

LAPORAN PRAKTIKUM
MODUL 1
CODE BLOCKS IDE & PENGENALAN C++ (BAGIAN PERTAMA)



Disusun Oleh:
Dhiemas Tulus Ikhsan 2311104046
SE-07-02

Dosen :
Wahyu Andi Saputra, S.Pd., M.Eng

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

Tujuan dari praktikum ini mencakup beberapa aspek penting dalam pemahaman dan penggunaan Code Blocks IDE serta penerapan konsep dasar pemrograman dalam bahasa C++. Pertama, mahasiswa diharapkan dapat mengenal lingkungan kerja Code Blocks dengan baik, sehingga mampu memanfaatkan fitur-fitur yang tersedia untuk menunjang pengembangan program secara efisien. Selain itu, mereka juga diharapkan mampu memahami cara penggunaan Code Blocks secara menyeluruh, termasuk keterampilan troubleshooting yang diperlukan untuk mengatasi berbagai masalah teknis yang mungkin muncul selama proses pengembangan.

Dalam praktikum ini, mahasiswa juga akan mempelajari dan mengimplementasikan berbagai operator dalam program, serta memahami konsep dasar seperti tipe data dan variabel dalam bahasa C++. Pemahaman mendalam tentang penggunaan operator input/output juga menjadi fokus, agar mahasiswa dapat menerapkannya dengan benar dalam program mereka. Selain itu, praktikum ini bertujuan untuk memperkenalkan cara membuat program sederhana dengan memanfaatkan fungsi kondisional, sehingga mahasiswa mampu mengimplementasikan logika keputusan dalam kode mereka secara efektif. Melalui semua ini, mahasiswa diharapkan dapat menguasai dasar-dasar pemrograman dalam C++ menggunakan Code Blocks dan menerapkan keterampilan tersebut dalam berbagai konteks pengembangan perangkat lunak.

2. Landasan Teori

Code C++ adalah bahasa pemrograman yang dikembangkan sebagai peningkatan dari bahasa C, yang memasukkan paradigma pemrograman berorientasi objek. C++ merupakan bahasa yang imperatif dan dikompilasi, dirancang untuk pemrograman sistem dan aplikasi.

Pada praktikum Struktur Data ini, digunakan teks editor atau Integrated Development Environment (IDE) yaitu Code::Blocks. Code::Blocks adalah lingkungan pengembangan terintegrasi yang mendukung pemrograman bahasa C, C++, dan Fortran. Berbagai compiler gratis tersedia di sana, seperti GNU GCC Compiler, Cygwin Compiler, Borland C++ Compiler, hingga compiler untuk mikrokontroler seperti GNU Compiler for AVR, ARM, MSP430, PowerPC, serta Keil C51 Compiler, IAR 8051 Compiler, dan IAR ARM Compiler.

Code::Blocks bersifat multi-platform atau cross-platform, yang artinya dapat diinstal di berbagai sistem operasi seperti Windows, Linux, dan MacOS tanpa kehilangan fitur-fitur utamanya. Semua fitur akan tetap sama di setiap sistem operasi yang digunakan. Code::Blocks merupakan IDE yang open-source, gratis, dan dapat diakses oleh siapa saja tanpa biaya. IDE ini berfokus pada dukungan untuk bahasa pemrograman C, C++, dan Fortran. Selain itu, Code::Blocks memiliki fleksibilitas yang tinggi karena dapat dikustomisasi dengan berbagai plugin, sehingga cocok untuk pengembang perangkat lunak dari berbagai tingkatan, mulai dari pemula hingga profesional.

