

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 - 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 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 ...

“Judul”



Disusun Oleh:

Reza Afiansyah Wibowo -2311104062

SE0702

Dosen :

WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

I. 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 kondisional dalam program.

II.Landasan Teori

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).

III.GUIDED

```
#include <iostream>
```

```
using namespace std;
```

```
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;*/
```

```
    /*int angka;
```

```
        cout << "Masukkan angka: ";
```

```
        cin >> angka;
```

```
        cout << "Angka: " << angka << endl;*/
```

```
    /*int angka1 = 8;
```

```
    int angka2 = 3;
```

```
    int hasil = angka1 + angka2;
```

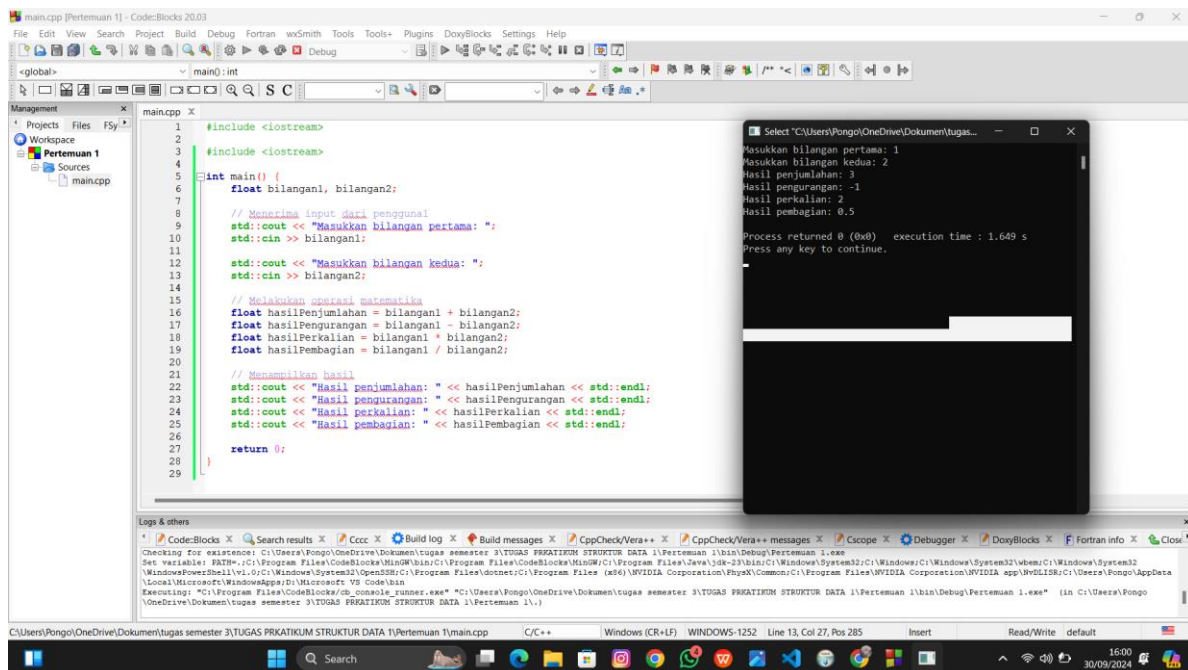
```
    cout << "Hasilnya adalah " << hasil << endl;
```

//operator perbandingan

IV. 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.

JAWAB:



```

1 #include <iostream>
2
3 #include <iostream>
4
5 int main() {
6     float bilangan1, bilangan2;
7
8     // Menerima input dari pengguna
9     std::cout << "Masukkan bilangan pertama: ";
10    std::cin >> bilangan1;
11
12    std::cout << "Masukkan bilangan kedua: ";
13    std::cin >> bilangan2;
14
15    // Melakukan operasi matematika
16    float hasilPenjumlahan = bilangan1 + bilangan2;
17    float hasilPengurangan = bilangan1 - bilangan2;
18    float hasilPerkalian = bilangan1 * bilangan2;
19    float hasilPembagian = bilangan1 / bilangan2;
20
21    // Menampilkan hasil
22    std::cout << "Hasil penjumlahan: " << hasilPenjumlahan << std::endl;
23    std::cout << "Hasil pengurangan: " << hasilPengurangan << std::endl;
24    std::cout << "Hasil perkalian: " << hasilPerkalian << std::endl;
25    std::cout << "Hasil pembagian: " << hasilPembagian << std::endl;
26
27    return 0;
28
29 }
  
