

Pertemuan 2

Pendahuluan Bahasa C



Mata Kuliah : Algoritma & Pemrograman
Dosen : Tessy Badriyah, SKom., MT., PhD.



Tujuan Pembelajaran

- Memahami Struktur dasar dari program bahasa C
- Memahami bagian2 dari dasar pemrograman bahasa C yaitu: Keyword, Identifier, Variable, Konstanta, Tipe data



Dari Algoritma ke Pemrograman

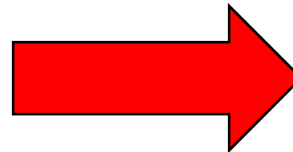
- Keduanya merupakan kumpulan instruksi bagaimana cara melakukan sesuatu tugas tertentu
- Algoritma:
 - Berbicara ke manusia, mudah dimengerti
 - Dengan Bahasa yang sederhana
- Program:
 - Berbicara ke komputer (compiler)
 - Dapat dinyatakan sebagai ekspresi formal dari suatu 'algoritma'

Bahasa Tingkat Tinggi (High-Level Language)

```
#include <stdio.h>

int main()
{
    printf("Hello World");
    return 0;
}
```

Source code dalam C



```
10100110 01110110
00100110 00000000
11111010 11111010
01001110 10100110
11100110 10010110
11001110 00101110
10100110 01001110
11111010 01100110
01001110 10000110
etc...
```

Executable code

- **Compilers** dan **linkers** men-translate program tingkat tinggi ke dalam kode Bahasa mesin yang dapat dijalankan



Mengapa C?

- Bersifat fleksibel
 - Bahasanya terstruktur
 - Memungkinkan dilakukan sesuatu yang sifatnya Low level
- Library nya bersifat standart dan portable
- Dapat menghasilkan kode program yang ringkas dan efisien
- Tersedia dalam berbagai macam jenis komputer
- Banyak digunakan



Sejarah C

- **CPL** Combined Programming Language (Barron et al., 1963)
- **BCPL** Basic CPL (Richards, 1969)
- **B** (Thompson, 1970)
- **C** K&R C (Ritchie, 1972)
- **ANSI C** American National Standards Institute C (X3J11, 1989)
- **C99** (JTC1/SC22/WG14, ISO/IEC 9899, 1999)



Struktur dasar dari Program Bahasa C

Contoh: Hello World

Algoritma:

output "Hello
World!"

C Program:

```
#include <stdio.h>

int main()
{
    printf("Hello World!");

    return 0;
}
```



Struktur dasar dari Program Bahasa C

Contoh: Hello world

include merupakan standard input/output library dari suatu procedures.

C Program:

```
#include <stdio.h>

int main()
{
    printf("Hello World!");

    return 0;
}
```




Struktur dasar dari Program Bahasa C

Contoh: Hello World

Tanda kurung kurawal awal dan akhir merupakan **permulaan** and **akhir** dari blok instruksi.

C Program:

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```



Struktur dasar dari Program Bahasa C

Contoh: Hello World

Instruksi (**fungsi pemanggilan dengan printf**) untuk menampilkan output "Hello World"

C Program:

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```



Struktur dasar dari Program Bahasa C

Contoh: Hello World

Statements” (baris instruksi)
selalu diakhiri dengan **semi-colon** (;)

C Program:

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```



C Keywords & Identifier

- Himpunan karakter
 - Yaitu himpunan dari alphabet, huruf, dan beberapa karakter special yang valid dalam Bahasa C.
- Alfabet
 - Huruf besar: A B C
 - Huruf kecil: a b c
- Digits / bilangan
 - 0 1 2 3 4 5 6 7 8 9
- Karakter khusus
 - Spasi, pindah baris, tab, tombol Enter, dll.

X V 7

Special Characters in C Programming

,	<	>	.	-
()	;	\$:
%	[]	#	?
'	&	{	}	"
^	!	*	/	
_	\	~	+	



C Keywords

- Keywords dalam Bahasa C adalah kata (word) yang digunakan dalam Bahasa pemrograman yang merupakan bagian dari aturan penulisan dan untuk itu maka keyword tidak dapat digunakan sebagai identifier.
- Contoh: `int luas;`
- Disini, `int` adalah keyword yang menunjukkan bahwa '`luas`' adalah variable bertipe integer.
- Dikarenakan C bersifat Bahasa pemrograman yang 'case sensitive', artinya huruf kecil dan besar dibedakan, maka semua keywords harus ditulis dalam huruf kecil semua.

