

# Algoritma & Pemrograman

Pertemuan ke 2

# Program Memasak → (Bahasa Manusia)

- Iris bawang bombay dan bawang putih kecil-kecil
- Potong potong semua bahan sayuran.
- Siapkan pengorengan dan masukkan mentega dengan nyala api kompor kecil hingga sedang.
- Tumis bawang bombay, bawang putih, hingga harum
- Tambahkan sedikit air sekitar 200 ml.
- Tunggu hingga mendidih, selanjutnya masukkan bahan sayurannya dan tunggu hingga matang, lantas masukkan juga irisan bakso sapi
- ... dst

# ... Program (Memasak)

- Rangkaian **LAMBANG** (Huruf dan Angka),
  - Disusun membentuk **PERINTAH** (Kata Kerja)
  - Dengan **URUTAN** tertentu
  - Untuk mengolah **SESUATU** (Kata Benda)
  - Yang **DITUJUKAN** ke seseorang (Koki)
  - Dengan **MEMANFAATKAN RESOURCE** yang dipunyai (Kompor, Panci, Penggorengan, Ketrampilan, dsb)
  - Untuk suatu **TUJUAN** (Memasak)
- Ditulis dalam bahasa manusia

# Program Bermain Musik → Bahasa Musik

The image shows the first page of a piano sheet music arrangement. The title at the top is "SYMPHONY No. 9 in D Minor Op. 125 Choral". Below the title, it says "Concert Piano Arrangement by F. Liszt". The composer of the original work is listed as "L. van Beethoven". The musical score consists of two staves. The upper staff is in treble clef and the lower staff is in bass clef. The tempo marking "Allegro ma non troppo, un poco maestoso" and the dynamic "pp" (pianissimo) are indicated. The key signature is one flat. The music features eighth-note patterns and sixteenth-note patterns. The bass staff includes a dynamic marking "sotto voce" and a tempo marking "92d.". The bottom staff continues the musical line with eighth-note patterns and a dynamic marking "sempre pp".

# ... Program (Main Musik)

- Rangkaian **LAMBANG** (Notasi Not Balok),
  - Disusun membentuk **PERINTAH** (Mengeluarkan Bunyi)
  - Dengan **URUTAN** tertentu
  - Untuk mengolah **SESUATU** (Partitur)
  - Yang **DITUJUKAN** ke seseorang (Pemusik/Pianis)
  - Dengan **MEMANFAATKAN RESOURCE** yang dipunyai (untuk Piano dengan Tuts, Pedal, dsb)
  - Untuk suatu **TUJUAN** (Main Musik)
- Ditulis dalam bahasa music (NOTE BALOK)

# Program Menulis Hello World → Bahasa Mesin

# ... Program (Hello, World...)

- Rangkaian **LAMBANG** (BINER atau HEKSADESIMAL untuk computer),
- Disusun membentuk **PERINTAH** (STATEMENT)
- Dengan **URUTAN** tertentu (ALGORITMA)
- Untuk mengolah **SESESUATU** (DATA)
- **DITUJUKAN** ke seseorang mesin (KOMPUTER)
- Dengan **MEMANFAATKAN RESOURCE** yang dipunya (CPU, RAM, Monitor, dsb)
- Untuk suatu **TUJUAN** (Menulis Hello, World...)

Ditulis dalam bahasa Pemrograman, yang kemudian diubah ke dalam bahsa mesin (biner)

# Sistem / Lambang Bilangan

Desimal	Biner	Oktal	Heksa
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8

Desimal	Biner	Oktal	Heksa
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	1 0000	20	10
17	1 0001	21	11

# Bilangan Biner (Digital)

- Hanya terdiri dari angka 0 & 1
- Satu simbol biner biasa disebut bit
  - Bilangan 1 bit → 0 atau 1
  - Bilangan 2 bit → 00 atau 01 atau 10 atau 11
  - Bilangan 3 bit → 000, 001, 010, 011 ... 111
- Bilangan 8 bit 0000 0000 ..s/d.. 1111 1111 → Byte
- Bentuk Heksa      0      0 ..s/d..      F      F

