

**LAPORAN PRAKTIKUM
STRUKTUR DATA**



Disusun Oleh :

Nurul Maulina Nainggolan

21104053/SE07-01

PROGRAM STUDI REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

MODUL 1

CODE BLOKS IDE & PENGENALAN BAHASA C++

I. TUJUAN

Setelah mempelajari materi dalam pertemuan ini, mahasiswa diharapkan mampu :

- 1) Mahasiswa mampu menginstal dan mengaplikasikan *code blocks* pada perangkat masing-masing.
- 2) Mahasiswa memahami penggunaan *code blocks* dan fitur yang digunakan.
- 3) Mengenal struktur program C++, mulai dari tipe data, variabel, konstanta dan input/output

II. LANDASAN TEORI

1. *Code Blocks*

Code Blocks merupakan *Integrated Development Environment* (IDE) yang dirancang untuk memfasilitasi proses pengembangan perangkat lunak. *Code blocks* memberikan infrastruktur yang solid dan alat yang dibutuhkan untuk menghasilkan perangkat lunak berkualitas. *Code blocks* terintegrasi untuk pemrograman C/C++ dan fortran. *Code blocks* bersifat multi-platform atau cross-platform, artinya dapat diinstal di berbagai sistem operasi seperti *windows*, *linux*, dan *MacOS* tanpa kehilangan fitur apapun. Keuntungan menggunakan IDE ini adalah *opensource* dengan ukuran *installer* yang ringan, output aplikasi yang *lightweight*, *code editor* yang *fancy* dan *useful*.

2. C++

C++ dirancang dan dikembangkan oleh Bjarne Stroustrup, seorang ilmuwan komputer Denmark yang ingin membuat bahasa fleksibel dan dinamis mirip dengan C, namun dengan kemampuan tambahan yang lebih canggih. C++ diupgrade sebanyak lima kali dari tahun 1998 sampai dengan 2017. Kelebihan C++ dibuat untuk mengkompilasi kode yang ramping dan efisien, sambil tetap dapat memberikan

abstraksi tingkat tinggi untuk mengelola proyek pengembangan besar dengan lebih baik.

3. Struktur Program C++

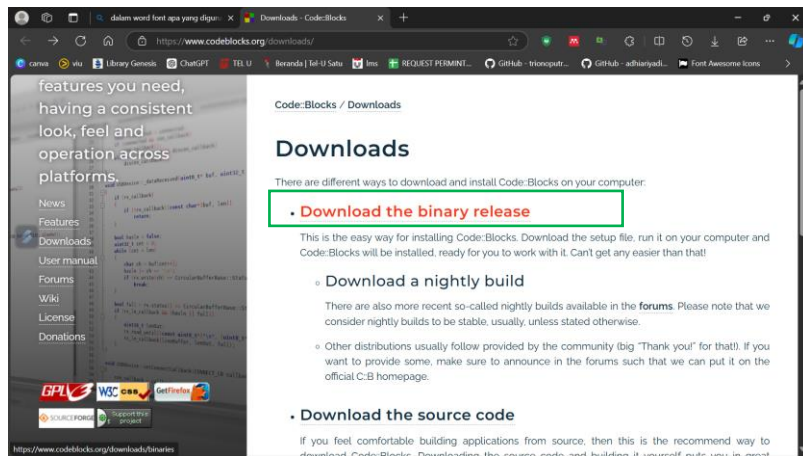
Struktur program C++ secara umum sebagai berikut.

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

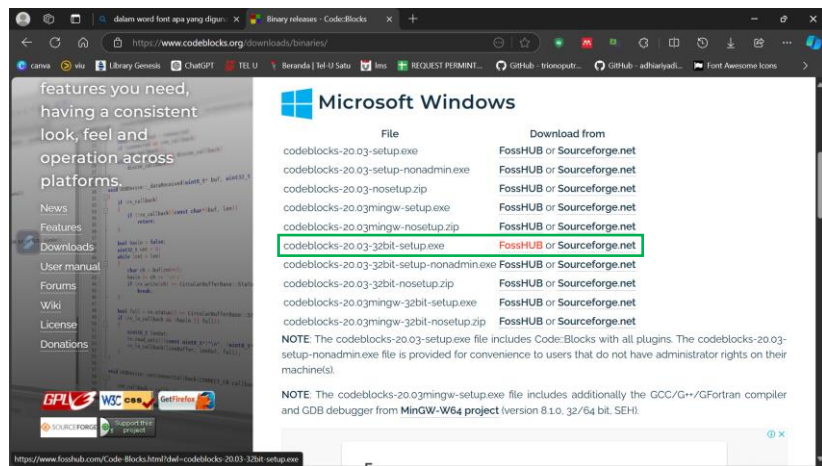
III. GUIDED

1. Instalasi Code Blocks

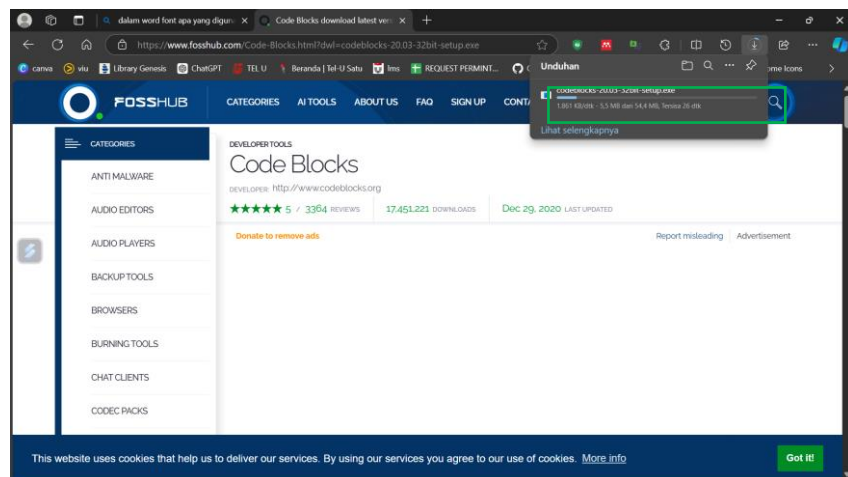
- Pertama-tama, kunjungi situs resmi Code Blocks pada <http://www.codeblocks.org/downloads> . Kemudian pilih *download the binary release* kemudian pilih *file* yang menggunakan *mingw-setup* (e.g.codeblocks-20.03mingw-setup.exe)



Gambar 1. Halaman situs code blocks



Gambar 2. Halaman download code blocks



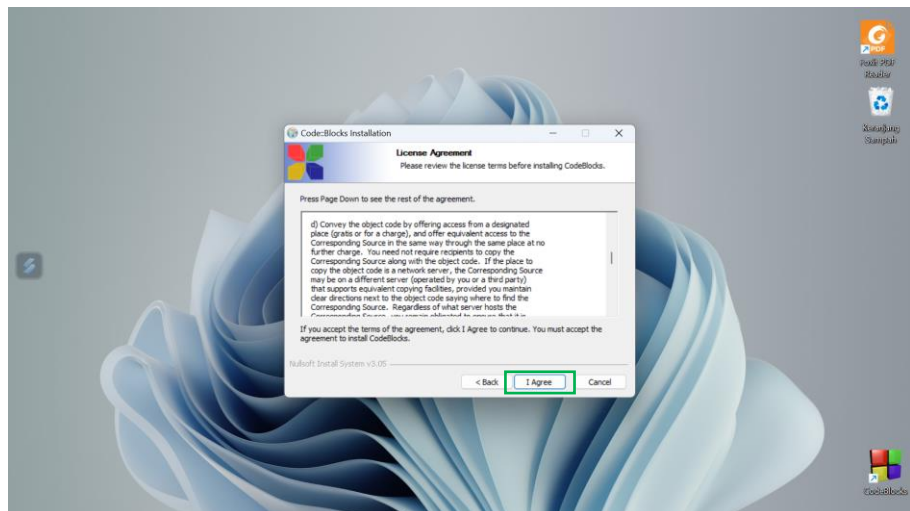
Gambar 3. Download Code Blocks sedang berlangsung

- b. Setelah aplikasi code blocks terdownload, selanjutnya ke bagian instalasi, klik next pada bagian *welcome to CodeBlocks Setup*.



