

**Laporan Praktikum
Modul 1
Pengenal CPP bagian 1**



Disusun Oleh:
Dwi Candra Pratama
2211104035

Dosen Pengampu:
Wahyu Andi Saputra

**Program Studi S1 Software Engineering
Fakultas Informatika
Universitas Telkom Purwokerto**

1. Tujuan

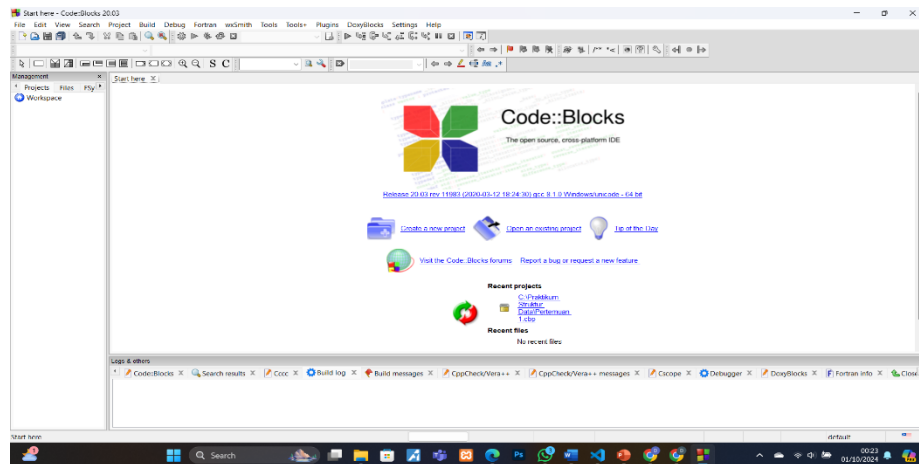
1. Mengetahui environment Code Blocks dengan baik.
2. Memahami cara menggunakan dan troubleshooting Code Blocks IDE.
3. Mengimplementasikan operator-operator dalam program.
4. Memahami cara membuat program sederhana dalam bahasa C++.
5. Memahami penggunaan tipe data dan variabel dalam bahasa C++.
6. Menggunakan operator-operator input/output dengan tepat.
7. Memahami dan mengimplementasikan fungsi kondisional dalam program.

