

**Aturan Praktikum Struktur Data**

1. **Akun GitHub:** Setiap praktikan wajib memiliki akun GitHubyang aktif dan digunakan selama praktikum berlangsung.
2. **Invite Collaborator:** Setiap praktikan diwajibkan untuk menambahkan collaborator di setiap repository
   1. Asisten Praktikum: AndiniNH
   2. AsistenPraktikum: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 | Penganalan 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 |



1. **Jam Praktikum:**

* Jam masuk praktikum adalah **1 jam lebih lambat** dari jadwal yang tercantum. Sebagai contoh, jika jadwalpraktikumadalah 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 - 0G.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.

1. **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**

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



1. **Struktur Laporan Praktikum**
2. **Cover :**

# LAPORAN PRAKTIKUM

**Modul 1**

# CODE BLOCKS IDE & PENGENALAN BAHASA C++ (BAGIAN PERTAMA)



**Disusun Oleh:**

# Izzaty Zahara Br Barus - 2311104041

**Kelas: SE 07-02**

**Dosen : WAHYU ANDI SAPUTRA**

# PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS INFORMATIKA

**TELKOM UNIVERSITY PURWOKERTO**

**2024**



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

1. **Landasan Teori**

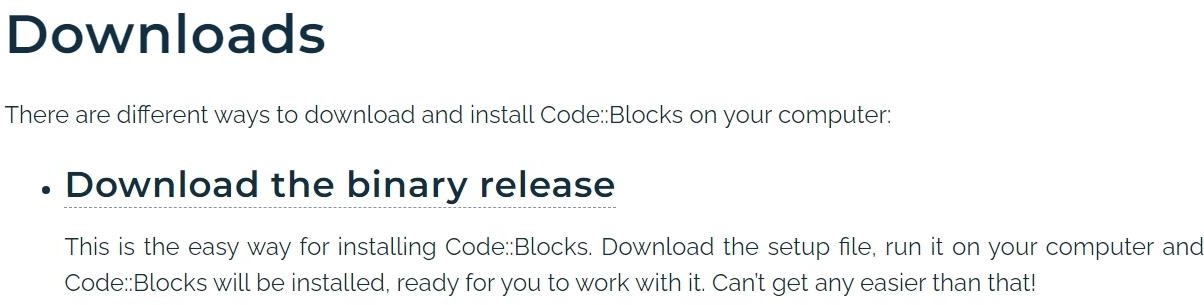
Code::Blocks adalah sebuah Integrated Development Environment (IDE), yaitu program yang digunakan untuk menulis, mengedit, mengompilasi, dan men-debug kode pemrograman, khususnya untuk bahasa seperti C, C++, dan Fortran. Code::Blocks menyediakan antarmuka yang ramah pengguna dengan berbagai fitur seperti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

