

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Rikza Nur Zaki

NIM : 103112430030

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

C++ adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Bjarne Stroustrup pada awal 1980-an sebagai ekstensi dari bahasa C. Bahasa ini menambahkan fitur-fitur seperti kelas, enkapsulasi, pewarisan, dan polimorfisme yang memungkinkan pemrograman modular dan terstruktur dengan kontrol langsung terhadap manajemen memori. C++ adalah bahasa yang dikompilasi sehingga memberikan performa tinggi dan digunakan luas dalam pengembangan perangkat lunak sistem dan aplikasi yang membutuhkan efisiensi dan kecepatan. Dengan adanya Standard Template Library (STL), C++ juga mendukung pemrograman generic yang memungkinkan reuse kode dan pengelolaan koleksi data yang efisien.

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

Guided 1

```
#include <iostream>
using namespace std;
int main()
{
    int hari;
    cout << "Hari 1-7: ";
    cin >> hari;
    if (hari == 7)
    {
        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

```
Hari 1-7: 7
Hari Minggu
Hari Minggu
PS D:\Kuliah\Struktur Data> cd "d:\Kuliah\Struktur Data\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Hari 1-7: 2
Hari Kerja
Hari Kerja
PS D:\Kuliah\Struktur Data> |
```

Deskripsi:

Ini adalah codingan untuk menentukan hari minggu dan hari kerja, jika hari kerja maka di hari ke 1-6, dan hari minggu adalah hari ke 7

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

Guided 2

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

int main()
{
    for (int i = 0; i < 10; i++)
    {
        cout << "Rikza Nur Zaki" << endl;
    }
    cout << endl;

    int i = 0;
    while (i < 10)
    {
        cout << "103112430030" << endl;
        i++;
    }
    cout << endl;

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

Screenshots Output

```
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki  
Rikza Nur Zaki
```

```
103112430030  
103112430030  
103112430030  
103112430030  
103112430030  
103112430030  
103112430030  
103112430030  
103112430030  
103112430030
```

```
0  
PS D:\Kuliah\Struktur Data> |
```

Deskripsi:

Ini adalah codingan untuk membuat looping (for, while, & do while) yang mencetak sebanyak 10x untuk nim dan nama, dan angka 0 1x.

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

Guided 3

```

#include <iostream>
using namespace std;

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    mhs.nama = "Rikza";
    mhs.NIM = "103112430030";

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

```

Screenshots Output

```

Nama : Rikza
NIM : 103112430030
PS D:\Kuliah\Struktur Data> 

```

Deskripsi:

Ini adalah codingan struktur data sederhana untuk menyimpan data mahasiswa.

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

Guided 4

```

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

struct Mahasiswa
{
    string nama;
    string NIM;
};

int main()
{
    Mahasiswa mhs;
    // mhs.nama = "Rikza";
    // mhs.NIM = "103112430030";

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

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

```

Screenshots Output

```

Nama : Rikza Nur Zaki
NIM : 103112430030
Nama : Rikza Nur Zaki
NIM : 103112430030
PS D:\Kuliah\Struktur Data>

```

Deskripsi:

Ini adalah codingan untuk user agar memasukan data(nim & nama) untuk struktur mahasiswa.

- G. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)
Unguided 1

```
using namespace std;

int main() {
    float a, b;
    cout << "Masukkan dua bilangan float: ";
    cin >> a >> b;

    cout << "Penjumlahan: " << a + b << endl;
    cout << "Pengurangan: " << a - b << endl;
    cout << "Perkalian : " << a * b << endl;
    cout << "Pembagian : " << a / b << endl;
}
```

Screenshots Output

```
Masukkan dua bilangan: 10 30
Penjumlahan: 40
Pengurangan: -20
Perkalian : 300
Pembagian : 0.333333
PS D:\Kuliah\Struktur Data>
```

Deskripsi:

Ini adalah codingan calculator sederhana yang berguna untuk mencari + - x / , pengguna cukup memasukkan 2 angka bilangan dan program akan mengeluarkan hasil dari 4 operasi dasar diatas.

- H. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)
Unguided 2

```

#include <iostream>
using namespace std;

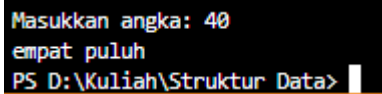
int main() {
    int n;
    cout << "Masukkan angka: ";
    cin >> n;

    switch (n) {
        case 0: cout << "nol"; break;
        case 1: cout << "satu"; break;
        case 2: cout << "dua"; break;
        case 3: cout << "tiga"; break;
        case 4: cout << "empat"; break;
        case 5: cout << "lima"; break;
        case 6: cout << "enam"; break;
        case 7: cout << "tujuh"; break;
        case 8: cout << "delapan"; break;
        case 9: cout << "sembilan"; break;
        case 10: cout << "sepuluh"; break;
        case 11: cout << "sebelas"; break;
        case 100: cout << "seratus"; break;
        default:
            if (n < 20) {
                cout << n % 10 << " belas";
            } else if (n < 100) {
                int puluhan = n / 10;
                int satuan = n % 10;
                switch (puluhan) {
                    case 2: cout << "dua puluh"; break;
                    case 3: cout << "tiga puluh"; break;
                    case 4: cout << "empat puluh"; break;
                    case 5: cout << "lima puluh"; break;
                    case 6: cout << "enam puluh"; break;
                    case 7: cout << "tujuh puluh"; break;
                    case 8: cout << "delapan puluh"; break;
                    case 9: cout << "sembilan puluh"; break;
                }
                if (satuan > 0) {
                    cout << " ";
                    switch (satuan) {
                        case 1: cout << "satu"; break;
                        case 2: cout << "dua"; break;
                        case 3: cout << "tiga"; break;
                        case 4: cout << "empat"; break;
                        case 5: cout << "lima"; break;
                        case 6: cout << "enam"; break;
                        case 7: cout << "tujuh"; break;
                        case 8: cout << "delapan"; break;
                        case 9: cout << "sembilan"; break;
                    }
                }
            }
    }

