

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



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

Laporan Praktikum ini bertujuan untuk memberikan pemahaman dasar mengenai penggunaan Code: Blocks IDE serta pengenalan bahasa pemrograman C++. Mahasiswa diharapkan mampu mengenal lingkungan pemrograman, memahami proses instalasi dan penggunaan IDE, serta mengimplementasikan konsep dasar pemrograman C++ seperti tipe data, variabel, operator, input/output, percabangan, dan perulangan.

Melalui Praktikum ini, mahasiswa tidak hanya belajar teori, tetapi juga mempraktikkan langsung bagaimana menulis, meng-compile, dan menjalankan program sederhana. Dengan demikian, praktikum ini menjadi dasar penting untuk memahami pemrograman C++ yang akan digunakan lebih lanjut dalam pembahasan struktur data dan topik-topik lanjutan di mata kuliah ini.

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

Guided 1

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

int main()
{
    int hari;
    cout << "Masukkan hari (1-7): ";
    cin >> hari;
    if (hari == 1)
    {
        cout << "Hari minggu\n";

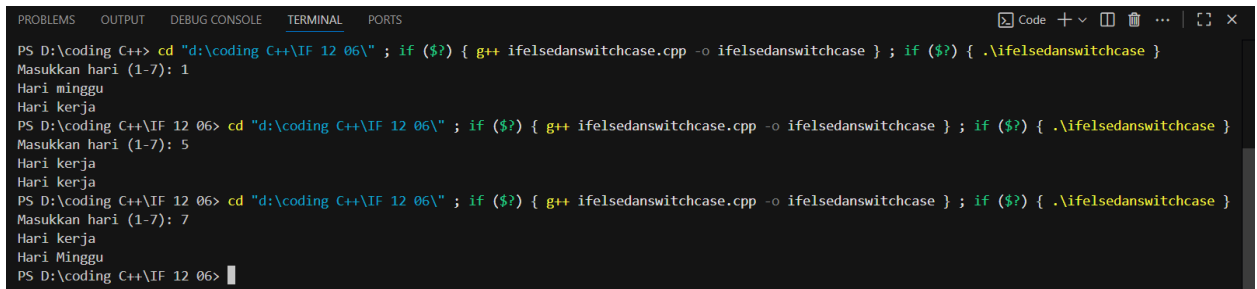
    }
    else
    {
        cout << "Hari kerja\n";
    }

    switch (hari)
    {
        case 7:
            cout << "Hari Minggu\n";
            break;

        default:
            cout << "Hari kerja\n";
            break;