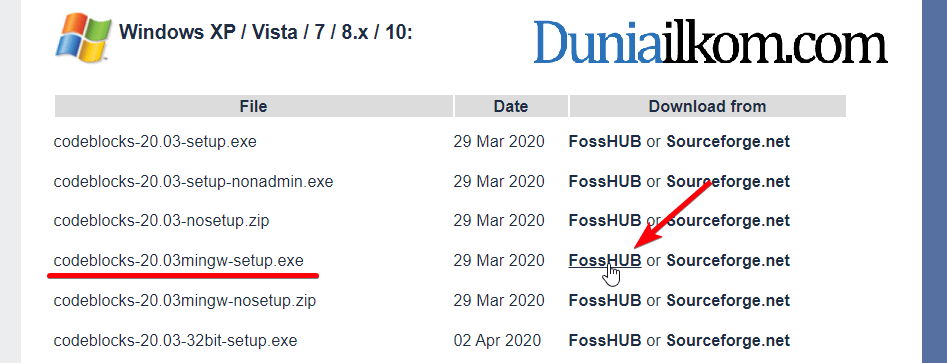
1. **Guided**

**Langkah Langkah instalasi code block**

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



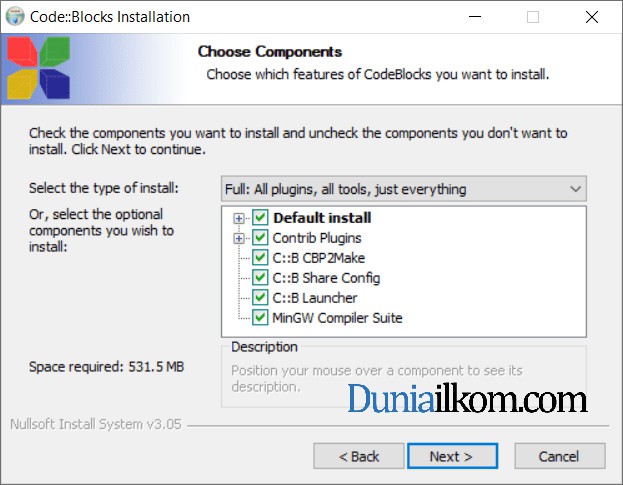
1. Setelah unduhan selesai, cari file .exe (misalnya codeblocks-XX.xxmingw-setup.exe) di folder unduhan kamu. Klik dua kali untuk menjalankan installer

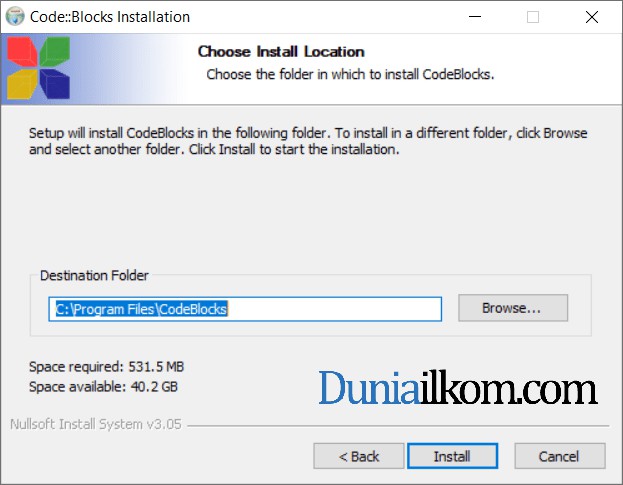




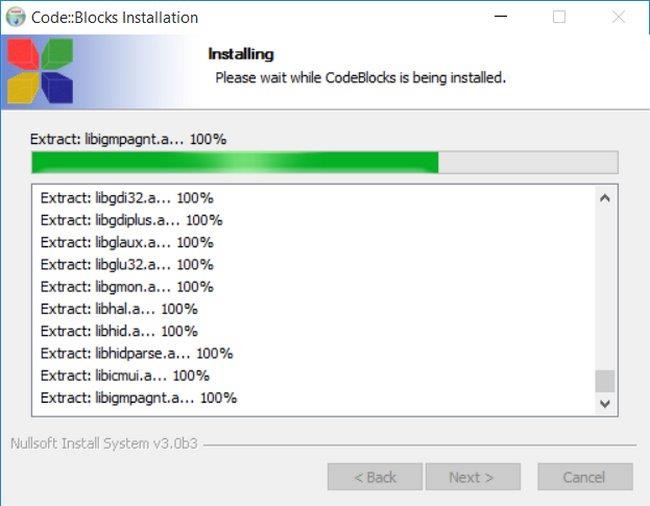
Pada jendela instalasi yang muncul, klik Next untuk memulai proses instalasi.Kamu akan melihat License Agreement. Klik I Agree untuk melanjutkan.

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



1. Pilih folder atau tempat untuk menyinpan apk nya, jika sudah tinggal install



Lalu tunggu untuk bebrapa saat agar terinstal.

