

# Algoritma & Pemrograman

Pertemuan 8:

- Pengenalan Bahasa Pemrograman
- Persiapan pembuatan program bahasa C
  - Struktur program bahasa C
- Tipe Data dan Variabel dalam bahasa C

# Pengenalan Bahasa Pemrograman

# Pendahuluan

- Ada berapa total jumlah bahasa pemrograman yang ada?
  - Ribuan, beberapa mengklaim ada sekitar 8500 bahasa beserta variannya

- 1951 – **Regional Assembly Language**
- 1952 – **Autocode**
- 1954 – **IPL** (forerunner to LISP)
- 1955 – **FLOW-MATIC** (led to COBOL)
- 1957 – **FORTRAN** (First compiler)
- 1957 – **COMTRAN** (precursor to COBOL)
- 1958 – **LISP**
- 1958 – **ALGOL 58**
- 1959 – **FACT** (forerunner to COBOL)
- 1959 – **COBOL**
- 1959 – **RPG**
- 1962 – **APL**
- 1962 – **Simula**
- 1962 – **SNOBOL**
- 1963 – **CPL** (forerunner to C)
- 1964 – **BASIC**
- 1964 – **PL/I**
- 1966 – **JOSS**
- 1967 – **BCPL** (forerunner to C)

- 1968 – Logo
- 1969 – B (forerunner to C)
- 1970 – Pascal
- 1970 – Forth
- 1972 – C
- 1972 – Smalltalk
- 1972 – Prolog
- 1973 – ML
- 1975 – Scheme
- 1978 – SQL (a query language, later extended)

- 1980 – C++ (as C with classes, renamed in 1983)
- 1983 – Ada
- 1984 – Common Lisp
- 1984 – MATLAB
- 1985 – Eiffel
- 1986 – Objective-C
- 1986 – Erlang
- 1987 – Perl
- 1988 – Tcl
- 1988 – Mathematica
- 1989 – FL (Backus)

- 1990 – Haskell
- 1991 – Python
- 1991 – Visual Basic
- 1993 – Ruby
- 1993 – Lua
- 1994 – CLOS (part of ANSI Common Lisp)
- 1995 – Ada 95
- 1995 – Java
- 1995 – Delphi (Object Pascal)
- 1995 – JavaScript
- 1995 – PHP
- 1996 – WebDNA
- 1997 – Rebol
- 1999 – D

- 2000 – ActionScript
- 2001 – C#
- 2001 – Visual Basic .NET
- 2003 – Groovy
- 2003 – Scala
- 2005 – F#

- 2006 – Windows PowerShell
- 2007 – Clojure
- 2009 – Go
- 2011 – Dart
- 2014 – Swift
- 2015 – Rust



