

Pengenalan Bahasa C++, Algoritma
Pemrograman, Integrated Development
Equipment (IDE) Visual C++ dan Dasar
Dasar Bahasa C++

Modul **Praktikum**
C++
Dasar Pemrograman Komputer

Semester Genap 2017/2018

MODUL 1

PENGENALAN BAHASA C++ ,ALGORITMA PEMROGRAMAN,INTEGRATED DEVELOPMENT EQUIPMENT (IDE) VISUAL C++ dan DASAR-DASAR BAHASA C++**A. Tujuan**

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu :

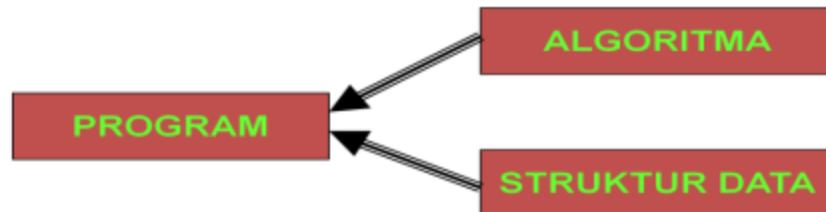
- Memahami sejarah perkembangan bahasa C++.
- Mengenal dan memahami algoritma program dengan flowchart.
- Mengenal dan menggunakan IDE C++ dengan baik.
- Mengenal dan memahami penggunaan Identifier, tipe data, variable, assignment, komentar, dan konstanta.
- Membuat program sederhana menggunakan bahasa C++.

B. Dasar Teori**Sejarah Perkembangan Bahasa C++**

C++ adalah bahasa pemrograman komputer yang di buat oleh **Bjarne Stroustrup**, yang merupakan perkembangan dari **bahasa C** dikembangkan di **Bong Labs (Dennis Ritchie)** pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu **B**, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standart Institute). Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal **1980-an**. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming).

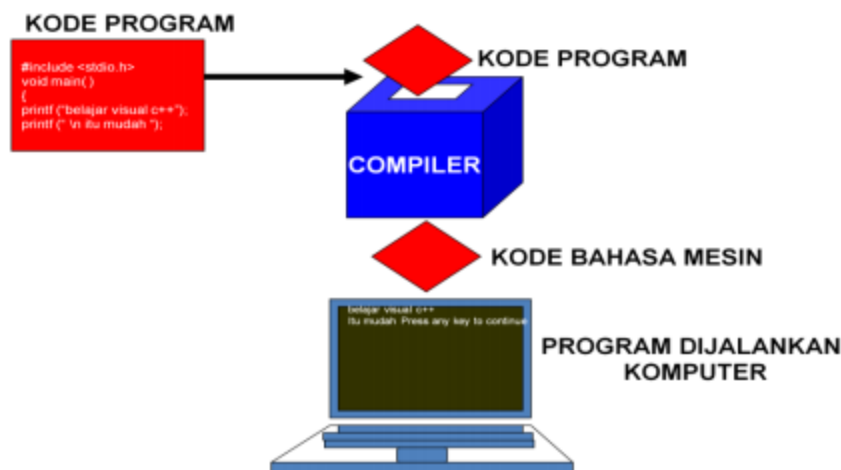
Algoritma Pemrograman

Algoritma adalah metode dan tahapan sistematis yang digunakan untuk memecahkan suatu permasalahan. Sedangkan pemrograman adalah proses pembuatan program dengan menerapkan algoritma dan struktur data tertentu menggunakan bahasa pemrograman. Struktur data adalah tempat tatanan penyimpanan data yang dibutuhkan program pada komputer.



Gambar 1.1 Bagan Struktur Program

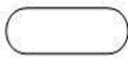

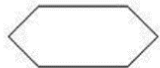

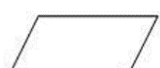




Kode program yang telah ditulis kemudian akan di-compile dan dieksekusi untuk menjalankannya. Compile adalah menerjemahkan kode program yang ditulis menggunakan bahasa pemrograman, ke bahasa mesin yang dapat dimengerti oleh komputer sehingga komputer mengerti apa yang diperintahkan.



