

LAPORAN PRAKTIKUM

Modul 1

Pengenalan CPP bagian 1



Disusun Oleh:

Yogi Hafidh Maulana - 2211104061

SE06-02

Dosen :

Yudha Islalmi Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

1. Tujuan

- Mengenal environment Code Blocks dengan baik.
- Memahami cara menggunakan dan troubleshooting Code Blocks IDE.
- Mengimplementasikan operator-operator dalam program.
- Memahami cara membuat program sederhana dalam bahasa C++.
- Memahami penggunaan tipe data dan variabel dalam bahasa C++.
- Menggunakan operator-operator input/output dengan tepat.
- Memahami dan mengimplementasikan fungsi kondisional dalam program.

2. Landasan Teori

- Bahasa C++

Bahasa C++ adalah salah satu bahasa pemrograman komputer yang paling populer di dunia dan dapat digunakan untuk membuat berbagai aplikasi berkinerja tinggi (general-purpose programming language) seperti aplikasi games, browsers, pengolah gambar, software gadget dan masih banyak lagi. Dikembangkan oleh Bjarne Stroustrup, bahasa ini merupakan pengembangan dari bahasa C. Bahasa C++ yang merupakan Object Oriented Programming (OOP) yang artinya C++ mempunyai data dan function yang bergabung dalam kelas dan objek yang sama untuk saling bekerjasama dalam memecahkan sebuah masalah. Pengubahan kode dalam hal ini akan menjadi lebih fleksibel, dikarenakan ketika Anda mengubah fungsi, Anda tidak perlu untuk mengubah keseluruhan dari program. Bahasa C++ merupakan bahasa pemrograman populer dan digunakan oleh perusahaan besar seperti Adobe dan Firefox.

- Code::Blocks

Code::Blocks adalah suatu program lingkungan pengembangan terpadu bebas, nirlaba, bersumber terbuka dan lintas platform. Program yang ditulis dalam C++ beserta wxWidgets untuk GUI-nya ini bisa digunakan bersama dengan berbagai macam kompilator, contohnya GCC dan Visual C++.

- Tipe data

Tipe data adalah pengklasifikasian data berdasarkan jenisnya yang berfungsi untuk memberi tahu komputer bagaimana menafsirkan nilai data. Tipe data merupakan konsep dasar yang penting untuk dipahami dalam dunia pemrograman.

3. Guided

a) Tipe data

Type data	Contoh	Ukuran	Jangkauan nilai
Char	char nama[20];	1 Byte	-128 s.d. +127
Int	int nilai; int jumlah = 0;	2 Byte	-32768 s.d +32767
Long	long selisih;	4 Byte	-2.147.438.648 s.d +2.147.438.647
Float	float jumlah;	4 Byte	3.4e-38 s.d 3.4e+38
Double	double hasil;	8 Byte	1.7e-308 s.d 1.7e+308

```
// Tipe data Integer
int angka = 10;

// Tipe data desimal
float desimal = 10.5;

// Tipe data String
string kalimat = "aldi";

// Tipe data double
double tinggi = 10.4;

// Tipe data char
char jenis_kelamin = 'L';

// Tipe data boolean
bool isSunny = true;

cout << "Angka: " << angka << endl;
```

b) Input Output

Fungsi cin() merupakan fungsi untuk meminta inputan keyboard dari user. Fungsi cout() ini digunakan untuk mencetak data baik yang bertipe numerik, ataupun teks, baik konstanta ataupun variabel.

```
int angka;
cout << "Masukan angka: ";
cin >> angka;
cout << "Angka: " << angka << endl;
getch();
```