Anders  
Hejlsberg

Turbo Pascal, Delphi,  
C#, TypeScript

Alan Cooper



Visual Basic



Yukihiro  
Matsumoto

Ruby

Rasmus Lerdorf



PHP



Dennis Ritchie, 2011

C

Guido van Rossum



Python



Bjarne  
Stroustrup

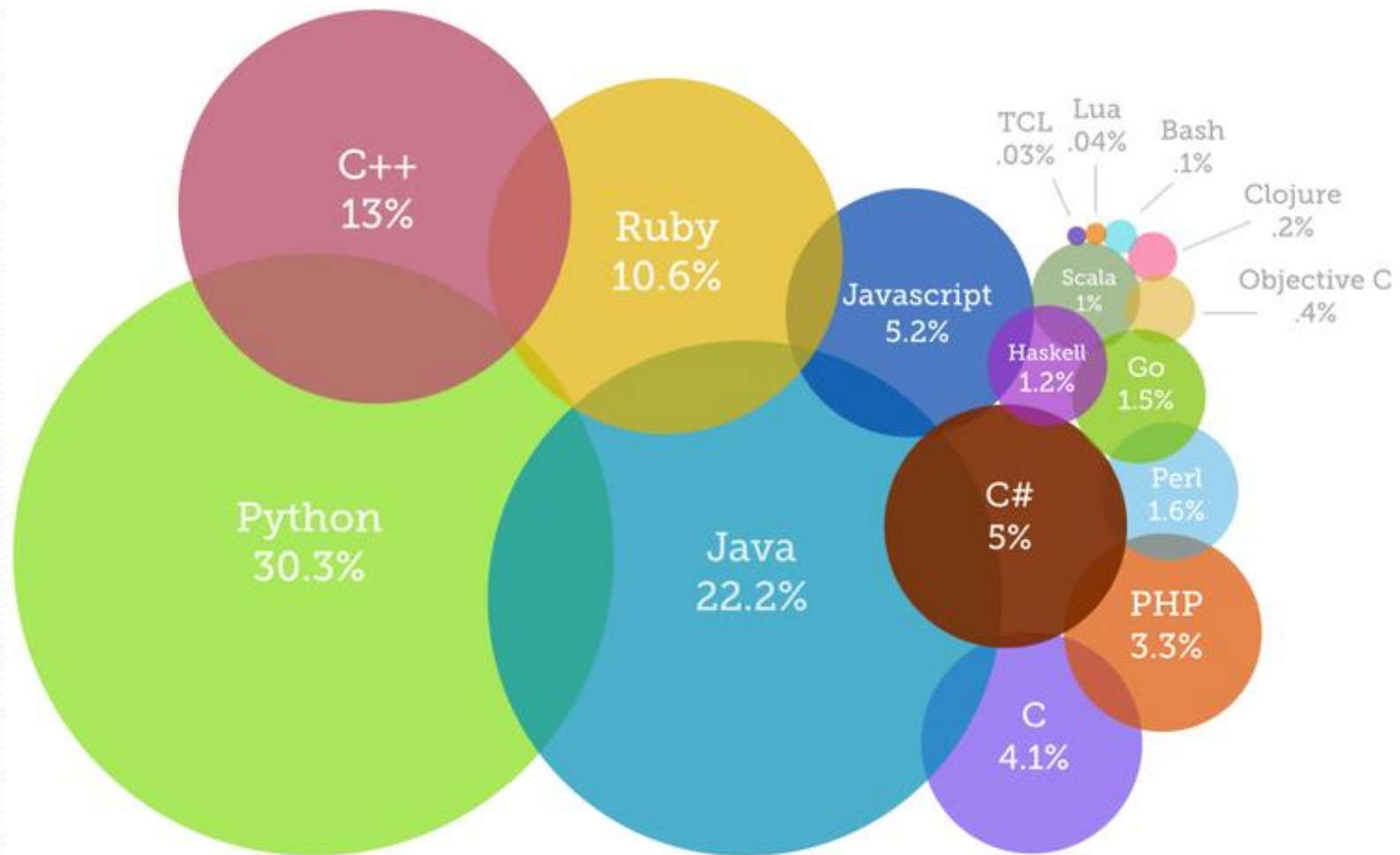
C++

James Gosling



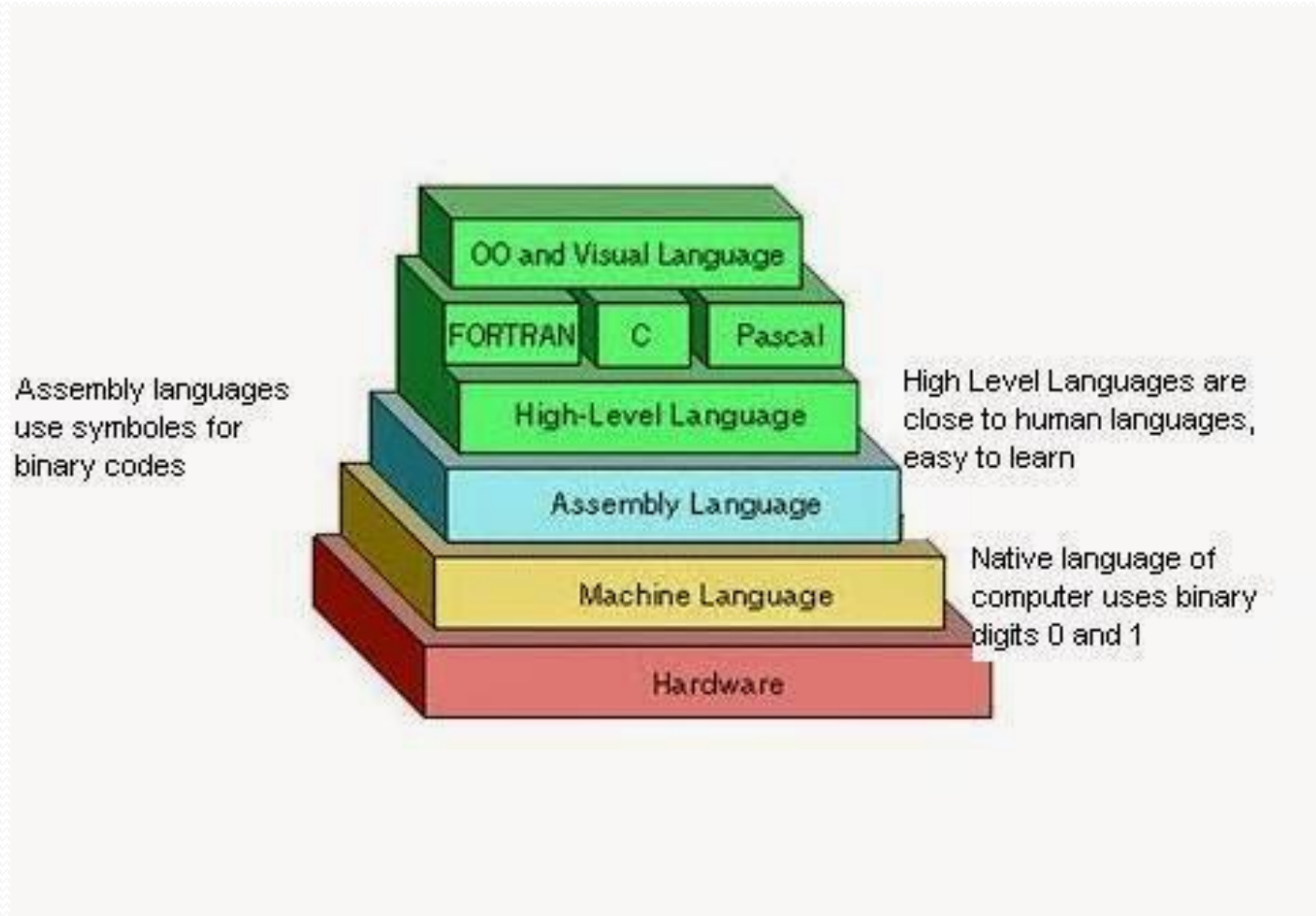
Oak (Java)

## Most Popular Coding Languages of 2014



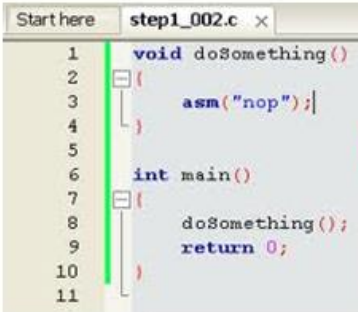


# Tingkatan Bahasa Pemrograman



# Compiler vs Interpreter

- Compiler:
  - Program komputer yang membaca *source code* dan menghasilkan *assembly code* atau *executable code*
  - Program yang menterjemahkan *software* dalam bentuk *source code* menjadi instruksi yang dapat dipahami oleh mesin komputer
  - *Software* yang menerima bahasa pemrograman tingkat tinggi (*high-level language*) dan menterjemahkannya menjadi *assembly code*
- Tahapan pemrosesan sebuah compiler:
  - Lexical Analysis Phase
  - Syntax Analysis
  - Semantic Analysis
  - Intermediate Code Generation
  - Code Optimizer
  - Code Generation



```
1 void doSomething()  
2 {  
3     asm("nop");  
4 }  
5  
6 int main()  
7 {  
8     doSomething();  
9     return 0;  
10 }  
11
```