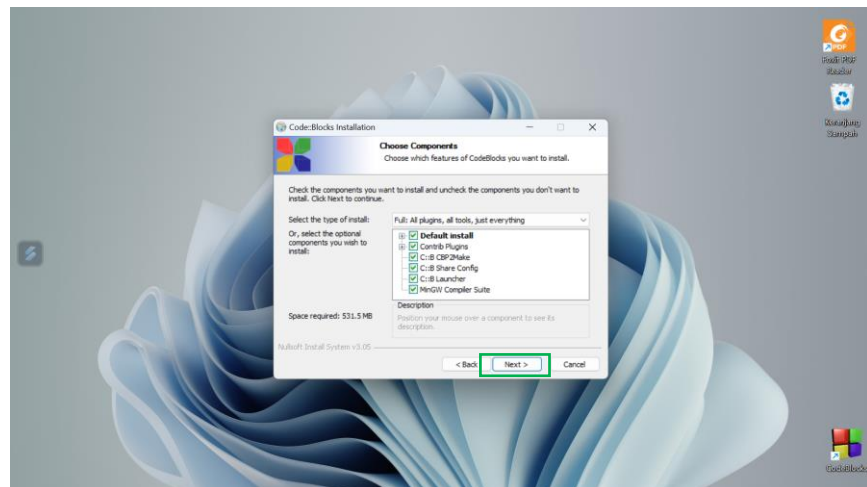
Gambar 4. Halaman Instalasi CodeBlocks

- c. Pilih I Agree, untuk melanjutkan proses instalasi



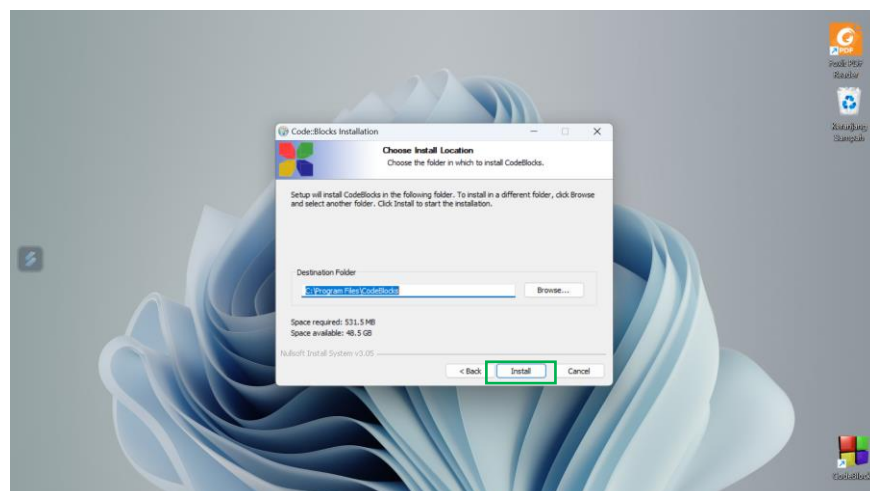
Gambar 5. Halaman Instalasi CodeBlocks

- d. Pilih next

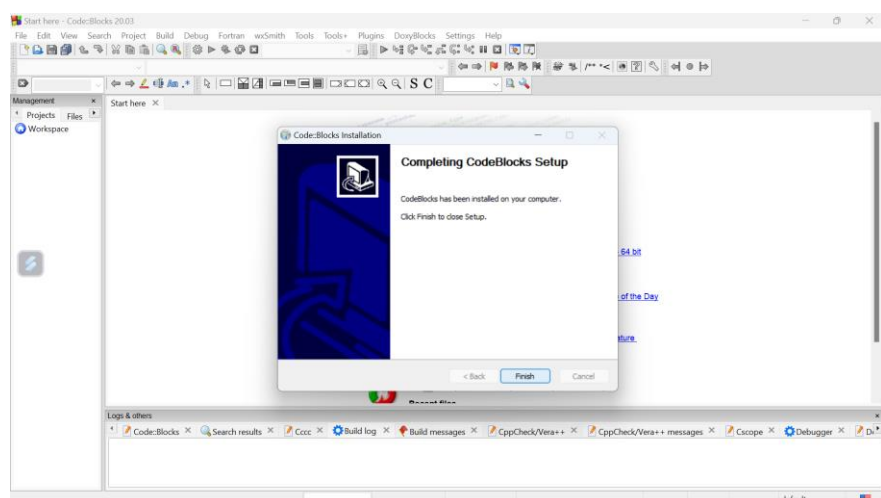


Gambar 6. Halaman Instalasi CodeBlocks

e. Lakukan pengintalan dengan pilih instal

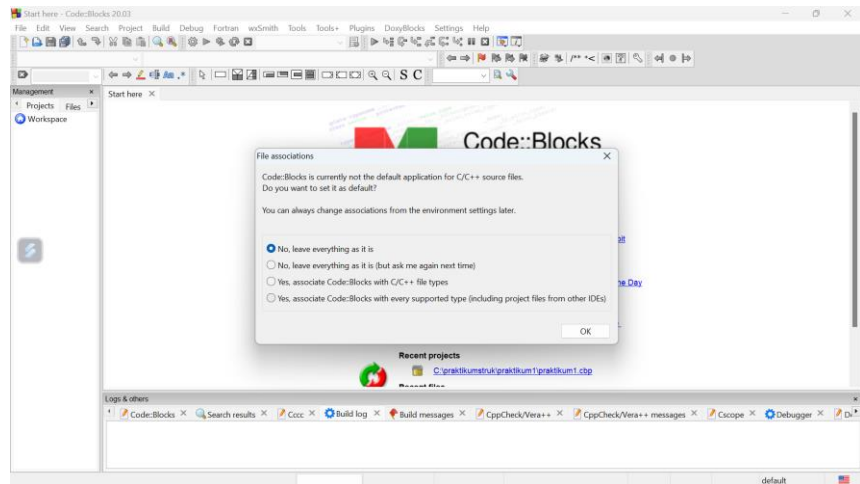


Gambar 7. Halaman Instalasi CodeBlocks



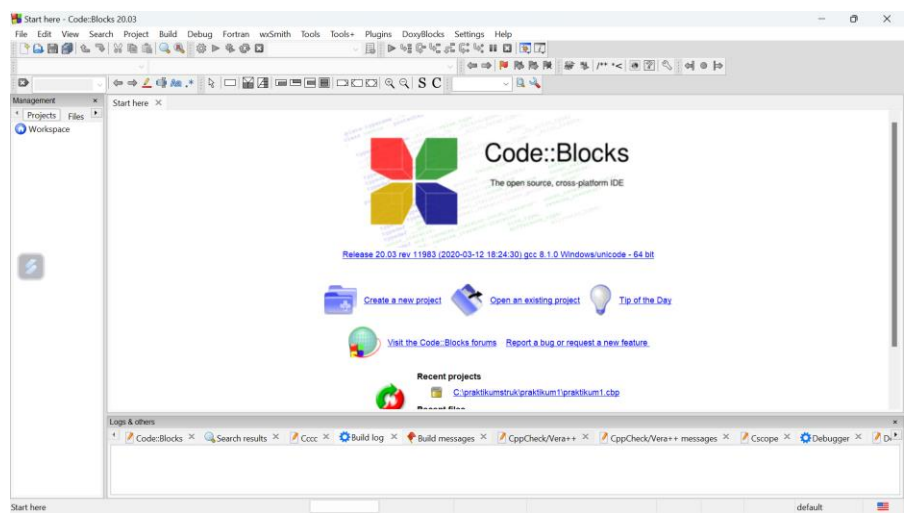
Gambar 8. Halaman Instalasi CodeBlocks

f. Setelah selesai melakukan instalasi, pilih Ok.