Gambar 1.2 Mekanisme Eksekusi Program

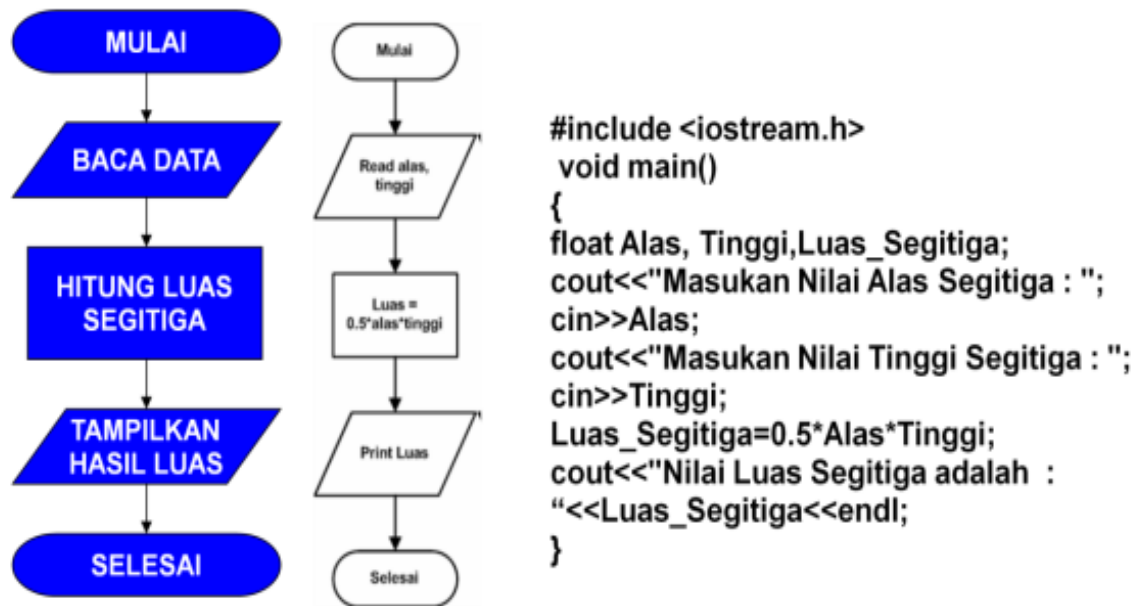
Flowchart

Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah, secara berurutan dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Gambar 1.3 Simbol-simbol flowchart

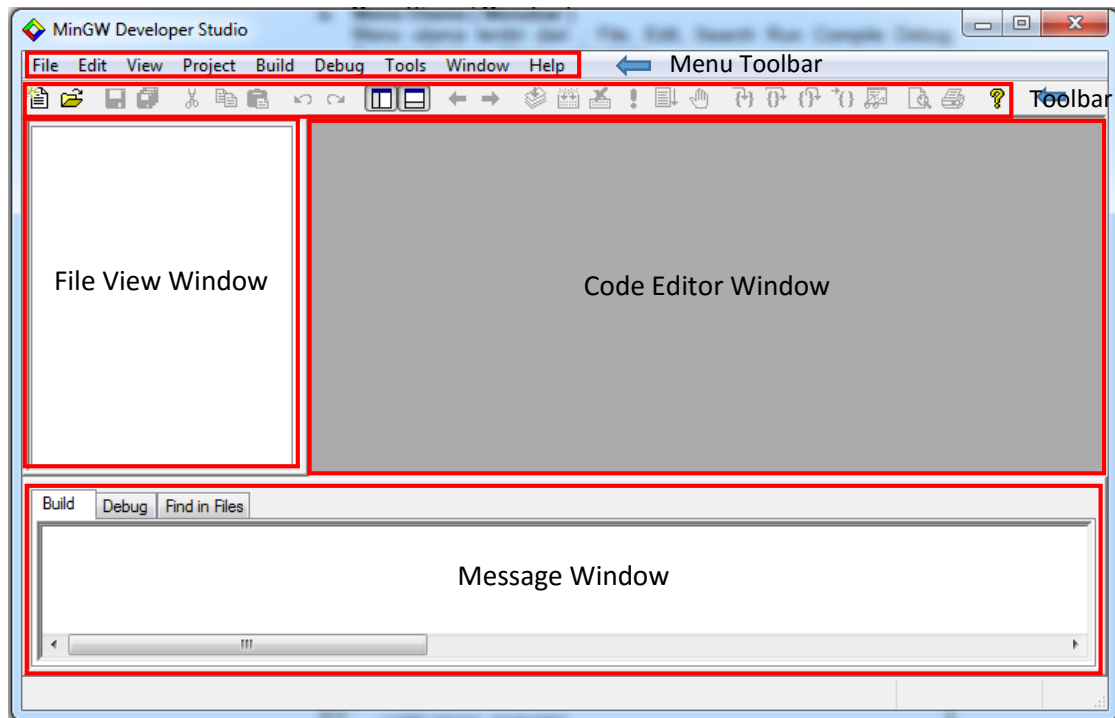
Berikut ini adalah contoh flowchart program sederhana :



