

LAPORAN PRAKTIKUM
Modul 1
“CODE BLOCKS IDE & PENGENALAN BAHASA C++ (BAGIAN
PERTAMA)”



Disusun Oleh:
Dewi Atika Muthi -2211104042
SE-07-02

Dosen:
Wahyu Andi Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

Berikut tujuan praktikum:

- 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

Berikut dasar teori yang dibahas dan digunakan pada praktikum:

- Pengenalan Code Blocks IDE
- Struktur dasar program C++
- Tipe data, variabel, dan konstanta
- Operator dalam C++
- Input/output dalam C++
- Pernyataan kondisional (if, if-else, switch)
- Perulangan (for, while, do-while)
- Struktur data dalam C++
- Fungsi dalam C++

3. Guided

A. Instalasi Code Blocks

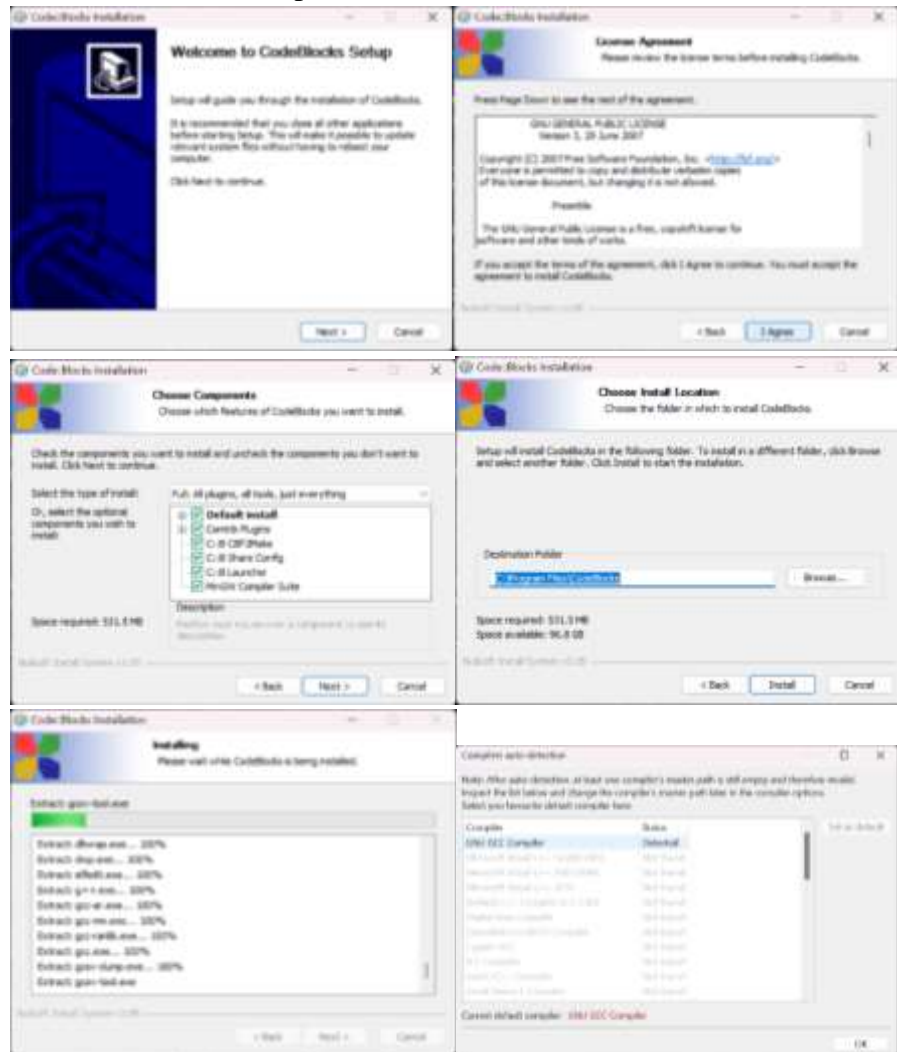
- 1) Download terlebih dahulu file exe pada <http://www.codeblocks.org/downloads>. Pilih Download the binary release kemudian pilih file yang menggunakan mingw-setup (e.g. codeblocks-20.03mingw-setup.exe).