Gambar 9. Halaman Instalasi CodeBlocks

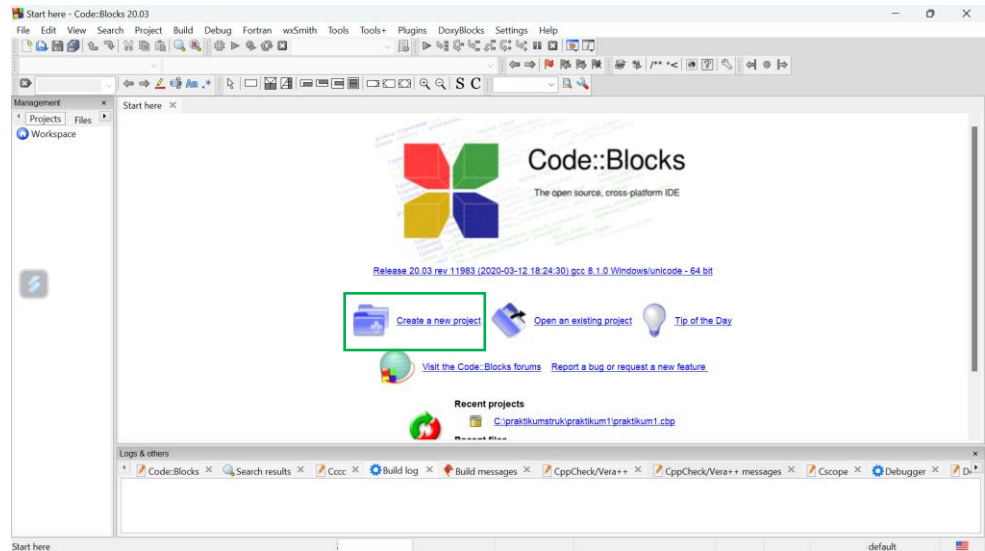
g. Tampilan Code Blocks



Gambar 10. Halaman CodeBlocks

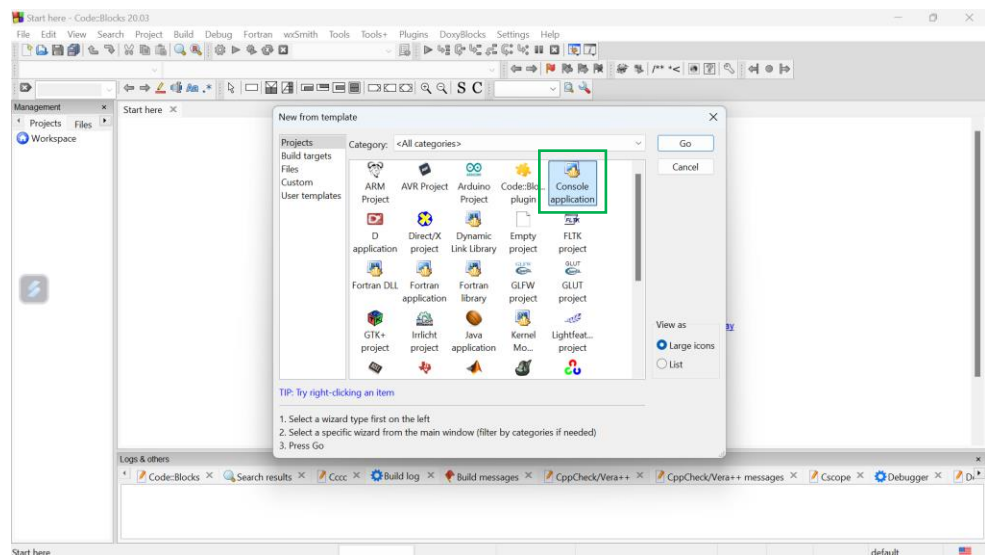
2. Penggunaan Code Blocks

a. Pilih *create a new project*



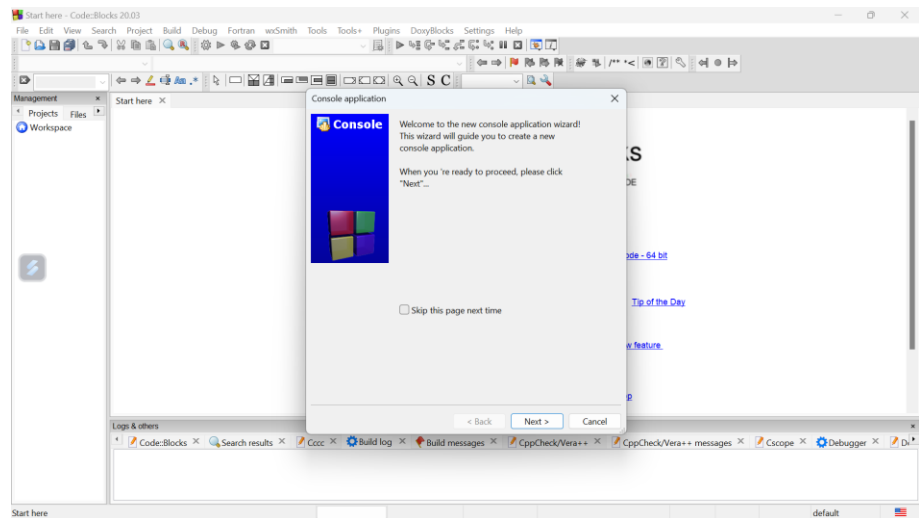
Gambar 11. Halaman CodeBlocks

b. Selanjutnya pilih *console application* dan Klik Go.



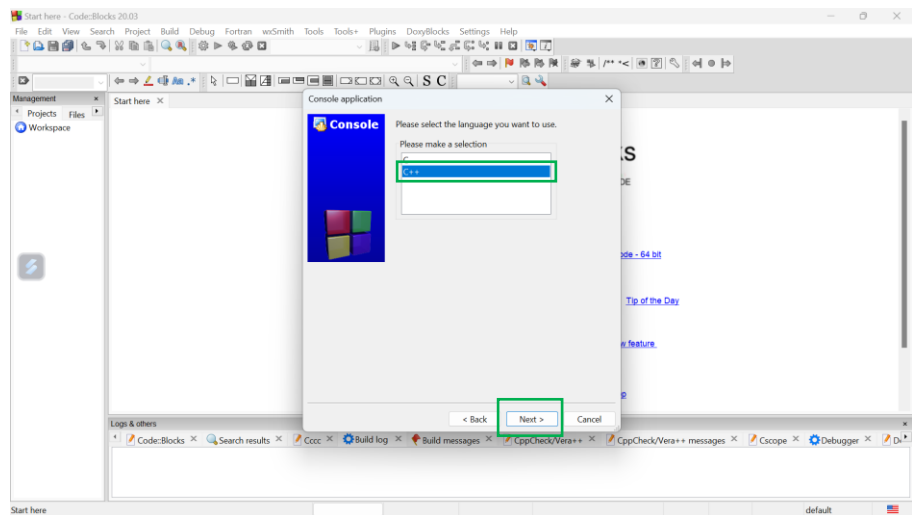
Gambar 12. Halaman CodeBlocks

c. Pilih Next



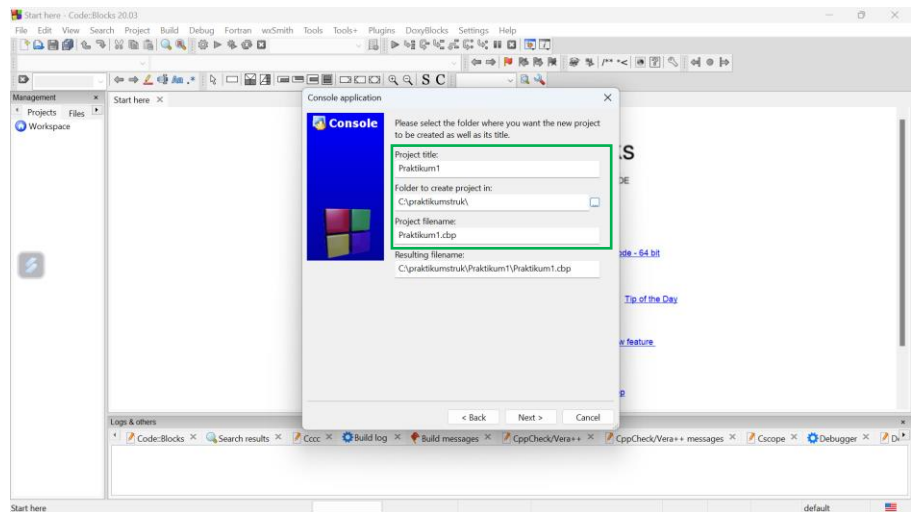
Gambar 13. Halaman CodeBlocks

d. Pilih C++ dan klik next



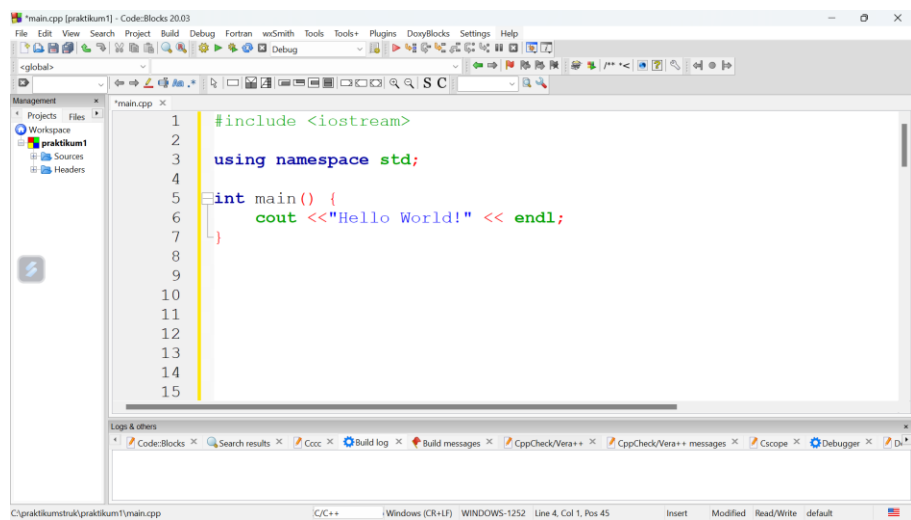
Gambar 14. Halaman CodeBlocks

e. isi nama project, dan file ditempatkan dimana.