Gambar 1.4 Contoh flowchat program sederhana

Mengenal IDE (Integrated Development Equipment) MinGW Developer Studio

IDE (*Integrated Development Equipment*) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. IDE pada MinGW Developer Studio terbagi menjadi lima bagian yaitu



Gambar 1.5 Tampilan IDE MinGW Developer Studio

a) *Menu Toolbar*

Menu utama yang terdiri dari ; File,Edit,View,Project, Build,Debug ,Tools ,Windows dan Help.

b) *Toolbar*

Menampilkan beberapa tool dari menu toolbar maupun sub menu-nya.

c) *Code Editor Window*

Tempat dituliskannya kode program.

d) *Message Window*

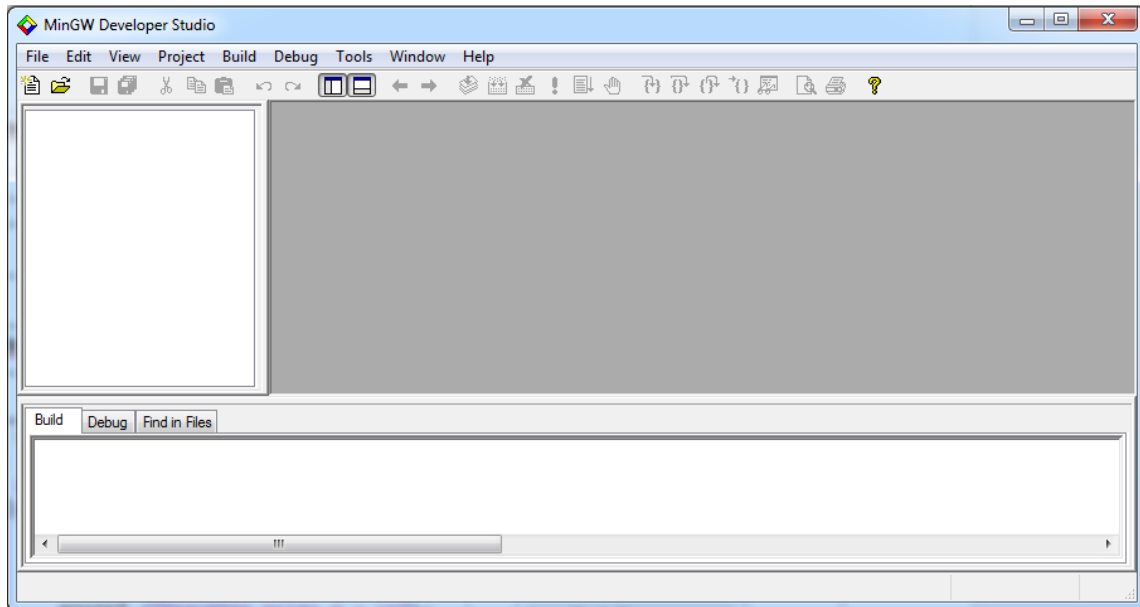
Menampilkan pesan-pesan pada proses kompilasi dan link program.

e) *File View Window*

Menampilkan program yang sedang dibuka dan dikerjakan.

