

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	Pengertian 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 - 09.30:** 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 Tugas 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:

Izzaty Zahara Br Barus - 2311104052

Kelas: SE 07-02

Dosen :

WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

2. Tujuan

1. Memahami proses debugging dan penggunaan breakpoints dalam CodeBlocks.
2. Mengimplementasikan loop (perulangan) dalam program C++.
3. Mengimplementasikan fungsi dan memecah program menjadi modul yang lebih kecil.
4. Memahami penggunaan pointer dalam C++.
5. Membuat dan menggunakan struktur data seperti struct dalam program C++.

3. Landasan Teori

Code::Blocks adalah sebuah Integrated Development Environment (IDE), yaitu program yang digunakan untuk menulis, mengedit, mengompilasi, dan men-debug kode pemrograman, khususnya untuk bahasa seperti C, C++, dan Fortran. Code::Blocks menyediakan antarmuka yang ramah pengguna dengan berbagai fitur seperti syntax highlighting (penyorotan sintaks), debugging, dan alat bantu untuk mempermudah pembuatan program.

Pada praktikum Struktur Data ini, kakas (*tool*) yang digunakan adalah Code Blocks. Kakas ini merupakan *free, open-source*, dan *cross-platform IDE*. Saat ini, Code Blocks berorientasi pada C/C++/Fortran (codeblocks, 2016).

4. Guided

Langkah Langkah instalasi code block

1. Download terlebih dahulu file exe nya pada link berikut ini [Downloads - Code::Blocks \(codeblocks.org\)](https://codeblocks.org/downloads)
2. Pilih opsi Download the binary release untuk mengunduh versi siap pakai (bukan dari source code).

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!

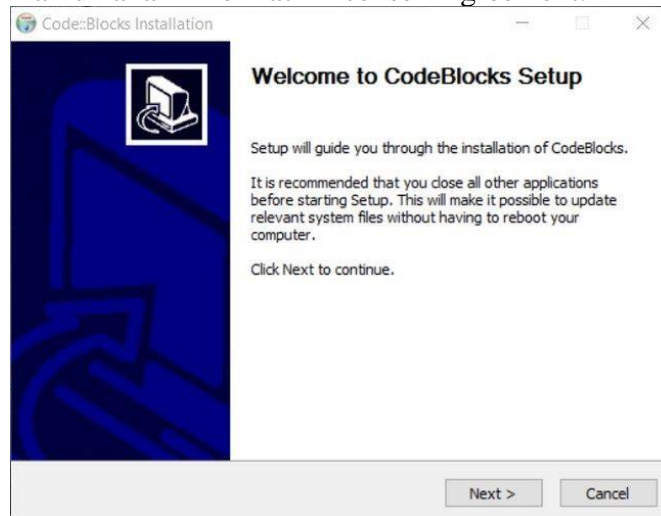
3. Setelah unduhan selesai, cari file .exe (misalnya codeblocks-XX.x mingw-setup.exe) di folder unduhan kamu. Klik dua kali untuk menjalankan installer



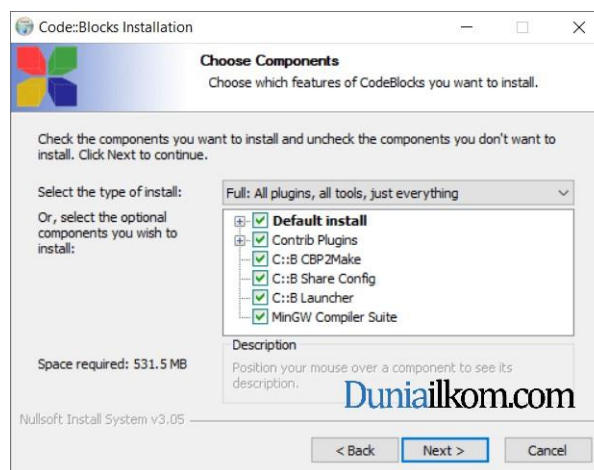
Windows XP / Vista / 7 / 8.x / 10: Duniailkom.com

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

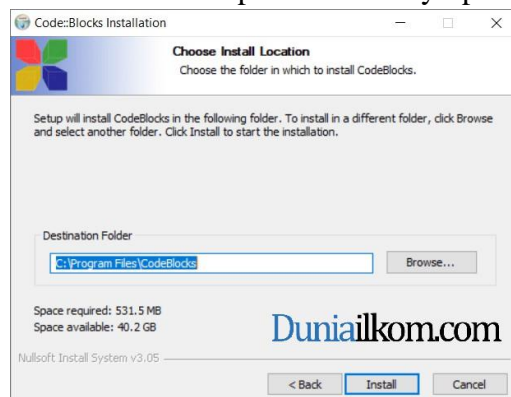
Pada jendela instalasi yang muncul, klik Next untuk memulai proses instalasi. Kamu akan melihat License Agreement. Klik I Agree untuk melanjutkan.