2. Landasan Teori

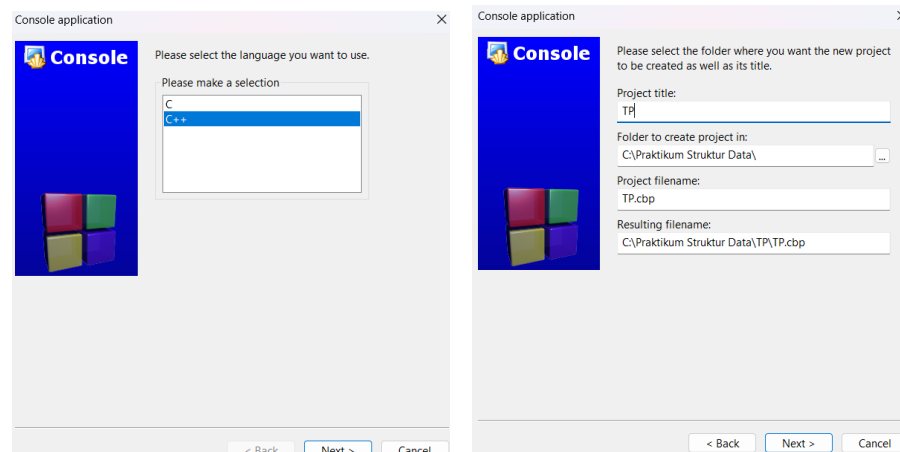
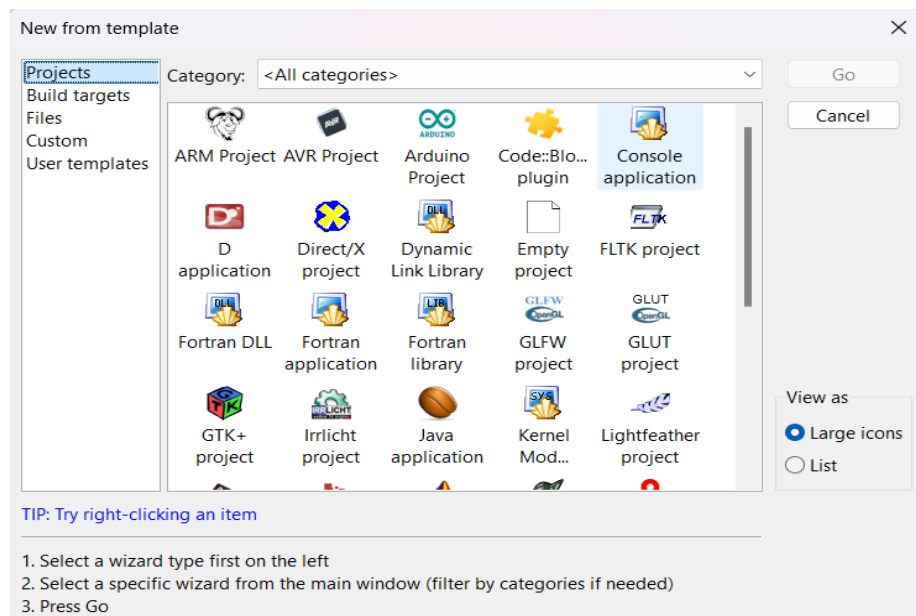
- Bahasa C++ adalah pemrograman yang dikembangkan oleh Bjarne Stroustrup di AT&T Bell Laboratories pada awal 1980-an sebagai perluasan dari bahasa C. Awalnya disebut "C with Classes," C++ memperkenalkan konsep pemrograman berorientasi objek (OOP) seperti enkapsulasi, pewarisan, dan polimorfisme. Nama C++ resmi diperkenalkan pada tahun 1983, mencerminkan peningkatan dari C. Pada tahun 1998, C++ distandarisasi oleh ISO dengan nama C++98, diikuti oleh beberapa pembaruan seperti C++11, C++14, C++17, dan C++20, yang terus memperkaya fitur bahasa ini. C++ digunakan luas di berbagai industri untuk pengembangan perangkat lunak, sistem operasi, dan aplikasi yang membutuhkan efisiensi tinggi.
- Code::Blocks adalah Integrated Development Environment (IDE) yang digunakan untuk mempermudah pengembangan program, terutama dalam bahasa C, C++, dan Fortran. IDE ini menyediakan fitur seperti editor kode yang ramah pengguna, dukungan untuk berbagai compiler (seperti GCC dan MinGW), serta debugger visual untuk mempermudah proses debugging. Dengan kemampuan untuk mengelola proyek, mendukung berbagai platform (Windows, Linux, macOS), dan memperluas fungsionalitas melalui plugin, Code::Blocks menjadi alat yang efisien dan fleksibel bagi pengembang perangkat lunak, baik untuk pemula maupun profesional.
- Tipe Data: kategori yang menentukan jenis nilai yang dapat disimpan oleh variabel. Tipe data dasar meliputi int untuk bilangan bulat, float dan double untuk angka desimal, char untuk karakter tunggal, serta bool untuk nilai logika true atau false. Selain itu, ada juga modifier seperti unsigned dan signed yang mengatur rentang nilai, serta tipe data kompleks seperti string untuk teks dan array untuk menyimpan kumpulan data sejenis. C++ juga memiliki tipe data seperti enum untuk daftar nilai konstan dan struct untuk menggabungkan beberapa tipe data dalam satu unit.