    }
}
```

Screenshots Output



```
PS D:\coding C++> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ ifelseswitchcase.cpp -o ifelseswitchcase } ; if ($?) { .\ifelseswitchcase }
Masukkan hari (1-7): 1
Hari minggu
Hari kerja
PS D:\coding C++\IF 12 06> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ ifelseswitchcase.cpp -o ifelseswitchcase } ; if ($?) { .\ifelseswitchcase }
Masukkan hari (1-7): 5
Hari kerja
Hari kerja
PS D:\coding C++\IF 12 06> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ ifelseswitchcase.cpp -o ifelseswitchcase } ; if ($?) { .\ifelseswitchcase }
Masukkan hari (1-7): 7
Hari kerja
Hari Minggu
PS D:\coding C++\IF 12 06> |
```

Deskripsi:

Program di atas bertujuan untuk menampilkan keterangan hari berdasarkan input angka dari pengguna. Pengguna diminta memasukkan angka antara 1 sampai 7 yang merepresentasikan hari dalam satu minggu.

Pada bagian **if-else**, program akan menampilkan “Hari Minggu” jika input bernilai 1, sedangkan input selain itu akan dianggap sebagai “Hari Kerja”.

Pada bagian **switch-case**, program memeriksa nilai yang dimasukkan. Jika nilai adalah 7, maka ditampilkan “Hari Minggu”. Jika tidak, maka akan ditampilkan “Hari Kerja”.

Program ini menunjukkan dua cara berbeda (if-else dan switch-case) dalam menangani kondisi untuk menentukan jenis hari.

Guided 2

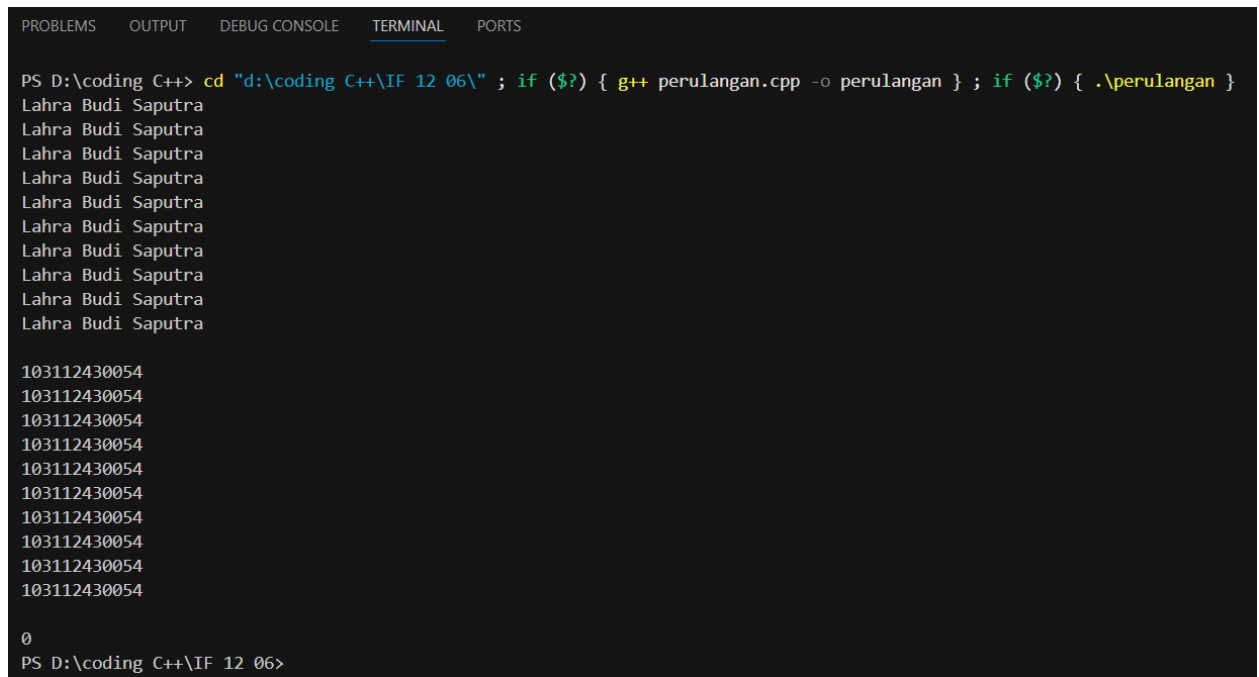
```
#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 10; i++)
    {
        cout << "Lahra Budi Saputra" << endl;
    }
    cout << endl;

    int i = 0;
    while (i < 10)
    {
        cout << "103112430054" << endl;
        i++;
    }
    cout << endl;
```