www.c4learn.com



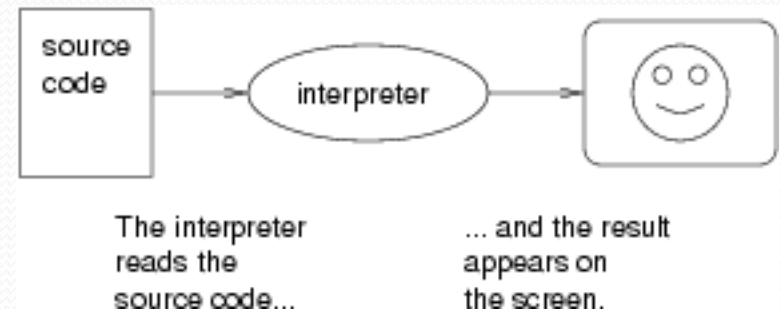
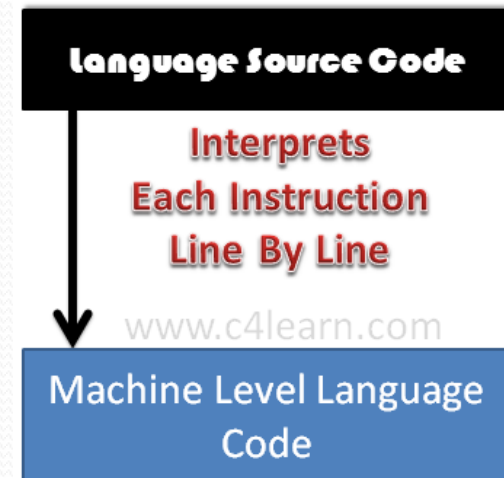
BINARY



# Compiler vs Interpreter

- Interpreter:

- Menerima 1 baris source code/instruksi sebagai input
- Tidak menghasilkan *object code*
- Membutuhkan lebih sedikit memori daripada compiler
- Error ditampilkan untuk setiap instruksi yang diterjemahkan
- Sering digunakan untuk pembelajaran bahasa pemrograman karena memungkinkan siswa untuk memprogram secara interaktif



# Compiler vs Interpreter

[www.technicalprogramming.com](http://www.technicalprogramming.com)

Sr No	Compiler	Interpreter
1	Compiler Takes <b>Entire</b> program as input	Interpreter Takes <b>Single</b> instruction as input .
2	Intermediate Object Code is <b>Generated</b>	<b>No</b> Intermediate Object Code is <b>Generated</b>
3	Conditional Control Statements are Executes <b>faster</b>	Conditional Control Statements are Executes <b>slower</b>
4	<b>Memory Requirement : More</b> (Since Object Code is Generated)	<b>Memory Requirement is Less</b>
5	Program need not be <b>compiled</b> every time	Every time higher level program is converted into lower level program
6	<b>Errors</b> are displayed after <b>entire program</b> is checked	<b>Errors</b> are displayed for <b>every instruction</b> interpreted (if any)
7	<b>Example</b> : C Compiler	<b>Example</b> : BASIC

# Bahasa Pemrograman C

# Pendahuluan

- Bahasa pemrograman C ditulis oleh Dennis Ritchie pada tahun 1973
- C merupakan bahasa pemrograman terstruktur
- C memberikan programmer akses *low level* terhadap memori, sehingga bermanfaat bagi aplikasi yang sebelumnya ditulis dengan bahasa Assembly
- C tersedia dalam berbagai macam *platform*, mulai dari *embedded microcontroller* sampai dengan *supercomputer*

# Persiapan Menjalankan C

- Pilih editor untuk menuliskan *source code*
  - Eclipse ([www.eclipse.org](http://www.eclipse.org)) versi rilis Juno, Kepler, Luna, atau Mars, editor Java yang dapat dikustomisasi untuk menjalankan bahasa lain
  - NewbieIDE ([newbieide.codeplex.com](http://newbieide.codeplex.com) versi 0.1.13), khusus C/C++, simpel, tanpa instalasi
  - Code::Blocks ([www.codeblocks.org](http://www.codeblocks.org) release 13.12), C/C++/Fortran, fitur lebih banyak
  - File program C memiliki ekstensi .c
- Pilih salah satu compiler C:
  - GCC (GNU Compiler Collection: <https://gcc.gnu.org/>)
  - MinGW (Minimalist GNU for Windows: <http://mingw.org/>)
  - Cygwin (Lingkungan pemrograman seperti Unix yang dijalankan di atas Windows <https://www.cygwin.com/>)
  - Lihat tutorial di [https://www3.ntu.edu.sg/home/ehchua/programming/howto/Cygwin\\_HowTo.html](https://www3.ntu.edu.sg/home/ehchua/programming/howto/Cygwin_HowTo.html) untuk cara instalasi MinGW/Cygwin