```

Output Window:

```

Masukkan bilangan pertama: 1
Masukkan bilangan kedua: 2
Hasil penjumlahan: 3
Hasil pengurangan: -1
Hasil perkalian: 2
Hasil pembagian: 0.5

Process returned 0 (0x0)   execution time : 1.649 s
Press any key to continue.
  
```

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

Gambar 1-15 Contoh

JAWAB

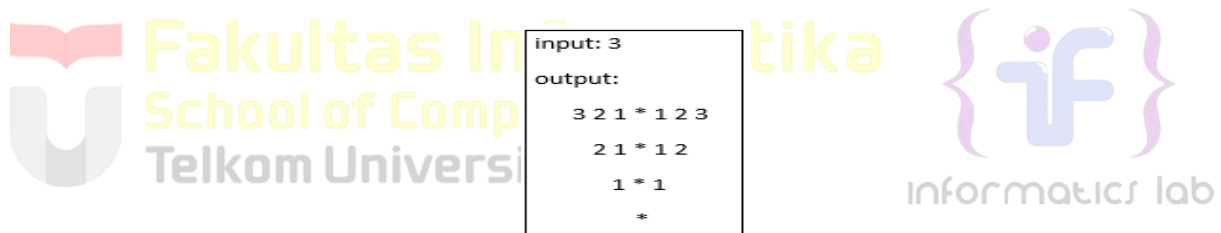
```

1 #include <iostream>
2 #include <iostream>
3 using namespace std;
4
5 // Fungsi untuk memodifikasi angka menjadi teks
6
7 string satuan[] = {"", "satu", "dua", "tiga", "empat", "lima", "enam", "tujuh", "delapan", "sembilan"};
8 string belasan[] = {"", "sebelas", "dua belas", "tiga belas", "empat belas", "lima belas", "enam belas", "tujuh belas", "delapan belas", "sembilan belas"};
9 string puluhan[] = {"", "sepuluh", "dua puluh", "tiga puluh", "empat puluh", "lima puluh", "enam puluh", "tujuh puluh", "delapan puluh", "sembilan puluh"};
10
11 if (angka < 10) {
12     return satuan[angka];
13 } else if (angka < 20) {
14     return belasan[angka - 10];
15 } else {
16     return puluhan[angka / 10] + " " + satuan[angka % 10];
17 }
18
19
20 int main() {
21     int inputAngka;
22     cout << "Masukkan angka (0-100): ";
23     cin >> inputAngka;
24
25     if (inputAngka < 0 || inputAngka > 100) {
26         cout << "Angka harus antara 0 hingga 100." << endl;
27     } else {
28         cout << "Angka " << inputAngka << " dalam bentuk teks: " << angkaKeTeks(inputAngka) << endl;
29     }
30 }