File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

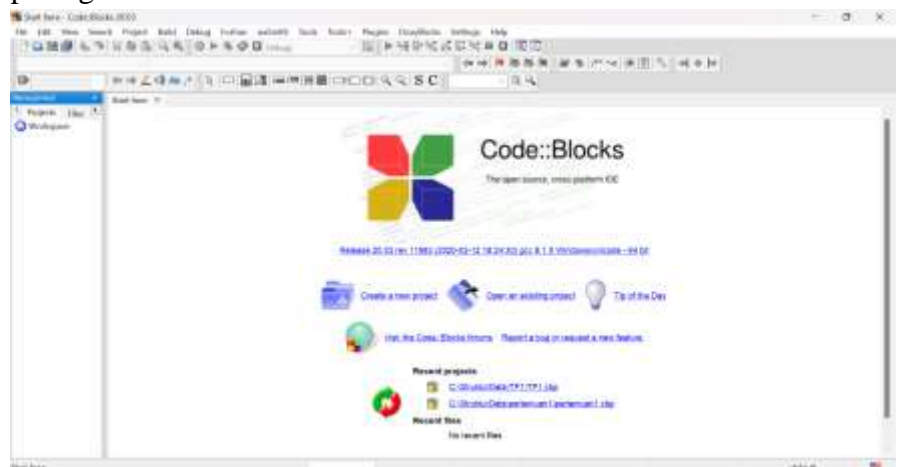
Gambar 1. Web Instalasi Code Blocks

2) Lakukan instalasi sampai selesai



Gambar 2. Proses Instalasi Code Blocks

3) Buka Code Blocks yang telah terinstall, akan muncul tampilan seperti pada gambar di bawah:

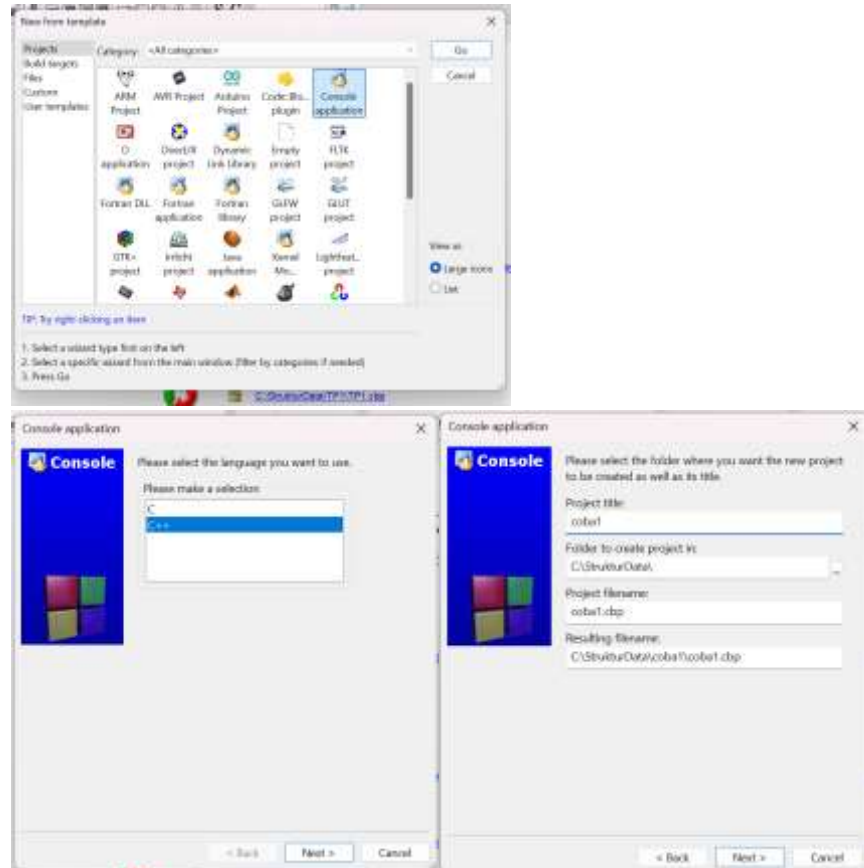


Gambar 3. Start Page Code Blocks

B. Penggunaan Dasar Code Blocks

1) Membuat Project Baru

dengan cara memilih File > New > Projects. Kemudian pada panel kiri pilih Project, pada panel kanan pilih Console application kemudian klik Go seperti pada gambar di bawah:



Gambar 4. Pembuatan Project Baru Code Blocks

Setelah itu klik Next > Pilih Bahasa Pemrograman yang akan digunakan > Isi Project title dan Folder to create project in (tempat menyimpan project) > Klik Finish

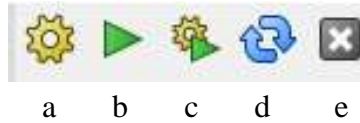
2) Menulis Sintak pada editor seperti pada gambar di bawah



Gambar 5. Mengisi Editor

Simpan perubahan dengan ctrl + s, file yang belum disimpan perubahannya akan muncul seperti `*main.cpp`

- 3) Compile program, yaitu aksi untuk menjalankan program yang telah ditulis sehingga penulis 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 CtrlF10.
- 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 membangun kembali program. Adapun shortcut yang digunakan adalah Ctrl-F11.
- e) Abort merupakan aksi untuk mematikan program yang sedang berjalan

C. Tentang C++ dan penggunaan tipe data dan variable

1. Tentang C++

C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories awal 1980-an, berbasis C ANSI. Awalnya disebut "C with Classes," kemudian disempurnakan pada 1983-1984 dengan penambahan fitur operator dan fungsi overloading, menjadi C++. Simbol "++" mengindikasikan versi yang lebih canggih dari C. Borland International merilis compiler Borland C++ dan Turbo C++, dengan Borland C++ mendukung pemrograman DOS dan Windows. Selain Borland, perusahaan lain juga merilis compiler C++ seperti Topspeed, Zortech, dan Code::Blocks. Dalam praktikum ini, kita akan menggunakan bahasa C++.

2. Tipe data, variable, input dan output

Variabel adalah tempat atau wadah dalam memori komputer yang digunakan untuk menyimpan data.

Tipe data adalah kategori atau jenis dari data yang disimpan di dalam variabel.

Misal:

- **int:** Menyimpan bilangan bulat, seperti 21.
- **float:** Menyimpan angka desimal, seperti 3.9.
- **char:** Menyimpan satu karakter, seperti 'A'.
- **bool:** Menyimpan nilai benar atau salah (true/false).
- **string:** Menyimpan teks, seperti "Dewi Atika."

Fungsi **cin()**, **cin()** merupakan salah satu fungsi yang digunakan untuk meminta inputan keyboard dari user.

Fungsi **cout()**, fungsi ini digunakan untuk mencetak data baik yang bertipe numerik, ataupun teks, baik konstanta ataupun variabel.

Contoh:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     string p;
8     int c = 21;
9     float des = 3.9;
10    bool s = true;
11    char jk = 'P';
12
13    cout << "Masukkan nama anda: ";
14    cin >> p;
15    cout << "Halo selamat datang " << p << "! Kamu berumur " << c
16    << " tahun dengan JK : " << jk << endl << endl;
17 }
18
```

Gambar 6. Contoh Masukan

Output:

```
C:\StrukturData\coba\bin\De x + v
Masukkan nama anda: Atika
Halo selamat datang Atika! Kamu berumur 21 tahun dengan JK : P

Process returned 0 (0x0)   execution time : 2.340 s
Press any key to continue.
```

Gambar 7. Hasil Output

Jadi, variabel adalah "wadah" untuk menyimpan data, dan tipe data menentukan jenis data yang dapat disimpan di dalam variabel tersebut.

D. Implementasi operator aritmatika

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

Tabel 1-4 Operator, Arah Proses, dan Jenjangnya

Kategori (Arti)	Operator	Kategori (Arti)	Operator
Panggilan fungsi, subscript array, dan elemen struktur data	() [] ->	Operator Hubungan (sama dengan, tidak sama dengan)	== !=
Operator Unary (NOT, komplemen, negasi, inkremen, dekremen , address, indirection)	!	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 (assignment, assignment perkalian, assignment pembagian, assignment mod, assignment penjumlahan, assignment pengurangan)	=
	/		*=
	%		/=
Operator Aritmatika (Pertambahan, Pengurangan)	+		%=
	-		+=
			-=

Operator Bitwise Pergeseran Bit (shift kiri, shift kanan)	<< >>	Operator Pengerjaan Bitwise (assignment AND bitwise, assignment OR bitwise, assignment XOR bitwise, assignment shift kanan)	&= ^= = <<= >>=
Operasi Hubungan (kurang dari, kurang dari atau sama dengan, lebih dari, lebih dari atau sama dengan)	< <= > >=	Operator Koma	,

Misal:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int ang1 = 20;
8      int ang2 = 2;
9
10     int hasilkali = ang1 * ang2;
11     int hasiltambah = ang1 + ang2;
12     int sisa = ang1 % ang2;
13     bool tf = (ang1 == ang2);
14
15
16     cout << "Jika " << ang1 << " x " << ang2 << " = " << hasilkali << endl;
17     cout << "Jika " << ang1 << " + " << ang2 << " = " << hasiltambah << endl;
18     cout << "Sisa hasil bagi " << ang1 << " oleh " << ang2 << " = " << sisa << endl;
19     cout << "Hasilnya adalah " << boolalpha << tf << endl;
20
21 }

