

Pengantar Bahasa C++

Review

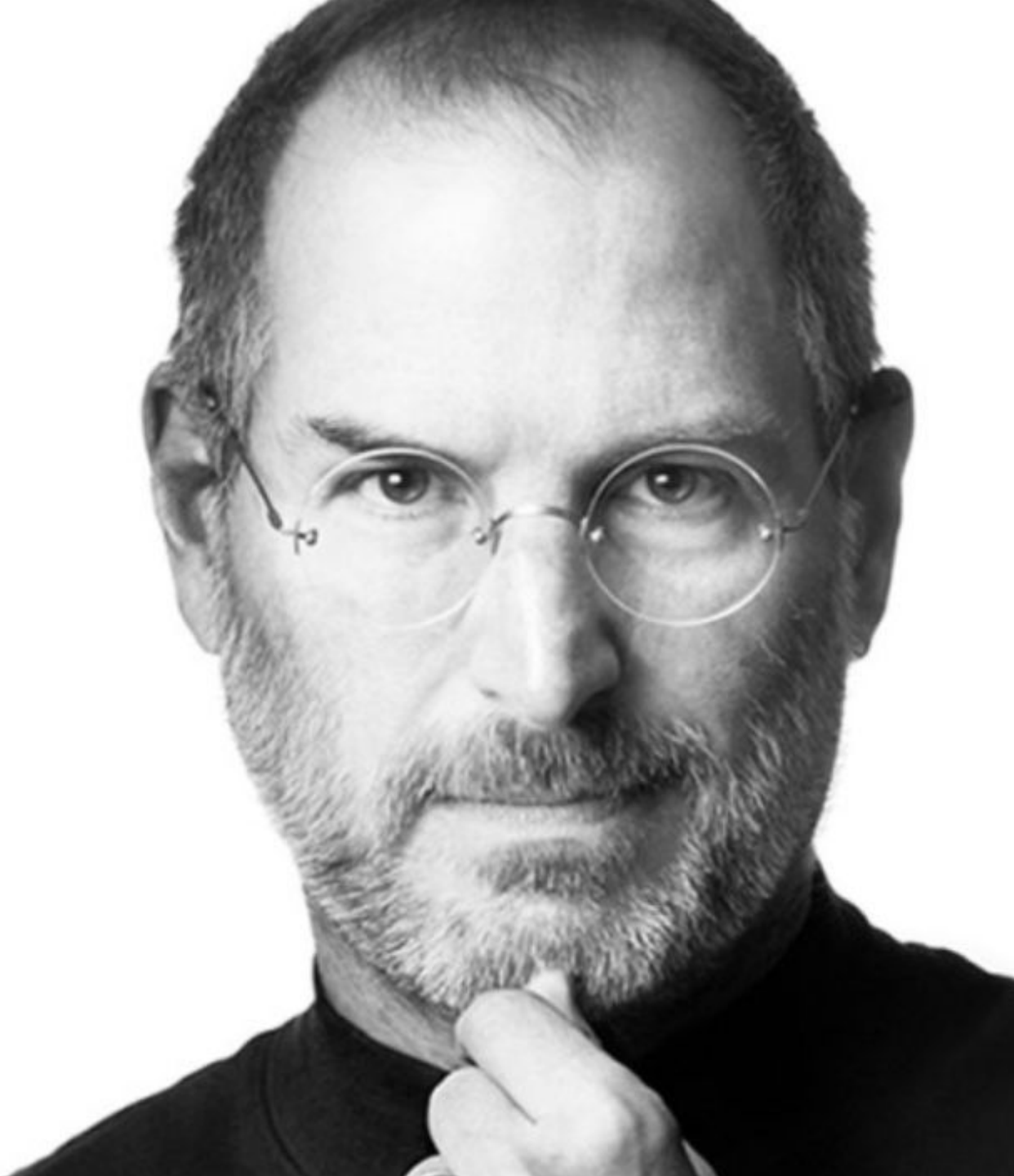
- ◆ Pengertian Flowchart
- ◆ Simbol Flowchart
- ◆ Pembahasan Tugas

Tugas minggu lalu

- ◆ Buatlah flowchart untuk mencari pangkat dari bilangan ($2^3 = 8$)
- ◆ Buatlah flowchart untuk menentukan kelulusan dari nilai ujian ($\text{nilai} = (\text{uts} + \text{uas}) / 2$). jika $\text{nilai} \geq 60 =$ "lulus". Nilai $< 60 =$ "Tidak lulus"
- ◆ Buatlah flowchart untuk menentukan bilangan factorial ($5! = 5 * 4 * 3 * 2 * 1 = 120$)