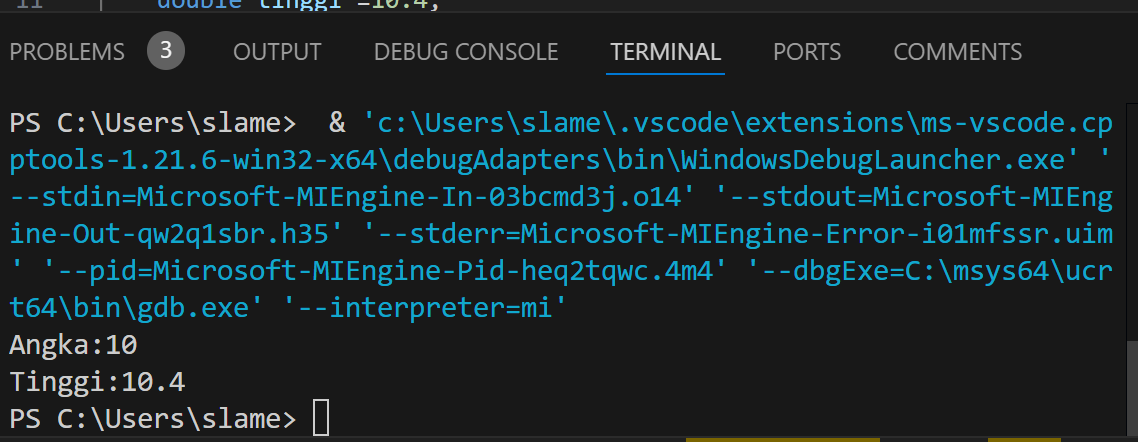
Setelah proses instalasi selesai, akan tampil jendela konfirmasi **"Do you want to run Code::Blocks now?"** Klik **Yes** agar setelah proses instalasi, IDE Code::Blocks langsung tampil.

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:

# 

Hasil Outputannya:



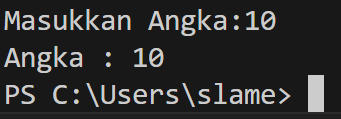
1. Fungsi cin() dan fungsi Getch()

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:

# 

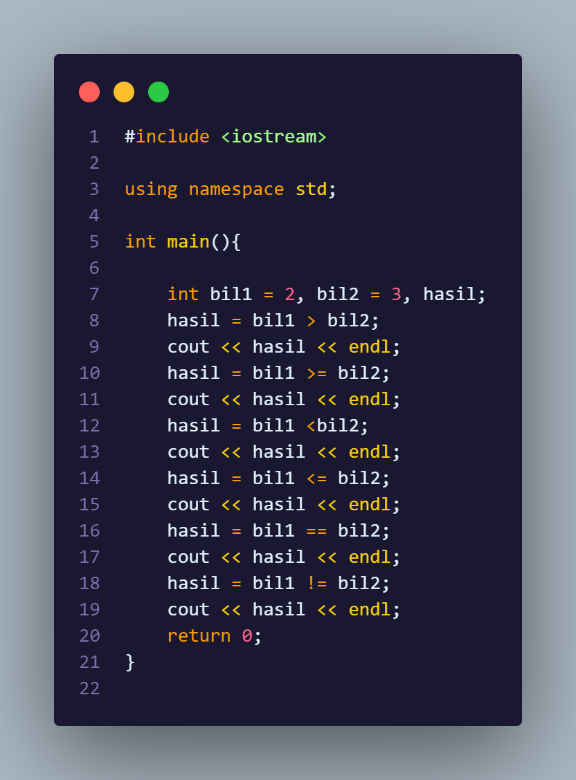
**Outputannya:**



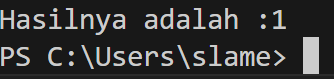
1. Operator perbandingan

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:





Hasil outputannya:

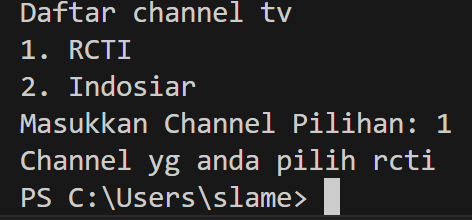
1. Kegunaan Fungsi Switch dan Case

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:**



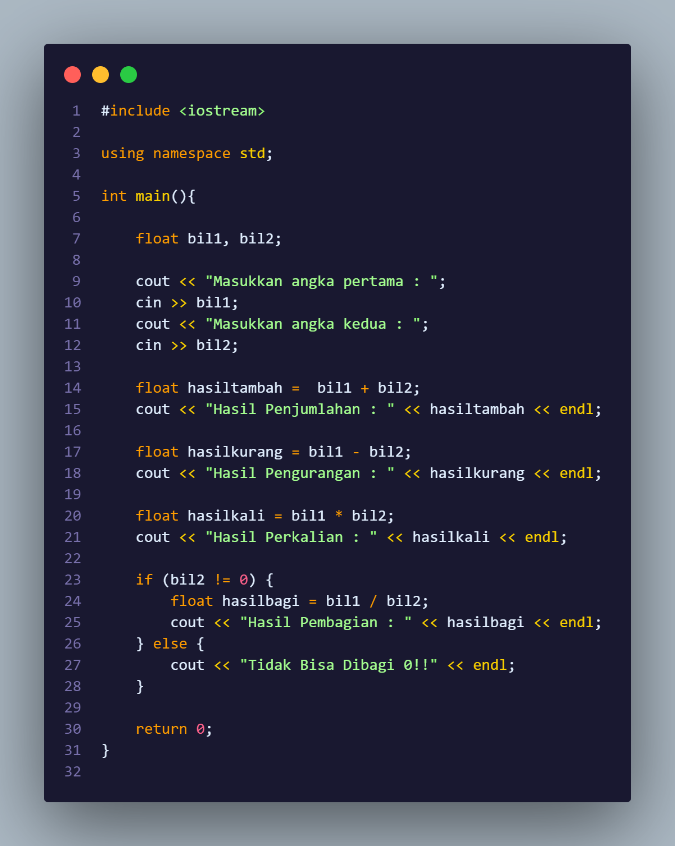
Hasil outpannya:



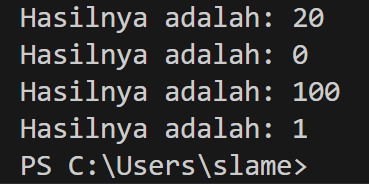


1. **Unguided**
   1. Buatlah program yang menerima *input*-an dua buah bilangan betipe float, kemudian memberikan *output*-an hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan tersebut

**Jawaban:**

****

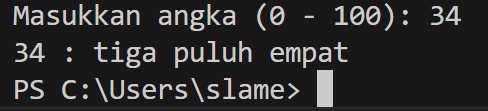
**Outputannya:**

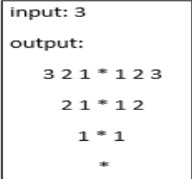


* 1. Buatlah sebuah program yang menerima masukan angka dan mengeluarkan *output* nilai angka tersebut dalam bentuk tulisan. Angka yang akan di- *input*-kan user adalah bilangan bulat positif mulai dari 0 s.d 100

