

LAPORAN PRAKTIKUM
PERTEMUAN 1
PROSES INSTALASI CODEBLOCKS DAN PENGENALAN BAHASA
C++



Nama :
Ryan Gabriel Togar Simamora (2311104045)

Kelas : SE07-02

Dosen :
Wahyu Andi Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. TUJUAN PRAKTIKUM

1. Mengenal 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.

II. Pengenalan Code Blocks

Pada praktikum Struktur Data ini, kertas (tool) yang digunakan adalah Code Blocks. Kertas ini merupakan free, open-source, dan cross-platform IDE. Saat ini, Code Blocks berorientasi pada C/C++/Fortran.

Code Blocks juga memiliki antarmuka yang user-friendly dan modular, yang memungkinkan penggunanya menambahkan berbagai plugin untuk memperluas fungsionalitasnya. IDE ini mendukung berbagai compiler seperti GCC, Clang, dan Visual C++, serta memiliki fitur debugging yang kuat menggunakan GDB. Dengan kemampuan untuk dikustomisasi, Code::Blocks cocok untuk pemula maupun profesional dalam pengembangan proyek C, C++, dan Fortran. Selain itu, karena bersifat cross-platform, Code::Blocks dapat dijalankan di berbagai sistem operasi seperti Windows, Linux, dan macOS, memberikan fleksibilitas lebih dalam pengembangan perangkat lunak.

A. Instalasi Code Blocks

Di sini, saya menggunakan OS Kali Linux, yang merupakan distribusi Linux berbasis Debian yang dirancang khusus untuk pengujian penetrasi dan keamanan informasi. Kali Linux menyediakan berbagai alat dan aplikasi yang berguna untuk analisis keamanan, termasuk pengembangan perangkat lunak. Salah satu IDE yang saya pilih untuk pemrograman C dan C++ adalah Code Blocks.

Berikut adalah langkah-langkah untuk menginstal Code Blocks di Kali Linux menggunakan terminal :

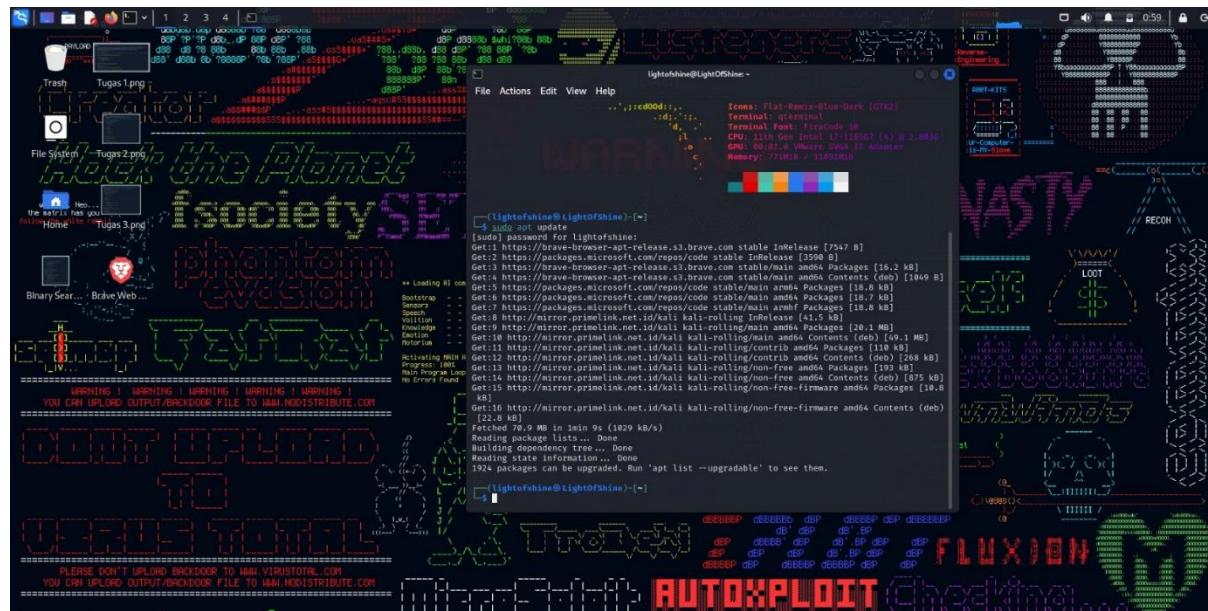
1. Membuka Terminal Kali Linux

Untuk membuka terminal, gunakan kombinasi tombol Ctrl + Alt + T. Terminal adalah antarmuka berbasis teks yang memungkinkan pengguna untuk berinteraksi langsung dengan sistem operasi. Dalam konteks pengembangan, terminal sering digunakan untuk menjalankan perintah dan mengelola perangkat lunak secara efisien.



2. Memperbarui Daftar Paket

Setelah terminal terbuka, langkah pertama yang disarankan adalah memperbarui daftar paket. Ini memastikan bahwa saya menginstal versi terbaru dari perangkat lunak. Jalankan perintah sudo apt update di terminal seperti gambar dibawah ini.



Setelah menjalankan perintah sudo apt update, sistem meminta password karena perintah tersebut memerlukan hak akses administrator atau superuser. Berikut penjelasannya:

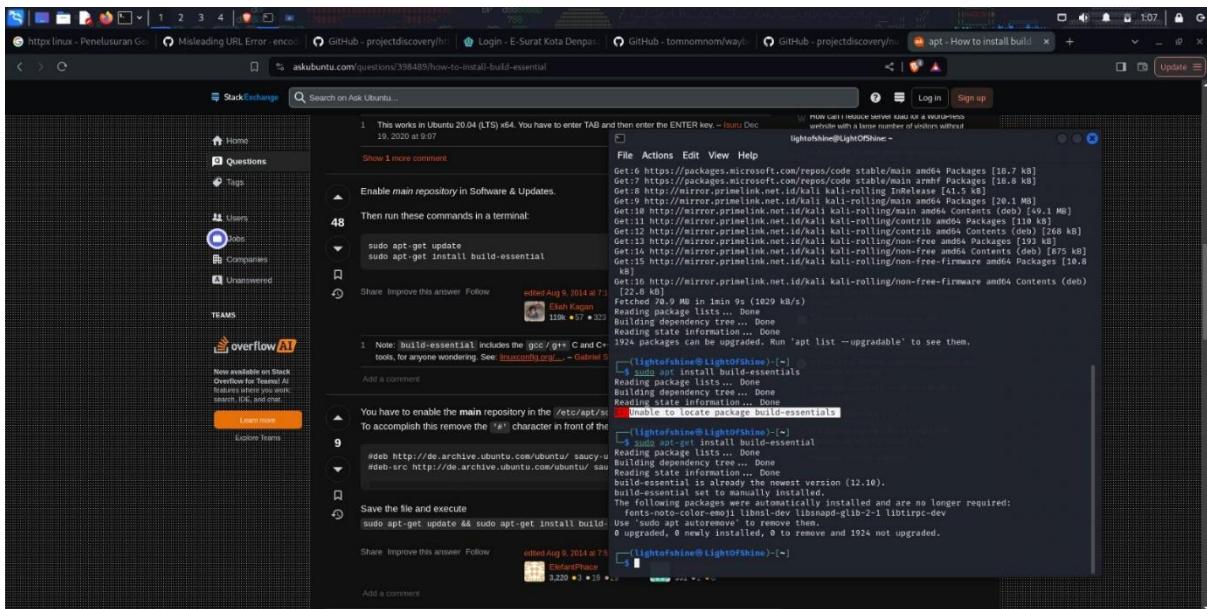
Perintah sudo (Superuser Do): Perintah sudo digunakan untuk memberikan hak akses sementara kepada pengguna non-administrator untuk menjalankan perintah yang memerlukan izin tingkat root atau administrator. Karena tindakan seperti memperbarui daftar paket atau menginstal perangkat lunak dapat mengubah sistem secara signifikan, hanya pengguna dengan izin khusus yang diizinkan untuk melakukannya. Oleh karena itu, sudo meminta password pengguna yang memiliki hak administrator.