**"Everyone should know
how to program a computer,
because it teaches you
how to think."**

Steve Jobs



Bahasa pemrograman

- ◆ Sistem komunikasi yang digunakan dalam memerintahkan komputer secara langsung untuk menjalankan perintah khusus (programming language). (*Otoritas Jasa Keuangan*)
- ◆ Instruksi standar untuk memerintah komputer. (*Wikipedia*)

Tingkatan Bahasa Pemrograman

- ◆ 3 tingkatan Bahasa pemrograman menurut tingkat kedekatannya dengan mesin komputer:
- ◆ Bahasa Program Tingkat Rendah (Low-Level)
- ◆ Bahasa Program Tingkat Menengah (Mid-Level)
- ◆ Bahasa Program Tingkat Tinggi (High-Level)

Bahasa Program Tingkat Rendah (Low-Level)

- ❖ Bahasa program tingkat rendah berisi instruksi-instruksi yang ditujukan kepada komputer dengan menggunakan kode biner (binary). Kode-kode tersebut akan langsung diterjemahkan oleh komputer tanpa harus melalui proses kompilasi.
- ❖ Contoh: 01100101100110

Bahasa Program Tingkat Menengah (Mid-Level)

- ❖ Bahasa program tingkat menengah adalah instruksi yang berupa kode mnemonic atau Statement List (STL) seperti ADD, SUB, DIV, STO, LOD, JMP dan lainnya.
- ❖ Bahasa ini memakai campuran instruksi dalam kata-kata bahasa manusia dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&, ||, dan lainnya
- ❖ Bahasa dalam tingkat menengah ini harus diterjemahkan terlebih dahulu ke dalam bahasa mesin, karena komputer hanya mengerti penggunaan bahasa mesin.

00000	LOAD	M 0000
00001	OUT	P 0063
00002	LOAD NOT	M 0000
00003	OUT	P 0064

Bahasa Program Tingkat Tinggi (High-Level)

- ◆ Bahasa program tingkat tinggi ini berisi instruksi-instruksi yang menggunakan bahasa alamiah yang dimengerti manusia, seperti bahasa Inggris atau matematika. Tingkatan ini diciptakan untuk mengatasi kekurangan yang dimiliki tingkat menengah dan tingkat rendah.
- ◆ Untuk menerjemahkannya diperlukan program **compiler** atau **interpreter**

Kompilasi dan Interpretasi

- ◆ Sebuah kode program tidak bisa langsung dijalankan oleh computer, perlu suatu mekanisme untuk menerjemahkan kode program menjadi file yang dapat dieksekusi oleh computer atau biasa disebut executable dengan ekstensi .exe
- ◆ Ada dua mekanisme yang dapat digunakan yaitu kompilasi (compiler) dan interpretasi (interpreter).

Kompilasi dan Interpretasi

- ❖ Kompilasi merupakan proses untuk merubah kode sumber menjadi file executable oleh **compiler** dari Bahasa pemrograman pada saat seluruh kode program selesai ditulis.
- ❖ Dengan menggunakan compiler kesalahan akan diketahui saat proses kompilasi dilakukan.
- ❖ Contoh Bahasa pemrograman yang menggunakan compiler: C++, java, cobol, pascal dan lainnya