```
int j = 0;
do
{
    cout << j;
} while (j != 0);
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\coding C++> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ perulangan.cpp -o perulangan } ; if ($?) { .\perulangan }
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
Lahra Budi Saputra
103112430054
103112430054
103112430054
103112430054
103112430054
103112430054
103112430054
103112430054
103112430054
103112430054
0
PS D:\coding C++\IF 12 06>
```

Deskripsi:

Program di atas bertujuan untuk mendemonstrasikan penggunaan tiga jenis perulangan, yaitu for, while, dan do-while. Pada bagian pertama, digunakan perulangan for yang mencetak teks “Lahra Budi Saputra” sebanyak sepuluh kali, di mana variabel kontrol i akan bertambah dari 0 hingga 9. Selanjutnya, program menggunakan perulangan while yang berfungsi mencetak “103112430054” juga sebanyak sepuluh kali. Kondisi perulangan ini akan terus berjalan selama nilai i lebih kecil dari 10, dan setiap iterasi i akan bertambah satu.

Pada bagian terakhir, program memanfaatkan perulangan do-while untuk mencetak angka 0. Perulangan ini selalu dijalankan minimal sekali meskipun kondisi yang diperiksa (j != 0) bernilai salah, sehingga hasil akhirnya hanya menampilkan angka nol satu kali. Program ini menggambarkan perbedaan mendasar dari ketiga struktur

perulangan, yaitu for digunakan ketika jumlah iterasi sudah diketahui, while mengecek kondisi sebelum perulangan dilakukan, dan do-while tetap mengeksekusi minimal satu kali meskipun kondisi tidak terpenuhi.

Guided 3

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

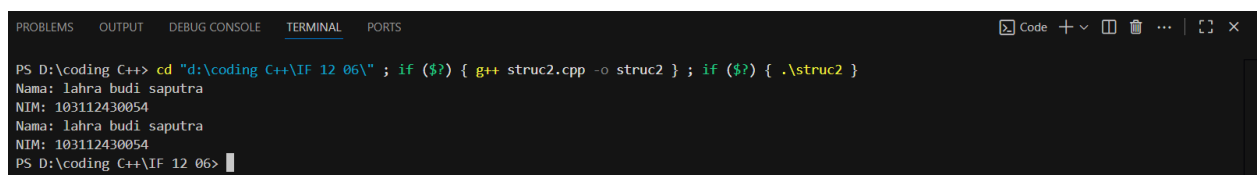
struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    // mhs.nama = "lahra";
    // mhs.NIM = "103112430054";

    cout << "Nama: ";
    getline(cin, mhs.nama);
    cout << "NIM: ";
    cin >> mhs.NIM;

    cout << "Nama: " << mhs.nama << endl
         << "NIM: " << mhs.NIM;
}
```

Screenshots Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\coding C++> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ struc2.cpp -o struc2 } ; if ($?) { .\struc2 }
Nama: lahra budi saputra
NIM: 103112430054
Nama: lahra budi saputra
NIM: 103112430054
PS D:\coding C++\IF 12 06> |
```

Deskripsi:

Program di atas berfungsi untuk mendemonstrasikan penggunaan struct sebagai wadah penyimpanan data. Struct bernama Mahasiswa dibuat dengan dua anggota, yaitu nama bertipe string dan NIM juga bertipe string. Pada fungsi main(), sebuah variabel mhs bertipe Mahasiswa dideklarasikan untuk menampung data mahasiswa. Program kemudian meminta pengguna untuk memasukkan nama melalui fungsi getline(cin, mhs.nama) agar dapat membaca input berupa string yang mengandung spasi. Setelah

itu, program meminta pengguna memasukkan NIM dengan perintah `cin >> mhs.NIM`. Terakhir, program menampilkan kembali data yang sudah diinput berupa Nama dan NIM mahasiswa ke layar. Dengan demikian, program ini memperlihatkan bagaimana struct dapat digunakan untuk menyimpan dan menampilkan data secara terorganisir dalam bentuk objek sederhana.

Guided 4

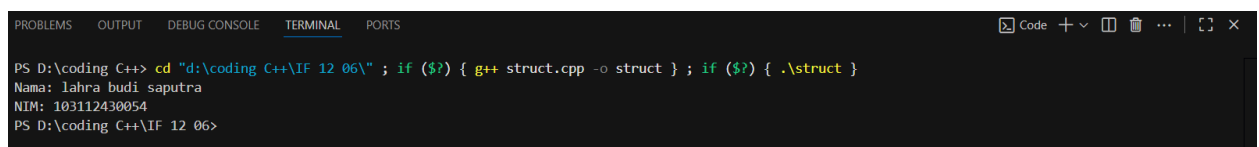
```
#include <iostream>
using namespace std;

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    mhs.nama = "lahra budi saputra";
    mhs.NIM = "103112430054";

    cout << "Nama: " << mhs.nama << endl
         << "NIM: " << mhs.NIM;
}
```

Screenshots Output