Catatan : bit → b kecil, Byte → B besar

# Lambang Manusia

- BUKAN HANYA Integer / Bilangan bulat positif (Desimal 0 s/d 9)
- TETAPI JUGA
  - bilangan Negatif contoh : -3  
Jadi lambang negatif (-) juga harus diterjemahkan ke dalam sistem biner
  - Bilangan Pecahan contoh : 3,14  
Jadi lambang koma (,) juga harus diterjemahkan ke dalam sistem biner
  - Huruf dan tanda baca yang juga harus ditermahkan ke dalam sistem biner / digital.  
Bahkan ada gambar, video, audio dsb.
- JUGA ADA PERINTAH (dalam bahasa Inggris) contoh : print, add, load, dsb

# Bahasa Manusia vs Mesin

## → Butuh Penterjemah

### Manusia

- Huruf : ABCDEFG ... XYZ
- Angka : 01..9
- TandaBaca : .,+,<spasi>!?%
  
- Kalimat :
  - Tumis Bawang
  - Tambahkan 2 Sedok Gula

### Komputer

- Biner : 01
- Heksa : 0123456789ABCDEF
  
- Kalimat :
  - 0101011000111010101011
  - 01Afd2345DE7800315FBA3

# Tipe Data Huruf / Karakter

- Unicode (16 bit) → Java, C
- ASCII (7 bit) → Pascal, C

Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT End of transmission	0x24	36	\$	0x44	68	D	0x64	100	d
0x05	5	ENQ Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1 Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2 Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3 Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4 Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS File separator	0x3C	60	<	0x5C	92	\	0x7C	124	
0x1D	29	GS Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	-
0x1F	31	US Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

# Interpreter vs Compiler

- Interpreter / Compiler adalah program aplikasi yang bertugas “menterjemahkan” bahasa level tinggi (Basic, C, Pascal, dll) ke bahasa mesin
- Interpreter :
  - menerjemahkan perintah bahasa tingkat tinggi satu (baris) per satu (baris) sesuai dengan alur perintah
  - Untuk menjalankan aplikasi interpreter harus jalan
  - Kekurangan Lambat
  - Contoh: Basic

# Interpreter vs Compiler

- Compiler:
  - Menterjemahkan seluruh source code bahasa level tinggi ke dalam bahasa mesin.
  - Bahasa mesin yang dihasilkan sesuai dengan platform hardware / system operasi yang dituju
  - Hasil terjemahan langsung menjadi aplikasi dan bisa dijalankan / dieksekusi oleh sistem operasi  
Pada Linux ditandai dengan atribut “Execute” atau pada Windows file berekstensi \*.exe
  - Contoh: Pascal, C, Assembly

# Java

- Java adalah interpreter
- Untuk mempercepat proses dari source code (\*.java) dicompile dengan java-compiler (jdk → **javac**) dulu menjadi bytecode (\*.class)
- Setelah menjadi bytecode (\*.class), maka bisa di jalankan oleh jre → **java**
- Keuntungan : Cross Platform → Selama ada java (jre) maka aplikasi bisa berjalan
- Catatan : Sun/Oracle sebagai menyediakan jre yang bisa jalan di berbagai platform sistem operasi.  
Bahkan lebih dari itu jre tersedia pula dalam aplikasi browser.

