolahan data menggunakan fungsi dengan parameter formal pada C++.

#### Unguided 4

```
/*Nama Lengkap : Lahra budi Saputra  
NIM : 103112430054  
Kelas : IF 12-06*/  
#include <iostream>  
  
using namespace std;  
  
void cetakSegitigaAngka(int n) {  
    for (int i = 1; i <= n; i++) {  
        for (int j = 1; j <= i; j++) {  
            cout << j << " ";  
        }  
        cout << endl;  
    }  
}  
  
int main() {  
    int tinggi;  
  
    cout << "Masukkan bilangan bulat (n): ";  
    cin >> tinggi;  
  
    cout << "Pola Segitiga Angka (n = " << tinggi << ")\\n";  
  
    // Memanggil Prosedur
```

```

    cetakSegitigaAngka(tinggi);

    return 0;
}

```

### Screenshots Output

```

PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)\\" ; if ($?) { g++ soal4.
cpp -o soal4 } ; if ($?) { .\soal4 }
Masukkan bilangan bulat (n): 5
Pola Segitiga Angka (n = 5)
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)\\" ; i
f ($?) { g++ soal4.cpp -o soal4 } ; if ($?) { .\soal4 }
Masukkan bilangan bulat (n): 7
Pola Segitiga Angka (n = 7)
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
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 2 (Unguided)>

```

### Deskripsi:

Program di atas merupakan contoh penggunaan prosedur dan perulangan bersarang (nested loop) dalam bahasa C++. Program ini digunakan untuk mencetak pola segitiga angka berdasarkan bilangan bulat n yang dimasukkan oleh pengguna.

Pertama, pengguna diminta untuk memasukkan nilai n yang menentukan tinggi segitiga. Kemudian, prosedur cetakSegitigaAngka() dipanggil dengan parameter n. Di dalam prosedur tersebut, terdapat dua perulangan for: perulangan luar (i) mengatur jumlah baris, sedangkan perulangan dalam (j) mencetak angka dari 1 hingga nilai i pada setiap baris. Setelah mencetak satu baris, program menambahkan baris baru menggunakan endl.

Sebagai hasilnya, program menampilkan pola segitiga angka yang dimulai dari baris pertama berisi angka 1, baris kedua 1 2, baris ketiga 1 2 3, dan seterusnya hingga mencapai nilai n.

### D. Kesimpulan

Modul 2 berfokus pada struktur data dasar C++ (Array, Pointer) dan struktur kendali (Fungsi, Prosedur), terutama bagaimana data dilewatkan di antara keduanya.

Konsep	Penjelasan Kunci (Sesuai Modul)	Penerapan dalam Soal
Array	Kumpulan data bertipe sama dengan nama yang sama, diakses	Digunakan untuk menyimpan daftar nilai N bilangan bulat (Soal 1) dan

	berdasarkan indeks (dimulai dari 0). Dapat berupa satu dimensi, dua dimensi (seperti tabel), atau berdimensi banyak	nilai ujian siswa (Soal 2 & 3).
Pointer	Variabel yang menyimpan alamat memori variabel lain (berisi <i>integer</i> dalam format heksadesimal). Digunakan untuk mengakses nilai variabel yang ditunjuk. Operator & mendapatkan alamat, dan operator * mendapatkan nilai (dereferencing).	Digunakan secara ekstensif pada Soal 1 untuk: 1) Mengalokasikan memori array secara dinamis (new). 2) Mengisi elemen array (cin >> *p). 3) Melakukan perhitungan (Jumlah, Min, Max).
Pointer Aritmatika	Ada hubungan kuat antara <i>Array</i> dan <i>Pointer</i> . Jika pa menunjuk ke a[0], maka pa + i adalah alamat dari a[i], dan *(pa + i) adalah isi dari elemen a[i]	Soal 1 adalah contoh utama, di mana perulangan dan akses elemen dilakukan menggunakan *(arr + i) atau ++p
Fungsi	Blok kode untuk tugas khusus, umumnya memerlukan parameter dan mengembalikan nilai balik (return value). Tipe keluaran (tipe_keluaran) harus didefinisikan	Fungsi rata (Soal 3) adalah contoh fungsi, karena mengembalikan nilai float (rata-rata)
Prosedur	Dalam C++, dikenal sebagai fungsi <i>void</i> . Melakukan tugas tertentu tetapi tidak mengembalikan nilai balik kepada pemanggilnya	Fungsi minmax (Soal 3) dan cetakSegitigaAngka (Soal 4) adalah contoh prosedur (void), karena tugas mereka adalah mengubah variabel eksternal (Min/Max) atau menampilkan output.

Secara keseluruhan, soal-soal yang dikerjakan berhasil mengimplementasikan dan menunjukkan perbedaan mendasar antara pointer aritmatika, fungsi (return value), dan prosedur (void), serta mekanisme Call by Reference untuk mengembalikan banyak nilai dari sebuah prosedur.

#### E. Referensi

Donyina, A., & Heckel, R. (2025). "Pedagogy of Teaching Pointers in the C Programming Language using Graph Transformations."

Logozar, R., Mikac, M., & Radošević, D. (2024). "Exploring the Access to the Static Array Elements via Indices and via Pointers — the Introductory C++ Case Expanded."