```
PS D:\coding C++> cd "d:\coding C++\IF 12 06\" ; if ($?) { g++ struct.cpp -o struct } ; if ($?) { .\struct }
Nama: lahra budi saputra
NIM: 103112430054
PS D:\coding C++\IF 12 06>
```

Deskripsi:

Program di atas merupakan contoh penggunaan struct untuk menyimpan dan menampilkan data mahasiswa. Struct bernama Mahasiswa memiliki dua anggota, yaitu nama dan NIM yang bertipe string. Pada fungsi `main()`, dibuat sebuah variabel `mhs` bertipe Mahasiswa. Nilai untuk anggota struct langsung diberikan melalui assignment, yaitu `mhs.nama = "lahra budi saputra";` dan `mhs.NIM = "103112430054";`. Setelah itu, program menampilkan data mahasiswa tersebut ke layar dengan menggunakan `cout`. Hasil output yang ditampilkan adalah nama lengkap mahasiswa dan NIM sesuai nilai yang telah diisikan sebelumnya.

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

Unguided 1

```
#include <iostream>

int main() {
    float num1, num2;


    std::cout << "Masukkan bilangan pertama: ";
    std::cin >> num1;
    std::cout << "Masukkan bilangan kedua: ";
    std::cin >> num2;

    float penjumlahan = num1 + num2;
    float pengurangan = num1 - num2;
    float perkalian = num1 * num2;
    float pembagian = (num2 != 0) ? (num1 / num2) : 0;

    std::cout << "\nHasil operasi:" << std::endl;
    std::cout << "Penjumlahan (" << num1 << " + " << num2 << "): " <<
penjumlahan << std::endl;
    std::cout << "Pengurangan (" << num1 << " - " << num2 << "): " <<
pengurangan << std::endl;
    std::cout << "Perkalian (" << num1 << " * " << num2 << "): " << perkalian <<
std::endl;
    std::cout << "Pembagian (" << num1 << " / " << num2 << "): " << pembagian
<< std::endl;

    return 0;
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\coding C++> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal01.cpp -o soal01 } ; if ($?) { .\soal01 }
Masukkan bilangan pertama: 5.5
Masukkan bilangan kedua: 2.0

Hasil operasi:
Penjumlahan (5.5 + 2): 7.5
Pengurangan (5.5 - 2): 3.5
Perkalian (5.5 * 2): 11
Pembagian (5.5 / 2): 2.75
PS D:\coding C++\Laprak dong> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal01.cpp -o soal01 } ; if ($?) { .\soal01 }
Masukkan bilangan pertama: 10.0
Masukkan bilangan kedua: 0

Hasil operasi:
Penjumlahan (10 + 0): 10
Pengurangan (10 - 0): 10
Perkalian (10 * 0): 0
Pembagian (10 / 0): 0
PS D:\coding C++\Laprak dong> |
```

Deskripsi:

Program diatas berfungsi untuk melakukan perhitungan aritmatika sederhana menggunakan dua bilangan bertipe float. Pertama, program meminta pengguna untuk memasukkan dua bilangan. Input tersebut disimpan dalam variabel num 1 dan num2. Setelah itu, program melakukan empat jenis operasi dasar, yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Pada bagian pembagian digunakan operator ternary (num2 != 0) ? (num1 / num2) : 0 untuk menghindari kesalahan saat bilangan kedua bernilai nol.

Hasil dari setiap operasi kemudian ditampilkan ke layar dengan format yang jelas, menampilkan operasi yang dilakukan sekaligus hasil perhitungannya.

Jika pengguna memasukkan bilangan kedua sebagai 0, maka hasil pembagian akan ditampilkan sebagai 0 untuk menghindari error pembagian dengan nol.

Unguided 2

```
#include <iostream>
#include <string>

int main() {
    int num;

    std::cout << "Masukkan hanya angka 0-100: ";
    std::cin >> num;

    if (num < 0 || num > 100) {
        std::cout << "Input salah. Harus berupa bilangan bulat positif 0-100." <<
std::endl;
        return 1;
    }

    std::string result;

    std::string ones[20] = {
        "", "satu", "dua", "tiga", "empat", "lima", "enam",
        "tujuh", "delapan", "sembilan", "sepuluh", "sebelas",
        "dua belas", "tiga belas", "empat belas", "lima belas",
        "enam belas", "tujuh belas", "delapan belas", "sembilan belas"
    };

    std::string tens[10] = {
        "", "", "dua puluh", "tiga puluh", "empat puluh",
        "lima puluh", "enam puluh", "tujuh puluh",

```



