

Aturan Praktikum Struktur Data

1. **Akun GitHub:** Setiap praktikan wajib memiliki akun GitHub yang aktif dan digunakan selama praktikum berlangsung.
2. **Invite Collaborator:** Setiap praktikan diwajibkan untuk menambahkan collaborator di setiap repository
 - a. Asisten Praktikum: AndiniNH
 - b. Asisten Praktikum: 4ldiputra
3. **Repository Praktikum:** Setiap praktikan diwajibkan untuk membuat satu repository di GitHub yang akan digunakan untuk seluruh tugas dan laporan praktikum. Repository ini harus diatur dengan rapi dan sesuai dengan instruksi yang akan diberikan di lampiran.
4. **Penamaan Folder:** Penamaan folder dalam repository akan dibahas secara rinci di lampiran. Praktikan wajib mengikuti aturan penamaan yang telah ditentukan.

Nomor	Pertemuan	Penamaan
1	Pengalaman Bahasa C++ Bagian Pertama	01_Pengenalan_CPP_Bagian_1
2	Pengenalan Bahasa C++ Bagian Kedua	02_Pengenalan_CPP_Bagian_2
3	Abstract Data Type	03_Abstract_Data_Type
4	Single Linked List Bagian Pertama	04_Single_Linked_List_Bagian_1
5	Single Linked List Bagian Kedua	05_Single_Linked_List_Bagian_2
6	Double Linked List Bagian Pertama	06_Double_Linked_List_Bagian_1
7	Stack	07_Stack
8	Queue	08_Queue
9	Assessment Bagian Pertama	09_Assessment_Bagian_1
10	Tree Bagian Pertama	10_Tree_Bagian_1
11	Tree Bagian Kedua	11_Tree_Bagian_2
12	Asistensi Tugas Besar	12_Asistensi_Tugas_Besar
13	Multi Linked List	13_Multi_Linked_List
14	Graph	14_Graph
15	Assessment Bagian Kedua	15_Assessment_Bagian_2
16	Tugas Besar	16_Tugas_Besar

5. Jam Praktikum:

- Jam masuk praktikum adalah **1 jam lebih lambat** dari jadwal yang tercantum. Sebagai contoh, jika jadwal praktikum adalah pukul 06.30 - 09.30, maka aturan praktikum akan diatur sebagai berikut:
 - **06.30 - 07.30:** Waktu ini digunakan untuk **Tugas Praktikum dan Laporan Praktikum** yang dilakukan di luar laboratorium.
 - **07.30 - 08.30:** Sesi ini mencakup **tutorial, diskusi, dan kasus problem-solving**. Kegiatan ini berlangsung di dalam laboratorium dengan alokasi waktu sebagai berikut:
 - **60 menit pertama:** Tugas terbimbing.
 - **60 menit kedua:** Tugas mandiri.

6. **Pengumpulan Tugasn Pendahuluan:** Tugas Pendahuluan (TP) wajib dikumpulkan melalui GitHub sesuai dengan format berikut:

nama_repo/nama_pertemuan/TP_Pertemuan_Ke.md

Sebagai contoh:

STD_Yudha_Islalmi_Sulistya_XXXXXXXX/01_Running_Modul/TP_01.md

7. **Pengecekan Tugas Pendahuluan:** Pengumpulan laporan praktikum akan diperiksa **1 hari sebelum praktikum selanjutnya** dimulai. Pastikan tugas telah diunggah tepat waktu untuk menghindari sanksi.

8. Struktur Laporan Praktikum

1. Cover :

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



Disusun Oleh:
Aulia Jasifa Br Ginting -2311104060
Kelas

Dosen :
WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

2. 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 koondisional dalam program.

3. Landasan Teori

Apa itu Code Blocks? Code Blocks adalah sebuah Integrated Development Environment (IDE) yang sangat populer dan gratis. IDE ini dirancang untuk membantu para programmer, terutama dalam mengembangkan aplikasi menggunakan bahasa pemrograman C dan C++, dan Fortan.