Kompilasi dan Interpretasi

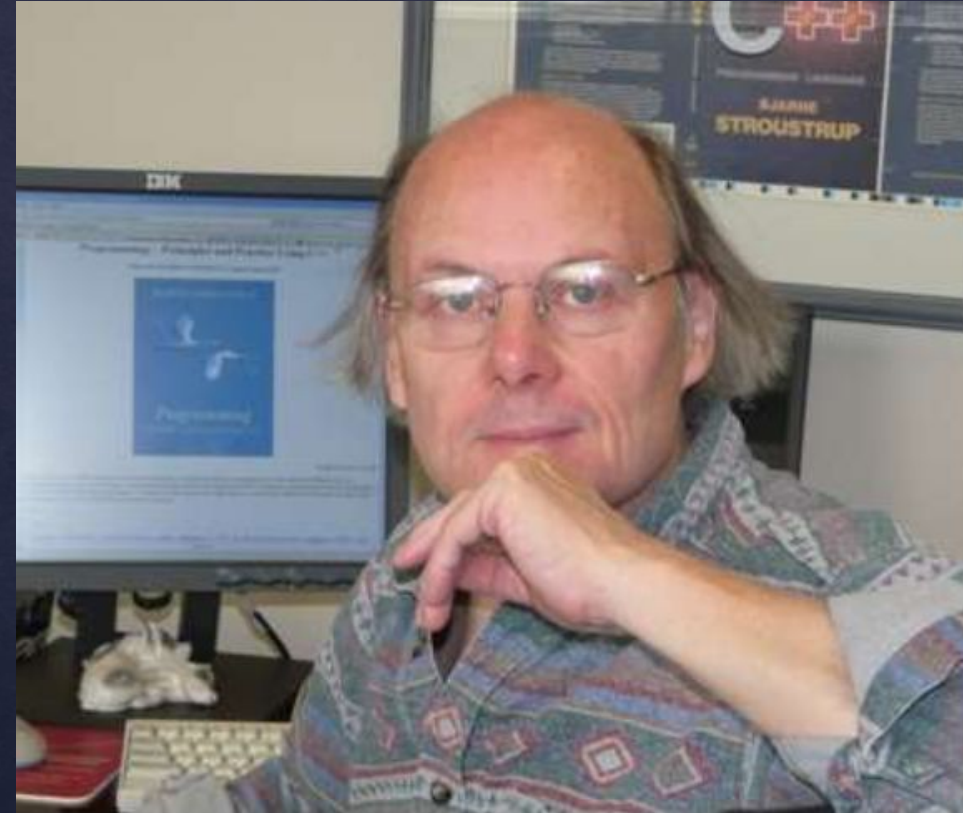
- ❖ Interpretasi adalah proses untuk merubah kode sumber menjadi file executable oleh interpreter Bahasa pemrograman secara baris per baris.
- ❖ jika menggunakan interpreter kesalahan akan ditemukan pada saat pindah ke baris berikutnya.
- ❖ Bahasa pemrograman yang menggunakan interpreter seperti Basic, Python, Ruby, dan MATLAB.

Kompilasi dan Interpretasi

Kita tidak dapat memilih menggunakan compiler atau interpreter karena proses ini merupakan karakteristik dari sebuah Bahasa pemrograman

Berkenalan Dengan C++

- ◆ C++ adalah bahasa pemrograman yang dibuat oleh Bjarne Stroustrup pada tahun 1979.
- ◆ Seperti namanya, Simbol “++” pada huruf C berarti increment dari C.
- ◆ Sebenarnya C++ sama seperti bahasa C, tapi memiliki fitur yang lebih banyak dibandingkan C. Karena itulah dinamakan C++ (dibaca si plus plus).
- ◆ Bahasa C++ sudah mendukung konsep pemrograman berorientasi objek



Berkenalan Dengan C++

- ◆ Banyak aplikasi yang kita gunakan saat ini dibuat dengan C++, misalnya:
 - MySQL Server, Mozilla Firefox, Thunderbird, Google Chrome, Microsoft Office, Adobe Photoshop, Adobe Illustrator, Java Virtual Machine (JVM) Desktop Environment: KDE dan Apple MacOS UI (Aqua) dan masih banyak lagi.
 - Kalian pernah gunakan yang mana?



Design and programming are human activities; forget that and all is lost.

— *Bjarne Stroustrup* —

AZ QUOTES

Mencoba C++

- ◆ Untuk dapat mencoba c++ ada dua aplikasi yang perlu kita siapkan yaitu compiler c++ dan editornya untuk memudahkan menulis.
- ◆ Pada perkuliahan ini kita akan menggunakan aplikasi DevC++ dimana aplikasi ini telah menggabungkan compiler dan editor di satu aplikasi.
- ◆ Selain itu DevC++ dapat diinstall pada computer dengan system operasi 32 ataupun 64 bit.
- ◆ Aplikasi ini dapat di unduh pada link berikut [Download](#)

DevC++

- ◆ Dev-C ++ adalah lingkungan pengembangan terintegrasi (IDE) yang lengkap dan gratis.
- ◆ DevC++ Itu ditulis dalam Delphi.
- ◆ DevC++ dibundel dengan compiler MinGW atau TDM-GCC dari GCC.
- ◆ DevC++ di kembangkan oleh Bloodshed Software pada tahun 2005, dan dilanjutkan oleh Orwell (Johan Mes) dari tahun 2011 hingga saat ini.
- ◆ Untuk menginstall aplikasi ini sangat mudah, jika ada yg mengalami kendala bisa cek tutorial [DISINI](#)