3. Menginstal Build Essential

Setelah menjalankan sudo apt update, saya menggunakan perintah sudo apt install build-essential untuk menginstal paket build-essential seperti gambar dibawah ini. Paket ini menyediakan alat-alat penting untuk pengembangan perangkat lunak di Kali Linux, termasuk:

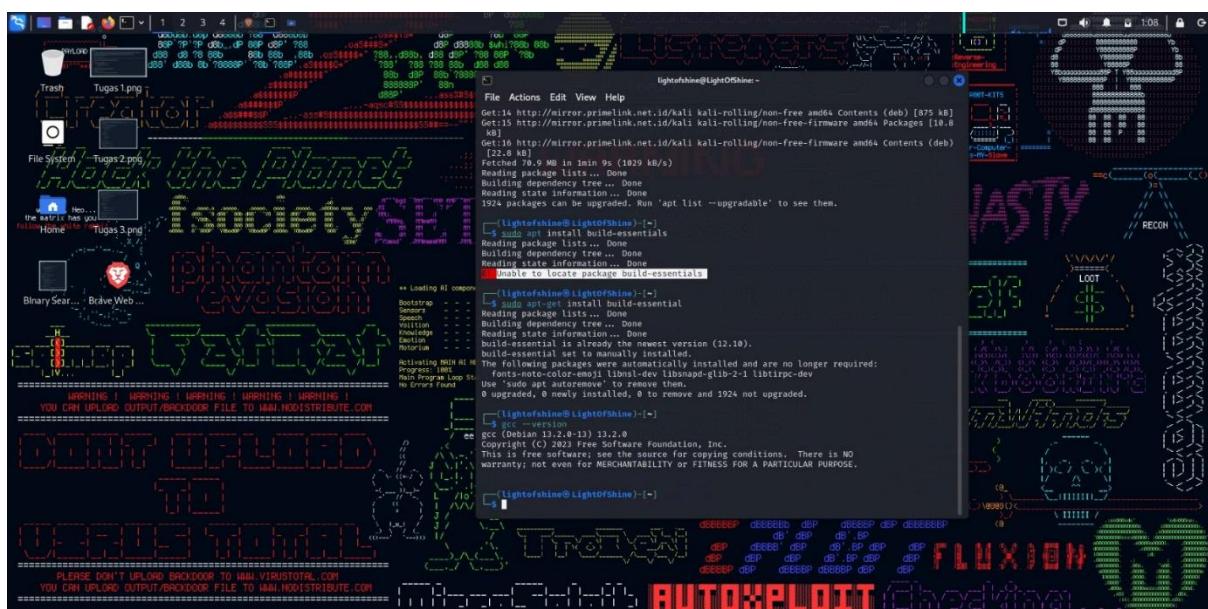
- ❖ GCC (GNU Compiler Collection): Compiler untuk bahasa C dan C++.
- ❖ make: Utilitas untuk mengotomatisasi proses kompilasi.
- ❖ libc6-dev: Header dan library yang diperlukan untuk pengembangan.

Menginstal build-essential memastikan bahwa lingkungan pengembangan dilengkapi dengan semua alat yang dibutuhkan untuk mengkompilasi dan membangun perangkat lunak dari kode sumber.



4. Mengecek Versi GCC

Setelah itu, saya mengecek versi GCC yang terinstal dengan menjalankan perintah gcc –version seperti gambar dibawah ini. Perintah ini menampilkan informasi tentang versi GCC yang saat ini terpasang di sistem, memastikan bahwa compiler telah terinstal dengan benar.



Setelah itu, saya langsung menginstal Code Blocks dengan menjalankan perintah sudo apt-get install codeblocks seperti gambar dibawah ini. Proses ini memastikan bahwa IDE Code Blocks terpasang dengan benar di Kali Linux, siap digunakan untuk pengembangan C/C++ dengan dukungan GCC yang sudah terinstal sebelumnya.



Setelah itu, proses instalasi Code Blocks sudah selesai, dan IDE siap digunakan untuk pengembangan perangkat lunak di Kali Linux.

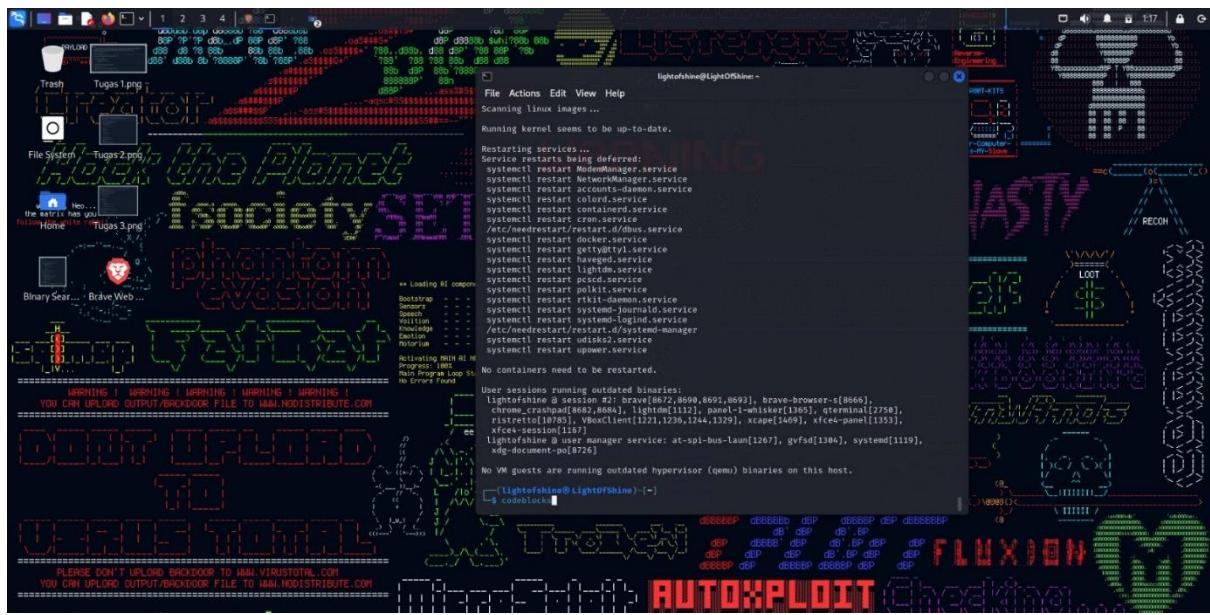
B. Cara Menjalankan dan Menggunakan Code Blocks Beserta Fiturnya

Di sini, saya akan menjalankan dan menjelaskan cara menggunakan Code Blocks. IDE ini dirancang untuk memudahkan pengembangan aplikasi C dan C++ dengan antarmuka yang intuitif. Saya akan menjelaskan langkah-langkah dasar dalam membuka, membuat, dan menjalankan project di Code Blocks.