# Memanggil compiler C dari Eclipse

- Pastikan compiler C dan Eclipse sudah terinstall dengan baik
- Pastikan Anda terhubung dengan internet, karena diperlukan untuk mengupdate Eclipse
- Jika Anda sudah install Eclipse for Java Developers, install C/C++ Development Tool (CDT) dengan cara berikut:
  - klik Help → Install New Software → pada field “Work with”, pilih update sesuai versi Eclipse Anda
  - Di kotak “Name”, klik tanda segitiga untuk mengekspansi menu “Programming Language” → beri tanda cek pada “C/C++ Development Tools” → Next → ... → Finish
  - Langkah di atas dapat dilakukan untuk menjalankan bahasa pemrograman lain di dalam Eclipse (misal: Ruby, Python)
- Jika Anda belum pernah install paket Eclipse apapun, Anda dapat langsung download versi instalasi Eclipse untuk C/C++ di <http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/mars1>
- Tutorial lanjut tentang menjalankan C/C++ di dalam Eclipse dapat dilihat di [https://www3.ntu.edu.sg/home/ehchua/programming/howto/EclipseCpp\\_HowTo.html](https://www3.ntu.edu.sg/home/ehchua/programming/howto/EclipseCpp_HowTo.html)



# Struktur Program C

- Sebuah program C pada dasarnya terdiri atas:
  - Perintah preprocessor
    - Biasanya merupakan perintah untuk memasukkan (*include*) *file header* bawaan standar compiler C. Contoh: `stdio.h` untuk memasukkan fungsi input dan output standar
  - Fungsi
    - Program C terdiri atas fungsi-fungsi yang saling berkomunikasi
  - Variabel
    - Merupakan tempat untuk menyimpan data program
  - Statement dan ekspresi
    - Merupakan pernyataan program
  - Komentar

# Contoh Program C sederhana

```
1 /*
2  Hello.c
3
4  Created on: Aug 25, 2015
5  Author: Intan
6  */
7
8 //ini juga komentar
9 //hanya berlaku 1 baris saja
10
11 #include<stdio.h>
12 int main() {
13     printf("Hello World");
14     return 0;
15 }
```

Komentar program, tidak akan dieksekusi

Perintah preprocessor utk memasukkan file header `stdio.h` sebelum memulai kompilasi  
Fungsi utama tempat eksekusi program dimulai  
Fungsi dalam C untuk menampilkan sesuatu di layar (`printf`)

Mengakhiri eksekusi dan mengembalikan nilai 0, jika tipe *return value* dari fungsi `main` adalah `int`

Tipe *return value* dari fungsi

# 1. Struktur Literal bahasa C

- **Tanda titik koma (;)**
  - Digunakan untuk mengakhiri sebuah statement/baris
  - Tanda ini mengindikasikan berakhirnya satu pernyataan logika
- **Komentar (// atau /\*...\*/)**
  - Komentar digunakan untuk memberikan keterangan tambahan pada baris program, tujuannya dapat untuk menjelaskan bagian program sehingga lebih dipahami oleh orang yang membaca.
  - Segala sesuatu yang ditulis setelah tanda Komentar tidak akan dieksekusi oleh C.
  - Tanda komentar (//) berlaku untuk 1 baris saja
  - Tanda komentar (/\*...\*/) berlaku untuk satu blok program yang diapit oleh kedua tanda tersebut

# Struktur Literal Bahasa C (lanjutan)

- **Identifier**

- Identifier digunakan untuk memberi nama pada variabel, method, dan lain-lain.
- Identifier dapat mengandung: huruf, angka, karakter underscore (\_).
- Penulisan identifier TIDAK dapat dimulai dengan angka.
- Contoh identifier:
  - `i`
  - `x2`
  - `nilai_awal`
  - `_interval`
  - `PI`      `//konstanta`
- C merupakan bahasa pemrograman yang ***case-sensitive***, yakni membedakan antara penulisan huruf kecil dan huruf besar pada identifier. Contoh:
  - `X2` tidak sama dengan `x2`
  - `nilai_awal` tidak sama dengan `Nilai_awal` atau `Nilai_Awal` atau `Nilai_awaL`, atau lain-lainnya

# Struktur Literal Bahasa C (lanjutan)

- Keywords

- Keywords adalah kata-kata yang merupakan bawaan bahasa C dan tidak dapat dijadikan sebagai nama identifier.