Pengoperasian MinGW Developer Studio

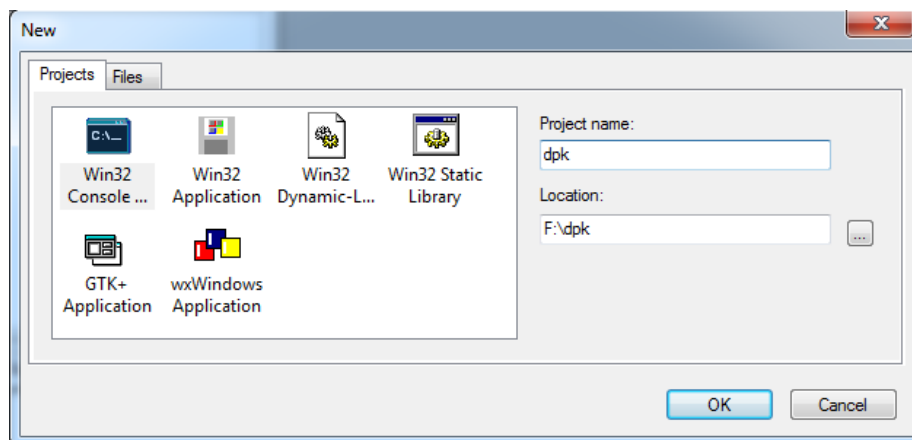
1. Membuka aplikasi MinGW



Gambar 1.6 Tampilan awal MinGW

2. Membuat Project baru

Klik menu File dan pilih new (Ctrl + N)

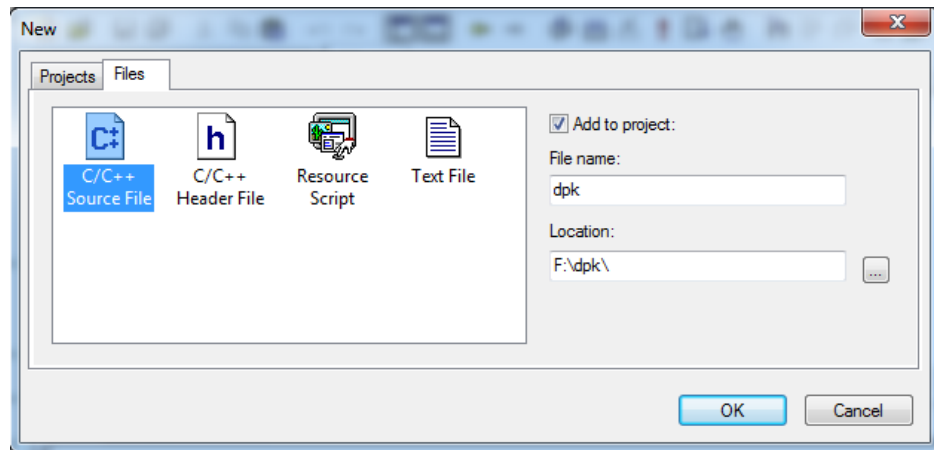


Gambar 1.7 Tampilan membuat project baru

Pilih tab “Projects” dan pilih “Win32Console..” . Beri nama project pada Project name dan pilih letak penyimpanan project tersebut di Location. Klik OK.

3. Membuat File baru

Klik menu File dan pilih new (Ctrl + N)



Gambar 1.8 Tampilan membuat file baru

Pilih tab “Files” dan pilih “C/C++ Source File” . Beri nama file pada File name dan pilih letak penyimpanan file tersebut di Location. Klik OK.