3. Guided

- a. Download terlebih dahulu *file.exe* nya pada link berikut ini [Downloads - Code::Blocks \(codeblocks.org\)](#)
- b. Kemudian klik pada “Download the binary release”

Downloads

There are different ways to download and install Code::Blocks on your computer:

• Download the binary release

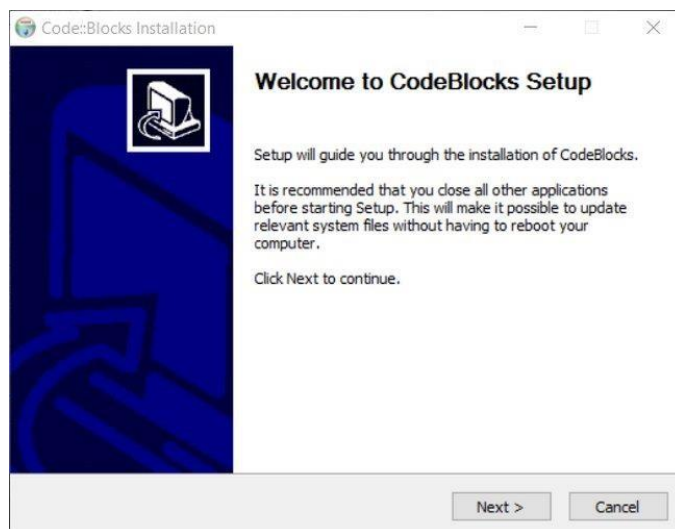
This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer and Code::Blocks will be installed, ready for you to work with it. Can't get any easier than that!

- c. Pilih file “codeblocks-20 03mingw-setup.exe dan klik FossHUBB

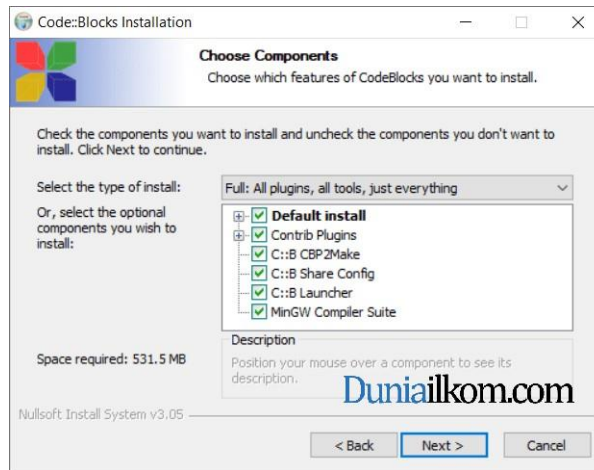


File	Date	Download from
codeblocks-20 03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20 03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20 03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
<u>codeblocks-20 03mingw-setup.exe</u>	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20 03mingw-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20 03-32bit-setup.exe	02 Apr 2020	FossHUB or Sourceforge.net

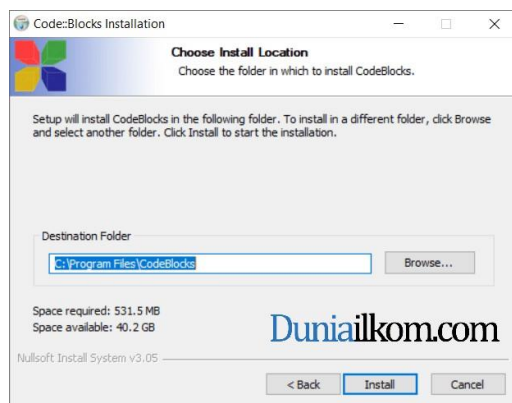
- d. Setelah instalasi selesai, klik file hasil instalasi untuk melanjutkan. Akan muncul tampilan seperti ini, lalu klik tombol "Next" untuk melanjutkan. Pada halaman "License Agreement", klik tombol "I Agree".