- Contoh:

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

## 2. Struktur Sintaks bahasa C

- Unit dasar dari sintaks C adalah ekspresi (expression)
- Ekspresi yang paling sederhana adalah *ekspresi primer* (*primary expression*). Beberapa *keyword* tertentu termasuk dalam ekspresi primer, seperti: `true`, `false`, `null`

```
x;           //Ekspresi primer
x=1;         //Ekspresi penugasan (assignment)
x=x+1;       //Ekspresi dengan dua operator
```



# Struktur Sintaks bahasa C

## (Lanjutan)

- Ekspresi dapat dikombinasikan dengan *keyword* C untuk membentuk sebuah pernyataan (*statement*), seperti statement `if` untuk seleksi kondisi dan statement `while` untuk perulangan (*loop*).

```
if (x<10) {           //if this expression is true
    x=x+1;            //then execute this statement
}                    //marks the end of the conditional

while (x<10) {        //While this expression is true...
    printf("%d",x);   //execute this statement
    x=x+1;            //then execute this statement
}                    //marks the end of the loop
```

# Tipe Data dan Variabel pada C

# Variabel dan tipe data C

- Bilangan bulat → integer

Tipe	Rentang Nilai	Format
byte (unsigned char)	0 ... 255	Unsigned 1 byte
shortint (signed char)	-128 ... 127	Signed 1 byte
word (unsigned int)	0 ... 65535	Unsigned 2 bytes
integer (int)	-32768 ... 32767	Signed 2 bytes
long int	-2177483648 ... 2177483647	Signed 4 bytes

- Operasi aritmetika dan perbandingan pada integer:

Aritmetika		Perbandingan	
x=3+10;	//Hasil=13	3<8;	//Hasil=true (1)
x=87-31;	//Hasil=56	74>101;	//Hasil=false (0)
x=5*10;	//Hasil=50	9<=9;	//Hasil=true (1)
x=5/2;	//Hasil=2	9<9;	//Hasil=false (0)
x=5%2;	//Hasil=1	17==17;	//Hasil=true (1)

# Variabel dan tipe data C

- Bilangan pecahan/floating-point → float

Tipe	Rentang Nilai	Format	Ketepatan (di belakang koma)
float	$1.2 \times 10^{-38} \dots 3.4 \times 10^{38}$	4 bytes	6 angka
double	$2.3 \times 10^{-308} \dots 1.7 \times 10^{308}$	8 bytes	15 angka
long double	$3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$	10 bytes	19 angka

- Operasi aritmetika dan perbandingan pada float:

Aritmetika		Perbandingan	
x=3.2+10.1;	//Hasil=13.300000	3.2<10.1;	//Hasil=true (1)
x=8.0-3.12;	//Hasil=13.120000	7.4>10.1;	//Hasil=false (0)
x=5.0*10.0;	//Hasil=50.000000	9.0<=9.0;	//Hasil=true (1)
x=10.0/3;	//Hasil=3.333333	9.1<9.0;	//Hasil=false (0)

# Variabel dan tipe data C

- Karakter → char
  - Semua huruf abjad kecil dan besar, semua tanda baca, angka '0', '1',..., '9', karakter khusus ('&', '^', '%', '#', dll)
  - Penulisannya diapit dengan petik tunggal atau *single quote* ('')
  - Karakter kosong (*null*): karakter dengan panjang nol dan dilambangkan dengan ""
  - Operasi perbandingan pada karakter:

Perbandingan	
'a' == 'a'	//Hasil=true (1)
'T' == 't';	//Hasil=false (0)
'y' != 'Y';	//Hasil=true (1)
'm' < 'z';	//Hasil=true (1)