Berikut ini Langkah Langkahnya :

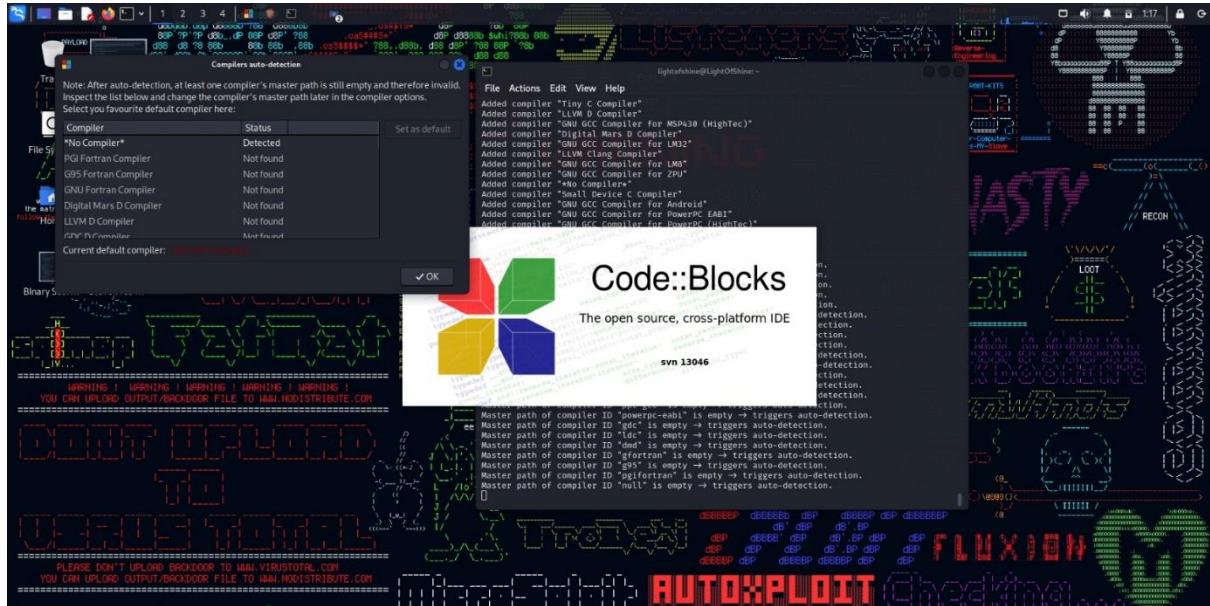
1. Menjalankan Code Blocks lewat Terminal

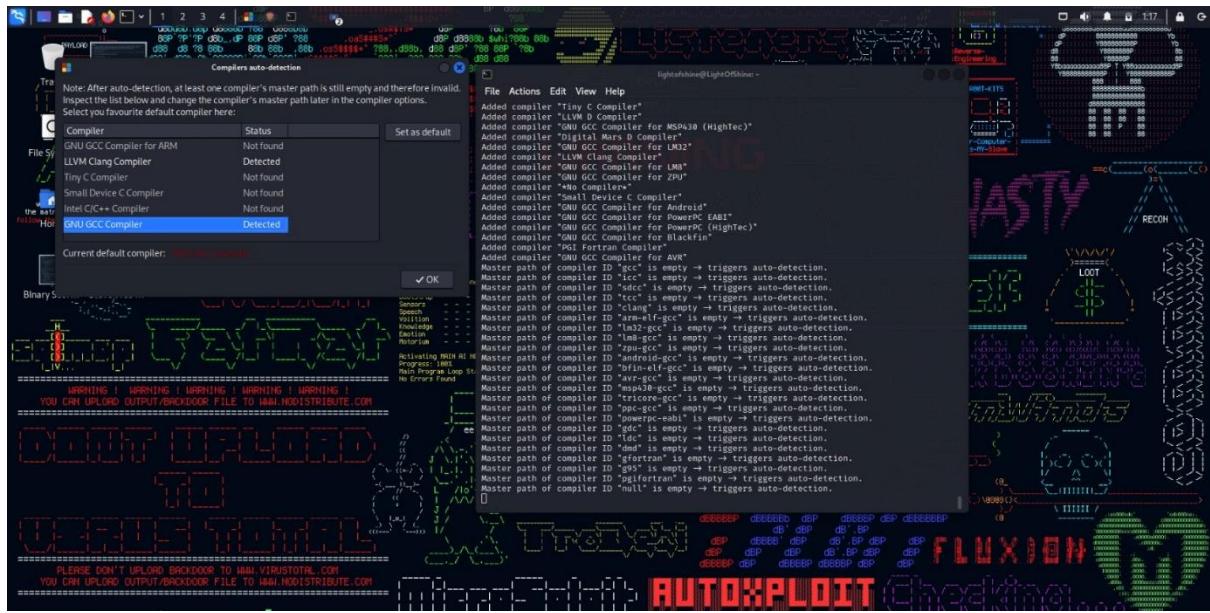
Untuk menjalankan Code Blocks di Kali Linux, buka terminal dan masukkan perintah codeblocks seperti gambar dibawah ini.



2. Setelah Code Blocks terbuka, langkah selanjutnya adalah memilih compiler yang tepat.

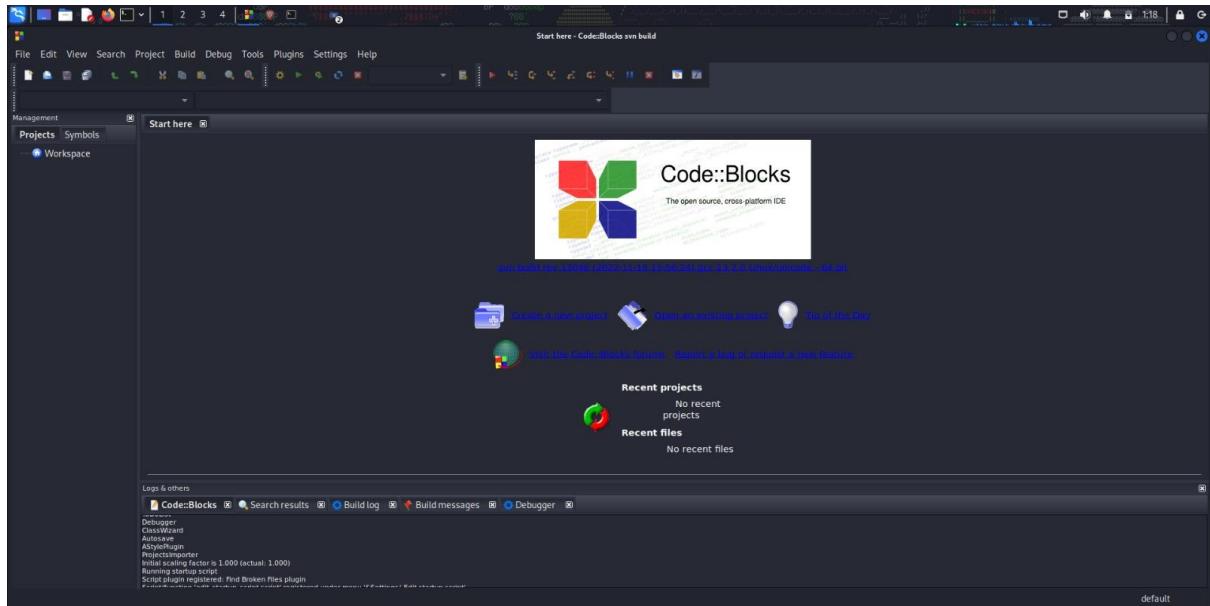
Ketika pertama kali menjalankan Code Blocks, Saya akan diminta untuk memilih compiler. Pilih GCC sebagai compiler default, yang telah diinstal sebelumnya sebagai bagian dari paket build-essential. GCC akan digunakan untuk mengompilasi proyek C/C++ di Kali Linux seperti pada gambar dibawah ini.