```

Gambar 8. Contoh Masukkan Operasi Aritmatika

Output:

```

C:\StrukturData\coba1\bin\De x + v
Jika 20 x 2 = 40
Jika 20 + 2 = 22
Sisa hasil bagi 20 oleh 2 = 0
Hasilnya adalah false

Process returned 0 (0x0)   execution time : 0.092 s
Press any key to continue.

```

Gambar 9. Hasil Output Masukkan Operasi Aritmatika

E. Penggunaan pernyataan if-else

Misal:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string kataa;
8      cout << "Masukkan kata = kinder" << endl;
9      cout << "Input: ";
10     cin >> kataa;
11
12     if(kataa == "kinder") {
13         cout << "p: Kata sesuai" << endl;
14     } else {
15         cout << "p: kata tidak sesuai" << endl << endl;
16     }
17 }

```

Gambar 10. Contoh Masukkan Percabangan If-Else

Output:

```

C:\StrukturData\coba1\bin\De x + v
Masukkan kata = kinder
Input: kinder
p: Kata sesuai

Process returned 0 (0x0)   execution time : 1.500 s
Press any key to continue.

```

Gambar 11. Hasil Output Percabangan If-Else

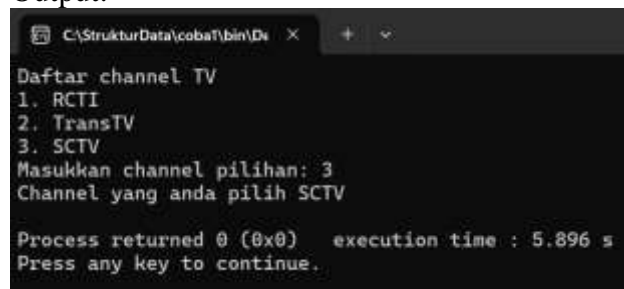
F. Penggunaan switch case

Misal:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int tv;
8     cout << "Daftar channel TV " << endl;
9     cout << "1. RCTI" << endl;
10    cout << "2. TransTV" << endl;
11    cout << "3. SCTV" << endl;
12
13    cout << "Masukkan channel pilihan: ";
14    cin >> tv;
15
16    switch (tv) {
17        case 1:
18            cout << "Channel yang anda pilih RCTI " << endl;
19            break;
20        case 2:
21            cout << "Channel yang anda pilih TransTV " << endl;
22            break;
23        case 3:
24            cout << "Channel yang anda pilih SCTV " << endl;
25            break;
26        default:
27            cout << "channel tidak tersedia" << endl;
28    }
29
30 }
```

Gambar 12. Contoh Masukan untuk Penggunaan Switch Case

Output:



```
C:\StrukturData\coba\bin\Ds x + v
Daftar channel TV
1. RCTI
2. TransTV
3. SCTV
Masukkan channel pilihan: 3
Channel yang anda pilih SCTV

Process returned 0 (0x0)   execution time : 5.896 s
Press any key to continue.
```

Gambar 13. Hasil Output Masukkan Switch Case

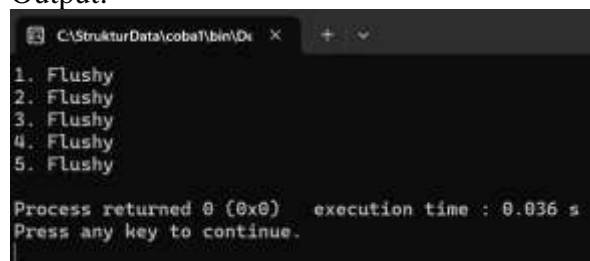
G. Membuat perulangan sederhana

Misal:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     for (int i=0; i<5; i++){
8         cout << 1+i << ". Flushy" << endl;
9     }
10
11 }
```

Gambar 14. Contoh Masukan Looping

Output:



```
C:\StrukturData\coba\bin\Ds x + v
1. Flushy
2. Flushy
3. Flushy
4. Flushy
5. Flushy

Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

Gambar 15. Hasil Output Looping

4. Unguided

- 1) Membuat program kalkulator sederhana

Jawab:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float bil1, bil2;
6
7     cout << "Masukkan dua bilangan: ";
8     cout << endl;
9     cout << "Bil 1 = ";
10    cin >> bil1;
11    cout << "Bil 2 = ";
12    cin >> bil2;
13
14    cout << "Hasil penjumlahan: " << bil1 + bil2 << endl;
15    cout << "Hasil pengurangan: " << bil1 - bil2 << endl;
16    cout << "Hasil perkalian: " << bil1 * bil2 << endl;
17    if (bil2 != 0) {
18        cout << "Hasil pembagian: " << bil1 / bil2 << endl;
19    } else {
20        cout << "Pembagian tidak dapat dilakukan (pembagi nol)" << endl;
21    }
22
23    return 0;
24 }
25
```

Output:

```
C:\StrukturData\cobaf\bin\Dr x + v
Masukkan dua bilangan:
Bil 1 = 10.8
Bil 2 = 2.4
Hasil penjumlahan: 13.2
Hasil pengurangan: 8.4
Hasil perkalian: 25.92
Hasil pembagian: 4.5

Process returned 0 (0x0)   execution time : 6.825 s
Press any key to continue.
```

- 2) Program konversi angka ke teks

```
1 #include <iostream>
2
3 using namespace std;
4
5 string satuan[] = {"", "satu", "dua", "tiga", "empat",
6 "lima", "enam", "tujuh", "delapan", "sembilan"};
7 string belasan[] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas",
8 "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
9
10 string angkaKeTulisan(int num) {
11     if (num == 0) return "nol";
12     else if (num < 10) return satuan[num];
13     else if (num < 20) return belasan[num - 10];
14     else if (num < 100) return satuan[num / 10] + " puluh " + satuan[num % 10];
15     else if (num == 100) return "seratus";
16     else return "Angka di luar jangkauan";
17 }
18
19 int main() {
20     int angka;
21     cout << "Masukkan angka (0-100): ";
22     cin >> angka;
23
24     if (angka >= 0 && angka <= 100) {
25         cout << angka << ": " << angkaKeTulisan(angka) << endl;
26     } else {
27         cout << "Angka di luar jangkauan." << endl;
28     }
29
30     return 0;
31 }
32
```

Output:

Angka <100:

```
C:\StrukturData\coba1\bin\De x + v
Masukkan angka (0-100): 89
89: delapan puluh sembilan

Process returned 0 (0x0)   execution time : 1.650 s
Press any key to continue.
```

Angka >100:

```
C:\StrukturData\coba1\bin\De x + v
Masukkan angka (0-100): 301
Angka di luar jangkauan.

Process returned 0 (0x0)   execution time : 2.699 s
Press any key to continue.
```

3) Membuat pola menggunakan perulangan

Jawab:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      cout << "Masukkan angka: ";
7      cin >> n;
8
9      for (int i = n; i >= 1; i--) {
10         // cetak angka menurun
11         for (int j = i; j >= 1; j--) {
12             cout << j << " ";
13         }
14
15         // cetak tanda *
16         cout << " * ";
17
18         // cetak angka menaik
19         for (int j = 1; j <= i; j++) {
20             cout << j << " ";
21         }
22
23         cout << endl;
24     }
25
26     return 0;
27 }
28
```

Output:

```
C:\StrukturData\coba1\bin\De x + v
Masukkan angka: 4
4 3 2 1 * 1 2 3 4
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1

Process returned 0 (0x0)   execution time : 0.973 s
Press any key to continue.
```

5. Kesimpulan

Praktikum ini memberikan pengenalan dasar tentang pemrograman C++ menggunakan Code Blocks IDE. Mahasiswa belajar tentang struktur program, tipe data, variabel, operator, input/output, pernyataan kondisional, perulangan, dan struktur data sederhana. Melalui latihan guided dan unguided, mahasiswa dapat mengaplikasikan konsep-konsep tersebut untuk membuat program-program sederhana.