



Pengantar Kompilasi

Wakhid Kurniawan, S.Kom., M.Kom.

Program Studi S1 Informatika

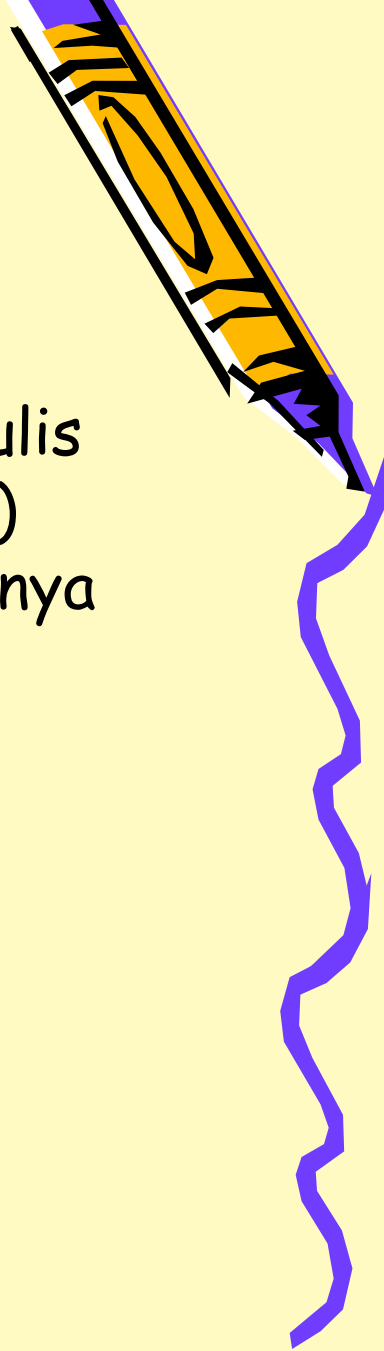
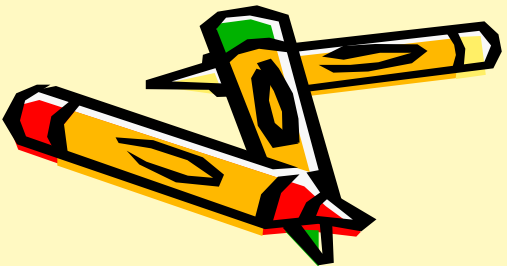
Fakultas Sains dan Teknologi

Universitas Muhammadiyah Karanganyar



1. Apa itu Kompilasi?

- **Kompilasi** adalah proses menerjemahkan kode program yang ditulis dalam bahasa pemrograman tingkat tinggi (High-Level Language) menjadi bentuk lain yang dapat dieksekusi oleh komputer (biasanya kode mesin).
- Perangkat lunak yang melakukan proses ini disebut **Compiler**.



2. Compiler

Pengertian:

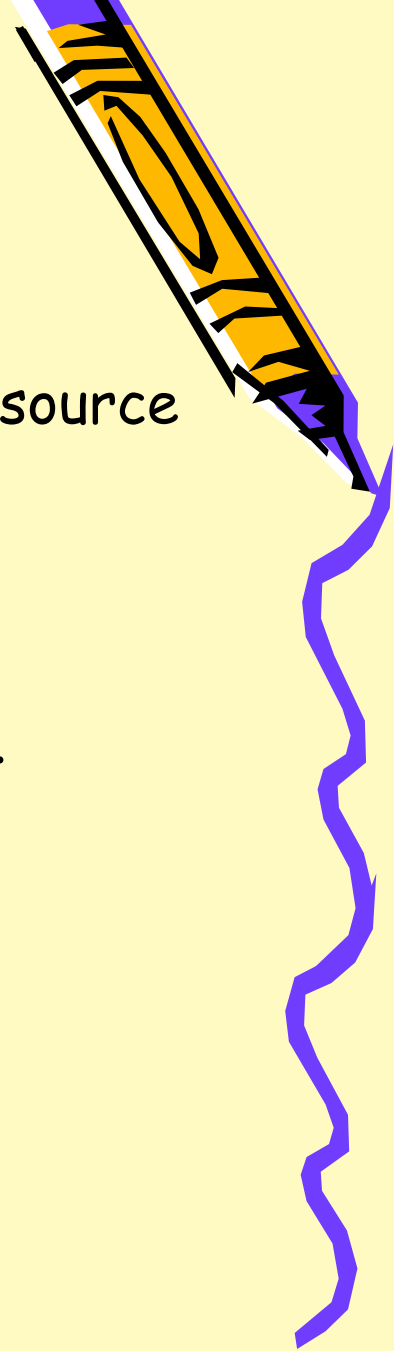
- **Compiler** adalah program yang menerjemahkan seluruh kode sumber (source code) ke dalam bahasa mesin atau bahasa objek sebelum dijalankan.