# Pemrograman

- Aplikasi dapat dibuat dengan cara membuat program komputer
- Program Komputer : adalah rangkaian perintah kepada komputer untuk menyelesaikan suatu masalah
  - dengan suatu bahasa programan tertentu (Basic, Pascal, C, Java, dll)
  - yang kemudian diterjemahkan ke dalam bahasa mesin
  - Selanjutkan akan dikerjakan oleh komputer dengan sumber daya yang dimiliki I/O, Memory dan CPU (ALU, Reigster, Controll Unit)
  - Pada program yang dibuat dapat memberikan perintah secara langsung ke Hardware, atau memanfaatkan fasilitas / Library yang disiapkan oleh BIOS ata Sistem Operasi

# Bahasa Pemrograman

**Jadi yang penting adalah RANGKAI PERINTAH (Algoritma) yang diberikan untuk menyelesaikan suatu masalah.**

- Bahasa Pemrograman hanya sebuah tools untuk membantu menterjemahkan → Bahas Mesin
- Tiap Bahasa (Basic, Pascal, C, Java, dll) mempunyai keunikan
  - Struktur → (Object Oriented Program)
  - Cara Penulisan / Sintaks
  - Reserved Word / Kosa kata
  - Type Data (Primitive)
  - Library / collection Perintah / Code
  - dsb

# Perintah Dasar

- Input
- Data Type → Variabel (Data Storage)
  - Numerical
  - Character / String
  - Logic
- Data Manipulation (Numerical, Character, Logic)
- Branching (IF ... Else)
- Looping (For, Do-While, While-Do)
- Output
- Sub Rute / Function / Object

# Contoh Program ...

**TUGAS → Cari persamaan dan perbedaan dari program dengan bahasa Basic, Pascal, C dan Java**

- **BASIC :**

```
10 PRINT "Hello, World..."
```

- **Python :**

```
print "Hello, World..."
```

- **Pascal :**

```
program Hello;
begin
  writeln ('Hello, world...');
end.
```

# Contoh Program ...

- **C :**

```
#include<stdio.h>
int main(int argc, char** argv) {
    printf("Hello, World...\n");
}
```

- **Java :**

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, World...");
    }
}
```

# Case Sensitive

- Pada beberapa bahasa huruf besar dan huruf kecil mempunya arti yang berbeda
- Contoh case sensitive
  - Agung != AGUNG != agung
- Contoh case in-sensitive
  - Agung == AGUNG == agung
- **Java, C → case sensitive**
- **Basic, Pascal → case in-sensitive**

# Warning & Error

- **Warning** : Peringatan yang dapat diabaikan
- **Error** : Kesalahan yan perlu diperbaiki agar aplikasi dapat jalan dengan baik
  - **Sintax** : Biasanya kesalahan ketik perintah / tanda baca. Diketahui pada saat di-compile / di-edit dengan ide
  - **Run Time** : Diketahui paa saat program berjalan. Contohnya ada pembagian dengan nilai 0 (nol)
  - **Logic / Semantik** : Kesalahan pemrogram yang memberikan perintah dengan logika yang terbalik-balik.

# Data Type

- Data biner 8 bit : 0101 0011 ← Apa ???
- **Numeris**
  - Bilangan **cacah** → (sign / unsigned) (short / long) **Integer**
  - Bilangan **nyata** → real, float, double
  - Operasi Matematika : + - / \*
- **Huruf**
  - ASCII (8 bit)
  - Unicode (16 bit)
  - Operasi Huruf/Kata : Gabung, Mirip, Selsih
- **Boolean**
  - 2 Keadaan → True (1) atau False (0)
  - Operasi Logika : AND, OR, NOT

# Tipe Data Numeris

C	Java	Bit	Range
(signed) <b>char</b>	<b>byte</b>	8	-128 ... +127
<b>unsigned char</b>	-	8	0 ... 255
(signed) <b>short</b> (int)	-	16	-32.768 ... +32.767
<b>unsigned short</b>	-	16	0 ... 65.535
(signed) <b>int</b>	<b>int</b>	32	-2,147,483,648 ... +2,147,483,647
<b>unsigned int</b>	-	32	0 ... 4.294.967.295
<b>float</b>	<b>float</b>	32	1e-37 ... 1e37
<b>double</b>	<b>double</b>	64	
<b>long double</b>	-	128	