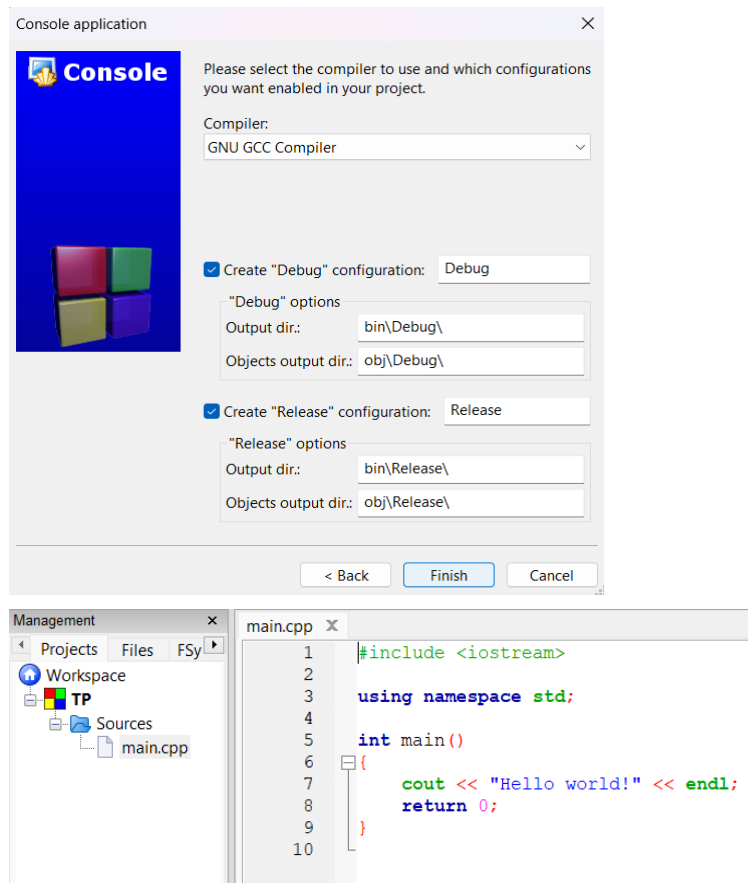
3. Guided

- I. Pertama Download Code::Blocks di Link ini: <http://www.codeblocks.org/downloads> . tampilan dalam pada CodeBlocks

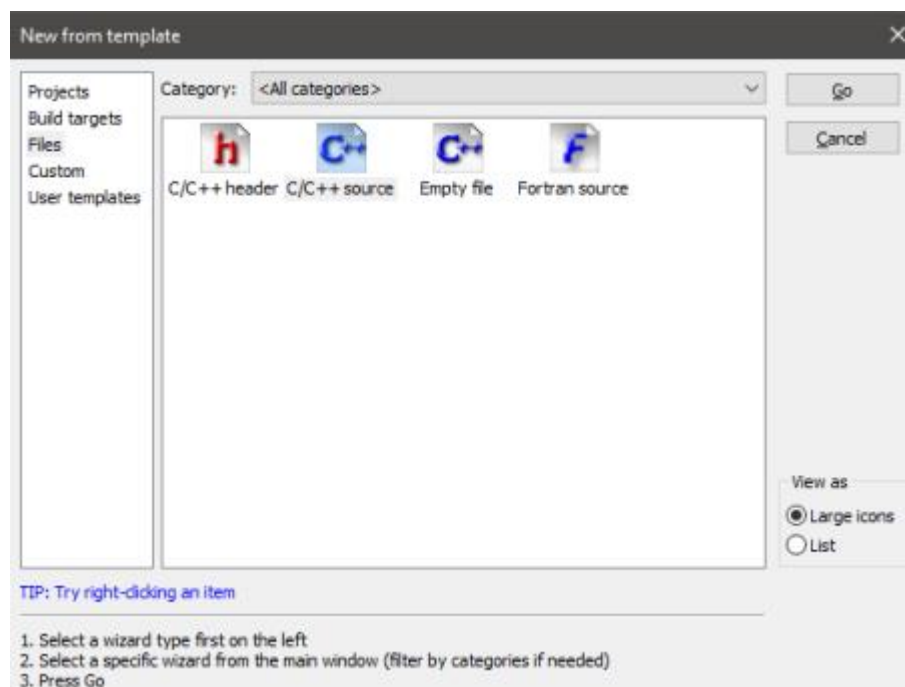


Buat Projek filenya dengan cara memilih **File > New > Projects**. Kemudian pada panel kiri pilih Project, pada panel kanan pilih Console application kemudian klik Go