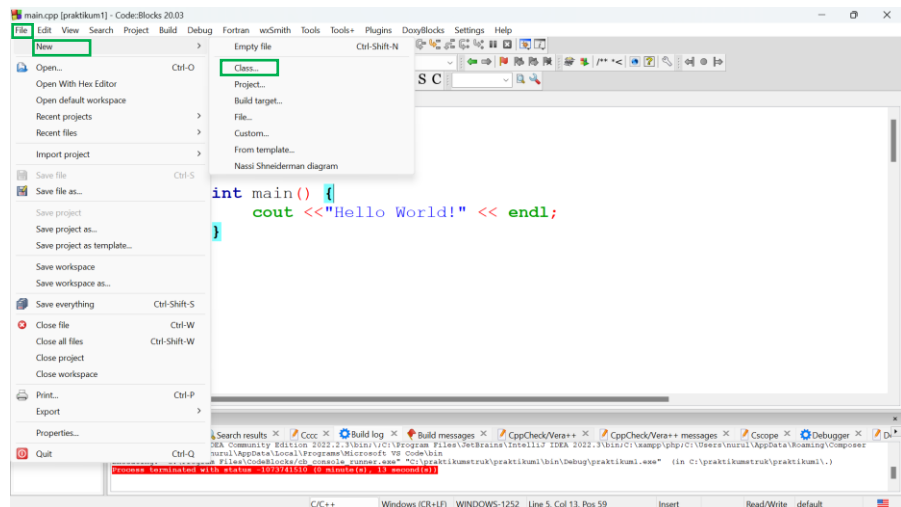
Gambar 15. Halaman CodeBlocks

f. Menulis sintak pada file main.cpp



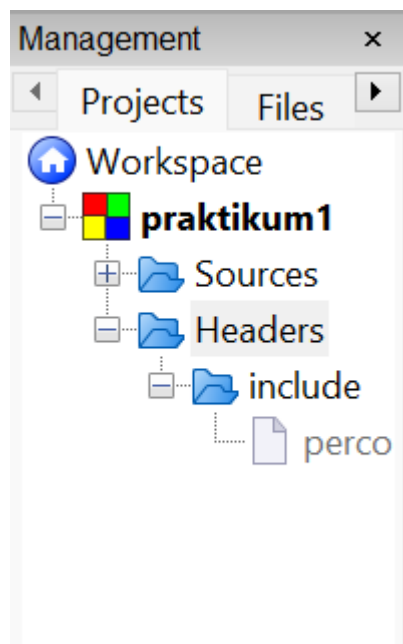
Gambar 16. Halaman Penulisan sintak

g. Membuat class baru dengan klik *File > New > File*. Pada panel kiri pilih *Files*, dan pada panel kanan pilih *C/C++ source* Kemudian Klik *Go*.



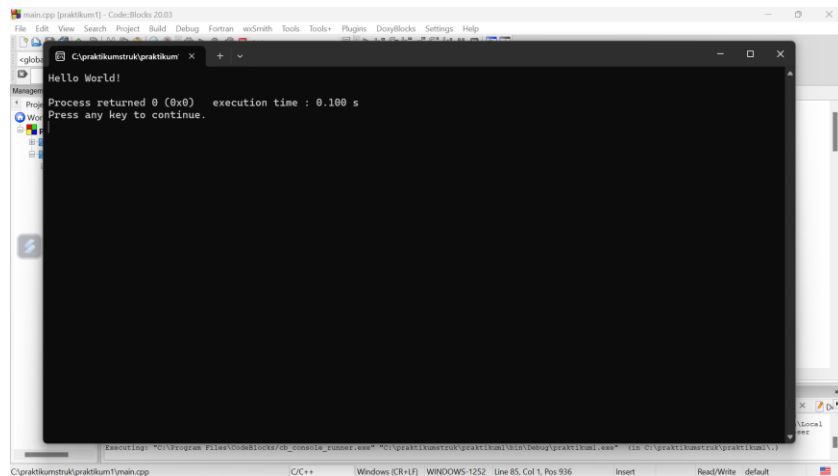
Gambar 17. Halaman Penulisan sintak

- h. Setelah itu, silahkan buat file dengan data sebagai berikut.



Gambar 18. Halaman Penulisan sintak

- i. Setelah menulis file sintak pada main.cpp seperti pada bagian (f), maka ketika dirun akan seperti berikut.



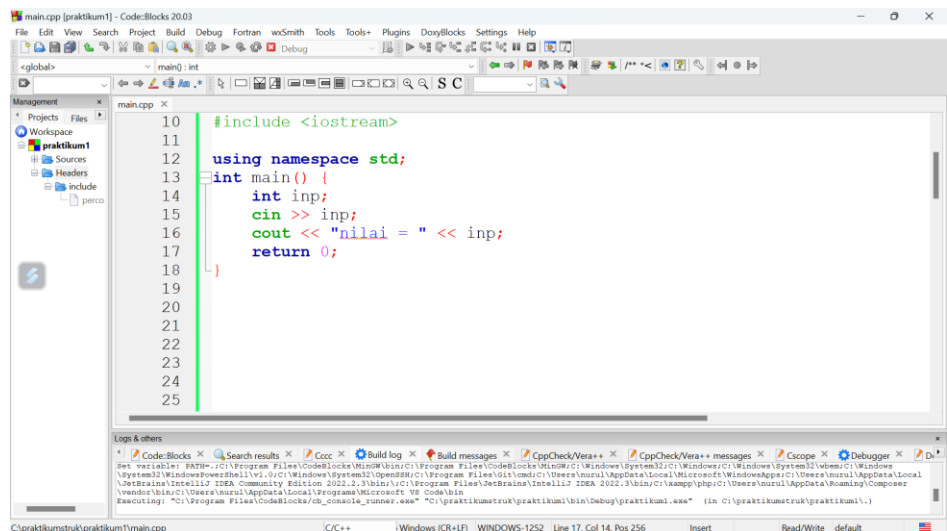
Gambar 18. Halaman Penulisan sintak

3. Input

- Penggunaan Fungsi Cin()

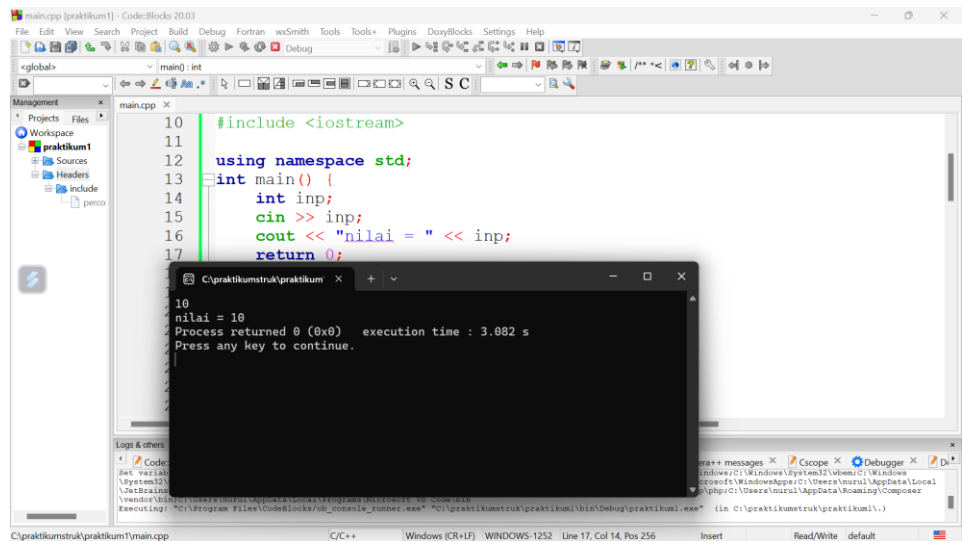
Cin() merupakan salah satu fungsi yang digunakan untuk meminta input keyboard dari user.