4. Membuat program sederhana

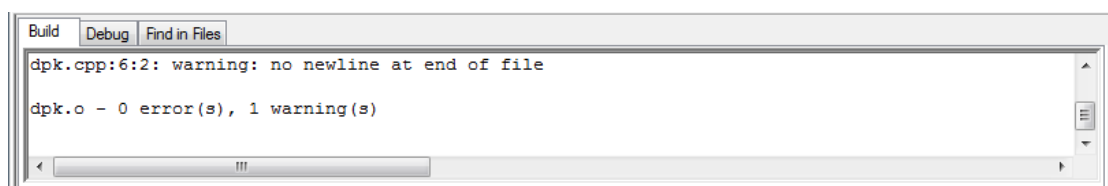
```
#include <stdio.h>

int main ()
{
    printf ("Selamat datang di Praktikum Pemrograman Komputer 2016 :) \n " ) ;
}
```

Gambar 1.9 Program Sederhana

5. Mengcompile dan eksekusi file

Klik tombol “compile” pada toolbar atau Ctrl + F7.



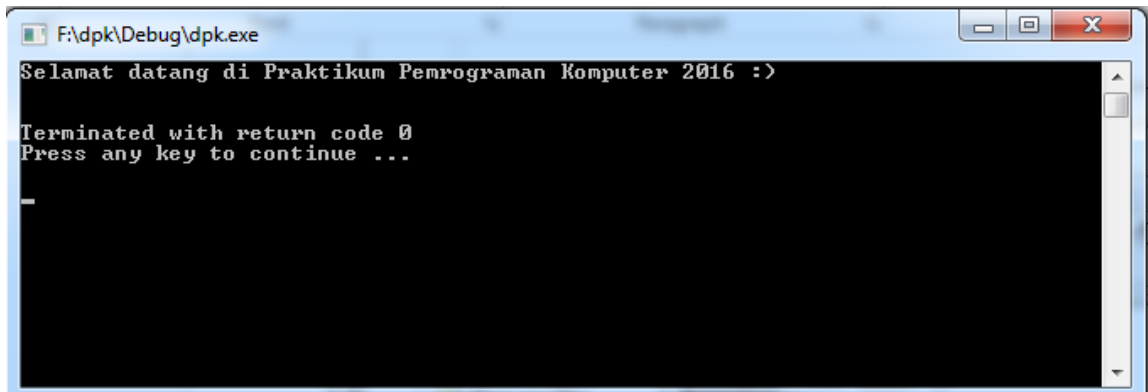
Gambar 1.10 Message Window

Pastikan tidak ada peringatan error atau warning yang membuat program tidak dapat di jalankan. Kemudian kik tombol “execute program” atau Ctrl + F5.