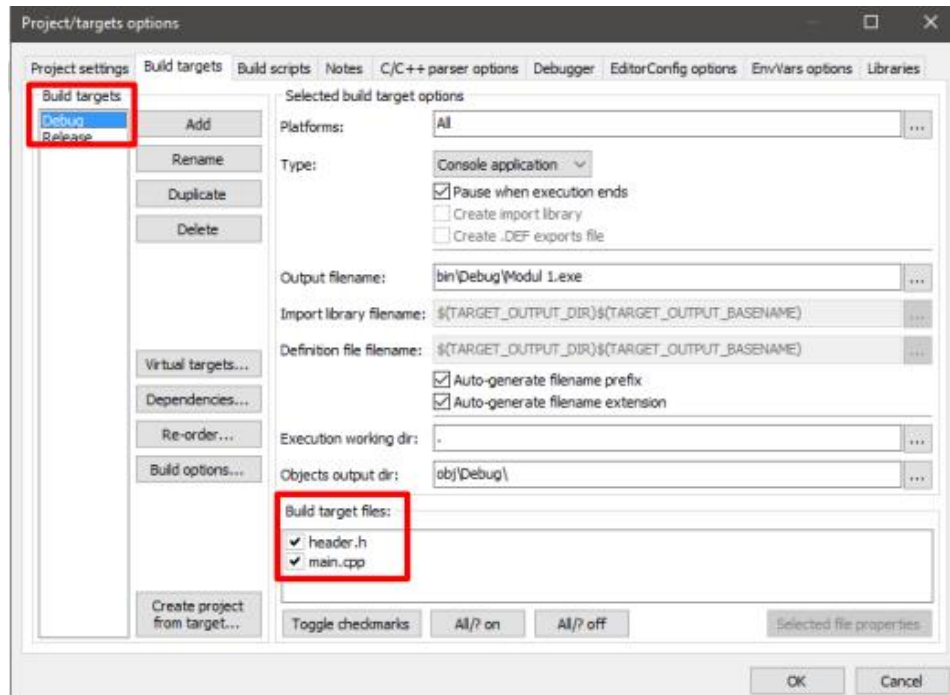




Membuat Class Baru dengan cara klik File > New > File. Pada panel kiri pilih Files, dan pada panel kanan pilih C/C++ source Kemudian Klik Go



Jika anda lupa mencentang build target, dapat dilakukan setting manual dengan cara klik kanan pada project > properties > Build targets > Debug > Centang semua target files seperti pada Error! Reference source not found



II. Tipe data

Type data	Conto h	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;

return 0;
```

III. Input/Output

Pasangan dinamis dalam bahasa pemrograman C++ yang memungkinkan program berinteraksi dengan pengguna. `cin` bertindak sebagai pintu masuk, memungkinkan program menerima data yang diinputkan oleh pengguna melalui keyboard. Data ini bisa berupa angka, teks, atau jenis data lainnya. Sementara itu, `cout` berfungsi sebagai jendela tampilan, menampilkan hasil perhitungan, pesan, atau informasi lainnya ke layar.

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

IV. Operator

Kategori (Arti)	Operator
Panggilan fungsi, subscript <i>array</i> , dan elemen struktur data	() [] ->
Operator Unary (NOT, komplemen, negasi, inkremen, dekremen, <i>address</i> , <i>indirection</i>)	! ~ - ++ -- & *
Operator Aritmatika (Perkalian, pembagian, Sisa Pembagian/mod)	* / %
Operator Aritmatika (Pertambahan, Pengurangan)	+ -

Kategori (Arti)	Operator
Operator Hubungan (sama dengan, tidak sama dengan)	== !=
Operator Bitwise AND	&
Operator Bitwise XOR	^
Operator Bitwise OR	
Operator Logika AND	&&
Operator Logika OR	
Operator Kondisional	?:
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 Bitwise Pergeseran Bit (shift kiri, shift kanan)	<< >>
Operasi Hubungan (kurang dari, kurang dari atau sama dengan lebih dari, lebih dari atau sama dengan)	< <= > >=

Operator Pengerjaan Bitwise (<i>assignment</i> AND bitwise, <i>assignment</i> OR bitwise, <i>assignment</i> XOR bitwise, <i>assignment</i> shift kanan)	&= ^= = <<= >>=
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: " << penjumlahan << endl;  
cout << "Hasilnya adalah: " << pengurangan << endl;  
cout << "Hasilnya adalah: " << perkalian << endl;  
cout << "Hasilnya adalah: " << pembagian << endl;  
cout << "Hasilnya adalah: " << modulus << endl;  
cout << "Hasilnya adalah: " << perbandingan << endl;
```