c) Oprator

Kategori (Arti)	Operator	Kategori (Arti)	Operator
Panggilan fungsi, subscript <i>array</i> , dan elemen struktur data	() [] ->	Operator Hubungan (sama dengan, tidak sama dengan)	== !=
Operator Unary (NOT, komplemen, negasi, inkremen, dekremen , <i>address</i> , <i>indirection</i>)	!	Operator Bitwise AND	&
	~	Operator Bitwise XOR	^
	-	Operator Bitwise OR	
	++	Operator Logika AND	&&
	--	Operator Logika OR	
	& *	Operator Kondisional	?:
Operator Aritmatika(Perkalian, pembagian, Sisa Pembagian/mod)	* / %	Operator Pengerjaan Aritmatika (<i>assignment</i> , <i>assignment</i> perkalian, <i>assignment</i> pembagian, <i>assignment</i> mod, <i>assignment</i> penjumlahan, <i>assignment</i> pengurangan)	= *= /= %= += -=
Operator Aritmatika (Pertambahan, Pengurangan)	+ -		
Operator Bitwise Pergeseran Bit (shift kiri, shift kanan)	<< >>		&= ^= = <<= >>=
Operasi Hubungan (kurang dari, kurang dari atau sama dengan lebih dari, lebih dari atau sama dengan)	< <= > >=		
		Operator Koma	,

```

int angka = 10;
int angka1 = 15;
int penjumlahan = angka + angka1;
int pengurangan = angka - angka1;
int perkalian = angka * angka1;
int pembagian = angka / angka1;
int modulus = angka % angka1;
int perbandingan = angka == angka1;

cout << "Hasilnya adalah: " << hasil<< endl;

```

d) Boolean

```

bool konsisi1 = true;
bool kondisi2 = true;

bool hasil = konsisi1 && kondisi2;
cout << "Hasilnya adalah: " << boolalpha << hasil << endl;

```

e) If Else (Percabangan)

```
string kata;

cout << "Masukan kata Hallo: " << endl;
cin >> kata;

if(kata == "Hallo"){
    cout << "Kata sesuai" << endl;
} else {
    cout << "Kata tidak sesuai" << endl;
}
```

f) Switch case (Percabangan)

```
int tv;

cout << "daftar chanel TV" << endl;
cout << "1. RCTI" << endl;
cout << "2. Indosiar" << endl;
cout << "Masukan chanel pilihan: ";
cin >> tv;

switch(tv) {
case 1 :
    cout << "Chanell yang dipilih rcti" << endl;
    break;
case 2 :
    cout << "Chanell yang dipilih indosiar" << endl;
    break;
default:
    cout << "Chanel tidak tersedia" << endl;
    break;
}
```

g) For Loop (Looping)

```
for(int i = 5; i > 0; i--){
    cout << i << "Hello Word!" << endl;
}
```

4. Unguided

- a) Buatlah program yang menerima *input*-an dua buah bilangan betipe float, kemudian memberikan *output*-an hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan tersebut.

```
#include <iostream>

using namespace std;

int main() {
    float bilangan1, bilangan2;
    float hasil_tambah, hasil_kurang, hasil_kali, hasil_bagi;

    // Meminta input dari pengguna
    cout << "Masukkan bilangan pertama: ";
    cin >> bilangan1;

    cout << "Masukkan bilangan kedua: ";
    cin >> bilangan2;

    // Melakukan operasi aritmatika
    hasil_tambah = bilangan1 + bilangan2;
    hasil_kurang = bilangan1 - bilangan2;
    hasil_kali = bilangan1 * bilangan2;
    hasil_bagi = bilangan1 / bilangan2;

    // Menampilkan hasil
    cout << "Hasil penjumlahan: " << hasil_tambah << endl;
    cout << "Hasil pengurangan: " << hasil_kurang << endl;
    cout << "Hasil perkalian: " << hasil_kali << endl;
    cout << "Hasil pembagian: " << hasil_bagi << endl;

    return 0;
}
```

Output

```
Masukkan bilangan pertama: 10
Masukkan bilangan kedua: 20
Hasil penjumlahan: 30
Hasil pengurangan: -10
Hasil perkalian: 200
Hasil pembagian: 0.5
PS D:\PROJECT\C++ Project\Pertemuan1>
```