Pendeklarasian Input : Fungsi Cin()



Gambar 19. Halaman Pendeklarasian Input Fungsi Cint()

Hasil Run :



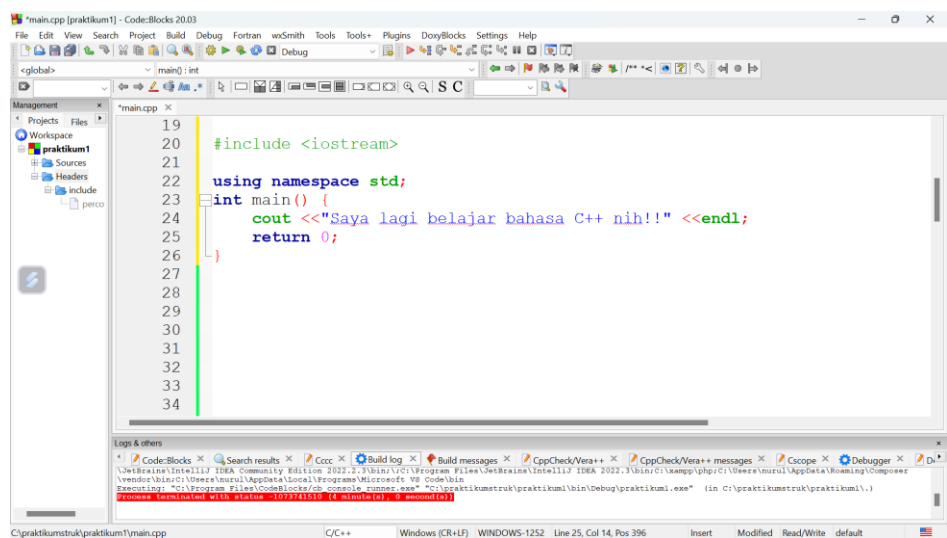
Gambar 20. Halaman Pendeklarasian Input Fungsi Cint()

4. Output

- Fungsi Cout()

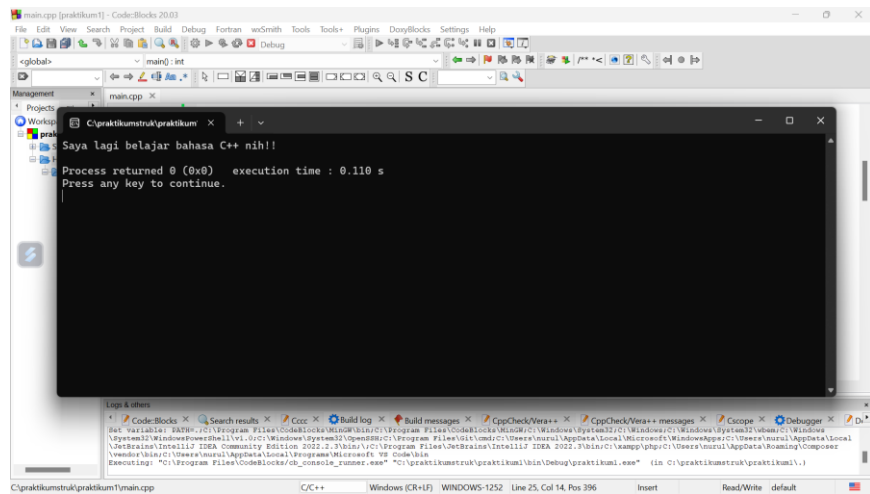
Fungsi ini digunakan untuk mencetak data baik bertipe numerik, ataupun teks, baik konstanta ataupun variabel.

Pendeklarasian Output : Fungsi Cout()



Gambar 21. Halaman Pendeklarasian Output Fungsi Cout()

Hasil Run

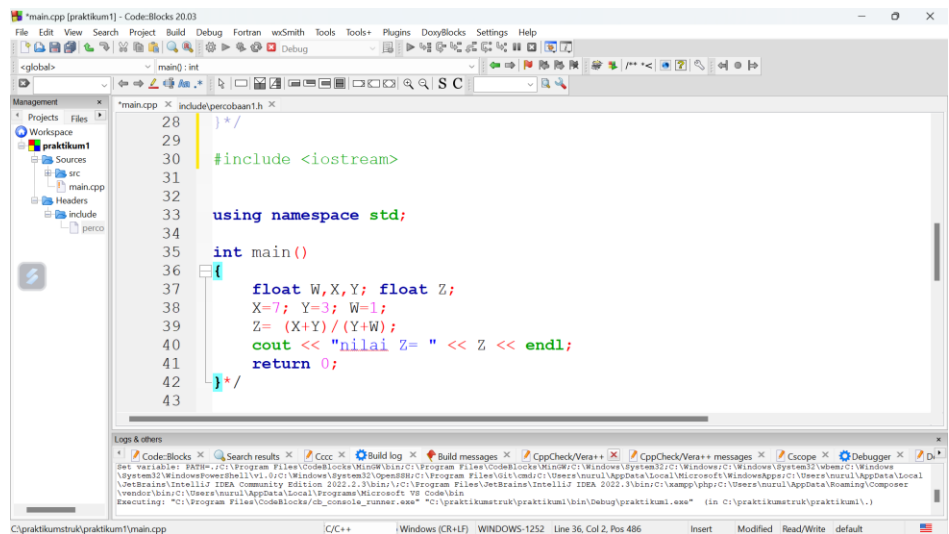


Gambar 21. Halaman Pendeklarasian Output Fungsi Cout()

5. Operator

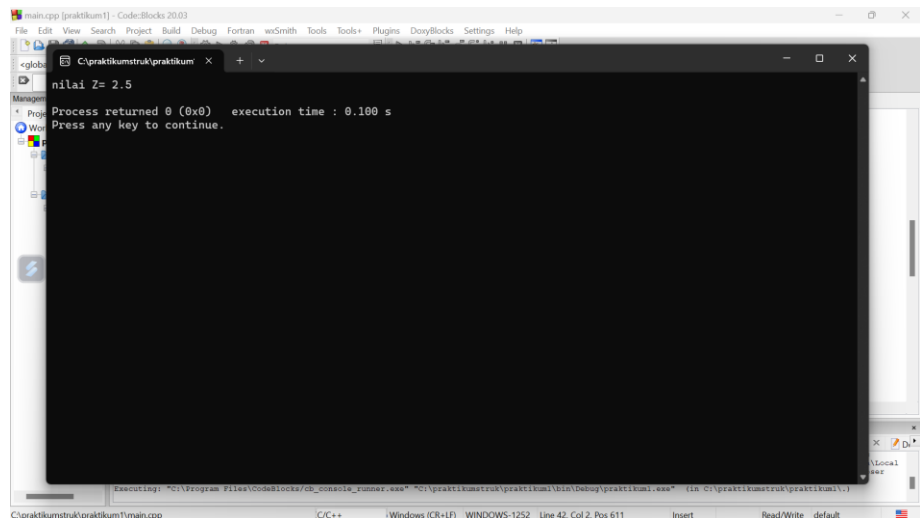
- Operator Aritmatika

Pendeklarasian Operator



Gambar 22. Halaman Pendeklarasian Operator aritmatika()

Hasil Run



Gambar 23. Halaman Pendeklarasian Operator aritmatika()

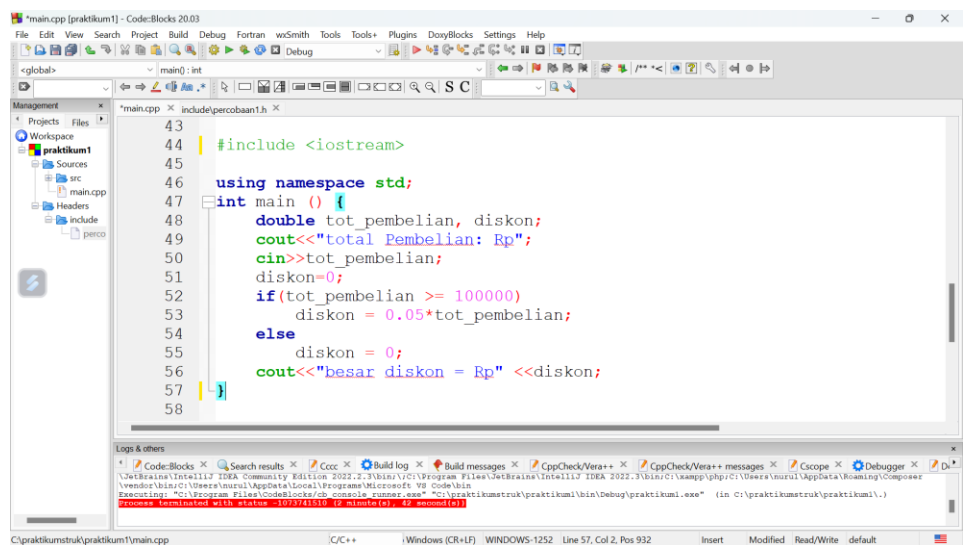
6. Kondisional

- Bentuk 1

if (kondisi) pernyataan ;