V. Boolean

```
bool kondisi1 = false;
bool kondisi2 = false;

bool hasil = ( kondisi1 || kondisi2);
cout << "Hasilnya adalah: " << boolalpha << hasil << endl;

bool kondisi1 = false;
bool hasil = !kondisi1;
cout << "Hasilnya adalah: " << boolalpha << hasil << endl;
```

VI. Percabangan

```
// //percabangan
string kata;
cout << " Masukkan kata = HALO" << endl;
cin >> kata;

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

VII. Switch Case

```
int tv;
cout << " daftar channel tv " << endl;
cout << "1. RCTI" << endl;
cout << "2. Indosiar" << endl;

cout << " Masukkan channel pilihan: ";
cin >> tv;

switch(tv){
case 1 :
    cout << "channel yg anda pilih rcti" << endl;
    break;
case 2 :
    cout << "channel yg anda pilih Indosiar" << endl;
    break;
default:
    cout << "channel tidak tersedia" << endl;
}
```

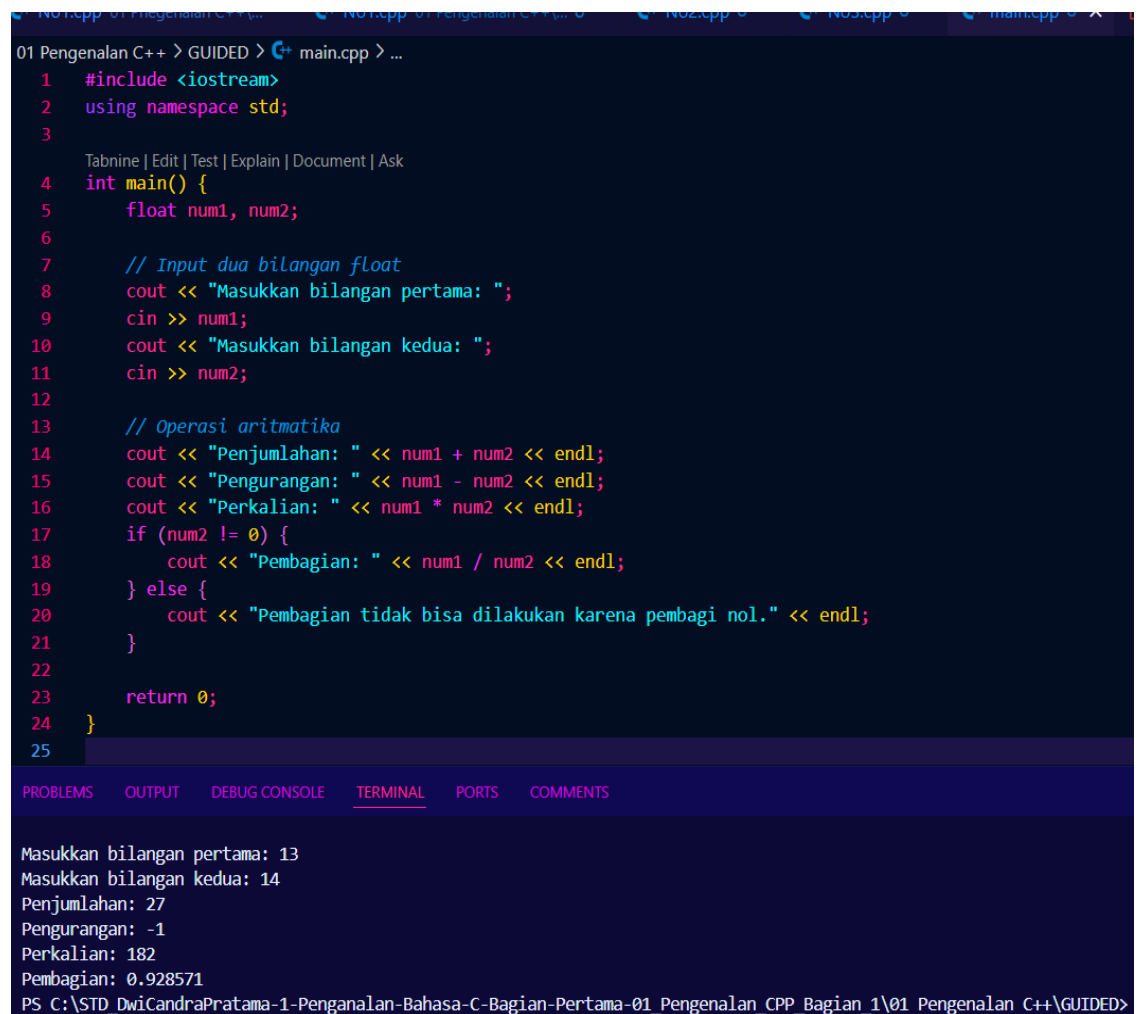
VIII. Looping