Struktur Bahasa C++

- ◆ Seperti Bahasa pemrograman lainnya, Bahasa c++ juga memiliki struktur program yang membentuknya dan harus ada saat proses kompilasi.
- ◆ Struktur bahasa C++ terdiri dari kumpulan baris pernyataan yang dibentuk dari sejumlah blok fungsi dan keyword yang digunakan untuk memberikan instruksi kepada komputer untuk melaksanakan tugas tertentu.

```
1. #Preprocessor Directive (library)
2. fungsi utama(){
3.     Definisi yang terdiri dari kumpulan pernyataan atau ekspresi;
4.     //komentar
5. }
```

Struktur Bahasa C++

- ◆ Baris ke1- adalah menambahkan file preprocessor directive atau file library. File ini ditambahkan agar fungsi yang digunakan pada aplikasi dapat dipanggil.
- ◆ Baris ke-3 digunakan untuk memberikan informasi kepada compiler bahwa aplikasi ini menggunakan seluruh berkas, class, atau fungsi yang menjadi bagian dari sebuah namespace std yang bersangkutan.

```
1. #include <iostream>
2.
3. using namespace std;
4.
5. int main()
6. {
7.     cout <<"Hallo Universitas Universal!!!"<<endl;
8.     return 0;
9. }
```


Struktur Bahasa C++

- ◆ Baris ke-5 merupakan fungsi utama dari c++. C++ dapat terdiri dari banyak fungsi.
- ◆ Baris ke-7 menampilkan tulisan dilayar "hallo Universitas Universal!!!" dengan fungsi cout. Perintah endl bermaksud pindah baris.
- ◆ Baris ke-8 adalah nilai kembalian untuk fungsi utama. Pembahasan ini akan dibahas pada pertemuan ke 11.
- ◆ Kurung kurawal buka ({) bermakna awalan fungsi dan kurawal tutup(}) bermakna akhir dari fungsi.

```
1. #include <iostream>
2.
3. using namespace std;
4.
5. int main()
6. {
7.     cout <<"Hallo Universitas Universal!!!"<<endl;
8.     return 0;
9. }
```

Variable

- ❖ Variabel merupakan sebuah nama atau pengenal yang digunakan untuk menampung data sesuai tipe tertentu
- ❖ Setiap variabel memiliki **Nama**, **alamat**, **tipe**, **size** dan **data**.
- ❖ Data atau isi variabel bisa dirubah pada saat dijalankan.
- ❖ Format deklarasi variabel:

<tipe data> <nama_variabel>;

<tipe data> <nama_variabel> = <initial_value>;

- ❖ Referensi tentang variable dan tipe data bisa cek [DISINI](#)
- ❖ Contoh:

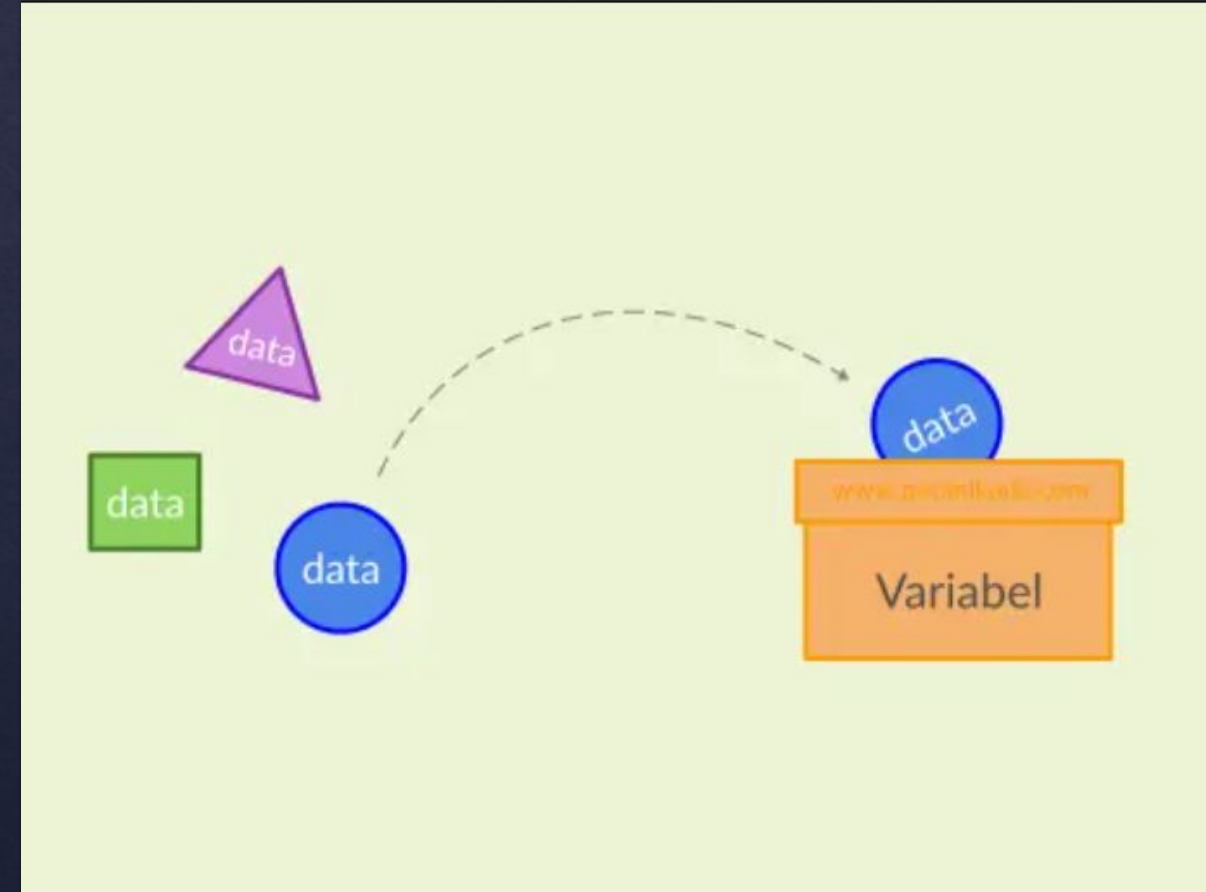
```
int jumlah;  
string nama;  
float harga = 10000;
```

Aturan penulisan Variable

- ❖ Tidak boleh menggunakan kata yang sudah dipesan (reserved word)
- ❖ Tidak boleh berawalan angka atau symbol
- ❖ Tidak boleh mengandung spasi
- ❖ C++ merupakan Bahasa yang ***case sensitive***. Artinya huruf besar dan kecil dianggap berbeda.

Ilustrasi Variable

- ◆ Variable ibaratkan tempat penampung data
- ◆ Data bisa di masukkan dan dikeluarkan dari variable
- ◆ Perlu di ingat bahwa setiap variable memiliki tipe / jenis masing – masing **tergantung dari data yang ingin ditampung.**
- ◆ Satu variable hanya dapat menampung satu data.
- ◆ *Misalnya kita ingin menampung air gunakan gelas jangan gunakan piring.*



Ilustrasi Variable

- ◆ Setiap variable yang dibuat akan ditempatkan di memory computer (RAM).
- ◆ Memory computer akan menyimpan nama variable, nilai dan alamat memorynya.
- ◆ Contoh jika kita mendeklarasikan variable pada c++
int angka = 90;

Nama Variabel	angka
Nilai Variabel	90
Alamat Memori	0x6ffe34



Tipe Data pada C++

- ◆ Tipe data digunakan untuk menentukan jenis data apa yang akan ditampung oleh variable.
- ◆ Tipe data akan memberitahukan kepada compiler mengenai jenis data dan seberapa lebar compiler mengalokasikan ruang memori untuk suatu variabel.
- ◆ Tipe data dapat dibedakan berdasarkan adanya tanda (signed) dan tidak adanya tanda (unsigned). Perbedaan antara keduanya adalah adanya tanda bilangan yang menunjukkan positif dan negatif. Untuk tipe data unsigned suatu bilangan hanya diawali dari 0 ke suatu jangkauan tertentu, sedangkan untuk tipe signed bilangan diawali dari nilai negatif (-) menuju ke jangkauan nilai positif (+).