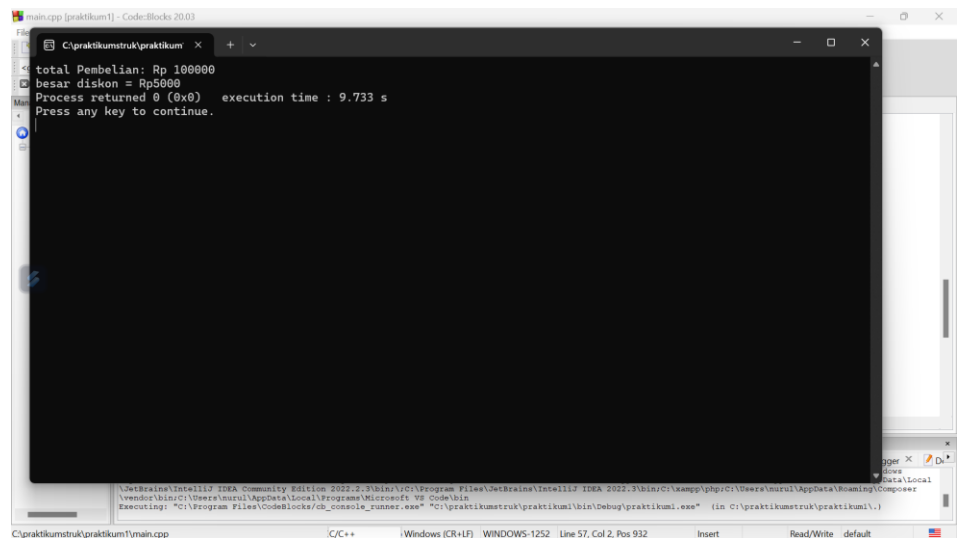
Arti dari perintah if di atas adalah jika kondisi benar maka pernyataan akan dijalankan. Sedangkan kondisi ditulis di antara tanda kurung, dapat berupa ungkapan yang memiliki nilai benar atau salah. Dan pernyataan berupa sebuah pernyataan tunggal pernyataan majemuk atau pernyataan kosong

Pendeklarasian Kondisional :



Gambar 24. Halaman Pendeklarasian Kondisional

Hasil Run :



```
total Pembelian: Rp 100000
besar diskon = Rp5000
Process returned 0 (0x0) execution time : 9.733 s
Press any key to continue.
```

Gambar 25. Halaman Kondisional

- Bentuk 2

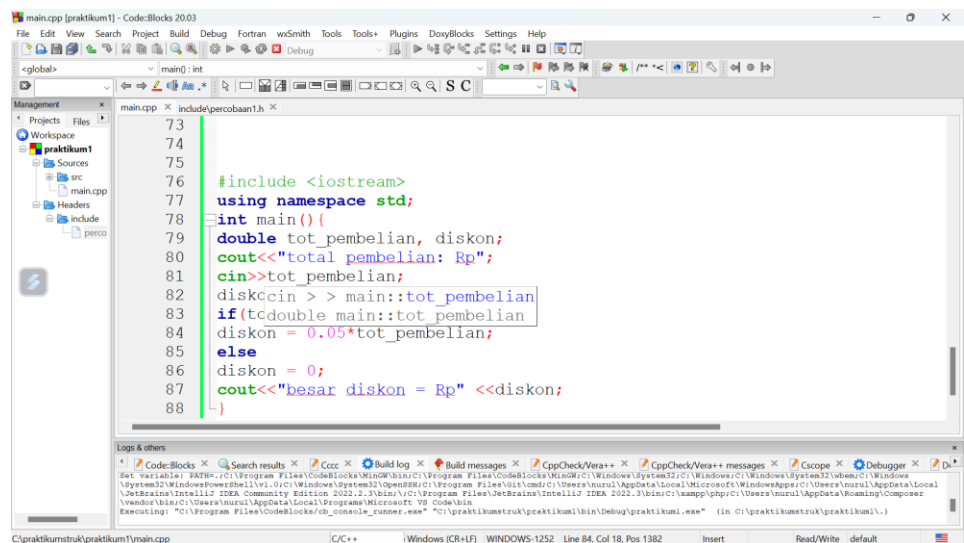
if (kondisi) pernyataan1 ;

else pernyataan2 ;

Arti dari pernyataan *if-else* di atas adalah:

1. Jika kondisi benar, maka pernyataan1 dijalankan.
2. Jika kondisi salah, maka pernyataan2 yang akan dijalankan

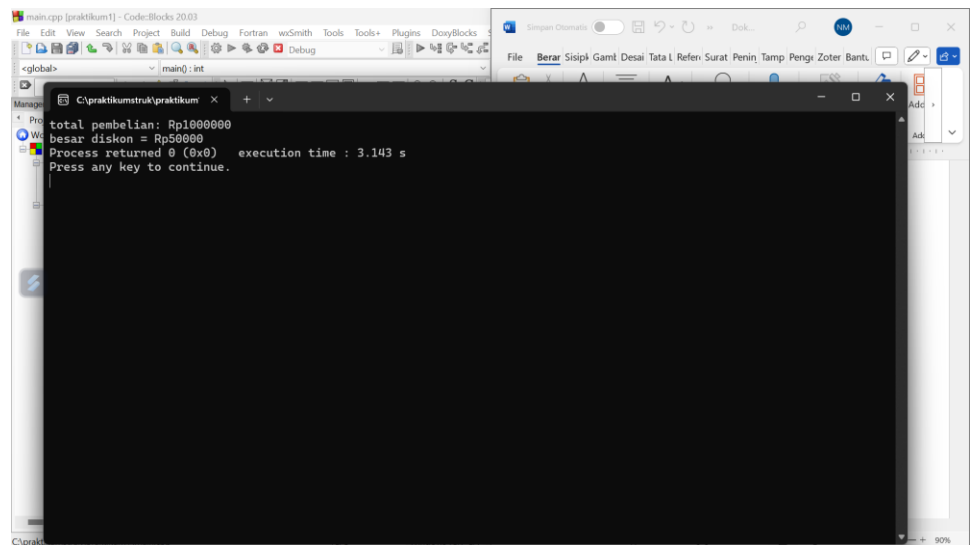
Pendeklarasin Bentuk 2



```
73
74
75
76 #include <iostream>
77 using namespace std;
78 int main(){
79     double tot_pembelian;
80     cout<<"total pembelian: Rp";
81     cin>>tot_pembelian;
82     diskon> > main::tot_pembelian
83     if(tot_pembelian > 0){
84         diskon = 0.05*tot_pembelian;
85     }
86     else
87         diskon = 0;
88     cout<<"besar diskon = Rp" <<diskon;
89 }
```

Gambar 26. Halaman Kondisional

Hasil Run :

