

LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL I
CODE BLOCKS DAN PENGENALAN BAHASA C++
(BAGIAN PERTAMA)



Disusun Oleh :
Nama : Izzah Minkhotun Fannisa
NIM : 103112400198

Dosen
FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

A. Dasar Teori

C++ merupakan bahasa pemrograman yang pertama kali dikembangkan oleh Bjarne Stroustrup, seorang ilmuwan komputer asal Denmark. Bahasa pemrograman C++ disempurnakan dari bahasa C yang di temukan oleh Dennis Ritchie pada awal 1970-an. Lalu resmi dirilis pada tahun 1985. C++ cukup banyak digunakan karena kemampuannya dalam memberikan kecepatan eksekusi tinggi serta dukungan terhadap manajemen memori tingkat rendah.

C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (menggunakan fungsi biasa), berorientasi objek (menggunakan *class* dan *object*), atau bahkan gabungan keduanya. C++ memiliki dasar-dasar seperti variabel, operator percabangan (*if*, *switch*), perulangan (*for*, *while*), dan bisa memakai *class* untuk membuat objek.

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

1. Guide 1

a. Source code

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

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

    int i = 0;
    while (i < 10)
    {
        cout << "103112400198" << endl;
        i++;
    }
    cout << endl;

    int j = 0;

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

}
```

b. Screenshot Output

```
.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-pggae40g.anh' '--stdout=Microsoft-MIEngine-Out-r50oq0ze.0ak' '--stderr=Microsoft-MIEngine-Error-cx4qbtqv.lfd' '--pid=Microscft-MIEngine-Pid-os1yvpey.t2u' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
Izzah

103112400198
103112400198
103112400198
103112400198
103112400198
103112400198
103112400198
103112400198
103112400198
103112400198
```

c. Deskripsi

Program di atas menggunakan 3 jenis perulangan (looping), berupa *for loop*, *while loop*, dan *do-while loop*.

1. For Loop

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

Bagian tersebut menggunakan perulangan for yang mencetak kata “Izzah” sebanyak 10 kali.

2. While Loop

```
int i = 0;
while (i < 10)
{
    cout << "103112400198" << endl;
    i++;
}
```

Bagian ini menggunakan perulangan while yang mencetak NIM 103112400198 sebanyak 10 kali. Perulangan while sering digunakan jika kondisi berhenti baru dipastikan saat program berjalan (runtime).

3. Do-While Loop

```
int j = 0;

do
{
    cout << j;
```

```
    } while (j !=0);
```

Bagian di atas digunakan untuk mencetak nilai variable j yang (nilai 0). Perulangan do-while selalu menjalankan isi blok minimal sekali, meskipun kondisi berhenti ($j \neq 0$) tidak terpenuhi. Program ini memperlihatkan 3 perbedaan dasar dalam bentuk loop C++, yaitu:

- for : iterasi dengan batas yang jelas.
- while : iterasi dengan kondisi yang di cek di awal.
- do-while : iterasi dengan kondisi yang dicek di akhir (selalu berjalan minimal sekali).

2. Guide 2

a. Source Code

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

int main() {

    int hari;
    cout << "hari 1-7" ;
    cin >> hari;
    if (hari == 7)
    {
        cout << "hari senin\n";
    }
    else
    {
        cout << "hari kerja\n";
    }

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

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

    }
    return 0;
}
```

b. Screenshot Output

```
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 1> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.27.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-wjtrxupw.2oz' '--stdout=Microsoft-MIEngine-Out-s1ukrnj0t.ael' '--stderr=Microsoft-MIEngine-Error-ne4yhquu.rds' '--pid=Microsoft-MIEngine-Pid-t4hwev5n.xve' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi' hari 1-7
```

c. Deskripsi

Program di atas digunakan untuk memasukan sebuah angka antara 1-7 yang mewakili hari dalam seminggu, menggunakan percabangan dengan if-else dan switch case.

1. If-Else

```
if (hari == 7)
{
    cout << "hari senin\n";
}
else
{
    cout << "hari kerja\n";
}
```

Jika input sama dengan 7 akan mencetak “hari senin”. Jika tidak, akan mencetak “hari kerja”.

2. Switch-Case

```
switch (hari)
{
    case 7:
        cout << "hari selasa\n";
        break;

    default:
        cout << "hari kerja\n";
        break;
}
```

Memeriksa kembali nilai hari dengan switch:

- Jika hari == 7, maka mencetak “hari selasa”
- Jika tidak, masuk ke default dan mencetak “hari kerja”

Output yang dihasilkan jika menginputkan 7 di if-else akan menampilkan “hari senin” dan switch-case “hari selasa”. Hal tersebut menunjukkan bahwa program belum konsisten, tetapi fungsinya memperlihatkan cara kerja dua bentuk percabangan.