- b) Buatlah sebuah program yang menerima masukan angka dan mengeluarkan *output* nilai angka tersebut dalam bentuk tulisan. Angka yang akan di- *input*-kan user adalah bilangan bulat positif mulai dari 0 s.d 100. contoh:

79 : tujuh puluh Sembilan

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

using namespace std;

string angkaKeTulisan(int angka) {
    string satuan[] = {"", "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 puluhan[] = {"", "", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh",
"delapan puluh", "sembilan puluh"};

    if (angka < 0 || angka > 100) {
        return "Angka harus antara 0 sampai 100";
    } else if (angka < 20) {
        return satuan[angka];
    } else {
        return puluhan[angka / 10] + (angka % 10 != 0 ? " " : "") + satuan[angka % 10];
    }
}

int main() {
    int angka;

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

    string tulisan = angkaKeTulisan(angka);
    cout << angka << " : " << tulisan << endl;

    return 0;
}
```

Output:

```
PS D:\PROJECT\C++ Project\Pertemuan1> & 'c:\Users\LENOVO\.vscode\extensions\ms-vscode.cpptools-1.22.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-tjyntydc.zie' '--stdout=Microsoft-MIEngine-Out-fbrfj2i3.ga3' '--stderr=Microsoft-MIEngine-Error-5cdyci4b.ww3' '--pid=Microsoft-MIEngine-Pid-ye30d2yi.hpw' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan angka (0-100): 69
69 : enam puluh sembilan
PS D:\PROJECT\C++ Project\Pertemuan1>
```

c) Buatlah program yang dapat memberikan *input* dan *output* sbb.

```
input: 3
output:
  3 2 1 * 1 2 3
    2 1 * 1 2
      1 * 1
        *
```

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

int main() {
    int n;

    // Input dari user
    cout << "Input: ";
    cin >> n;

    // Loop untuk setiap baris
    for (int i = n; i >= 1; i--) {
        // Mencetak bagian kiri (menurun)
        for (int j = i; j >= 1; j--) {
            cout << j << " ";
        }

        // Mencetak tanda '*'
        cout << "* ";

        // Mencetak bagian kanan (meningkat)
        for (int j = 1; j <= i; j++) {
            cout << j << " ";
        }

        cout << endl;
    }

    // Mencetak '*' terakhir
    cout << "*" << endl;

    return 0;
}
```

Output :

```
Input: 3
3 2 1 * 1 2 3
  2 1 * 1 2
    1 * 1
      *

PS D:\PROJECT\C++ Project\Pertemuan1>
```


5. Kesimpulan

C++ merupakan bahasa yang dapat memberikan pemahaman mendalam tentang struktur pemrograman. C++ mendukung pemrograman berorientasi objek (OOP) yang memungkinkan penggunaan kembali kode dan membuat program lebih dapat diandalkan. C++ memiliki fitur-fitur seperti fungsi anggota, konstruktor, penentu akses, dan pewarisan yang memperkaya konsep struktur data. C++ merupakan bahasa tingkat tinggi yang memungkinkan programmer untuk mengekspresikan algoritma dan instruksi dengan cara yang lebih dekat dengan pemikiran manusia. C++ dapat mengajarkan keterampilan manajemen memori dan instruksi kompleksitas waktu secara efisien.

C++ memiliki sistem tipe yang ketat yang memastikan keamanan tipe dan memaksa programmer untuk menjelaskan maksud mereka dengan program mereka. C++ juga mendukung konsep tingkat tinggi dengan prinsip zero-overhead . Ini adalah prinsip desain C++ yang menyatakan bahwa Anda tidak membayar apa yang tidak Anda gunakan dan apa yang Anda gunakan sama efisiennya dengan apa yang dapat Anda tulis dengan tangan .