Karakteristik:

- Menerjemahkan **sekali** seluruh program.
- Hasil terjemahan berupa **file executable**.
- Waktu kompilasi biasanya lama, tetapi waktu eksekusi program cepat.

Contoh:

- GCC (GNU Compiler Collection) untuk C/C++
- Javac untuk Java
- Go compiler (golang)



3. Interpreter

Pengertian:

- **Interpreter** adalah program yang membaca dan mengeksekusi kode sumber baris demi baris secara langsung tanpa menghasilkan file executable terpisah.

Karakteristik:

- Tidak menghasilkan file executable.
- Eksekusi bisa langsung, cocok untuk debugging.
- Waktu eksekusi lebih lambat dibanding compiler karena penerjemahan dilakukan saat runtime.

Contoh:

- Python Interpreter
- Ruby Interpreter
- PHP Engine



4. Assembler

Pengertian:

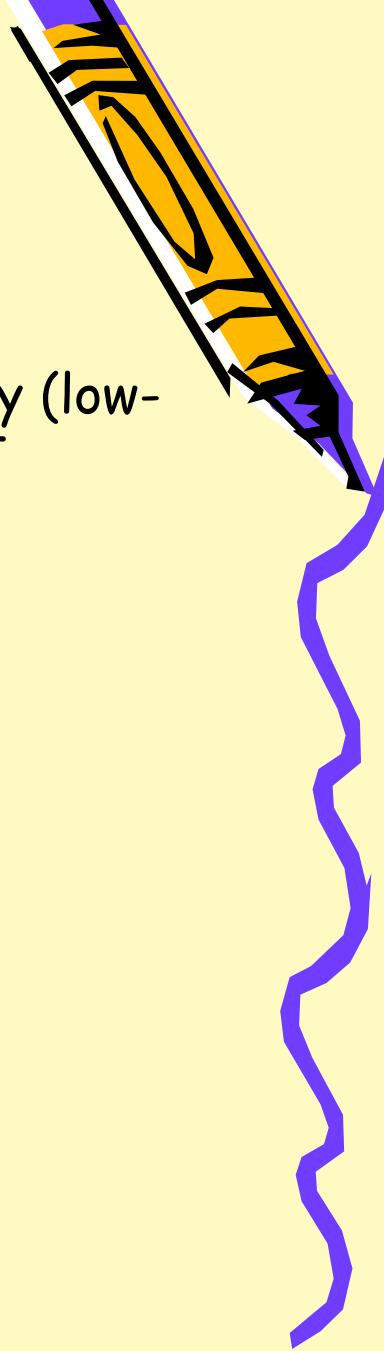
- **Assembler** adalah program yang menerjemahkan kode dalam bahasa assembly (low-level language, dekat dengan instruksi mesin) menjadi kode mesin yang dapat langsung dijalankan oleh komputer.

Karakteristik:

- Input: Kode assembly (misalnya `MOV AX, BX`).
- Output: Kode mesin (biner).
- Digunakan dalam pemrograman sistem, driver, atau perangkat keras.