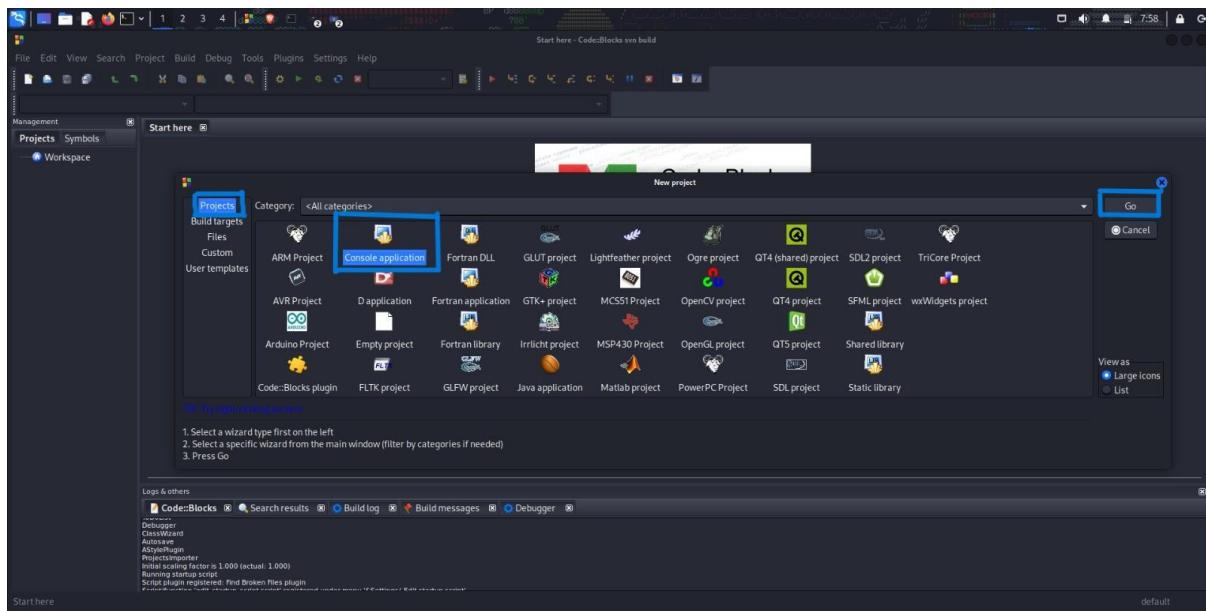
3. Setelah itu klik ok dan Code Blocks siap digunakan

Setelah memilih GCC sebagai compiler default, Code Blocks siap digunakan untuk pengembangan proyek C/C++. Dengan konfigurasi yang telah selesai, Saya dapat mulai membuat, mengelola, dan menjalankan proyek perangkat lunak secara efisien menggunakan IDE ini.

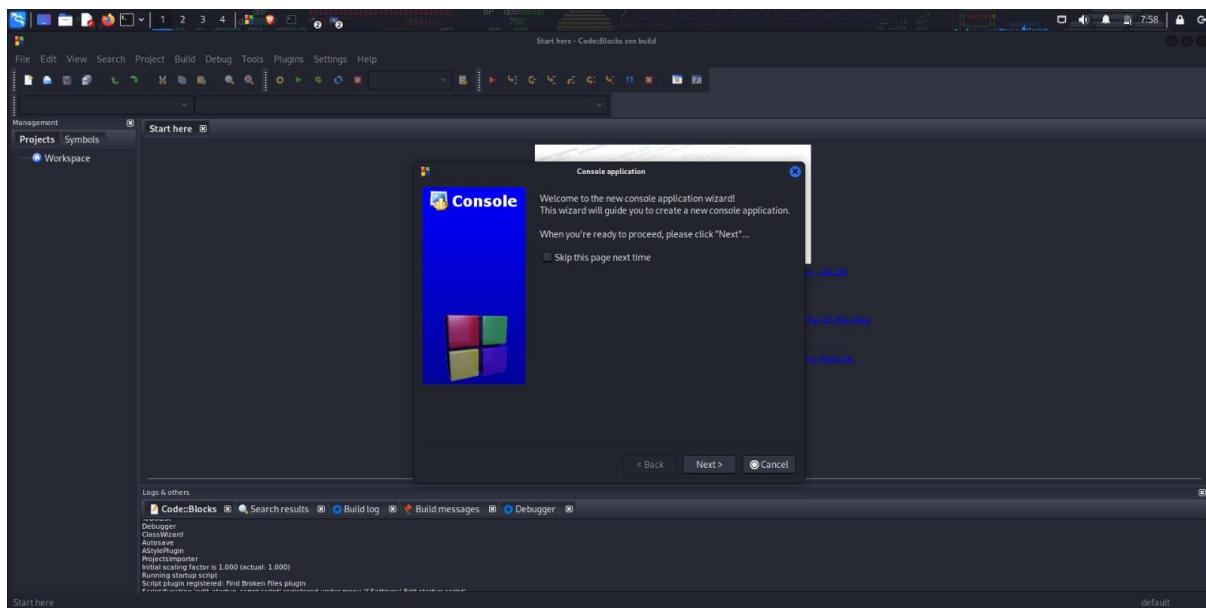


4. Membuat Project Baru

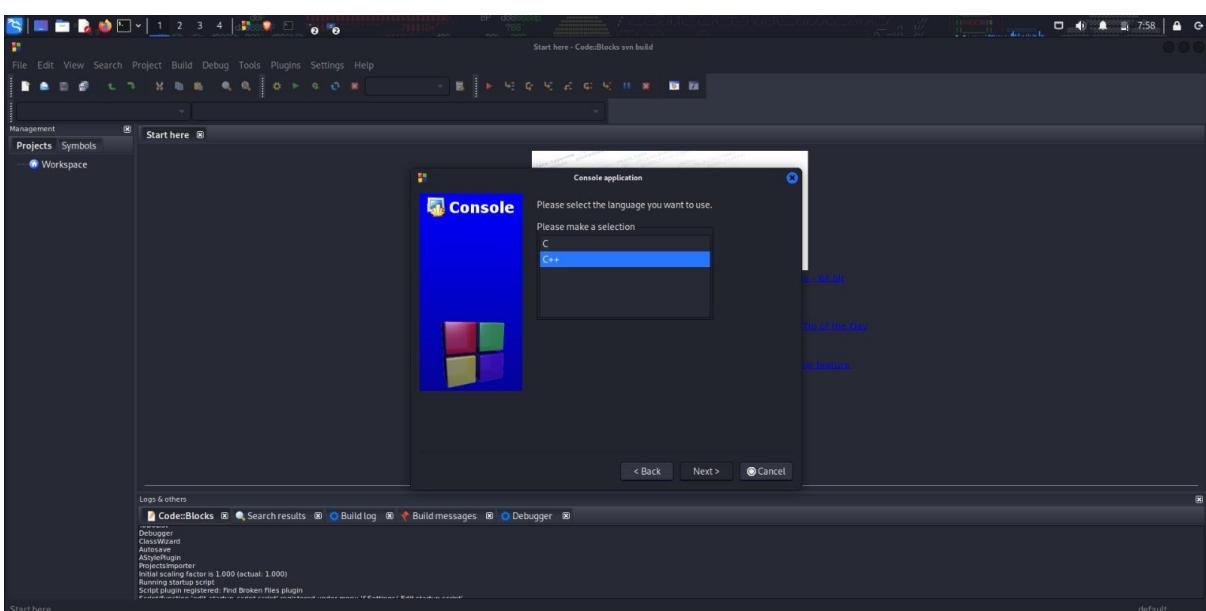
Untuk membuat proyek baru, pilih File > New > Project. Pada panel sebelah kiri, pilih opsi Project, dan pada panel sebelah kanan, pilih Console Application, lalu klik Go untuk melanjutkan ke langkah berikutnya dalam pembuatan proyek.seperti pada gambar di bawah ini.



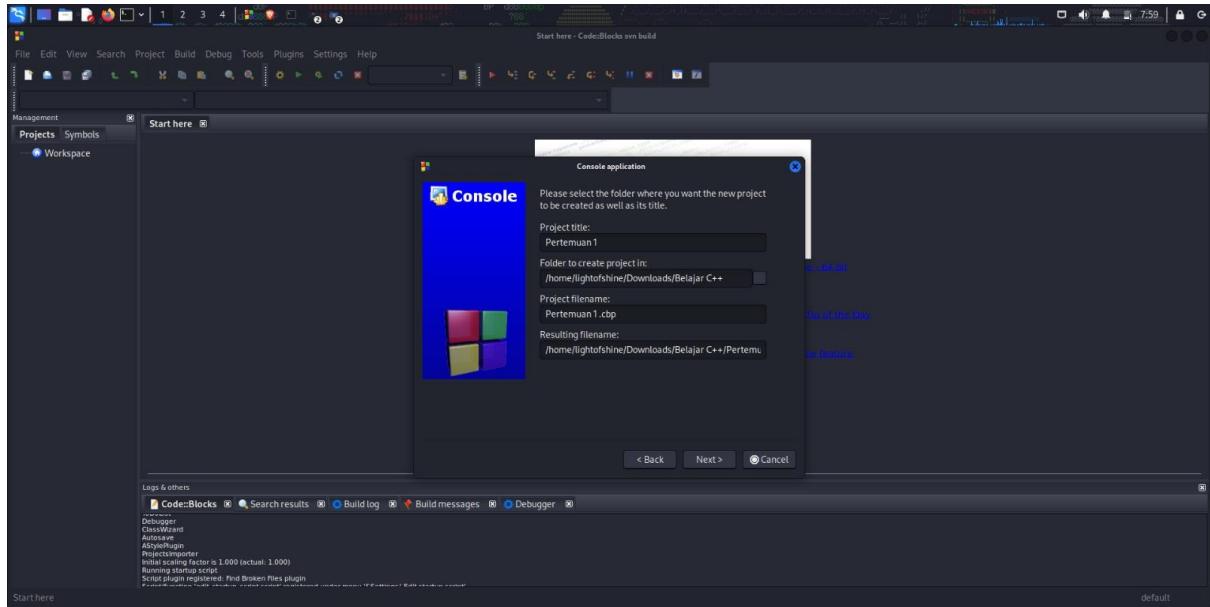
Lalu akan muncul tampilan seperti pada gambar dibawah ini.