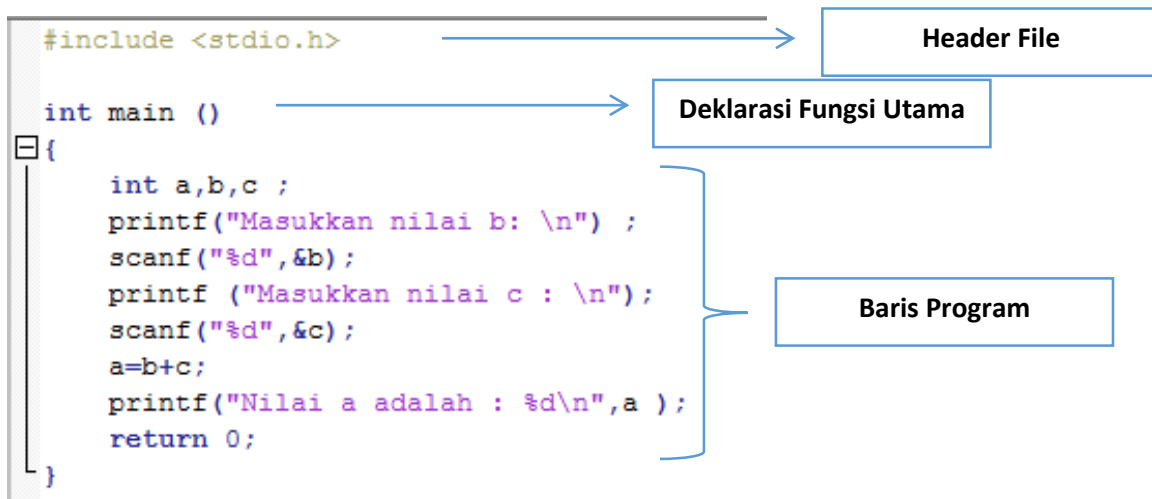
Gambar 1.11 Compile dan Execute

6. Hasilnya



Gambar 1.12 Output program

Struktur Sederhana Program C++



Gambar 1.13 Stuktur sederhana program C/C++

Komponen Bahasa C++

1. Header

File Header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax. File Header dalam C++ digunakan untuk memanggil library-library yang ada sehingga suatu fungsi dapat digunakan secara baik dan benar. Beberapa contoh file header di dalam C++ sebagai berikut :

File header	Input	Output	fungsi
<code>iostream</code>	<code>cin</code>	<code>cout</code>	
<code>conio.h</code>	<code>Getchar,</code> <code>getch,</code> <code>getche</code>		<code>clrscr()</code>
<code>stdio.h</code>	<code>Scanf,</code> <code>gets</code>	<code>Printf,</code> <code>puts,</code> <code>putchar</code>	
<code>windows.h</code>			<code>system()</code>
<code>math.h</code>			<code>sqrt(),</code> <code>pow(),</code> <code>log10(),</code> <code>sin()</code>
<code>string.h</code>			<code>strcpy(),</code> <code>strlen(),</code> <code>strcat()</code>

Gambar 1.14 Contoh file header

2. Fungsi Utama

Setiap program yang ditulis dengan menggunakan bahasa C harus mempunyai fungsi utama, fungsi tersebut bernama `main()`. Fungsi inilah yang akan dipanggil pertama kali pada saat eksekusi program. Apabila ada fungsi lain yang dibuat, maka fungsi tersebut akan dijalankan ketika dipanggil di fungsi utama. Hal ini karena bahasa C/C++ merupakan bahasa procedural yang menerapkan konsep runtutan (program dieksekusi baris-perbaris dari atas ke bawah secara berurutan).

3. Identifier / Pengenal

Identifier adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan variabel, konstanta, tipe data, dan fungsi. Aturan untuk penulisan *identifier* antara lain:

1. Tidak boleh dimulai dengan karakter non huruf
2. Tidak boleh ada spasi
3. Tidak boleh menggunakan karakter-karakter `~ ! @ # $ % ^ & * () + ` - = { }`
`[] : " ; ' < > ? , . / |`
4. Tidak boleh menggunakan *reserved words* yang ada dalam C/C++.

Reserved Word atau *Keyword* merupakan kata-kata yang telah ada/didefinisikan oleh bahasa pemrograman yang bersangkutan. Kata-kata tersebut telah memiliki definisi yang sudah tetap dan tidak dapat diubah. Karena telah memiliki definisi tertentu, maka kata-kata ini tidak dapat digunakan sebagai identifier.

asm	auto	bool	break	case	catch
char	class	const	const_cast	condition	default
delete	do	double	dynamic_cast	else	enum
explicit	export	extern	false	float	for
friend	goto	if	inline	int	long
mutable	namespace	new	operator	private	protected
public	register	reinterpret_cast	return	short	signed
sizeof	static	static_cast	struct	switch	template
this	throw	true	try	typedef	typeid
typename	union	unsigned	using	virtual	void
volatile	wchar_t	while			

Gambar 1.15 Reserved Words C/C++

4. Tipe Data

Tipe Data adalah jenis nilai yang dapat ditampung oleh suatu variabel. Berikut ini tipe data yang terdapat dalam C++ :