4. Guided

Langkah Langkah Instalasi Code Block

1. Download terlebih dahulu file exe nya pada link ini Downloads - Code::Blocks (codeblocks.org)
2. Lanjut klik “Download the binary release

Downloads

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

• Download the binary release

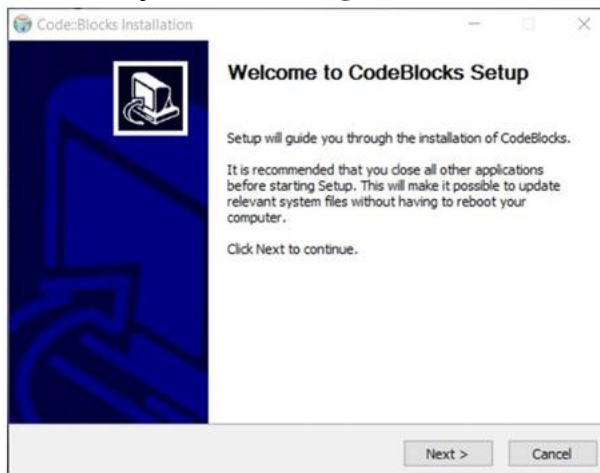
This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer and Code::Blocks will be installed, ready for you to work with it. Can't get any easier than that!

3. Pilih file “codeblocks-20.03mingw-setup.exe dan klik FossHUBB

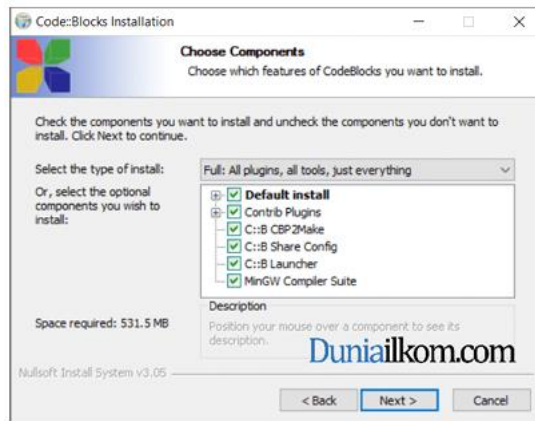


File	Date	Download from
codeblocks-20.03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
<u>codeblocks-20.03mingw-setup.exe</u>	29 Mar 2020	FossHUB or Sourceforge.net
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

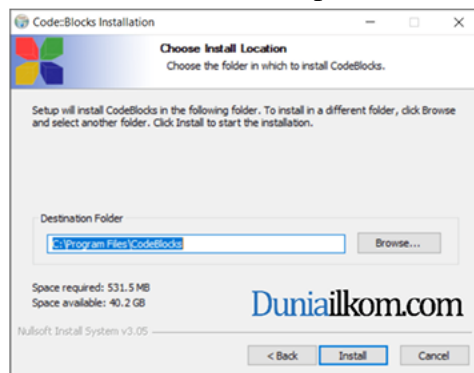
4. Jika sudah terinstal langsung klik hasilnya maka akan keluar tampilan seperti ini, lanjut Klik tombol “Next” untuk lanjut. Lalu di halaman “License Agreement” Klik saja tombol “I Agree”.



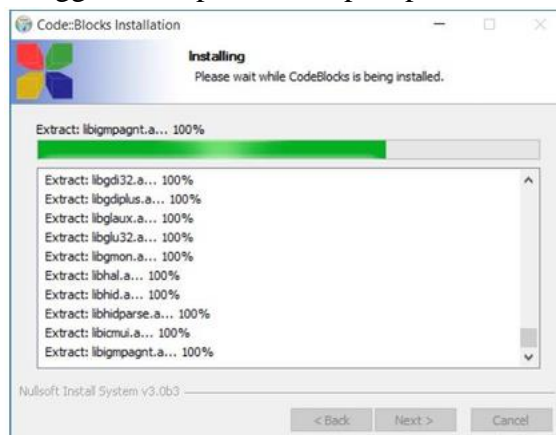
5. Halaman selanjutnya adalah “**Chosee Components**” biarkan pilihan default lalu klik “**next**”



6. Pilih folder atau tempat untuk menyimpan aplikasi nya, jika sudah tinggal instal



Tunggu beberapa saat sampai aplikasi terinstal



Setelah proses instalasi selesai, akan tampil jendela konfirmasi “**Do you want to run Code::Blocks now?**” Klik **Yes** agar setelah proses instalasi, IDE Code::Blocks langsung tampil.