```

        "delapan puluh", "sembilan puluh"
    };

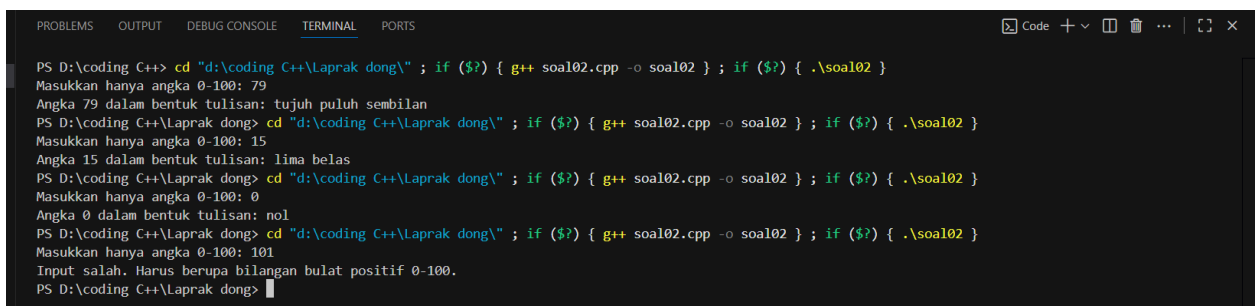
    if (num == 0) {
        result = "nol";
    } else if (num == 100) {
        result = "seratus";
    } else if (num < 20) {
        result = ones[num];
    } else {
        int ten = num / 10;
        int one = num % 10;
        result = tens[ten];
        if (one != 0) {
            result += " " + ones[one];
        }
    }
}

std::cout << "Angka " << num << " dalam bentuk tulisan: " << result <<
std::endl;

return 0;
}

```

Screenshots Output



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\coding C++> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal02.cpp -o soal02 } ; if ($?) { .\soal02 }
Masukkan hanya angka 0-100: 79
Angka 79 dalam bentuk tulisan: tujuh puluh sembilan
PS D:\coding C++\Laprak dong> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal02.cpp -o soal02 } ; if ($?) { .\soal02 }
Masukkan hanya angka 0-100: 15
Angka 15 dalam bentuk tulisan: lima belas
PS D:\coding C++\Laprak dong> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal02.cpp -o soal02 } ; if ($?) { .\soal02 }
Masukkan hanya angka 0-100: 0
Angka 0 dalam bentuk tulisan: nol
PS D:\coding C++\Laprak dong> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal02.cpp -o soal02 } ; if ($?) { .\soal02 }
Masukkan hanya angka 0-100: 101
Input salah. Harus berupa bilangan bulat positif 0-100.
PS D:\coding C++\Laprak dong>

```

Deskripsi:

Program diatas berfungsi untuk mengubah bilangan bulat dalam rentang 0 sampai 100 menjadi bentuk tulisan dalam bahasa Indonesia. Prosesnya dimulai dengan meminta input dari pengguna, kemudian dilakukan validasi agar hanya angka yang sesuai ketentuan yang dapat diproses. Jika input berada di luar rentang 0–100, maka program menampilkan pesan kesalahan dan langsung berhenti. Untuk mengonversi angka ke tulisan, program menggunakan dua array string, yaitu ones yang berisi kata-kata bilangan dari 1 hingga 19, serta tens yang berisi kata-kata bilangan puluhan dari 20 hingga 90. Logika konversi dibagi menjadi beberapa kondisi: jika angka yang

dimasukkan adalah 0, maka hasilnya “nol”; jika angka sama dengan 100, hasilnya “seratus”; jika angka kurang dari 20, hasil langsung diambil dari array ones; sedangkan jika angka 20 ke atas, nilai dibagi menjadi puluhan dan satuan, lalu digabungkan sesuai dengan kata yang ada pada array. Dengan cara ini, program mampu menampilkan angka dalam bentuk kata, misalnya input 17 akan menghasilkan “tujuh belas”, input 42 menjadi “empat puluh dua”, dan input 100 terbaca sebagai “seratus”.