TYPE	Bit width		Common range
Char	8	(1byte)	-128 to -127
unsigned char	8	(1byte)	0 to 255
signed char	8	(1byte)	-128 to -127
int	32	(4bytes)	-2,147,483,648 to 2,147,483,647
unsigned int	32	(4bytes)	0 to 4,294,967,295
signed int	32	(4bytes)	-2,147,483,648 to 2,147,483,647
short int	16	(2bytes)	-32,768 to 32,767
unsigned short int	16	(2bytes)	0 to 65,535
signed short int	16	(2bytes)	-32,768 to 32,767
long int	32	(4bytes)	Sama as int
unsigned long int	32	(4bytes)	Same as unsigned int
signed long int	32	(4bytes)	Same as signed int
float	32	(4bytes)	3.4E-38 to 3.4E+38
double	64		1.7E-308 to 1.7E+308
long double	80		3.4E-4932 to 1.1E+4932
bool	N/A		true or false

Gambar 1.16 Jenis – jenis tipe data

5. Deklarasi Variabel

Variabel merupakan suatu tempat untuk menampung data atau konstanta di memori yang mempunyai nilai atau data yang dapat berubah – ubah selama proses program. Seperti halnya Pascal, variabel yang digunakan dalam

program harus dideklarasikan terlebih dahulu. Pengertian deklarasi di sini yaitu mengenalkan variabel ke program dan menentukan tipe datanya. Berikut ini contoh pendeklarasian variabel:

int jumlah;

float harga_satuan, variabel1;

char saya, kamu;

6. Assignment

Proses **assignment** adalah proses pemberian nilai kepada suatu variabel yang telah dideklarasikan. Berikut adalah contoh assignment:

Jumlah = 10;

Harga_satuan = 23.456;

Saya = 'B';

Kamu = '2';

Berikut ini contoh program yang menggambarkan deklarasi variabel dan assignment:

Contoh 1 :

```
#include <iostream.h>
#include <conio.h>

int main ()
{
    int var1, var2, var3;
    char karakter;
    var1 = 10;
    var2 = 5;
    var3 = var1 + var2;
    karakter = 'D';
    cout <<"Nilai var3= " << var3 << "\n";
    cout <<"Nilai karakter = " << karakter;
}
```

Gambar 1.17 Contoh deklarasi variable dan assignment

Contoh 2 :

```
#include "conio.h";
#include "iostream.h";
#include "iomanip.h";
void main()
{
    clrscr();
    double real;
    real = 182.2182713674821746;
    cout << "Nilai real = " << real;
}
```

Gambar 1.18 Contoh deklarasi variable dan assignment

7. Konstanta

Konstanta adalah identifier yang terkait nilai tidak bisa biasanya diubah oleh program selama pelaksanaannya. Untuk pendeklarasian konstanta dalam C++ mirip dengan Pascal. Contoh penulisan kode program (sintaks) adalah seperti berikut :

const tipe_data nama_konstanta = value;

const int mhs = 20;

8. Komentar

Komentar digunakan untuk memberikan penjelasan pada program dan tidak akan dieksekusi dalam program. Contoh penulisan komentar dalam program adalah sebagai berikut:

```
// -----
// ini adalah komentarku yang pertama
// -----
atau
/* -----
ini adalah komentarku yang kedua
----- */
```

Gambar 1.19 Contoh komentar

LATIHAN

1. Tuliskan kembali kode program di bawah ini dan identifikasi (header, variable, assignment dsb) ?

A.

```
#include <iostream.h>
#include <conio.h>

int main ()
{
    int var1, var2, var3;
    char karakter;
    var1 = 10;
    var2 = 5;
    var3 = var1 + var2;
    karakter = 'D';
    cout << "Nilai var3= " << var3 << "\n";
    cout << "Nilai karakter = " << karakter;
}
```

B.

```
#include "conio.h";
#include "iostream.h";
#include "iomanip.h";
void main()
{
    clrscr();
    double real;
    real = 182.2182713674821746;
    cout << "Nilai real = " << real;
}
```