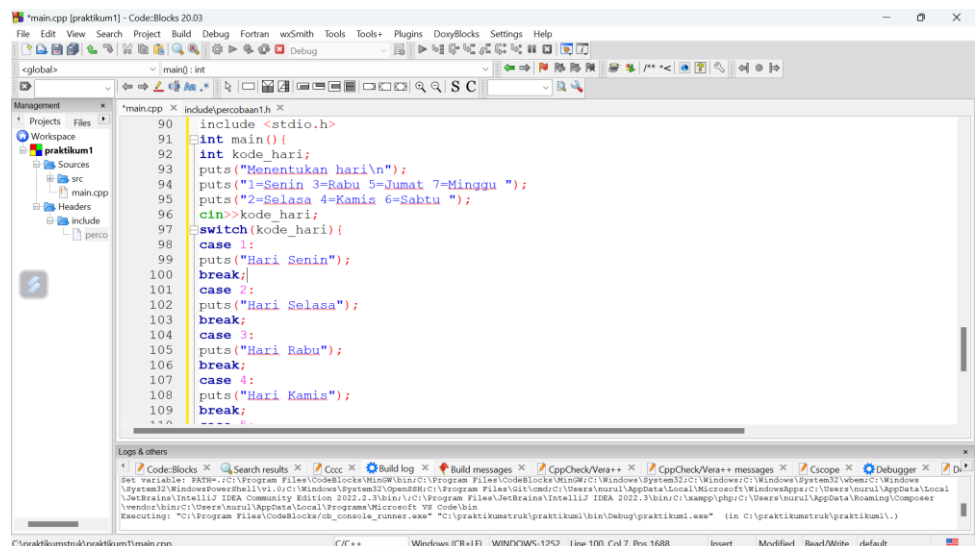


Gambar 27. Halaman Kondisional

- Bentuk 3

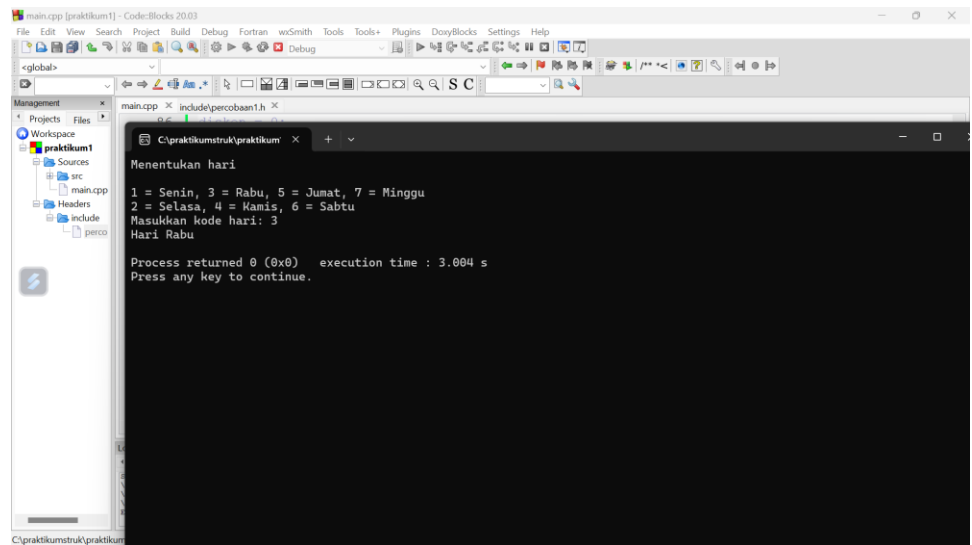
Pernyataan *switch*, merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan banyak alternatif.

Pendeklarasian :



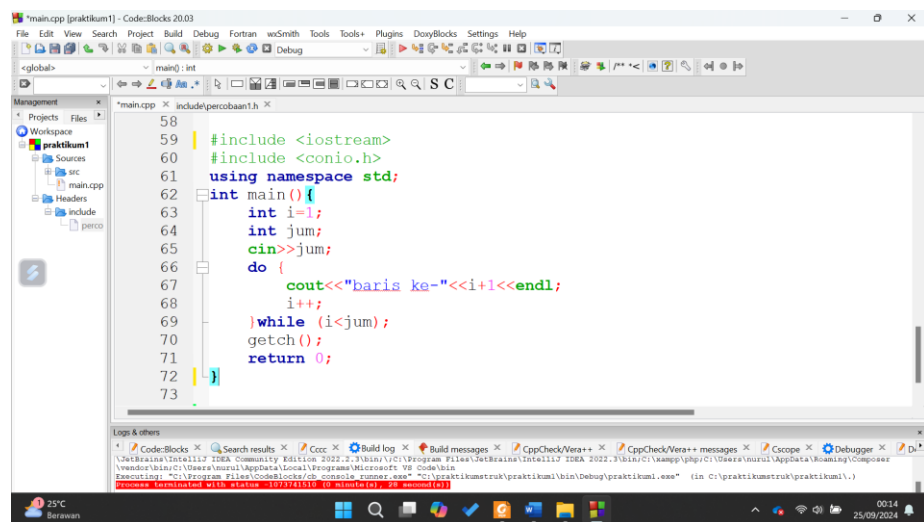
Gambar 28. Halaman Kondisional

Hasil Run :



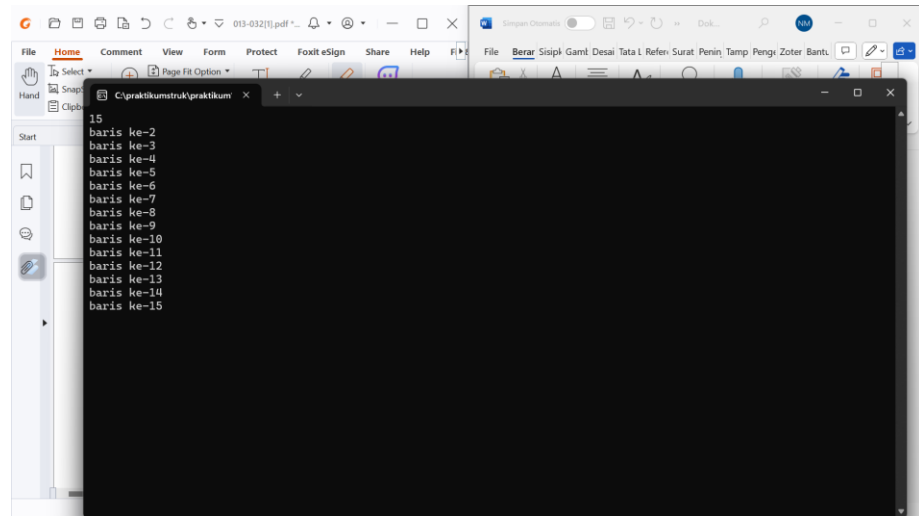
Gambar 29. Halaman Kondisional

7. Perulangan



Gambar 30. Halaman Perulangan

Hasil Run



Gambar 31. Halaman Perulangan

IV. UNGUIDED

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.

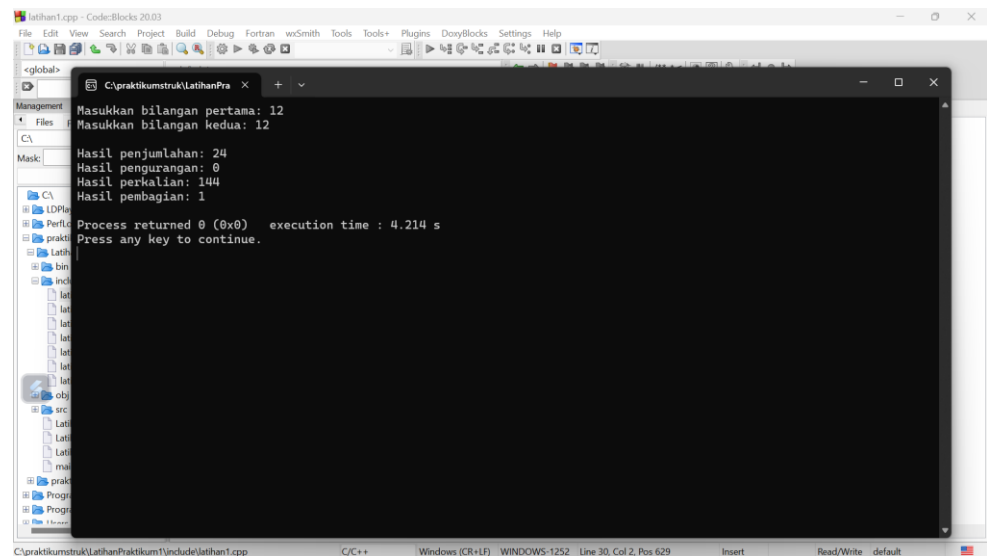
Pendeklarasian (input) :

```

Start here x *latihan3.cpp x *latihan1.cpp x
1 | #ifndef LATIHAN1_CPP
2 | #define LATIHAN1_CPP
3 |
4 |
5 | #include <iostream>
6 | using namespace std;
7 |
8 | int main() {
9 |     float bill1, bill2;
10 |
11 |     // Input bilangan pertama
12 |     cout << "Masukkan bilangan pertama: ";
13 |     cin >> bill1;
14 |
15 |
16 |
17 |     // Input bilangan kedua
18 |     cout << "Masukkan bilangan kedua: ";
19 |     cin >> bill2;
20 |
21 |
22 |
23 |     // Menampilkan hasil operasi jika input valid
24 |     cout << "\nHasil penjumlahan: " << bill1 + bill2 << endl;
25 |     cout << "Hasil pengurangan: " << bill1 - bill2 << endl;
26 |     cout << "Hasil perkalian: " << bill1 * bill2 << endl;
27 |     cout << "Hasil pembagian: " << bill1 / bill2 << endl;
28 |
29 |     return 0;
30 | }*/
31 |
32 |
33 | #endif // LATIHAN1_CPP
34 |
  
```

Gambar 32. Halaman input Latihan 1

Hasil Run (Output) :



```
latihan1.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoryBlocks Settings Help
<global>
C:\praktikumstruktur\LatihanPra
Masukkan bilangan pertama: 12
Masukkan bilangan kedua: 12
Hasil penjumlahan: 24
Hasil pengurangan: 0
Hasil perkalian: 144
Hasil pembagian: 1
Process returned 0 (0x0)   execution time : 4.214 s
Press any key to continue.
```