1. Fungsi Float

Di dalam bahasa pemrograman, float dipakai guna menyimpan bilangan desimal maupun bilangan pecahan. Tipe data float juga memberikan akurasi yang dibutuhkan

dalam perhitungan yang memerlukan bilangan desimal yang tak dapat dicapai dengan data tipe bilangan bulat.

```
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;
```

2. 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;
```

3. 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;*/
```

4. 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;
}
```

5. 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;
}
```

5. 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

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

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

    // Menhitung 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 adalah

```
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
*
```

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

```
string angkaTeks(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 21: return "tiga puluh";
        case 22: return "empat puluh";
        case 23: return "lima puluh";
        case 24: return "enam puluh";
        case 25: return "tujuh puluh";
        case 26: return "delapan puluh";
        case 27: return "sembilan puluh";
        case 28: return "sepuluh";
        case 29: return "sebelas";
        case 30: return "dua belas";
        case 31: return "tiga belas";
        case 32: return "empat belas";
        case 33: return "lima belas";
        case 34: return "enam belas";
        case 35: return "tujuh belas";
        case 36: return "delapan belas";
        case 37: return "sembilan belas";
        case 38: return "sepuluh";
        case 39: return "sebelas";
        case 40: return "dua belas";
        case 41: return "tiga belas";
        case 42: return "empat belas";
        case 43: return "lima belas";
        case 44: return "enam belas";
        case 45: return "tujuh belas";
        case 46: return "delapan belas";
        case 47: return "sembilan belas";
        case 48: return "sepuluh";
        case 49: return "sebelas";
        case 50: return "dua belas";
        case 51: return "tiga belas";
        case 52: return "empat belas";
        case 53: return "lima belas";
        case 54: return "enam belas";
        case 55: return "tujuh belas";
        case 56: return "delapan belas";
        case 57: return "sembilan belas";
        case 58: return "sepuluh";
        case 59: return "sebelas";
        case 60: return "dua belas";
        case 61: return "tiga belas";
        case 62: return "empat belas";
        case 63: return "lima belas";
        case 64: return "enam belas";
        case 65: return "tujuh belas";
        case 66: return "delapan belas";
        case 67: return "sembilan belas";
        case 68: return "sepuluh";
        case 69: return "sebelas";
        case 70: return "dua belas";
        case 71: return "tiga belas";
        case 72: return "empat belas";
        case 73: return "lima belas";
        case 74: return "enam belas";
        case 75: return "tujuh belas";
        case 76: return "delapan belas";
        case 77: return "sembilan belas";
        case 78: return "sepuluh";
        case 79: return "sebelas";
        case 80: return "dua belas";
        case 81: return "tiga belas";
        case 82: return "empat belas";
        case 83: return "lima belas";
        case 84: return "enam belas";
        case 85: return "tujuh belas";
        case 86: return "delapan belas";
        case 87: return "sembilan belas";
        case 88: return "sepuluh";
        case 89: return "sebelas";
        case 90: return "dua belas";
        case 91: return "tiga belas";
        case 92: return "empat belas";
        case 93: return "lima belas";
        case 94: return "enam belas";
        case 95: return "tujuh belas";
        case 96: return "delapan belas";
        case 97: return "sembilan belas";
        case 98: return "sepuluh";
        case 99: return "sebelas";
        case 100: return "seratus";
        default:
            if (angka < 100) {
                return angkaTeks(angka - (angka % 10)) + " " + angkaTeks(angka % 10);
            }
            return "";
    }
}

int main() {
    int angka;

    // Tryen 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 {
        // Menampilkan angka ke dalam bentuk tulisan dan memisalkan
        cout << "Angka " << angka << " dalam bentuk tulisan adalah: " << angkaTeks(angka) << endl;
    }
}
```

Outputnya adalah

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


3. 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 adalah

```
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
*
```

6.