```
int i;
for(i=0; i<5; i++){
    cout << "Hello World" << endl
}

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

4. Unguided

1. 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.



```
01 Pengenalan C++ > GUIDED > main.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      float num1, num2;
6
7      // Input dua bilangan float
8      cout << "Masukkan bilangan pertama: ";
9      cin >> num1;
10     cout << "Masukkan bilangan kedua: ";
11     cin >> num2;
12
13     // Operasi aritmatika
14     cout << "Penjumlahan: " << num1 + num2 << endl;
15     cout << "Pengurangan: " << num1 - num2 << endl;
16     cout << "Perkalian: " << num1 * num2 << endl;
17     if (num2 != 0) {
18         cout << "Pembagian: " << num1 / num2 << endl;
19     } else {
20         cout << "Pembagian tidak bisa dilakukan karena pembagi nol." << endl;
21     }
22
23     return 0;
24 }
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
Masukkan bilangan pertama: 13
Masukkan bilangan kedua: 14
Penjumlahan: 27
Pengurangan: -1
Perkalian: 182
Pembagian: 0.928571
PS C:\STD_DwiCandraPratama-1-Pengalan-Bahasa-C-Bagian-Pertama-01_Pengenalan_CPP_Bagian_1\01 Pengenalan C++\GUIDED>
```