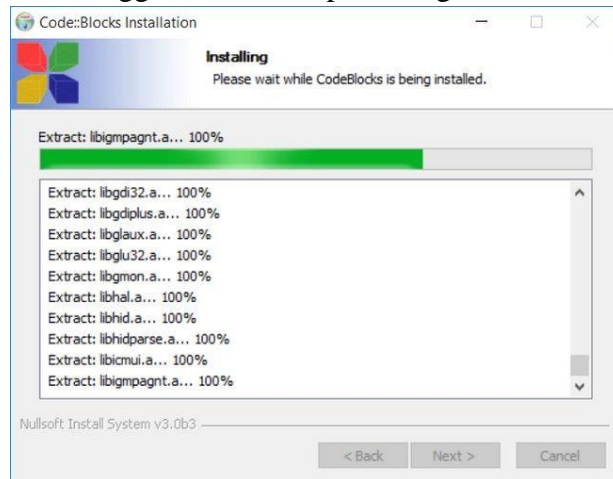
4. Halaman selanjutnya adalah “Choose Components” biarkan pilihan default lalu klik “next”



5. Pilih folder atau tempat untuk menyimpan apk nya, jika sudah tinggal install



Lalu tunggu untuk beberapa saat agar 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

- Menyimpan angka yang membutuhkan presisi desimal.

- Digunakan untuk operasi matematika yang melibatkan pecahan atau angka desimal.

Namun, tipe float memiliki keterbatasan dalam hal presisi, biasanya hingga sekitar 6-7 digit desimal. Untuk angka yang lebih presisi, tipe data double sering digunakan.

CONTOH:



```
1 #include <iostream>
2 #include <conio.h>
3
4 using namespace std;
5
6 int main()
7 {
8     // Fungsi Floata
9     int angka = 10;
10    float desimal = 3.50;
11    string kalimat = "izzaty";
12    double tinggi = 10.4;
13    char jenis = 'L';
14    bool isSunny = true;
15
16    cout << "Angka:" << angka << endl;
17    cout << "Tinggi:" << tinggi << endl;
18
19 }
20
```

Hasil Outputannya:

```
11 // double tinggi = 10.4;

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\slame> & 'c:\Users\slame\.vscode\extensions\ms-vscode.cp
ptools-1.21.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '
--stdin=Microsoft-MIEngine-In-03bcmd3j.o14' '--stdout=Microsoft-MIEng
ine-Out-qw2q1sbr.h35' '--stderr=Microsoft-MIEngine-Error-i01mfssr.uim
' '--pid=Microsoft-MIEngine-Pid-heq2tqwc.4m4' '--dbgExe=C:\msys64\ucr
t64\bin\gdb.exe' '--interpreter=mi'
Angka:10
Tinggi:10.4
PS C:\Users\slame> 
```

2. Fungsi cin() dan fungsi Getch()

cin adalah singkatan dari "Character Input" dalam C++ dan digunakan untuk menerima input dari pengguna melalui keyboard. Biasanya digunakan bersama dengan operator >> untuk membaca berbagai tipe data (seperti int, float, char, dll.).

CONTOH:

```
1 #include <iostream>
2 #include <conio.h>
3
4 using namespace std;
5
6 int main()
7 {
8     int angka;
9
10    cout << "Masukkan Angka:";
11    cin >> angka;
12
13    cout << "Angka : " << angka << endl;
14
15    getch();
16
17
18 }
```

Outputannya:

```
Masukkan Angka:10
Angka : 10
PS C:\Users\slame> 
```

3. Operator perbandingan

Operator perbandingan adalah operator yang digunakan untuk membandingkan dua nilai dalam pemrograman dan menghasilkan hasil berupa nilai true (benar) atau false (salah), tergantung dari apakah kondisi perbandingan terpenuhi.