Unguided 3

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

int main() {
    int n;
    cout << "Masukkan angka: ";
    cin >> n;

    for (int i = n; i >= 1; i--) {

        for (int s = 0; s < n - i; s++) cout << " ";

        for (int j = i; j >= 1; j--) cout << j << " ";

        cout << "* ";

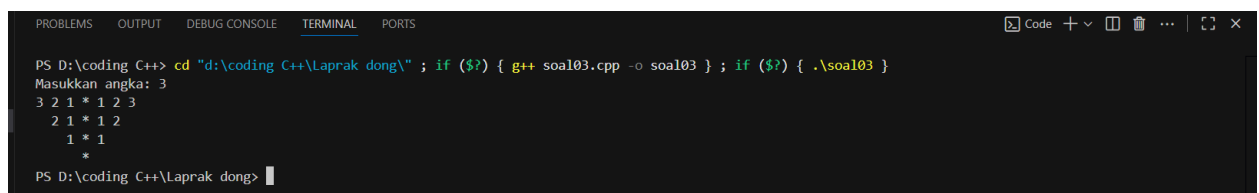
        for (int j = 1; j <= i; j++) cout << j << " ";

        cout << endl;
    }

    for (int s = 0; s < n; s++) cout << " ";
    cout << "*" << endl;

    return 0;
}
```

Screenshots Output



```
PS D:\coding C++\Laprak dong> cd "d:\coding C++\Laprak dong\" ; if ($?) { g++ soal03.cpp -o soal03 } ; if ($?) { .\soal03 }
Masukkan angka: 3
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*

PS D:\coding C++\Laprak dong>
```

Deskripsi:

Program di atas berfungsi untuk menampilkan pola angka simetris dengan tanda bintang (*) yang berada tepat di tengah. Program dimulai dengan meminta pengguna memasukkan sebuah bilangan bulat n yang akan menentukan ukuran pola. Pola dicetak dengan menggunakan perulangan `for` yang berjalan dari n hingga 1. Pada setiap iterasi, bagian pertama program mencetak sejumlah spasi agar susunan pola berbentuk segitiga terbalik dan tetap simetris. Setelah itu, program mencetak deret angka menurun dari i hingga 1, lalu mencetak sebuah tanda * di tengah, kemudian diikuti oleh deret angka menaik dari 1 hingga i . Setelah semua baris selesai dicetak, program menambahkan satu baris terakhir berupa tanda * yang juga disejajarkan dengan pola di atasnya. Dengan demikian, program ini menghasilkan pola angka yang teratur, di mana angka menurun berada di sisi kiri, angka menaik berada di sisi kanan, dan tanda * selalu berada di tengah sebagai pemisah.

D. Kesimpulan

Dari rangkaian praktikum ini dapat disimpulkan bahwa pemahaman mengenai dasar pemrograman C++ melalui penggunaan Code:Blocks IDE sangat penting sebagai fondasi dalam mempelajari konsep lanjutan seperti struktur data dan algoritma. Melalui *guided* maupun *unguided*, mahasiswa telah mempraktikkan berbagai topik dasar, seperti penggunaan percabangan (*if-else*, *switch-case*), perulangan (*for*, *while*, *do-while*), struktur data sederhana (*struct*), operasi aritmatika, konversi angka ke bentuk teks, hingga pembuatan pola dengan *nested loop*. Praktikum ini tidak hanya melatih logika berpikir komputasional, tetapi juga memberikan pemahaman mengenai implementasi nyata dari konsep teori. Dengan demikian, mahasiswa diharapkan mampu mengembangkan keterampilan pemrograman dasar sebagai bekal untuk mempelajari materi lanjutan.

E. Referensi

TutorialsPoint. (2023). C++ Programming Language.
https://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm

TutorialsPoint. (2023). C++ Programming Language.
https://www.tutorialspoint.com/cplusplus/cpp_pointers.htm