Keywords in C Language

<code>auto</code>	<code>double</code>	<code>int</code>	<code>struct</code>
<code>break</code>	<code>else</code>	<code>long</code>	<code>switch</code>
<code>case</code>	<code>enum</code>	<code>register</code>	<code>typedef</code>
<code>char</code>	<code>extern</code>	<code>return</code>	<code>union</code>
<code>continue</code>	<code>for</code>	<code>signed</code>	<code>void</code>
<code>do</code>	<code>if</code>	<code>static</code>	<code>while</code>
<code>default</code>	<code>goto</code>	<code>sizeof</code>	<code>volatile</code>
<code>const</code>	<code>float</code>	<code>short</code>	<code>unsigned</code>



C Identifiers

- Identifier mengacu pada nama yang diberikan pada variable, fungsi, struktur atau segala yang berkaitan dengan pemrograman dalam C yang perlu diidentifikasi dengan sebuah nama.
- Identifier harus bersifat unique, artinya tidak boleh ada yang sama namanya.
- Contoh :

```
int luas;  
double volumeKubus;
```
- Disini, **luas** dan **volumeKubus** adalah identifiers.
- Nama identifier juga tidak boleh sama namanya dengan keywords. Jadi, kita tidak bisa menggunakan **int** sebagai identifier karena **int** adalah sebuah keyword.



Aturan Penulisan identifier

1. Nama identifier yang valid dapat terdiri dari huruf (baik huruf kecil maupun huruf besar), digits dan underscores.
 2. Karakter pertama dari identifier harus huruf atau underscore. Bagaimanapun, sebaiknya tidak menggunakan underscore sebagai karakter pertama dari identifier.
 3. Tidak ada aturan pada panjang identifier. Compiler akan membedakan suatu identifier dengan identifier lainnya berdasarkan 31 karakter pertama.
- Sebaiknya:
 - Buat kata yang mengandung arti untuk nama identifier (sesuai dengan fungsinya), sehingga mudah dipahami penggunaannya untuk apa.



Variabel dalam C

- Dalam pemrograman, variable adalah tempat untuk menyimpan data (atau sebuah nilai).
- Nama dari variable merupakan representasi simbolis dari lokasi memori tempat dimana data dalam variable disimpan.
- Contoh: `int luasLingkaran = 60;`
- Disini, **luasLingkaran** adalah variable bertipe integer, yang memiliki nilai : 60.
- Nilai dari sebuah variable dapat diubah-ubah..
- Dalam Bahasa C, sebuah variable perlu dideklarasikan terlebih dahulu sebelum digunakan.



Aturan penamaan variabel

Sama seperti aturan penamaan identifier, yaitu :

1. Nama variable yang valid dapat terdiri dari huruf (baik huruf kecil maupun huruf besar), digits dan underscores.
2. Karakter pertama dari variable harus huruf atau underscore. Bagaimanapun, sebaiknya tidak menggunakan underscore sebagai karakter pertama dari variable.
3. Tidak ada aturan pada panjang variable. Compiler akan membedakan suatu variable dengan variable lainnya berdasarkan 31 karakter pertama.



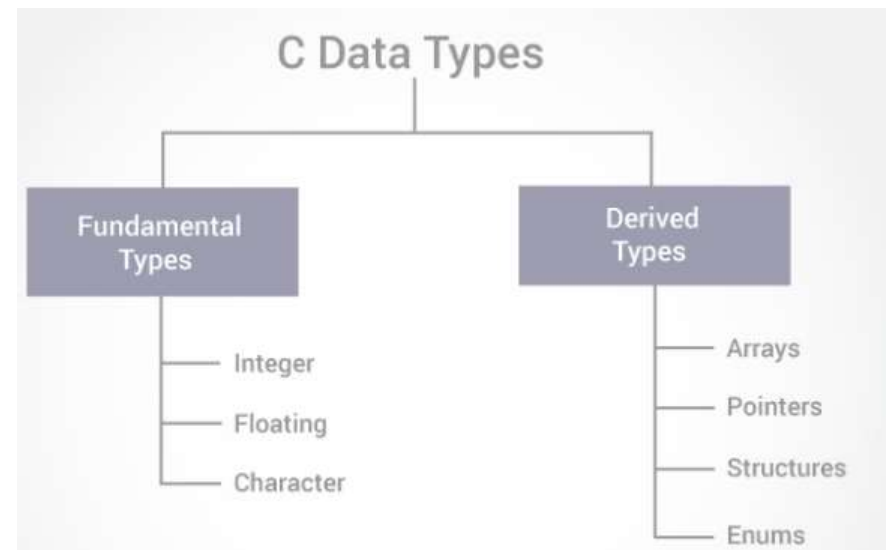
Konstanta

- Konstanta adalah nilai dari suatu identifier yang tidak dapat diubah.
- Contoh: `const double PI=3.14`
- Disini, **PI** adalah sebuah konstanta. Dimana nilainya selalu tetap yaitu **3.14**