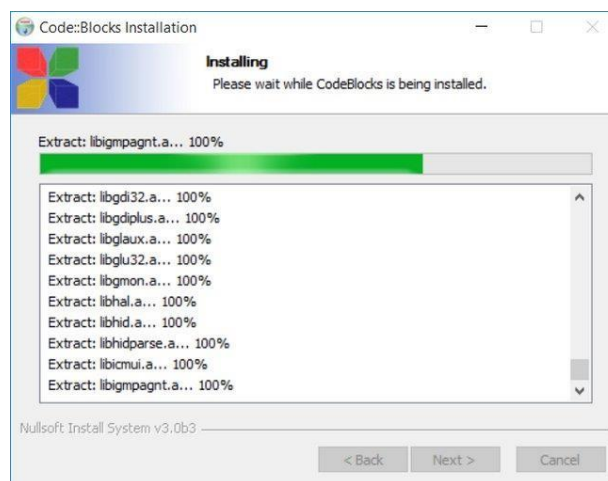
- e. Pada halaman berikutnya yang bertuliskan “Pilih Komponen”, biarkan pilihan yang telah disediakan secara default dan kemudian klik “selanjutnya”



- f. Pilih folder atau tempat untuk menyimpan aplikasi tersebut, apabila sudah kemudian install



- g. Tunggu beberapa saat sampai aplikasi terinstall



Setelah instalasi selesai, akan muncul jendela konfirmasi dengan pesan "Do you want to run Code::Blocks now?" Pilih Yes agar IDE Code::Blocks langsung terbuka setelah proses instalasi.

Kemudian Percobaan di codeblock mencakup beberapa operator

a. Operator Float

Operator float di C++ merujuk pada tipe data yang digunakan untuk menyimpan bilangan desimal, yaitu angka yang memiliki bagian pecahan. Tipe data ini memungkinkan program untuk melakukan operasi aritmatika dengan angka non-bulat, seperti 3.14 atau 2.5. Operator yang umum digunakan dengan tipe float mencakup operator aritmatika seperti +, -, *, dan /, yang memungkinkan pengguna melakukan perhitungan matematika dengan angka desimal.

```
int main()
{
    int angka = 10;
    float desimal = 10.5;
    string kalimat = "aldi"
    double tinggi = 10.4;
    char jenis_kelamin = 'M';
    bool isSunny = true;

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

b. Operator Perbandingan

Dalam C++, operator perbandingan berfungsi untuk membandingkan dua nilai atau ekspresi. Hasil dari operasi ini berupa nilai boolean, yang dapat bernilai true (benar) atau false (salah). Operator ini umumnya digunakan dalam struktur kontrol seperti pernyataan if, loop while, dan lainnya, untuk menentukan alur eksekusi program berdasarkan kondisi tertentu.

```
//operator perbandingan

int angka1 = 0;
int angka2 = 1;

bool hasil = (angka1 != angka2);

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

c. Operator Logika

Dalam C++, operator logika berfungsi untuk menggabungkan atau memodifikasi ekspresi boolean yang menghasilkan nilai true atau false. Operator ini sering diterapkan dalam struktur kontrol seperti pernyataan if, loop while, dan lain-lain, untuk mengambil keputusan berdasarkan beberapa kondisi yang ada.

```
//operator logika

bool kondisi1 = true;
bool kondisi2 = true;

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

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

d. Operator Percabangan

Dalam C++, percabangan berfungsi untuk mengatur alur eksekusi program, memungkinkan penanganan berbagai situasi yang mungkin muncul saat program dijalankan. Terdapat berbagai jenis pernyataan percabangan yang dapat digunakan, seperti if statement, if-else statement, else-if ladder, nested if-else, switch-case statement, serta operator ternary (?:).

```

/*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;
}

```

e. Operator Perulangan

Pada C++, erdapat berbagai jenis perulangan yang dapat dipilih sesuai dengan kebutuhan program Anda. Dengan memahami berbagai tipe perulangan dan cara penerapannya, Anda dapat menulis program yang lebih efisien dan adaptif. Beberapa jenis perulangan dalam C++ antara lain for loop, while loop, dan do-while loop.

