

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :
NAMA : IVAN RAMADHAN
NIM : 103112400186

Dosen
WAHYU ANDI SAPUTRA

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

A. Dasar Teori

C++ adalah pengembangan dari bahasa C yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

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

Guided 1

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

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

    cout << "\nHasil Operasi Aritmatika:" << endl;
    cout << a << " + " << b << " = " << a + b << endl;
    cout << a << " - " << b << " = " << a - b << endl;
    cout << a << " * " << b << " = " << a * b << endl;
    if(b != 0)
        cout << a << " / " << b << " = " << a / b << endl;
    else
        cout << "Tidak bisa dibagi dengan nol" << endl;

    return 0;
}
```

Screenshots Output

```
pyter-m11
Masukkan bilangan pertama: 17
Masukkan bilangan kedua: 35

Hasil Operasi Aritmatika:
17 + 35 = 52
17 - 35 = -18
17 * 35 = 595
17 / 35 = 0.485714
PS C:\xampp\htdocs\Modul Alpro\C++ SMT 3> 
```

Deskripsi:

Program ini bertujuan untuk menerima **dua buah bilangan float** dari pengguna. Setelah itu, program menghitung dan menampilkan hasil dari:

- Penjumlahan ($a + b$)
- Pengurangan ($a - b$)
- Perkalian ($a * b$)
- Pembagian (a / b)

Jika bilangan kedua bernilai **0**, program akan menampilkan pesan bahwa pembagian tidak bisa dilakukan (untuk menghindari error pembagian dengan nol).

Intinya: Program ini melatih penggunaan **tipe data float**, **operasi aritmatika dasar**, serta **penggunaan kondisi if** dalam pemrograman.

Guided 2

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

int main() {
    string satuan[] = {"nol", "satu", "dua", "tiga", "empat", "lima",
                       "enam", "tujuh", "delapan", "sembilan"};
    string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas",
                        "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan
belas"};
    string puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh",
                        "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};

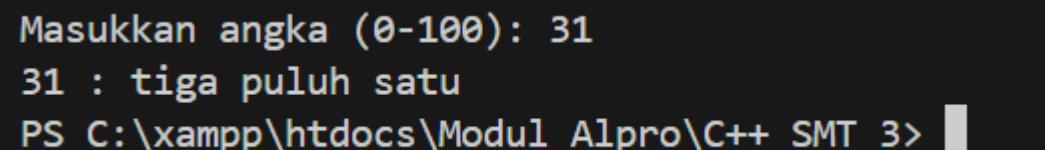
    int n;
    cout << "Masukkan angka (0-100): ";
    cin >> n;

    string hasil;
    if(n < 10) {
        hasil = satuan[n];
    } else if(n < 20) {
        hasil = belasan[n - 10];
    } else if(n < 100) {
        hasil = puluhan[n / 10];
        if(n % 10 != 0)
            hasil += " " + satuan[n % 10];
    } else if(n == 100) {
        hasil = "seratus";
    } else {
        hasil = "Angka di luar jangkauan";
    }

    cout << n << " : " << hasil << endl;

    return 0;
}
```

Screenshots Output



```
Masukkan angka (0-100): 31
31 : tiga puluh satu
PS C:\xampp\htdocs\Modul Alpro\C++ SMT 3>
```

Deskripsi:

Program menerima input berupa **bilangan bulat (0 sampai 100)** dari pengguna. Kemudian, angka tersebut akan dikonversi menjadi **bentuk tulisan dalam bahasa Indonesia**.

Contoh:

- Input: 79
- Output: tujuh puluh sembilan

Program ini menggunakan:

- Array/string untuk menyimpan kata dasar (satuan, belasan, puluhan).
- Logika **if-else** untuk menentukan kategori angka:
 - 0–9 → langsung ambil dari array satuan.
 - 10–19 → gunakan array belasan.
 - 20–99 → ambil kata puluhan dan tambahkan satuan.
 - 100 → khusus ditulis “seratus”.

Intinya: Program ini melatih **struktur kontrol if-else, array/string, dan pemrosesan angka menjadi kata**.

Guided 3

```
#include <iostream>

using namespace std;

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

    cout << "Output:" << endl;
    for (int i = n; i >= 1; i--) {
        // bagian kiri
        for (int j = i; j >= 1; j--) {
            cout << j << " ";
        }
        // tanda *
        cout << "* ";
        // bagian kanan
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }
        cout << endl;
    }
    // tambahan bintang terakhir
    cout << "*" << endl;

    return 0;
}
```

Screenshots Output

```
preter=mi'
Input: 3
Output:
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*
PS C:\xampp\htdocs\Modul Alpro\C++ SMT 3> 
```

Deskripsi:

Program ini meminta satu bilangan bulat positif (**n**) dari pengguna dan kemudian menggunakan perulangan bersarang (*nested loops*) untuk membuat pola pada output konsol.

Program menerima input berupa **bilangan bulat positif n**.

Program mencetak pola angka berbentuk **mirror** dengan aturan:

- Bagian kiri menampilkan angka menurun dari **n** ke **1**.
- Di tengah selalu ada simbol *****.
- Bagian kanan menampilkan angka naik dari **1** ke **n**.
- Setelah satu baris selesai, nilai **n** berkurang sehingga pola makin kecil sampai hanya ***** yang tersisa di baris terakhir.

Intinya: Program ini melatih penggunaan **nested loop (for di dalam for)**, kontrol urutan angka **menurun dan menaik**, serta **logika pola teks..**

C. Kesimpulan

Dari praktikum modul ini dapat disimpulkan bahwa mahasiswa telah memahami elemen-elemen dasar pemrograman C++, meliputi deklarasi variabel, tipe data, operator, struktur kendali (percabangan), dan perulangan. Pemahaman dasar ini menjadi fondasi penting dalam membangun program yang lebih kompleks.

Meskipun konsep percabangan dan perulangan sering kali menjadi tantangan bagi pemula, pendekatan visualisasi dan contoh konkret terbukti efektif untuk mempermudah pemahaman.

Dengan demikian, modul ini berhasil membekali mahasiswa dengan kompetensi dasar yang krusial, khususnya dalam pengembangan logika algoritma, yang nantinya akan menjadi landasan kuat untuk mempelajari struktur data dan topik lanjutan lainnya.

D. Referensi

- Cetin, I. (2020). *Teaching Loops Concept through Visualization Construction*. Informatics in Education.
- Rudder, A., Bernard, M., & Mohammed, S. (2007). *Teaching Programming Using Visualization*.
- Bellström, P., & Thorén, C. (2010). *On the Importance of Visualizing in Programming Education*.
- Yamashita, K. et al. (2016). *Code-reading support environment visualizing nested loops*.