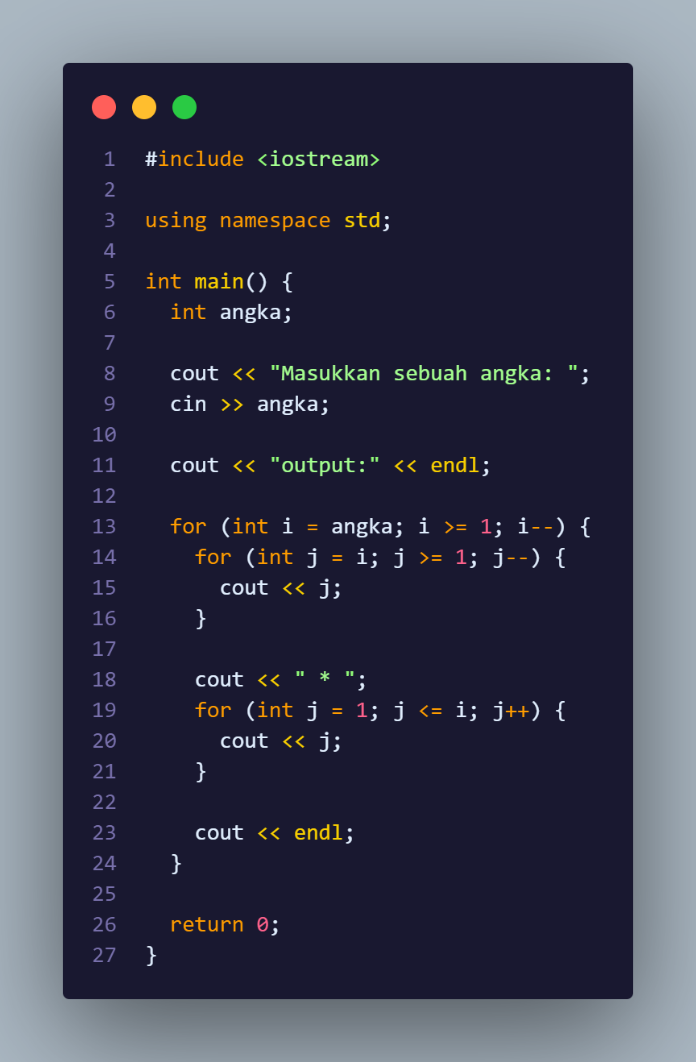
**Jawaban:**

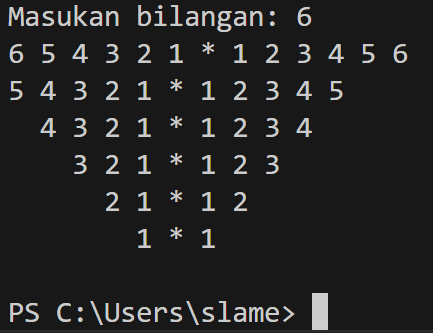


**Outputannya:**

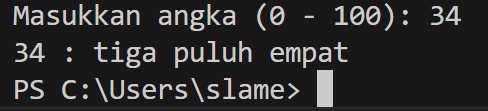
1. Buatlah program yang dapat memberikan *input* dan *output* sbb

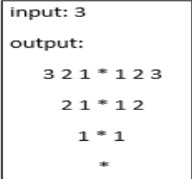
**Jawaban:**



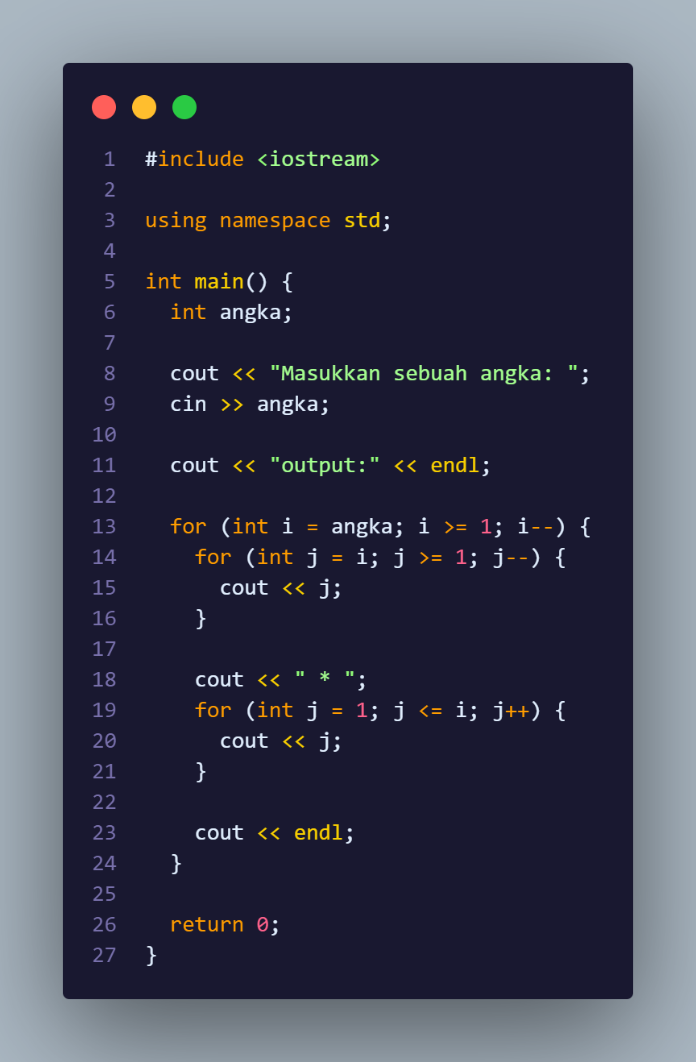
**Outputannya:**

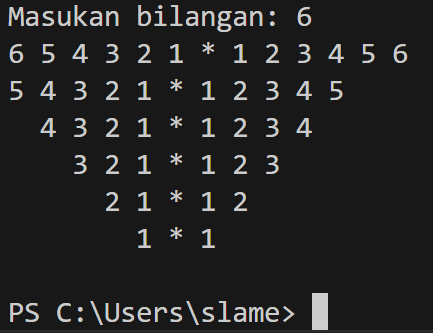


**Outputannya:**

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

**Jawaban:**



**Outputannya:**

1. **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.