    cout << endl;
    return 0;
}

```


Screenshots Output

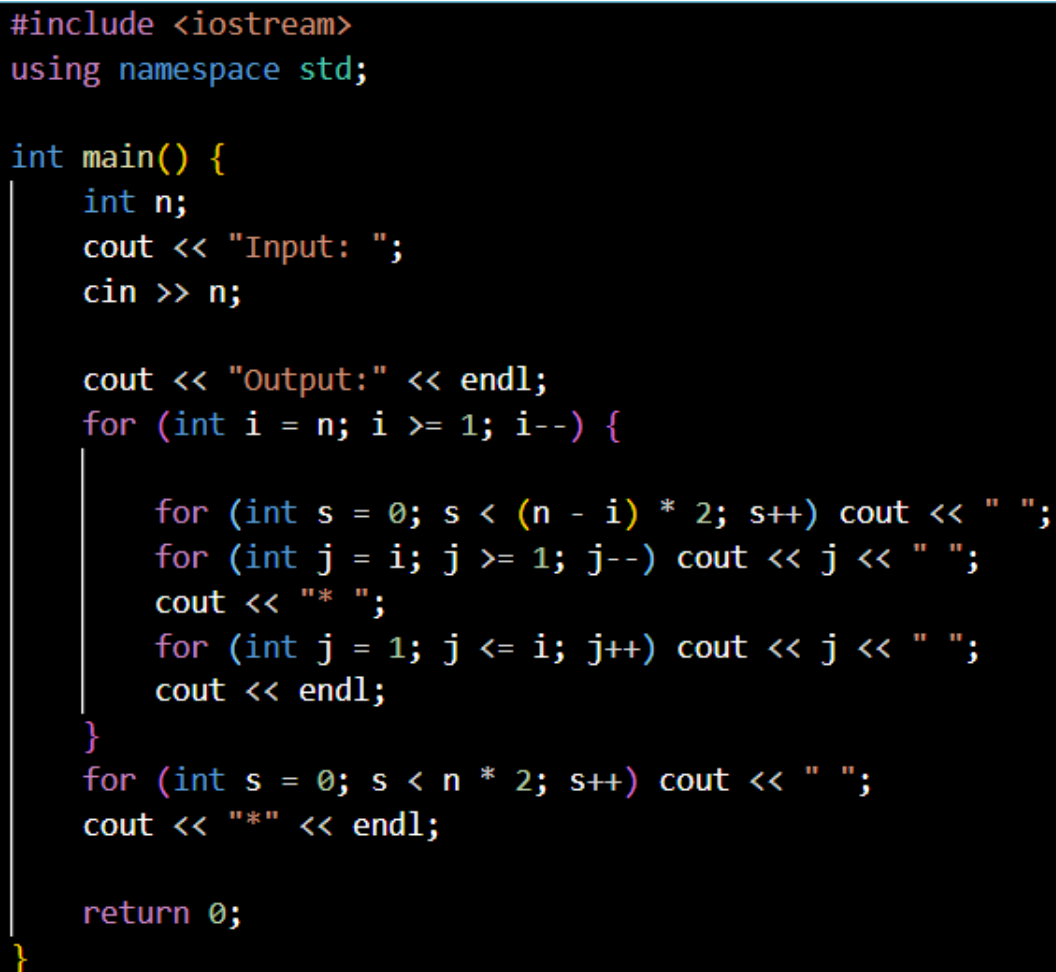


```
Masukkan angka: 40
empat puluh
PS D:\Kuliah\Struktur Data>
```

Deskripsi:

Codingan diatas bertujuan untuk mengubah angka menjadi tulisan dengan menggunakan metode switch case dan if else.

- I. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)
Unguided 3



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

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

    cout << "Output:" << endl;
    for (int i = n; i >= 1; i--) {
        for (int s = 0; s < (n - i) * 2; 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 * 2; s++) cout << " ";
    cout << "*" << endl;

    return 0;
}
```

Screenshots Output

```

Input: 4
Output:
4 3 2 1 * 1 2 3 4
  3 2 1 * 1 2 3
    2 1 * 1 2
      1 * 1
        *
PS D:\Kuliah\Struktur Data>

```

Deskripsi:

Codingan diatas merupakan codingan yang membuat tampilan berbentuk seperti segitiga kebalik, dengan menggunakan perulangan pada bagian baris, spasi, dan angka kiri kanan.

J. Kesimpulan

C++ merupakan bahasa yang sangat kuat dan fleksibel, ideal untuk aplikasi yang menuntut performa dan kontrol sumber daya yang tinggi. Kelebihan utamanya meliputi kemampuan pemrograman berorientasi objek, pengelolaan memori manual, dan dukungan pustaka yang luas. Namun, kompleksitas bahasa ini menyebabkan kurva pembelajaran yang cukup tinggi, terutama untuk pemula. C++ tetap menjadi pilihan utama untuk pengembangan perangkat lunak tingkat rendah dan aplikasi berperforma tinggi, meskipun kini banyak alternatif bahasa pemrograman yang lebih mudah digunakan untuk pengembangan aplikasi tingkat tinggi.

K. Referensi

- Stroustrup, Bjarne. The C++ Programming Language
- Gurtovoy, Aleksey (2005). C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond. Addison-Wesley