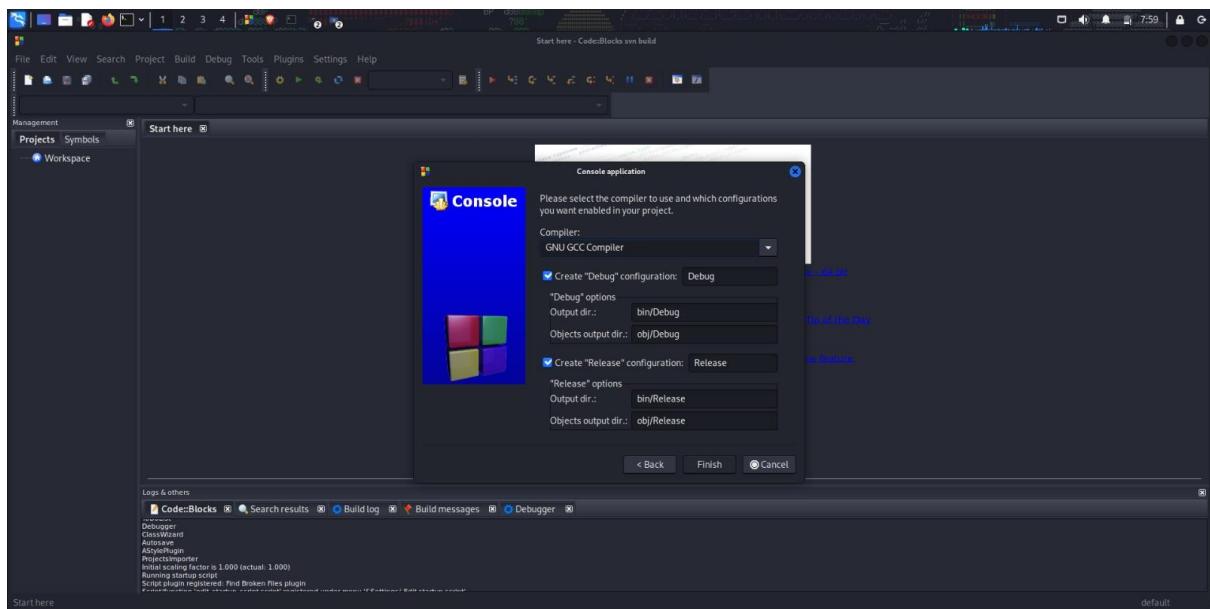
Kemudian pilih c++ seperti pada gambar dibawah ini lalu klik next



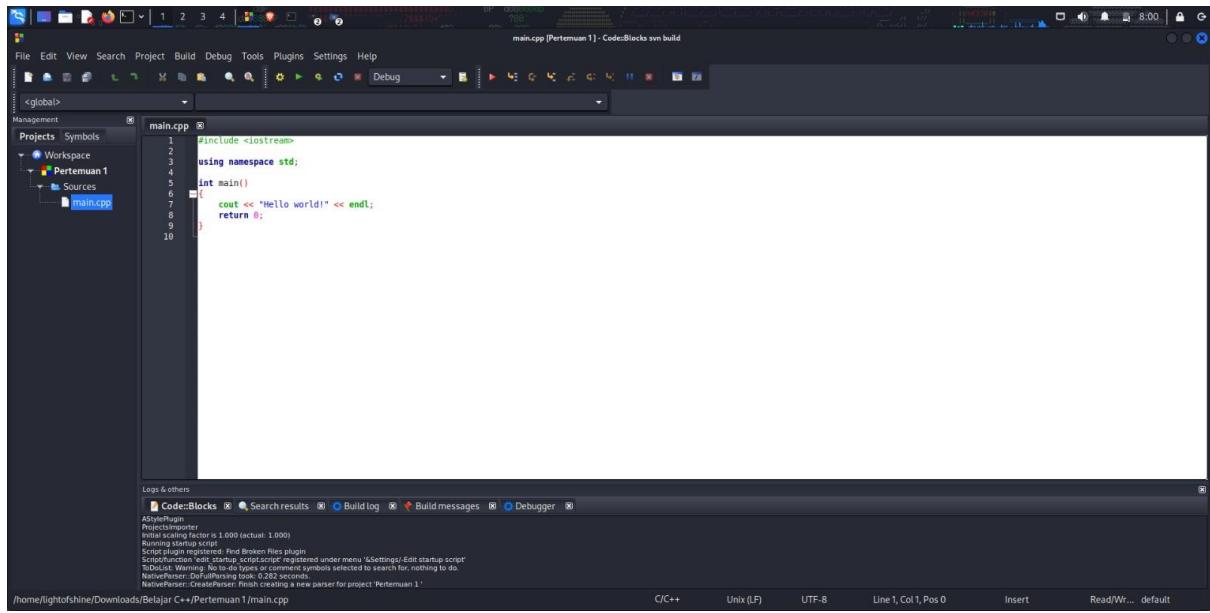
Kemudian buat judul projectnya dan pilih tempat projectnya disimpan lalu klik next seperti pada gambar dibawah ini.



Kemudian klik finish seperti pada gambar dibawah ini.