# Variabel dan tipe data C

- Sekumpulan karakter akan membentuk sebuah string
- Bahasa C tidak memiliki tipe khusus untuk menampung sebuah string
- String dapat disimpan dalam *array of characters*
- Penulisan sebuah string diapit oleh petik ganda atau *double quotes* (“”)
- Contoh:
  - `char kota[] = "Surabaya";`
- Operasi pada string meliputi operasi penyambungan (*concatenation*) dan perbandingan
  - Contoh *concatenation*: “kota” + “Surabaya” = “kotaSurabaya”
  - Contoh perbandingan:
    - “abcd” > “acbd” → Hasil: false (0)
    - “aku” < “AKU” → Hasil: true (1)



# Variabel dan tipe data C

- Tipe data Boolean (`true`, `false`)
  - C tidak memiliki tipe data boolean
  - Kita dapat menggantinya dengan tipe `int`, misal `true=1` dan `false=0`

# Penulisan dan Pengisian Nilai pada C

# 1. Penulisan Nilai ke Layar

- Untuk mengeluarkan output ke layar digunakan perintah `printf` (dengan include `stdio.h`)
- Pada bahasa C, penulisan output ke layar dari nilai sebuah variabel disertai dengan menentukan format dari output yang akan ditulis (jika *include library* yang digunakan adalah `stdio.h`)
- Contoh penggunaan `printf`:

```
#include <stdio.h>
int main() {

    int umur=20;
    printf("Hello World\n");
    printf("Isi variabel umur: %d",umur);
    return 0;
}
```

## 2. Pengisian Nilai ke Variabel

- Pengisian secara langsung (*assignment*):
  - `int x=10;`
  - `char abjad='a';`
  - `char kata[]="alfabet";`
- Pembacaan dari input user (keyboard):
  - Pada bahasa C, pembacaan input dari keyboard disertai dengan menentukan format dari input yang akan dibaca (jika *include library* yang digunakan adalah `stdio.h`)

Tipe data	Penentu format
<code>int</code>	<code>%d</code>
<code>unsigned int</code>	<code>%u</code>
<code>long int</code>	<code>%ld</code>
<code>long unsigned int</code>	<code>%lu</code>
<code>float</code>	<code>%f</code>
<code>double</code>	<code>%lf</code>
<code>char</code>	<code>%c</code>
<code>char[n]</code>	<code>%s</code>

Untuk membaca input dari keyboard digunakan perintah `scanf` (dengan `include stdio.h`)

## 2. Pengisian Nilai ke Variabel

- Contoh penggunaan scanf:

```
#include <stdio.h>
int main() {
    int umur;
    char nama[10];
    printf("Masukkan nama Anda:");
    scanf("%s",nama);
    printf("Masukkan umur Anda:");
    scanf("%d",&umur);
    printf("Nama Anda dan Umur Anda: %s %d",nama,umur);
    return 0;
}
```

## 2. Pengisian Nilai ke Variabel

- Untuk membaca sebuah nilai dan menyimpan ke sebuah variabel perlu diberi awalan “&”
- Khusus untuk variabel bertipe string, perintah `scanf` tidak memerlukan awalan “&”
- `scanf` tidak dapat membaca string yang mengandung spasi. Untuk kasus tersebut, gunakan perintah `gets`
- Simbol `\n` menyatakan ganti baris (*new line*)



## 2. Pengisian Nilai ke Variabel dengan perintah lain

Perintah	Keterangan
<code>getche()</code>	<ul style="list-style-type: none"><li>• membaca data karakter diikuti penekanan tombol Enter</li><li>• <code>include &lt;conio.h&gt;</code></li><li>• <b>Contoh:</b> <code>char x=getche();</code></li></ul>
<code>getch()</code>	<ul style="list-style-type: none"><li>• membaca data karakter tanpa penekanan tombol Enter</li><li>• <code>include &lt;conio.h&gt;</code></li><li>• <b>Contoh:</b> <code>char x=getch();</code></li></ul>
<code>gets()</code>	<ul style="list-style-type: none"><li>• membaca string (dengan spasi) dari keyboard</li><li>• <code>include &lt;stdio.h&gt;</code></li><li>• <b>Contoh membaca input dari keyboard dan menyimpan ke variabel alamat:</b><ul style="list-style-type: none"><li>• <code>gets(alamat);</code></li></ul></li></ul>