3. Guide 3

a. Source Code

```

#include <iostream>
using namespace std;

struct mahasiswa
{
    string nama;
    string nim;
};

int main()
{
    mahasiswa mhs;
    mhs.nama = "Izzah";
    mhs.nim = "103112400198";

    cout << "nama :" << mhs.nama << endl;
    cout << "nim :" << mhs.nim << endl;
}

```

b. Screenshot Output

```

// MinGW 5.0.2 (GDB) 2015-02-26 (cvs-20150226-000)
// Microsoft-MIEngine-Out-xif1on5s.vm4' '--stderr=Microsoft-MIEngine-Pid-c4c50frr.00e' '--dbgExe=C:\Users\ACER\mingw32\b:
nama :Izzah
nim :103112400198
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 1>

```

c. Deskripsi

Program ini digunakan untuk menyimpan data mahasiswa yang terdiri dari atribut nama dan NIM. Pada fungsi main, sebuah mahasiswa dibuat, kemudian nilainya diisi dengan "Izzah" untuk nama dan "103112400198" untuk NIM. Data tersebut kemudian ditampilkan ke layar menggunakan cout. Program ini menunjukkan konsep dasar bagaimana struct digunakan untuk mengelompokkan beberapa variabel dalam satu kesatuan data.

4. Guide 4

a. Source code

```

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

struct mahasiswa {
    string nama;
    string nim;
};

```

```

int main() {
    mahasiswa mhs;

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

    cout << "nim : ";
    cin >> mhs.nim;

    cout << "\n--- Data Mahasiswa ---\n";
    cout << "nama : " << mhs.nama << endl;
    cout << "nim : " << mhs.nim << endl;

    return 0;
}

```

b. Screenshot Output

```

ft-MIEngine-Pid-3aitka3o.fra' '--dbgExe=C:\Users\ACER'
nama : Izzah
nim : 103112400198
nama : Izzah
nim : 103112400198
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 1> 

```

c. Deskripsi

Program ini menggunakan struct bernama mahasiswa yang memiliki dua atribut, yaitu nama dan nim. Pada fungsi main, dibuat sebuah objek mhs bertipe mahasiswa. Program kemudian meminta pengguna untuk memasukkan nama menggunakan getline dan NIM menggunakan cin. Data yang sudah dimasukkan kemudian ditampilkan kembali ke layar.

Alur program ini dimulai dengan pembuatan variabel mhs, lalu pengguna diminta mengisi data. Nilai yang diinput akan disimpan pada atribut mhs.nama dan mhs.nim, kemudian ditampilkan dengan perintah cout. Dengan cara ini, program memperlihatkan penggunaan struct sebagai wadah data serta penerapan input-output sederhana di C++.

5. Guide 5

a. Source code

```

#include <iostream>
using namespace std;

int main() {
    float a, b;
    cin >> a >> b;
    cout << "jumlah: " << a + b << endl;
}

```

```

cout << "kurang: " << a - b << endl;
cout << "kali " << a * b << endl;
if (b != 0) {
    cout << "bagi: " << a / b << endl;
} else {
    cout << "bagi: undefined" << endl;
}

}
return 0;
}

```

b. Screenshot Output

```

MIEngine-Pid-1odbxuof.edq' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe'
16
2
jumlah: 18
kurang: 14
kali 32
bagi: 8
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 1> █

```

c. Deskripsi

Program ini meminta dua bilangan sebagai input dari pengguna. Setelah itu, program melakukan operasi aritmatika berupa penjumlahan, pengurangan, dan perkalian. Untuk pembagian, program memeriksa apakah bilangan kedua bernilai nol. Jika tidak nol, hasil pembagian ditampilkan; jika nol, maka ditampilkan pesan "*undefined*"

```

10
0
jumlah: 10
kurang: 10
kali 0
bagi: undefined

```

karena pembagian dengan nol tidak terdefinisi. Program diakhiri dengan menampilkan semua hasil operasi tersebut lalu selesai dijalankan.

6. Guide 6

a. Source code

```

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

// Fungsi untuk mengubah angka menjadi teks
string ubahKeTulisan(int n) {
    string kataSatuan[] = {
        "nol", "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"
};

string kataPuluhan[] = {
    "", "sepuluh", "dua puluh", "tiga puluh", "empat
puluhan",
    "lima puluh", "enam puluh", "tujuh puluh",
"delapan puluh", "sembilan puluh"
};

if (n < 20) {
    return kataSatuan[n];
} else if (n < 100) {
    int puluh = n / 10;
    int sisa = n % 10;
    if (sisa == 0) {
        return kataPuluhan[puluhan];
    } else {
        return kataPuluhan[puluhan] + " " +
kataSatuan[sisa];
    }
} else if (n == 100) {
    return "seratus";
} else {
    return "angka tidak tersedia";
}
}

int main() {
    int angka;
    cin >> angka;
    cout << angka << " : " << ubahKeTulisan(angka) <<
endl;
    return 0;
}

```

b. Screenshot Output

```

MIEngine-Pid-3qmdys4b.vly' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.e
19
19 : sembilan belas
PS C:\Users\ACER\OneDrive\Desktop\L-Minggu 1> █

```

c. Deskripsi

Program tersebut digunakan untuk mengubah bilangan bulat menjadi bentuk tulisan dengan rentang 0-100.