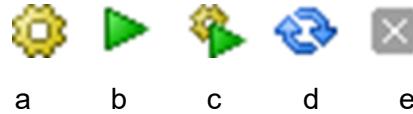
Setelah proses pembuatan project selesai, Code Blocks siap digunakan untuk menulis dan mengeksekusi kode seperti pada gambar dibawah ini.



```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

5. Compile program

yaitu aksi untuk menjalankan program yang telah ditulis sehingga programmer dapat melihat output dari program.

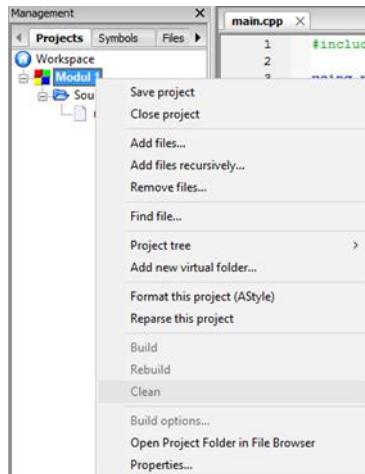


- a. Build merupakan aksi untuk membangun sintak menjadi sebuah program. Adapun shortcut yang digunakan adalah Ctrl-F9.
- b. Run merupakan aksi untuk menjalankan program yang telah di-build. Program tidak akan berjalan sebelum dilakukan build. Adapun shortcut yang digunakan adalah Ctrl-F10.
- c. Build and Run adalah aksi yang mengizinkan Build dan Run berjalan berurutan secara otomatis. Adapun shortcut yang digunakan adalah F9.
- d. Rebuild merupakan aksi untuk membagun kembali program. Adapun shortcut yang digunakan adalah Ctrl-F11.
- e. Abort merupakan aksi untuk mematikan program yang sedang berjalan.

6. Clean

Terkadang program yang kita buat tidak dapat di run, untuk itu perlu dilakukan clean project dengan cara klik kanan pada project kemudian klik Clean seperti pada gambar dibawah ini.

Setelah dilakukan clean, program akan dapat berjalan kembali.



7. Close dan Open Project

Untuk menutup project dilakukan dengan cara klik kanan pada project kemudian pilih close. Project yang di close tidak akan terhapus dan tetap ada pada directory. Sedangkan untuk membuka project yang telah di close dilakukan dengan cara memilih File > Open > Pilih File *.cbp

8. Error Message

Message akan muncul ketika terjadi kesalahan pada penulisan sintak. Contoh seperti pada gambar dibawah ini. terdapat error pada line 8. Pada error message diharapkan semicolon (;) sebelum return dan ternyata pada line 7 dapat dilihat bahwa penulisan sintak belum diakhiri untuk fungsi cout.

The screenshot shows the 'main.cpp' file in the code editor. Line 8 contains the code 'cout << "Hello world!" << endl'. A red box highlights this line, indicating a syntax error. In the 'Build messages' tab, an error message is displayed: 'error: expected ';' before 'return'' at line 8. This message corresponds to the missing semicolon after 'endl' in the code.

File	Line	Message
C:\Users\tst\D...	8	error: expected ';' before 'return'

III. Pengenalan Bahasa Pemrograman C++

A. Pengertian dan Sejarah C++

a. Apa Itu C++?

C++ adalah bahasa pemrograman komputer yang merupakan evolusi dari keluarga bahasa C yang sudah ada. Sebagai bahasa yang berorientasi pada objek yang memberikan struktur jelas pada program dan memungkinkan kode untuk digunakan ulang, C++ dapat menurunkan biaya pengembangan.

Contoh program C++ bisa ditemukan dalam sistem operasi yang kita kenal sekarang karena C++ adalah pemrograman yang bersifat portabel dan bisa digunakan untuk menciptakan berbagai aplikasi yang bisa beradaptasi dengan beragam platform.

b. Sejarah C++

Delapan tahun setelah bahasa pemrograman C dikembangkan di Bell Lab pada tahun 1972, muncul bahasa C++ yang dirancang oleh Bjarne Stroustrup di AT&T Bell Laboratories pada tahun 1980.

Stroustrup yang saat itu mengejar gelar Ph.D berkesempatan bekerja dengan bahasa Simula. Simula disebut sebagai bahasa pertama yang mendukung paradigma pemrograman berorientasi obyek.

Kemudian, muncul keinginan pada diri Stroustrup untuk menciptakan bahasa pemrograman dengan tambahan fitur berorientasi objek. Ia menyadari paradigma pemrograman berorientasi obyek akan berguna untuk pengembangan perangkat lunak. Sementara menurutnya, bahasa Simula masih terlalu lambat untuk penggunaan praktis.

Prototype C++ kali pertama muncul sebagai C yang dipercanggih dengan fasilitas kelas, atau dikenal sebagai C with Classes. C with Classes kemudian disempurnakan kembali dengan menambah fasilitas overloading operator, dan kemudian disebut sebagai a better C, yang kemudian berganti nama lagi menjadi C++ di tahun 1983.

Simbol ++ yang ditambahkan pada penamaan tersebut adalah operator penaikan C yang digunakan untuk memperlihatkan bahwa bahasa ini adalah versi paling canggih dan paling baru dari C.

B. Dasar Pemrograman C++

1. Struktur Bahasa Pemrograman C++

Secara umum, pembagian struktur bahasa pemrograman C++ adalah sebagai berikut.

1.3.2 Pengenal (Identifier)

C++	Keterangan
#include <iostream> #include <conio.h> #include "newlibrary.h"	Pendeklarasian <i>library</i> yang akan digunakan di dalam program
#define PHI 3.14 const int constant1; const float constant2 = 0.5;	Pendefinisan konstanta
struct new_record_type { int element1; float element2; }	Pendefinisan tipe data bentukan / <i>record type</i> / struktur
int var1; float var2[2];	Pendeklarasian variabel
int function_A(){ } void procedure_B(){ }	Pendeklarasian fungsi dan prosedur
int main(){ // blok program return 0; }	Program utama

2. Pengenal (Identifier)

Identifier merupakan nama yang biasa digunakan untuk variabel, konstanta, fungsi atau objek lain yang didefinisikan oleh program.

Aturan yang digunakan untuk menentukan identifier:

1. Harus diawali dengan huruf (A....Z, a....z) atau garis bawah (_).
2. Karakter selanjutnya bisa berupa huruf, digit atau karakter garis bawah (_) atau dollar (\$).
3. Panjang maksimal identifier adalah 32 karakter, jika lebih maka yang dianggap adalah 32 karakter awal.
4. Tidak boleh mengandung spasi.
5. Tidak boleh menggunakan operator aritmatika (+ - * %).

Bahasa C bersifat case sensitive, jadi huruf besar dan huruf kecil dianggap berbeda. Contoh:

- ❖ Panjang (berbeda dengan: Panjang)
- ❖ nilai4
- ❖ luas_total
- ❖ harga_beli\$

3. Tipe data dasar

Data merupakan suatu nilai yang dapat dinyatakan dalam bentuk konstanta atau variabel. Data berdasarkan jenisnya dibagi dalam 5 kelompok, yang dinamakan sebagai tipe data dasar. Kelima kelompok tersebut:

1. Bilangan bulat (integer).
2. Bilangan real presisi – tunggal.

3. Bilangan real presisi – ganda.
4. Karakter.
5. Tak-bertipe.

4. Variabel

Variabel dalam program digunakan untuk menyimpan nilai, nilai variabel bisa berubah – ubah selama program berjalan. Aturan penamaan variabel sesuai dengan aturan penamaan identifier. Bentuk umum pendeklarasian suatu variabel dalam bahasa pemrograman C dapat ditulis sebagai berikut.

tipe_data nama_variabel;

Contoh :

```

1 #include <iostream>
2 #include <curses.h>
3
4 using namespace std;
5
6 int main()
7 {
8     int angka = 10;
9     float desimal = 10.5;
10    string kalimat = "Gabriel";
11    double tinggi = 10.4;
12    char jenis_kelamin = 'L';
13
14    cout << "Angka : " << angka << endl;
15
16    /* int angka;
17     cout << "Masukkan angka : ";
18     cin >> angka;
19
20     cout << "Angka : " << angka << endl;
21
22     getch();
23
24 */
25

```

5. Konstanta

Konstanta menyatakan nilai yang selalu tetap. Seperti halnya dengan variabel, konstanta juga mempunyai tipe.

Untuk mendeklarasikan suatu nilai yang sifatnya konstan kita cukup menambahkan kata **const** di depan tipe data dan variabel.

Contoh:

```

const float phi = 3.14;
const int n = 20;

```

C.Input/Output

1. Output

❖ Fungsi cout()

Fungsi ini digunakan untuk mencetak data baik yang bertipe numerik, ataupun teks, baik konstanta ataupun variabel.

Algoritma	Program coba_output Algoritma Output("saya lagi belajar bahasa C++ nih!!!")
C++	#include <iostream> using namespace std; int main(){ cout<<"saya lagi belajar bahasa C++ nih!!!"<<endl; return 0; }

Output : saya lagi belajar bahasa C++ nih!!!

2. Input

❖ Fungsi cin()

Cin() merupakan salah satu fungsi yang digunakan untuk meminta inputan keyboard dari user. Bentuk umumnya pendeklarasiannya adalah sebagai berikut :

cin>>nama_variabel;

Algoritma	C++
Program coba_input <u>Kamus</u> inp : int <u>Algoritma</u> <u>Input</u> (inp) <u>Output</u> ("nilai = ", inp)	#include <iostream> using namespace std; int main(){ int inp; cin >> inp; cout << "nilai = " << inp; return 0; }

Contoh :

The screenshot shows the Code::Blocks IDE interface. On the left, the code editor displays a file named 'main.cpp' with the following content:

```
string kalimat = "Gabriel";
double tinggi = 10.4;
char jenis_kelamin = 'L';

cout << "Angka : " << angka << endl;/*

int angka;
cout << "Masukkan angka : ";
cin >> angka;

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

// operator aritmatika
/* int angkal = 10;
int angka2 = 2;

int hasil = angkal % angka2;
cout << "Hasilnya adalah " << hasil << endl;*/
```

To the right of the code editor is a terminal window titled 'Pertemuan 1' showing the program's output:

```
Hasilnya angka : 100
Angka : 100
Process returned 0 (0x0)   execution time : 5.639 s
Press ENTER to continue.
```

At the bottom of the interface, there is a status bar with various information including the file path '/home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp' and the current line 'Line 31, Col 33, Pos 469'.

3. Fungsi getche() dan getch()

Fungsi input getche() memiliki sifat yang sedikit berbeda dari cin(). Jika dalam cin() jumlah karakter yang diinputkan boleh bebas, dalam getche() hanya sebuah karakter yang dapat diterima. Bila cin() membutuhkan return (enter) untuk mengakhiri input, getche tidak membutuhkan return (enter). Input akan dianggap selesai begitu anda selesai mengetikkan satu karakter. Dan secara otomatis akan melanjutkan perintah-perintah berikutnya.

```
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4 int main(){
5     char ch;
6     ch = getche();
7     cout << " anda telah menekan tombol "<< ch << endl;
8     return 0;
9 }
```

Fungsi input lain yang mirip dengan getche() adalah getch(). Satu-satunya perbedaan antara getche() dan getch() adalah bahwa getche() akan menampilkan karakter yang diinput ke layer sedangkan getch() hanya akan menyimpan karakter masukan ke memori tanpa menampilkannya ke layar. Untuk pendeklarasiannya sama seperti fungsi getche().

4. Operator

Operator adalah suatu simbol yang digunakan untuk melakukan suatu operasi atau manipulasi. Bahasa C merupakan bahasa yang kaya dengan operator yaitu: Operator Aritmatika, Operator Penggeraan (assignment operator), Operator Logika, Operator Unary, Operator Bitwise, Operator Kondisional, dan lain-lain.

A. Operator Aritmatika

Operator aritmatika dalam C++ digunakan untuk melakukan operasi matematika dasar pada variabel numerik.

Contoh :

The screenshot shows the Code::Blocks IDE interface. On the left, the code editor displays a file named main.cpp with the following content:

```
24
25
26
27 // operator aritmatika
28 int angka1 = 10;
29 int angka2 = 2;
30
31 int hasil = angka1 % angka2;
32
33 cout << "Hasilnya adalah " << hasil << endl;
34
35
36 // operasi perbandingan
37 /* int angka1 = 5;
38 int angka2 = 5;
39
40 bool hasil = (angka1 == angka2);
41
42 cout << "Hasil adalah " << hasil << endl;
43 */
44
45 // operator logika
46 /* bool kondisi1 = true;
47 bool kondisi2 = true;
48
49 bool hasil = (kondisi1 || kondisi2);
50 cout << "Hasilnya adalah " << boolalpha << hasil << endl;
51
52
53 // percabangan
54 /*string kata;
55 cout << "Masukkan kata : Halo " << endl;
56
57
58
```

To the right of the code editor is a terminal window titled "Pertemuan 1" showing the execution results:

```
Hasilnya adalah 0
Process returned 0 (0x0) execution time : 0.007 s
Press ENTER to continue.
```

At the bottom of the interface, the status bar shows the path /home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp, the file type C/C++, the encoding Unix (LF), and the current line number Line 34, Col 1, Pos 522.

B. Operator Logika

Operasi logika membandingkan dua buah nilai logika. Nilai logika adalah nilai benar atau salah.

Contoh :

The screenshot shows the Code::Blocks IDE interface. On the left, the code editor displays a file named main.cpp with the following content:

```
34
35
36 /* operasi perbandingan
37 int angka1 = 5;
38 int angka2 = 5;
39
40 bool hasil = (angka1 == angka2);
41
42 cout << "Hasil adalah " << hasil << endl;*/
43
44
45 // operasi logika
46 /* bool kondisi1 = true;
47 bool kondisi2 = true;
48
49 bool hasil = (kondisi1 || kondisi2);
50 cout << "Hasilnya adalah " << boolalpha << hasil << endl;
51
52
53 // percabangan
54 /*string kata;
55 cout << "Masukkan kata : Halo " << endl;
56
57
58
```

To the right of the code editor is a terminal window titled "Pertemuan 1" showing the execution results:

```
Hasilnya adalah true
Process returned 0 (0x0) execution time : 0.007 s
Press ENTER to continue.
```

At the bottom of the interface, the status bar shows the path /home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp, the file type C/C++, the encoding Unix (LF), and the current line number Line 52, Col 1, Pos 864.

C. Operator Perbandingan

Operator perbandingan dalam C++ digunakan untuk membandingkan dua nilai atau variabel dan mengembalikan hasil dalam bentuk nilai boolean (true atau false)

Contoh :

The screenshot shows the Code::Blocks IDE interface. On the left, the code editor displays a file named 'main.cpp' with Pertemuan 1 selected in the project tree. The code contains various operators and conditions. On the right, a terminal window titled 'Pertemuan 1' shows the execution of the program, which outputs 'Hasil adalah 1'. Below the terminal is the build log, which includes compiler settings like LD_LIBRARY_PATH and the command used to run the program.

```
// operator aritmatika
/* int angka1 = 10;
int angka2 = 2;

int hasil = angka1 % angka2;
cout << "Hasilnya adalah " << hasil << endl; */

// operasi perbandingan
int angka1 = 5;
int angka2 = 5;

bool hasil = (angka1 == angka2);
cout << "Hasil adalah " << hasil << endl;

// operator logika
/* bool kondisi1 = true;
bool kondisi2 = true;
```

```
Pertemuan 1
Hasil adalah 1
Process returned 0 (0x0) execution time : 0,008 s
Press ENTER to continue.
```

```
..... Run: Debug in Pertemuan 1 (compiler: GNU GCC Compiler).....
Using for compilation: /home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp
Set variable: LD_LIBRARY_PATH=.:./bin/Debug/Pertemuan 1
Executing: xterm -T Pertemuan 1 -e ./usr/bin/db.console_runner LD_LIBRARY_PATH=.:/home/lightofshine/Downloads/Belajar C++/Pertemuan 1/./bin/Debug/Pertemuan 1 (in /home/lightofshine/Downloads/Belajar C++/Pertemuan 1 /.)
```

5. Kondisional / Percabangan

Untuk menyelesaikan suatu masalah diperlukan pengambilan keputusan, Bahasa C menyediakan beberapa jenis pernyataan berupa operator kondisi sebagai berikut.

1. Pernyataan if
2. Pernyataan if-else
3. Pernyataan switch

Pernyataan pengambilan keputusan di atas memerlukan suatu kondisi sebagai basis pada pengambilan keputusan. Kondisi umum yang dipakai keadaan benar atau salah.

Contoh :

The screenshot shows the Code::Blocks IDE interface. On the left, the code editor displays a file named 'main.cpp' with Pertemuan 1 selected in the project tree. The code includes a switch statement based on user input for channel selection. On the right, a terminal window titled 'Pertemuan 1' shows the execution of the program, which asks for a word and then prints 'Halo' or 'Kata sesuai' based on the input. Below the terminal is the build log, which includes compiler settings like LD_LIBRARY_PATH and the command used to run the program.