Tipe Data Signed

Tipe Data	Memory	Jangkauan	Contoh	Keterangan
bool	1 byte	True atau false	True	Hanya berisi true atau false (1/0). Biasanya digunakan untuk kondisi tertentu.
char	1 Byte	-127 to 127 or 0 to 255	'Y'	Bilangan bulat dengan kapasitas memory 1 byte. Biasanya digunakan menampung untuk konstanta.
int	4 Byte	-2147483648 to 2147483647	1	Bilangan bulat yang berisi angka positif dan negative. Biasanya digunakan untuk melakukan perhitungan sederhana.
long	4 Byte	-2,147,483,648 to 2,147,483,647	40	Digunakan untuk pengganti int dengan nilai yang besar
float	4 Byte	+/- 3.4e +/- 38 (~7 digits)	4.5	Digunakan untuk perhitungan dengan nilai pecahan
double	8 Byte	+/- 1.7e +/- 308 (~15 digits)	1.0	Digunakan untuk perhitungan yang besar
string	8 Byte	Sebuah kalimat	"Keluarga Uvers"	Digunakan untuk menyimpan sebuah kalimat

Tipe Data Unsigned

Tipe Data	Memory	Jangkauan	Contoh	Keterangan
Unsigned char	1 byte	0-255	'B'	Menampung karakter
Unsigned int	4 byte	0-4294967295	20	Bilangan bulat tak bertanda atau positif
Unsigned long int	4 byte	0-4294967295	5000	Long integer
Unsigned short int	2 byte	0-65535	90	Short integer

Konstanta

- ◆ Konstanya merupakan variabel dengan nilai tetap dan tidak dapat dilakukan perubahan.
- ◆ Pendeklarasian konstanta dapat dilakukan dengan perintah `const` atau `#define`
- ◆ Jika menggunakan `const` maka dituliskan didalam fungsi utama.
- ◆ Jika menggunakan `#define` dituliskan pada bagian preprocessor (library) dan tidak dituliskan tipe datanya.

```
const tipeData namaKonstanta = nilai;
```

```
#define namaKonstanta nilai;
```

- ◆ Contoh:

```
#define phi = 3,14;  
  
const float phi = 3.14;
```


Reserved Words

- ◆ C++ mempunyai 32 buah kata yang dipesan (reserved words)
- ◆ Reserved words dibagi menjadi dua kelompok yaitu kelompok pertama yang merupakan turunan dari bahasa C dan kelompok kedua yang merupakan kata baru dari c++

Reserved Words

◆ Reserved words kelompok pertama

auto	const	double	float	int	short	Struct	unsigned
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	sizeof	typedef	Volatile
char	do	extern	if	return	static	union	while

Reserved Words

◆ Reserved words kelompok kedua

asm	dynamic_cast	namespace	reinterpret_cast	try	delete
bool	explicit	new	static_cast	typeid	mutable
catch	false	operator	template	typename	protected
class	friend	private	this	using	true
const_cast	inline	public	throw	virtual	wchar_t

**“BEAUTY IS VARIABLE, UGLINESS IS
CONSTANT.”**

DOUGLAS HORTON

Fungsi input dan output

- ◈ Agar user dapat memasukkan data kedalam variable diperlukan sebuah fungsi.
- ◈ Banyak fungsi input dan output yang dapat digunakan.
- ◈ Fungsi ini tergantung dari preprocessor derivative yang (library) digunakan.

iostream

- ❖ Untuk dapat menggunakan fungsi input dari iostream terlebih dahulu perlu di deklarasikan pada bagian awal program. `#include <iostream>`
- ❖ Ada beberapa perintah yang bisa di tangani oleh library ini antara lain:

No	Fungsi	Contoh	Keterangan
1	Cin	<code>cin>>nama;</code>	Perintah untuk menampung inputan user kedalam variable
2	Cout	<code>cout<<"uvers";</code>	Perintah untuk menampilkan output ke layar
3	Endl	<code>cout<<endl;</code>	Perintah untuk pindah baris (end line)

Contoh Program

```
1. #include <iostream>
2. using namespace std;
3.
4. #define tinggi 10
5. #define garis "-----"
6.
7. int main () {
8.
9.     const double pi = 3.14;
10.
11.     double vol, r;
12.
13.     cout<<"Program mencari volume kerucut"<<endl;
14.     cout<<garis<<endl;
15.
16.     cout<<"Masukkan jari-jari: "; cin>>r;
17.
18.     vol = (pi * r * r * tinggi) / 3;
19.
20.     cout<<"Volume kerucut adalah: "<<vol;
21.
22.     return 0;
23. }
```

Semangattt!!!