CONTOH:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      int bil1 = 2, bil2 = 3, hasil;
8      hasil = bil1 > bil2;
9      cout << hasil << endl;
10     hasil = bil1 >= bil2;
11     cout << hasil << endl;
12     hasil = bil1 < bil2;
13     cout << hasil << endl;
14     hasil = bil1 <= bil2;
15     cout << hasil << endl;
16     hasil = bil1 == bil2;
17     cout << hasil << endl;
18     hasil = bil1 != bil2;
19     cout << hasil << endl;
20     return 0;
21 }
22
```


Hasil outputannya:

```
Hasilnya adalah :1  
PS C:\Users\slame>
```

4. Kegunaan Fungsi Switch dan Case

Fungsi switch dan case digunakan dalam C++ untuk membuat keputusan berdasarkan nilai dari sebuah variabel. Ini berguna untuk menggantikan banyak pernyataan if-else yang berturut-turut, sehingga kode lebih rapi dan mudah dibaca.

Contoh:



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int nilai;
7     cout << "Masukkan nilai: ";
8     while (!(cin >> nilai)) {
9         cout << "Input salah. Silakan masukkan nilai lagi: ";
10        cin.clear();
11    }
12    if (nilai > 80) {
13        cout << "A" << endl;
14    } else {
15        cout << "Bukan A" << endl;
16    }
17    return 0;
18 }
```

Hasil outputannya:

```
Daftar channel tv
1. RCTI
2. Indosiar
Masukkan Channel Pilihan: 1
Channel yg anda pilih rcti
PS C:\Users\slame>
```

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

Jawaban:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      float bil1, bil2;
8
9      cout << "Masukkan angka pertama : ";
10     cin >> bil1;
11     cout << "Masukkan angka kedua : ";
12     cin >> bil2;
13
14     float hasiltambah = bil1 + bil2;
15     cout << "Hasil Penjumlahan : " << hasiltambah << endl;
16
17     float hasilkurang = bil1 - bil2;
18     cout << "Hasil Pengurangan : " << hasilkurang << endl;
19
20     float hasilkali = bil1 * bil2;
21     cout << "Hasil Perkalian : " << hasilkali << endl;
22
23     if (bil2 != 0) {
24         float hasilbagi = bil1 / bil2;
25         cout << "Hasil Pembagian : " << hasilbagi << endl;
26     } else {
27         cout << "Tidak Bisa Dibagi 0!!" << endl;
28     }
29
30     return 0;
31 }
32
```

Outputannya:

```
Hasilnya adalah: 20
Hasilnya adalah: 0
Hasilnya adalah: 100
Hasilnya adalah: 1
PS C:\Users\slame>
```

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

Jawaban:

```

1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 string terbilang(int angka) {
7     string satuan[10] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
8     string belasan[10] = {"sepuluh", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
9     string puluhan[10] = {"", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
10
11     if (angka < 0 || angka > 100) {
12         return "Angka di luar rentang!";
13     } else if (angka == 0) {
14         return "nol";
15     } else if (angka < 10) {
16         return satuan[angka];
17     } else if (angka < 20) {
18         return belasan[angka - 10];
19     } else {
20         return puluhan[angka / 10] + ((angka % 10 != 0) ? " " + satuan[angka % 10] : "");
21     }
22 }
23
24 int main() {
25     int angka;
26
27     cout << "Masukkan angka (0-100): ";
28     cin >> angka;
29
30     cout << angka << " : " << terbilang(angka) << endl;
31
32     return 0;
33 }

```

Outputannya:

```

Masukkan angka (0 - 100): 34
34 : tiga puluh empat
PS C:\Users\slame>

```

1. Buatlah program yang dapat memberikan *input* dan *output* sbb

```

input: 3
output:
  3 2 1 * 1 2 3
    2 1 * 1 2
      1 * 1
        *

```

Jawaban:

```

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

```

Outputannya:

```

Masukan bilangan: 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

PS C:\Users\slame>

```

Outputannya:

```
Masukkan angka (0 - 100): 34
34 : tiga puluh empat
PS C:\Users\slame>
```

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
        *
```

Jawaban:

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

Outputannya:

```
Masukan bilangan: 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

PS C:\Users\slame>
```

6. Kesimpulan

Kesimpulan dari laporan yang telah dibuat adalah bahwa setelah mempelajari dasar-dasar bahasa pemrograman C++, penulis mendapatkan wawasan lebih mengenai penggunaan fungsi-fungsi dalam bahasa tersebut. Terutama, penulis mampu memahami dan mengimplementasikan konsep seperti debugging, perulangan (loop), modularisasi dengan fungsi, penggunaan pointer, dan struktur data seperti struct dalam program C++. Laporan ini juga menunjukkan bagaimana penggunaan Code::Blocks IDE dapat mempermudah proses penulisan, kompilasi, dan debugging program.