```
// percabangan
string kata;
cout << "Masukkan kata : Halo " << endl;
cin >> kata;

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

/*int tv;
cout << "Daftar Chanel Tv : " << endl;
cout << "1.RCTI" << endl;
cout << "2.Indosiar" << endl;

cout << "Masukkan Channel Pilihan : " << endl;
cin >> tv;
switch (tv) {
```

```
Pertemuan 1
Masukkan kata : Halo
Halo
Kata sesuai
Process returned 0 (0x0) execution time : 5,202 s
Press ENTER to continue.
```

```
..... Run: Debug in Pertemuan 1 (compiler: GNU GCC Compiler).....
Using for compilation: /home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp
Set variable: LD_LIBRARY_PATH=.:./bin/Debug/Pertemuan 1
Executing: xterm -T Pertemuan 1 -e ./usr/bin/db.console_runner LD_LIBRARY_PATH=.:/home/lightofshine/Downloads/Belajar C++/Pertemuan 1/./bin/Debug/Pertemuan 1 (in /home/lightofshine/Downloads/Belajar C++/Pertemuan 1 /.)
```

```

main.cpp [Pertemuan 1] - Code::Blocks svn build

File Edit View Search Project Build Debug Tools Plugins Settings Help
Management Projects Symbols
  <global>
    main.cpp
      main():int
        /*/
        ...
        int tv;
        cout << "Daftar Chanel Tv : " << endl;
        cout << "1.RCTI" << endl;
        cout << "2.Indosiar" << endl;
        cout << "Masukkan Channel Pilihan : " << endl;
        cin >> tv;
        switch (tv) {
        case 1:
            cout << "Channel yang anda pilih rcti" << endl;
            break;
        case 2:
            cout << "Channel yang anda pilih indosiar" << endl;
            break;
        default:
            cout << "Channel Tidak Tersedia" << endl;
        }
        // Perulangan
        //for (int i = 0; i<5; i++){
        //}
    }
}

Logs & others
Code:Blocks Search results Build log Buildmessages Debugger
/home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp
C/C++ Unix (LF) UTF-8 Line 87, Col 1, Pos 1491 Insert Read/Wr... default

```

6. Perulangan

Perulangan digunakan untuk mengefisienkan waktu dan meringkas kode program dalam pengeksekusian sub-program yang sama. Hal yang terpenting dalam perulangan adalah harus ada kondisi berhenti.

Perulangan dalam C++ digunakan untuk mengeksekusi satu blok kode berkali-kali, selama kondisi tertentu terpenuhi. Terdapat tiga jenis utama perulangan dalam C++: for loop, while loop, dan do-while loop. Masing-masing memiliki cara penggunaan yang berbeda, tergantung pada kebutuhan program.

Contoh :

```

main.cpp [Pertemuan 1] - Code::Blocks svn build

File Edit View Search Project Build Debug Tools Plugins Settings Help
Management Projects Symbols
  <global>
    main.cpp
      main():int
        ...
        cout << "1.RCTI" << endl;
        cout << "2.Indosiar" << endl;
        cout << "Masukkan Channel Pilihan : " << endl;
        cin >> tv;
        switch (tv) {
        case 1:
            cout << "Channel yang anda pilih rcti" << endl;
            break;
        case 2:
            cout << "Channel yang anda pilih indosiar" << endl;
            break;
        default:
            cout << "Channel Tidak Tersedia" << endl;
        }
        // Perulangan
        for (int i = 0; i<5; i++){
            cout << i << "Hello World" << endl;
        }
    }
}

Logs & others
Code:Blocks Search results Build log Buildmessages Debugger
/home/lightofshine/Downloads/Belajar C++/Pertemuan 1/main.cpp
C/C++ Unix (LF) UTF-8 Line 92, Col 6, Pos 1591 Insert Read/Wr... default

```

IV. Latihan

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

Jawab :

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for a program named 'main.cpp' under the project 'Pertemuan 1'. The code performs arithmetic operations on two floating-point numbers. The output window shows the results of the operations: addition, subtraction, multiplication, and division. The terminal window shows the command to run the program and its execution time.

```
// Latihan No 1
// Minta input dari pengguna
cout << "Masukkan bilangan pertama: ";
cin >> bilangan1;
cout << "Masukkan bilangan kedua: ";
cin >> bilangan2;

// Melakukan operasi aritmatika
float penjumlahan = bilangan1 + bilangan2;
float pengurangan = bilangan1 - bilangan2;
float perkalian = bilangan1 * bilangan2;
float pembagian = bilangan1 / bilangan2;

// Menampilkan hasil
cout << "Hasil penjumlahan: " << penjumlahan << endl;
cout << "Hasil pengurangan: " << pengurangan << endl;
cout << "Hasil perkalian: " << perkalian << endl;
cout << "Hasil pembagian: " << pembagian << endl;

return 0;
```

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

Jawab :

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for a program named 'main.cpp' under the project 'Pertemuan 1'. The code takes an integer input from the user and prints its corresponding word representation. The output window shows the input and the resulting word output. The terminal window shows the command to run the program and its execution time.

```
int angka;
cout << "Masukkan bilangan bulat positif (0 - 100): ";
cin >> angka;

// Memerlukan validitas input
if (angka < 0 || angka > 100) {
    cout << "Input tidak valid." << endl;
    return 1; // Keluar dari program jika input tidak valid
}

// Menampilkan hasil dalam bentuk tulisan
if (angka == 0) cout << "nol";
else if (angka == 1) cout << "satu";
else if (angka == 2) cout << "dua";
else if (angka == 3) cout << "tiga";
else if (angka == 4) cout << "empat";
else if (angka == 5) cout << "lima";
else if (angka == 6) cout << "enam";
else if (angka == 7) cout << "tujuh";
else if (angka == 8) cout << "delapan";
else if (angka == 9) cout << "sembilan";
else if (angka == 10) cout << "sepuluh";
```

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
  *
```

Jawab :

The screenshot shows the Code::Blocks IDE interface. On the left, the project tree shows 'Pertemuan 1' with a single source file 'main.cpp'. The main editor window displays the following C++ code:

```
176     cout << "sembilan"; // satuan
177 }
178
179 cout << endl; // Untuk membuat baris baru
180 return 0;/*
181
182 // Latihan No 3
183 int n;
184 cout << "Input: ";
185 cin >> n;
186
187 // Bagian atas pola
188 for (int i = n; i >= 1; i--) {
189     // Bagian kiri (angka menurun)
190     for (int j = i; j >= 1; j--) {
191         cout << j;
192     }
193     // Bintang
194     cout << "*";
195     // Bagian kanan (angka naik)
196     for (int j = 1; j <= i; j++) {
197         cout << j;
198     }
199     cout << endl;
200 }
```

To the right of the editor is a terminal window titled 'Pertemuan 1' showing the program's output:

```
Process returned 0 (0x0)   execution time : 2.115 s
Press ENTER to continue.
```

At the bottom of the interface, the status bar indicates the current file is 'main.cpp' and the line number is 'Line 183, Col 14, Pos 4761'.

V. Kesimpulan

Dalam praktikum ini, dilakukan instalasi Code Blocks dan pengenalan dasar terhadap bahasa pemrograman C++. Beberapa kesimpulan yang dapat diambil dari praktikum ini adalah:

Instalasi Code Blocks:

- ❖ Code Blocks berhasil diinstal sebagai IDE untuk pengembangan aplikasi menggunakan C++.
- ❖ Pada sistem operasi Linux, setelah instalasi Code Blocks, perlu dilakukan pemasangan build-essential yang mencakup GCC (GNU Compiler Collection) untuk mendukung kompilasi program C++.
- ❖ Setelah instalasi, konfigurasi compiler dilakukan untuk memastikan Code Blocks dapat mendeteksi dan menggunakan GCC dengan benar.
- ❖ Konfigurasi Compiler :

Pengaturan compiler di Code::Blocks merupakan langkah penting untuk menjalankan dan mengkompilasi kode C++.

- ❖ Praktikan memastikan bahwa path ke GCC Compiler telah diatur dengan benar melalui Toolchain Executables agar proses kompilasi dan eksekusi program dapat berjalan lancar.

Pemrograman C++ Dasar:

- ❖ Dasar-dasar bahasa C++ yang diperkenalkan mencakup struktur dasar program, penggunaan fungsi main(), sintaks output (cout), dan penggunaan perintah return untuk menandakan akhir dari program ,serta basic c++ dalam menggunakan variabel ,percabangan, perulangan,operator dll.