```

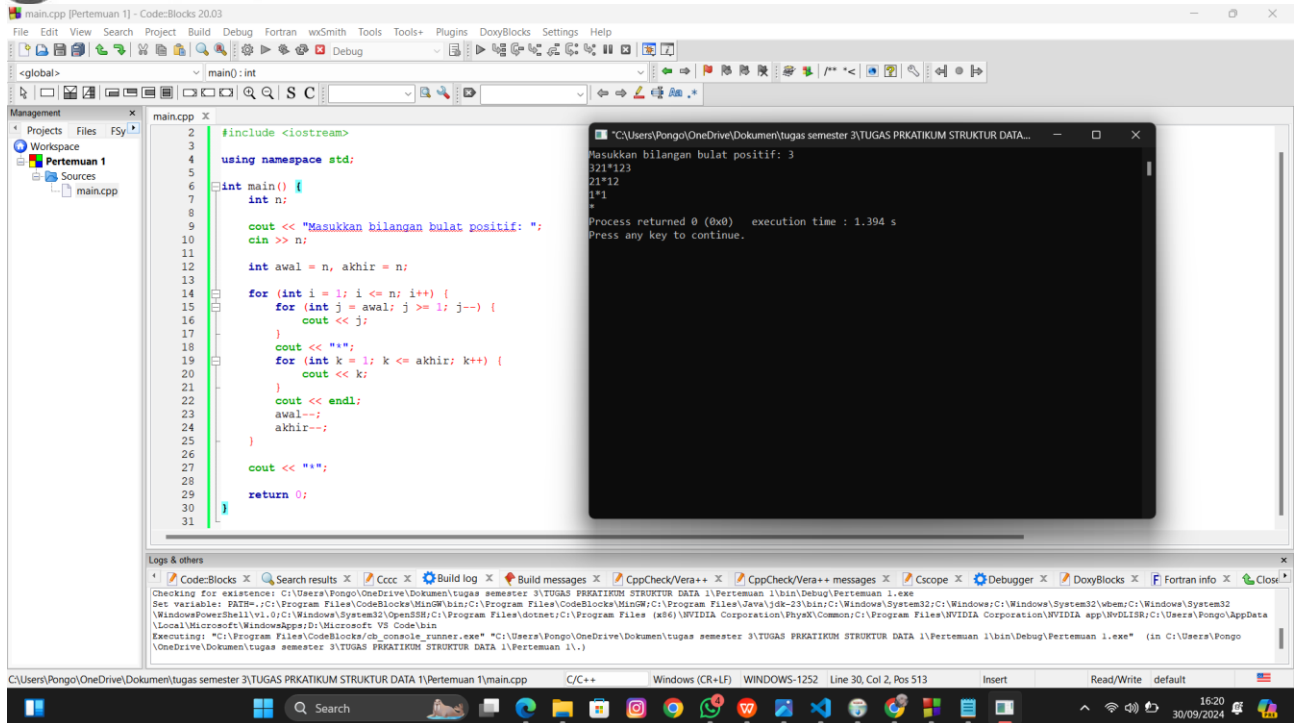
Masukkan angka (0-100): 90
Angka 90 dalam bentuk teks: sembilan puluh
Process returned 0 (0x0) execution time : 3.026 s
Press any key to continue.

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



Gambar 1-16 Mirror

JAWAB:



The screenshot shows the Code::Blocks IDE with a C++ program in `main.cpp` and its execution output in a terminal window.

main.cpp Code:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int n;
7
8      cout << "Masukkan bilangan bulat positif: ";
9      cin >> n;
10
11      int awal = n, akhir = n;
12
13      for (int i = 1; i <= n; i++) {
14          for (int j = awal; j >= 1; j--) {
15              cout << j;
16          }
17          cout << " * ";
18          for (int k = 1; k <= akhir; k++) {
19              cout << k;
20          }
21          cout << endl;
22          awal--;
23          akhir--;
24      }
25
26      cout << " * ";
27
28      return 0;
29
30 }

```

Terminal Output:

```

Masukkan bilangan bulat positif: 3
121*123
21*12
1*1
*
Process returned 0 (0x0)   execution time : 1.394 s
Press any key to continue.

```

The terminal output shows the program's execution for `n=3`, displaying a pattern of numbers and asterisks. The program returns 0 and execution time is 1.394 s.

V.KESIMPULAN

Kesimpulannya saya jadi memahami bahasa c++

Dan tahu fitur2 dalam aplikasi codeblock