2. 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

```
01 Pengenalan C++ > UNGUIDED > C:\TP_ModulNo2.cpp > ...
1 //Buatlah sebuah program yang menerima masukan angka dan mengeluarkan output nilai angka tersebut dalam bentuk tulisan. Angka yang akan di- input-ka
2
3 #include <iostream>
4 using namespace std;
5
6 // Fungsi untuk mengubah satuan menjadi kata
7 string convertSatuan(int number) {
8     string satuan[] = {"", "Satu", "Dua", "Tiga", "Empat", "Lima", "Enam", "Tujuh", "Delapan", "Sembilan"};
9     return satuan[number];
10 }
11
12 // Fungsi untuk mengubah puluhan menjadi kata
13 string convertPuluhan(int number) {
14     if (number == 10) return "Sepuluh";
15     if (number == 11) return "Sebelas";
16     if (number < 20) return convertSatuan(number % 10) + " Belas";
17     string puluhan[] = {"", "", "Dua Puluh", "Tiga Puluh", "Empat Puluh", "Lima Puluh",
18                         "Enam Puluh", "Tujuh Puluh", "Delapan Puluh", "Sembilan Puluh"};
19     return puluhan[number / 10] + " " + convertSatuan(number % 10);
20 }
21
22 // Fungsi utama untuk mengonversi angka ke kata
23 string numberToWords(int number) {
24     if (number == 0) return "Nol";
25     if (number == 100) return "Seratus";
26     return convertPuluhan(number);
27 }
28
29 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
30
31 Masukkan angka (0-100): 69
32 69 : Enam Puluh Sembilan
33 PS C:\STD_DwiCandraPratama-1-Pengalan-Bahasa-C-Bagian-Pertama-01_Pengalan_CPP_Bagian_1\01 Pengalan C++\UNGUIDED> cd "c:\STD_DwiCandraPratama-1-Pengalan-Bahas
rtama-01_Pengalan_CPP_Bagian_1\01 Pengalan C++\UNGUIDED\" ; if ($?) { g++ TP_ModulNo2.cpp -o TP_ModulNo2 } ; if ($?) { .\TP_ModulNo2 }
34 Masukkan angka (0-100): 34
35 34 : Tiga Puluh Empat
36 PS C:\STD_DwiCandraPratama-1-Pengalan-Bahasa-C-Bagian-Pertama-01_Pengalan_CPP_Bagian_1\01 Pengalan C++\UNGUIDED>
```

3. Buatlah program yang dapat memberikan input dan output sbb.

Input: 3
output:
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*

```
01 Pengenalan C++ > UNGUIDED > C:\TP_ModulNo3.cpp > main()
1 // 3. Buatlah program yang dapat memberikan input dan output sbb.
2
3 #include <iostream>
4 #include <string>
5
6 using namespace std;
7
8 void printPattern(int n) {
9     for (int i = n; i >= 1; i--) {
10         // Print numbers descending
11         for (int j = i; j >= 1; j--) {
12             cout << j << " ";
13         }
14
15         // Print single asterisk
16         cout << "* ";
17
18         // Print numbers ascending
19         for (int j = 1; j <= i; j++) {
20
21 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
22
23 PS C:\STD_DwiCandraPratama-1-Pengalan-Bahasa-C-Bagian-Pertama-01_Pengalan_CPP_Bagian_1> cd "c:\STD_DwiCandraPratama
alan_CPP_Bagian_1\01 Pengalan C++\UNGUIDED\" ; if ($?) { g++ TP_ModulNo3.cpp -o TP_ModulNo3 } ; if ($?) { .\TP_ModulNo3 }
24 Input: 4
25 Output:
26 4 3 2 1 * 1 2 3 4
27 3 2 1 * 1 2 3
28 2 1 * 1 2
29 1 * 1
30 PS C:\STD_DwiCandraPratama-1-Pengalan-Bahasa-C-Bagian-Pertama-01_Pengalan_CPP_Bagian_1\01 Pengalan C++\UNGUIDED>
```

5. Kesimpulan

C++ adalah bahasa pemrograman yang kuat dan efisien, sering digunakan untuk pengembangan aplikasi sistem, perangkat lunak yang membutuhkan performa tinggi, serta game. Dengan dukungan tipe data yang beragam, mulai dari bilangan bulat, desimal, karakter, hingga struktur dan enumerasi, C++ memungkinkan pengembang untuk mengelola memori secara efisien dan menjalankan operasi kompleks dengan lebih optimal. Fleksibilitas dalam mengontrol alokasi memori juga menjadi kelebihan utama C++, terutama untuk aplikasi skala besar.

Penggunaan Code::Blocks sebagai Integrated Development Environment (IDE) mempermudah proses pengembangan program dalam C++. Dengan fitur-fitur seperti editor kode yang interaktif, dukungan untuk berbagai compiler, dan kemampuan debugging, Code::Blocks sangat membantu dalam menulis, menguji, serta memelihara kode C++. Kombinasi antara C++ dan Code::Blocks memberikan pengalaman pengembangan yang efisien dan terstruktur, baik bagi pemula maupun pengembang profesional.