1. Input berupa bilangan bulat

2. Jika angka < 20 → langsung diambil dari array kataSatuan. Jika angka 20–99 → angka dibagi menjadi puluhan (puluhan) dan sisa satuan (sisa). Jika sisa nol, cukup tampilkan puluhan, jika tidak maka puluhan digabungkan dengan satuan. Jika angka = 100 → ditulis sebagai “seratus”. Jika di luar rentang → ditampilkan pesan “angka tidak tersedia”.
3. Output : angka yang diinput beserta hasil yang telah diubah dalam teks.

7. Guide 7

a. Source code

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

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

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

        for (int gap = 0; gap < n - i; gap++) {
            cout << " ";
        }

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

        cout << "* ";

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

        cout << endl;
    }

    for (int gap = 0; gap < n; gap++) {
        cout << " ";
    }
    cout << "*" << endl;

    return 0;
}
```

b. Screenshot Output

```
MIEngine-Pid-qmzpnji.15p' '--dbgExe=C:\Users\ACER\mingw32\b:
Masukkan angka: 3
3 2 1 * 1 2 3
 2 1 * 1 2
    1 * 1
      *
```

c. Deskripsi

Program ini menampilkan pola angka berbentuk cermin (mirror) sesuai bilangan bulat yang diinputkan user.

1. Menginputkan sebuah bilangan bulat n
2. Lalu, program mencetak angka menurun dari n ke 1 di sisi kiri.
3. Di tengah terdapat tanda * sebagai pemisah.
4. Angka dicetak naik dari 1 ke n di sisi kanan.
5. Setiap baris bergeser ke kanan menggunakan spasi sehingga membentuk pola segitiga cermin.
6. Setelah baris 1 * 1, program menambahkan satu baris berisi tanda * di bagian bawah.
7. Pola seperti cermin berbentuk segitiga dari angka sesuai input.

C. Kesimpulan

Dari praktikum Modul 1, saya dapat menyimpulkan bahwa saya dapat memahami cara menggunakan Code::Blocks IDE untuk menuliskan, meng-compile, dan menjalankan program C++. Bahasa C++ memiliki struktur program yang jelas, mulai dari deklarasi library, fungsi utama main(), hingga proses input/output menggunakan cin dan cout. Melalui latihan yang diberikan, saya dapat mengimplementasikan berbagai konsep dasar pemrograman, seperti operasi aritmatika, konversi angka ke tulisan, penggunaan kondisi if-else, serta penerapan perulangan (loop) untuk membentuk pola. Selain itu, kesalahan umum seperti pembagian dengan nol dapat saya tangani agar program tetap berjalan tanpa error. Praktikum ini menjadi dasar penting sebelum mempelajari struktur data yang lebih kompleks, karena pemahaman mengenai variabel, tipe data, operator, input/output, dan kontrol alur program harus saya kuasai terlebih dahulu.

D. Referensi

Domainesia. (2023). Belajar bahasa pemrograman C++.

<https://www.domainesia.com/berita/belajar-bahasa-pemrograman-cpp/>

GeeksforGeeks. (2023). C++.

<https://www.geeksforgeeks.org/cpp/c-plus-plus/>

GeeksforGeeks. (2023). Decision Making in C++ (if, if-else, switch, nested).

<https://www.geeksforgeeks.org/decision-making-cpp/>

GeeksforGeeks. (2023). Loops in C++ (For, While, Do-While).
<https://www.geeksforgeeks.org/cpp-loops/>

GeeksforGeeks. (2023). Structures in C++.
<https://www.geeksforgeeks.org/structures-in-cpp/>

Josuttis, N. M. (2012). The C++ Standard Library: A Tutorial and Reference. Addison-Wesley.

Onero. (2023). C adalah....
<https://onero.id/insight/detail/c-adalah/>

Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley.

W3Schools. (2023). C++ if...else statement.
https://www.w3schools.com/cpp/cpp_conditions.asp

W3Schools. (2023). C++ Loops.
https://www.w3schools.com/cpp/cpp_loops.asp

W3Schools. (2023). C++ Structures (struct).
https://www.w3schools.com/cpp/cpp_structs.asp