Tipe Data dalam C

- Dalam Bahasa pemrograman C, variable atau lokasi memory harus dideklarasikan sebelum digunakan.
- Tipe data mengaju pada tipe dan ukuran dari data yang tersimpan dalam variable.





Tipe Data dalam C

1. Tipe Data dasar

- Integer
- Floating
- Character

2. Tipe data turunan

- Arrays
- Pointers
- Structures
- Enumeration



Tipe data dasar: Integer

1. int - Integer

- Integers adalah tipe data dari bilangan bulat (bukan pecahan). Contoh : 0, -5, 10
- Dalam Bahasa pemrograman C, keyword **int** digunakan untuk mendeklarasikan variable.
- Contoh: `int id;`
- Disini, **id** adalah variable bertipe integer.
- Pada bagian deklarasi, sekaligus dapat dideklarasikan beberapa variable, contoh : `int luas, keliling, volume;`



Tipe data dasar: Float

2. float - Floating types

- Variabel bertipe data float dapat menyimpan bilangan pecahan, semisal: 2.34, -9.382, 5.0 etc. Kita dapat mendeklarasikan variable yang dapat menyimpan bilangan pecahan dengan menggunakan keyword **float** atau **double**.

- Contoh :

```
float jumlahTotal;  
double hargaBuku;
```

- Disini, baik **jumlahTotal** maupun **hargaBuku** dapat menyimpan bilangan pecahan..
- Dalam C, bilangan pecahan dapat direpresentasikan dalam bentuk eksponensial, contohnya :

```
float normalizationFactor = 22.442e2;
```



Tipe data dasar: Double dan char

- Perbedaan antara tipe data float dan double
 - Ukuran dari **float** adalah 4 bytes. Sedangkan ukuran dari **double** adalah 8 bytes. Presisi dari variable floating point adalah 6 digits sementara itu presisi dari variable double adalah 14 digits.

3. char – tipe data Karakter

- Keyword **char** digunakan untuk mendeklarasikan variable bertipe karakter.
- Contoh:

```
char test = 'h'
```
- Disini, variable karakter **test** memiliki nilai 'h'.
- Ukuran dari variable karakter ini adalah 1 byte.



Yang sudah dipelajari

- Struktur dasar dari program Bahasa C
- Penggunaan Keyword, Identifier, Variable, Konstanta, dan tipe data dasar dalam program Bahasa C.



Review: Algoritma Kabisat

1. Jika angka tahun itu habis dibagi 400, maka tahun itu sudah pasti tahun kabisat.
2. Jika angka tahun itu tidak habis dibagi 400 tetapi habis dibagi 100, maka tahun itu sudah pasti **bukan** merupakan tahun kabisat.
3. Jika angka tahun itu tidak habis dibagi 400, tidak habis dibagi 100 akan tetapi habis dibagi 4, maka tahun itu merupakan tahun kabisat.
4. Jika angka tahun tidak habis dibagi 400, tidak habis dibagi 100, dan tidak habis dibagi 4, maka tahun tersebut **bukan** merupakan tahun kabisat.



Review: Pseudocode Kabisat

- **if** year is divisible by 400 **then** is_leap_year
else if year is divisible by 100 **then** not_leap_year
else if year is divisible by 4 **then** is_leap_year
else not_leap_year



Tugas 2

- Buat Flowchart untuk Tahun Kabisat berdasarkan algoritma dan pseudo code yang ada pada slide sebelumnya.



Referensi

- Robertson, Lesley Anne. (1992). *Students' guide to program design*. Oxford : Newnes
- Santner, Williams, and Notz (2003), *Design and Analysis of Computer Experiments*, Springer.
- Deitel & Deitel, *C How to Program*, Prentice Hall 1994 (2nd edition)
- Brookshear, J.G., *Computer Science: An Overview*, Benjamin-Cummings 2000 (6th edition)
- Kernighan & Ritchie, *The C Programming Language*, Prentice Hall