# 3. Ekspresi

- Ekspresi Aritmetik

Benar	Salah	Keterangan
<pre>int x=4; int y=x+3; //output:7</pre>	<pre>float x=4.5; int y=x+3.4; //output:7</pre>	Tipe data yang dioperasikan harus sama

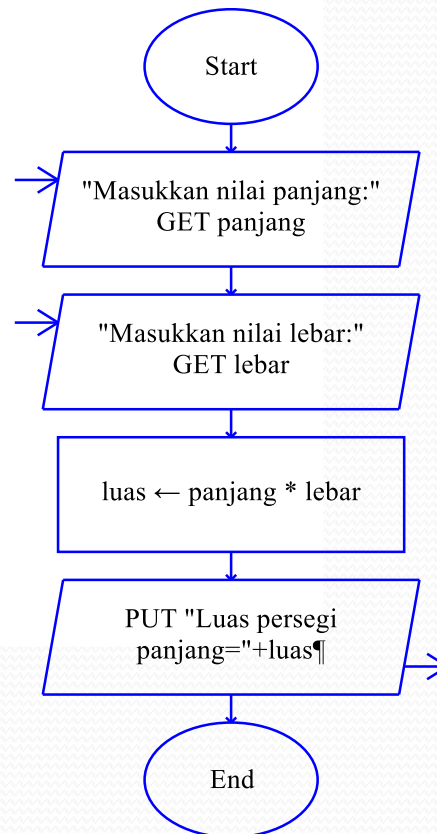
- Ekspresi Relasional

- Contoh:

```
int ada=1;  
int ketemu=0;  
int x=8;  
int y=12;  
not ada;           //hasil: 0 (false)  
ada or ketemu;     //hasil: 1 (true)  
ada and true;      //hasil: 0 (false)  
x<5;               //hasil: 0 (false)  
ada or (x==y);     //hasil: 1 (true)
```

# Contoh Soal Pemrograman Sederhana

- Buat program dari flowchart berikut:



# Solusi

```
1  #include <stdio.h>
2
3  int main() {
4      int p,l;
5      printf("PROGRAM MENGHITUNG LUAS PERSEGI PANJANG\n");
6      printf("Masukkan nilai panjang:");
7      scanf("%d",&p);
8      printf("Masukkan nilai lebar:");
9      scanf("%d",&l);
10     printf("\nLuasnya=%d",p*l);
11     return 0;
12 }
```

Contoh Output:

```
PROGRAM MENGHITUNG LUAS PERSEGI PANJANG
Masukkan nilai panjang:4
Masukkan nilai lebar:2

Luasnya=8
```

# Latihan

- Buatlah program dengan memasukkan nilai tes mata kuliah kemudian menampilkan hasil nilai rata-rata. Misal diasumsikan ada 3 mata kuliah yang diambil oleh mahasiswa (Catatan: Anda belum perlu menggunakan array).
- Buatlah program untuk menghitung jarak peluru yang ditembakkan dari suatu lokasi dengan jarak awal ( $S_0$ ), kecepatan awal ( $v_0$ ), dan waktu ( $t$ ) dimasukkan dari keyboard. Jarak yang diperoleh ditampilkan ke layar monitor. (Rumus  $S = S_0 + v_0.t + \frac{1}{2}.a.t^2$ )
- Buatlah program untuk menghitung luas dan keliling lingkaran kemudian ditampilkan ke layar monitor.
- Tulislah program sekuensial untuk menghitung sisi miring, sinus, cosinus, dan tangent dari suatu segitiga dengan masukan dari pengguna berupa panjang alas dan tingginya.
- Buatlah program untuk menghitung Luas dan Keliling Segi Tiga, Bujur Sangkar dan Persegi Panjang.
- Buatlah program untuk menghitung Volume dan Luas Permukaan Kubus, Limas dan Prisma.
- Buatlah program untuk menghitung  $Y = 5X^4 + 3X^3 - X^2 + 7$ .
- Buatlah program untuk menghitung  $D = 2A^4 - 6B^3 + 8C^2 - 10$ .
- Buatlah program untuk menghitung  $Z = \sin 2X + 3.\cos 3X - 5.\tan 4X$ .