2. Buatlah program sederhana untuk menghitung luas lingkaran !
3. Tuliskan kembali program di bawah ini dan simpan dengan ekstensi :
 - a. Ekstensi .cpp
 - b. Ekstensi .c

```
#include <iostream.h>

int main ()
{
    float Alas, Tinggi, Luas_Segitiga;

    cout<<"Masukkan Nilai Alas Segitiga: ">>;
    cin>>Alas;
    cout<<"Masukkan Nilai Tinggi Segitiga: ">>;
    cin>>Tinggi;
    Luas_Segitiga = 0.5*Alas*Tinggi;
    cout<<"Nilai Luas Segitiga adalah : "<<Luas_Segitiga<<endl;
}
```

4. Adakah perbedaan output kedua program pada no 3 ? Jelaskan !

TUGAS PRAKTIKUM

1. Identifikasi kode program di bawah ini.

```
#include <stdio.h>
#include <math.h>

int main ()
{
    double p, result;
    float param = 6.5;

    result = log (param);

    printf ("ln(%f) = %lf\n", param, result); /*cetak output program */
    return 0;
}
```

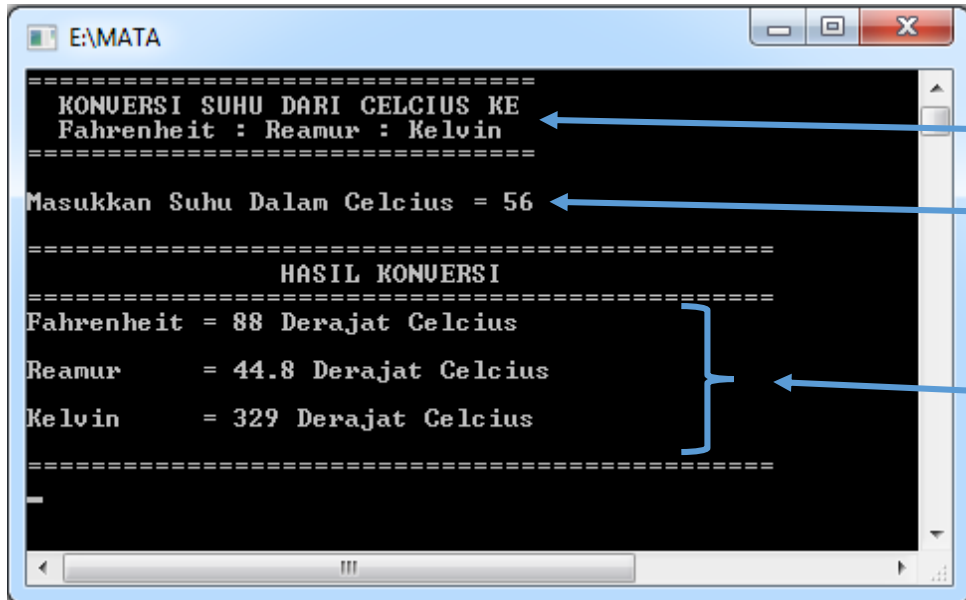
2. Identifikasi kesalahan kode program di bawah ini. Kemudian tulislah kembali dengan kode program yang benar.

```
#include (stdio,h)

void main ()
{
    long break = 0L; /*sejumlah angka dari ketinggian*/
    printf("Masukkan panjang: ");
    scanf("%ld", &break);
    printf("Angka yang Anda masukkan adalah %ld", break);
}
```


TUGAS RUMAH

1. Buatlah program konversi suhu sederhana dari Celcius di ubah menjadi Fahrenheit, Kelvin dan Reamur dengan tampilan program di bawah ini :



```
=====
KONVERSI SUHU DARI CELCIUS KE
Fahrenheit : Reamur : Kelvin
=====
Masukkan Suhu Dalam Celcius = 56
=====
                        HASIL KONVERSI
=====
Fahrenheit = 88 Derajat Celcius
Reamur      = 44.8 Derajat Celcius
Kelvin      = 329 Derajat Celcius
=====
```

Keterangan Program

Inputan dari keyboard

Hasil konversi