```

//perulangan

```

```

int i;
for(i=0; i<5, i){
cout << "hello world" << endl;
}

```

```

for(int i =0; i<5; i++){
cout << "hello world" << 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.

```
int main() {
    float bilangan1, bilangan2;

    // Input dua bilangan float
    cout << "Masukkan bilangan pertama: ";
    cin >> bilangan1;
    cout << "Masukkan bilangan kedua: ";
    cin >> bilangan2;

    // Menghitung hasil operasi
    float penjumlahan = bilangan1 + bilangan2;
    float pengurangan = bilangan1 - bilangan2;
    float perkalian = bilangan1 * bilangan2;
    float pembagian;

    // Memastikan pembagian tidak oleh nol
    if (bilangan2 != 0) {
        pembagian = bilangan1 / bilangan2;
        cout << "Hasil Pembagian: " << pembagian << endl;
    } else {
        cout << "Pembagian tidak dapat dilakukan (pembagi nol)." << endl;
    }

    // Output hasil operasi
    cout << "Hasil Penjumlahan: " << penjumlahan << endl;
    cout << "Hasil Pengurangan: " << pengurangan << endl;
    cout << "Hasil Perkalian: " << perkalian << endl;

    return 0;
}
```

Outputnya akan berupa

```
Masukan Bilangan pertama: 10
Masukan bilangan kedua: 2
Hasil:
Penjumlahan: 12
Pengurangan: 8
Perkalian: 20
Pembagian: 5
```

- 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

```
string angkaKeTeks(int angka) {
    switch (angka) {
        case 0: return "nol";
        case 1: return "satu";
        case 2: return "dua";
        case 3: return "tiga";
        case 4: return "empat";
        case 5: return "lima";
        case 6: return "enam";
        case 7: return "tujuh";
        case 8: return "delapan";
        case 9: return "sembilan";
        case 10: return "sepuluh";
        case 11: return "sebelas";
        case 12: return "dua belas";
        case 13: return "tiga belas";
        case 14: return "empat belas";
        case 15: return "lima belas";
        case 16: return "enam belas";
        case 17: return "tujuh belas";
        case 18: return "delapan belas";
        case 19: return "sembilan belas";
        case 20: return "dua puluh";
        case 30: return "tiga puluh";
        case 40: return "empat puluh";
        case 50: return "lima puluh";
        case 60: return "enam puluh";
        case 70: return "tujuh puluh";
        case 80: return "delapan puluh";
        case 90: return "sembilan puluh";
        case 100: return "seratus";
        default:
            if (angka < 100) {
                return angkaKeTeks(angka - (angka % 10)) + " " + angkaKeTeks(angka % 10);
            }
            return "";
    }
}

int main() {
    int angka;

    // Input angka dari pengguna
    cout << "Masukkan angka (0 - 100): ";
    cin >> angka;

    // Validasi input
    if (angka < 0 || angka > 100) {
        cout << "Angka harus dalam rentang 0 hingga 100." << endl;
    } else {
        // Mengubah angka ke dalam bentuk tulisan dan menampilkannya
        cout << "Angka " << angka << " dalam bentuk tulisan adalah: " << angkaKeTeks(angka) << endl;
    }
}
```

Outputnya akan berupa

```
Masukkan angka (0-100): 77
77 : Tujuh Puluh Tujuh
```


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

```
int main() {
    int n;

    // Meminta input dari pengguna
    cout << "Masukkan angka: ";
    cin >> n;

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

        // Mencetak tanda bintang di tengah
        cout << "* ";

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

        cout << endl; // Pindah ke baris berikutnya
    }

    // Mencetak satu bintang di bagian paling bawah
    cout << "* " << endl;

    return 0;
}
```

Outputnya akan berupa

```
Masukkan angka: 6
6 5 4 3 2 1 * 1 2 3 4 5 6
5 4 3 2 1 * 1 2 3 4 5
4 3 2 1 * 1 2 3 4
3 2 1 * 1 2 3
2 1 * 1 2
1 * 1
*
```