Gambar 33. Halaman Output Latihan 1

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

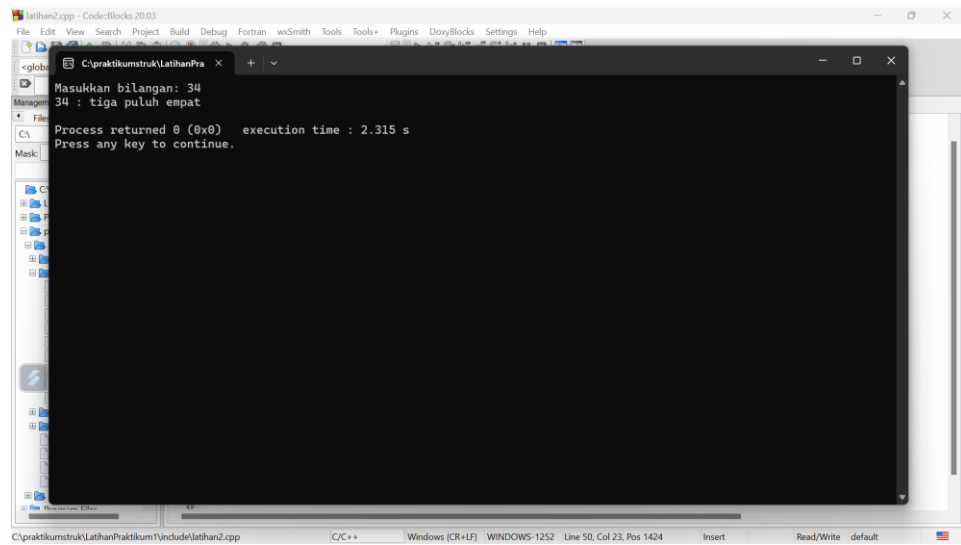
Pendeklarasin (input) :



```
Start here x *latihan3.cpp x *latihan1.cpp x *latihan2.cpp x
3
4
5 #include <iostream>
6 #include <string>
7 using namespace std;
8
9
10 string bilTeks(int bil) {
11     string satuan[] = {"satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
12     string puluhan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
13     string ratusan[] = {"", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
14
15     if (bil < 10) {
16         return satuan[bil];
17     } else if (bil < 20) {
18         return puluhan[bil - 10];
19     } else if (bil < 100) {
20         int puluh = bil / 10;
21         int sisa = bil % 10;
22         return ratusan[puluh] + (sisa != 0 ? " " + satuan[sisa] : "");
23     }
24     return "Diluar jangkauan";
25 }
26
27
28 int main() {
29     float bil;
30     int bilInt;
31
32     // Input bilangan pertama
33     cout << "Masukkan bilangan: ";
34     cin >> bil;
35
36     // Cek apakah input valid
37     if (cin.fail()) {
38         cout << "Error: Input bukan angka" << endl;
39         return 1; // Mengakhiri program dengan kode error
40     }
41
42     // Ubah bilangan pertama ke bentuk integer untuk diubah ke teks
43     bilInt = static_cast<int>(bil);
44     cout << bilInt << " : " << bilTeks(bilInt) << endl;
45
46     return 0;
47 }
```

Gambar 34. Halaman input Latihan 2

Hasil Run (output) :



Gambar 35. Halaman output latihan 2

3. Buatlah program yang dapat memberikan input dan output sebagai berikut.

Input : 3

Output :

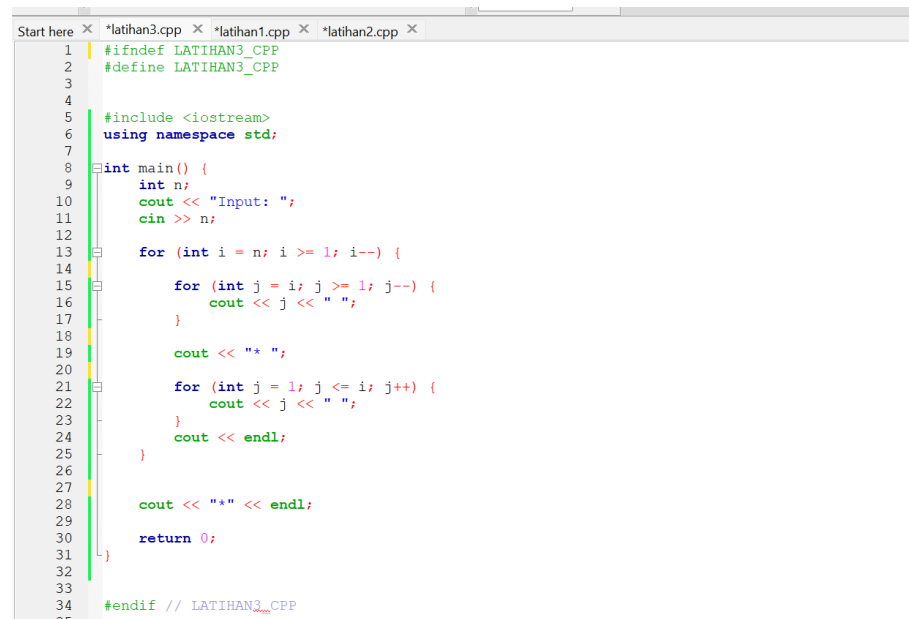
3 2 1 * 1 2 3

2 1 * 1 2

1 * 1

*

Pendeklarasian (input) :



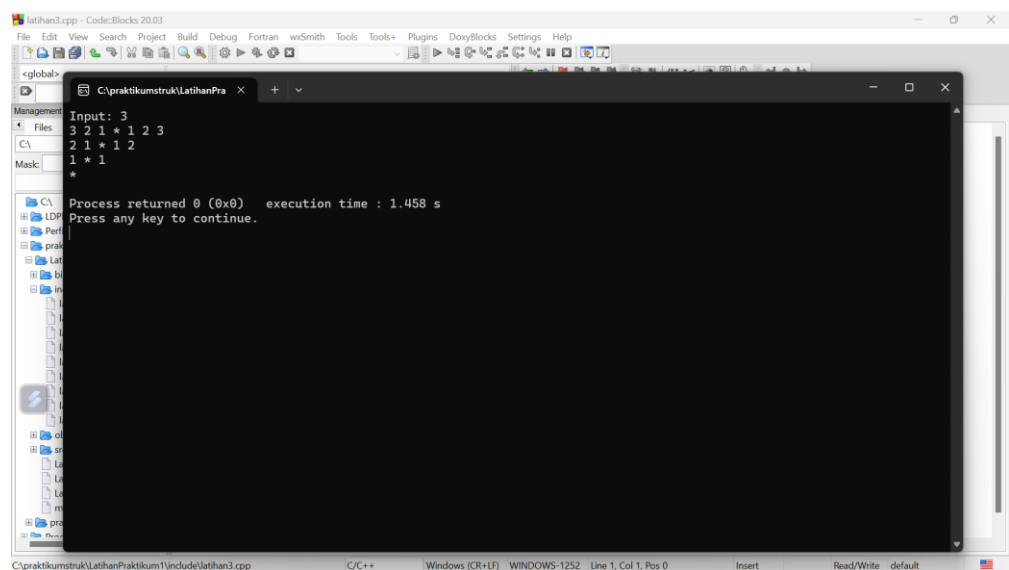
```

1  #ifndef LATIHAN3_CPP
2  #define LATIHAN3_CPP
3
4
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      int n;
10     cout << "Input: ";
11     cin >> n;
12
13     for (int i = n; i >= 1; i--) {
14         for (int j = i; j >= 1; j--) {
15             cout << j << " ";
16         }
17         cout << "\n";
18
19         for (int j = 1; j <= i; j++) {
20             cout << j << " ";
21         }
22         cout << endl;
23     }
24
25     cout << "\n" << endl;
26
27     return 0;
28 }
29
30 #endif // LATIHAN3_CPP

```

Gambar 36. Halaman Input Latihan 3

Hasil Run (Output) :



```

Input: 3
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*
Process returned 0 (0x0)   execution time : 1.458 s
Press any key to continue.

```

Gambar 37. Halaman Output Latihan 3

V. KESIMPULAN

Pada pembelajaran pengenalan C++ pada code block memberikan pemahaman mendasar mengenai penginputan dan output yang digunakan mulai dari perulangan, aritmatik hingga perbandingan. Pada pembelajaran ini dijelaskan bagaimana program C++ berjalan dalam menstruktur data.