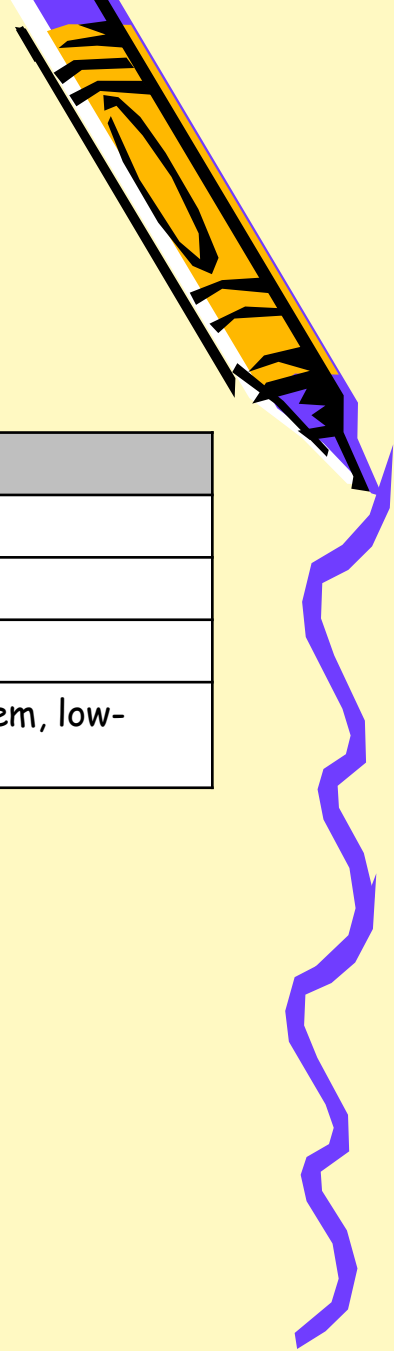
Contoh:

- NASM (Netwide Assembler)
- MASM (Microsoft Assembler)
- GAS (GNU Assembler)

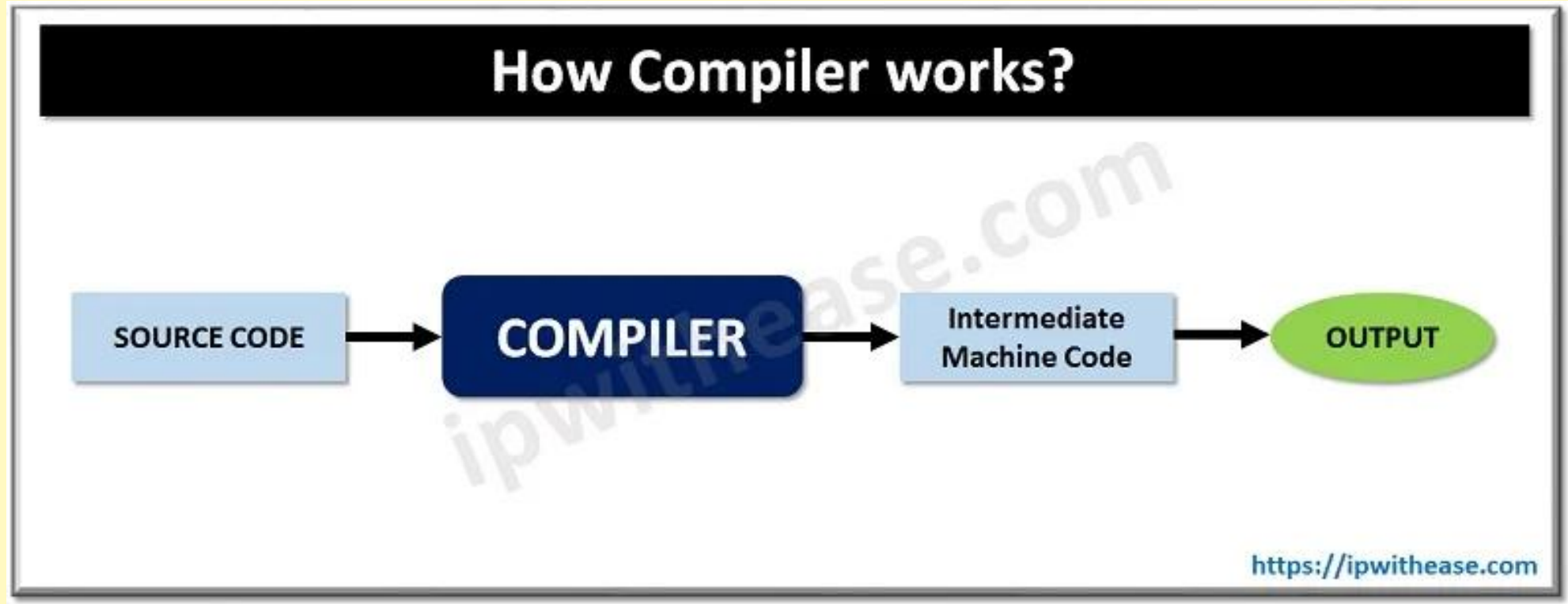


5. Perbandingan Compiler, Interpreter, Assembler

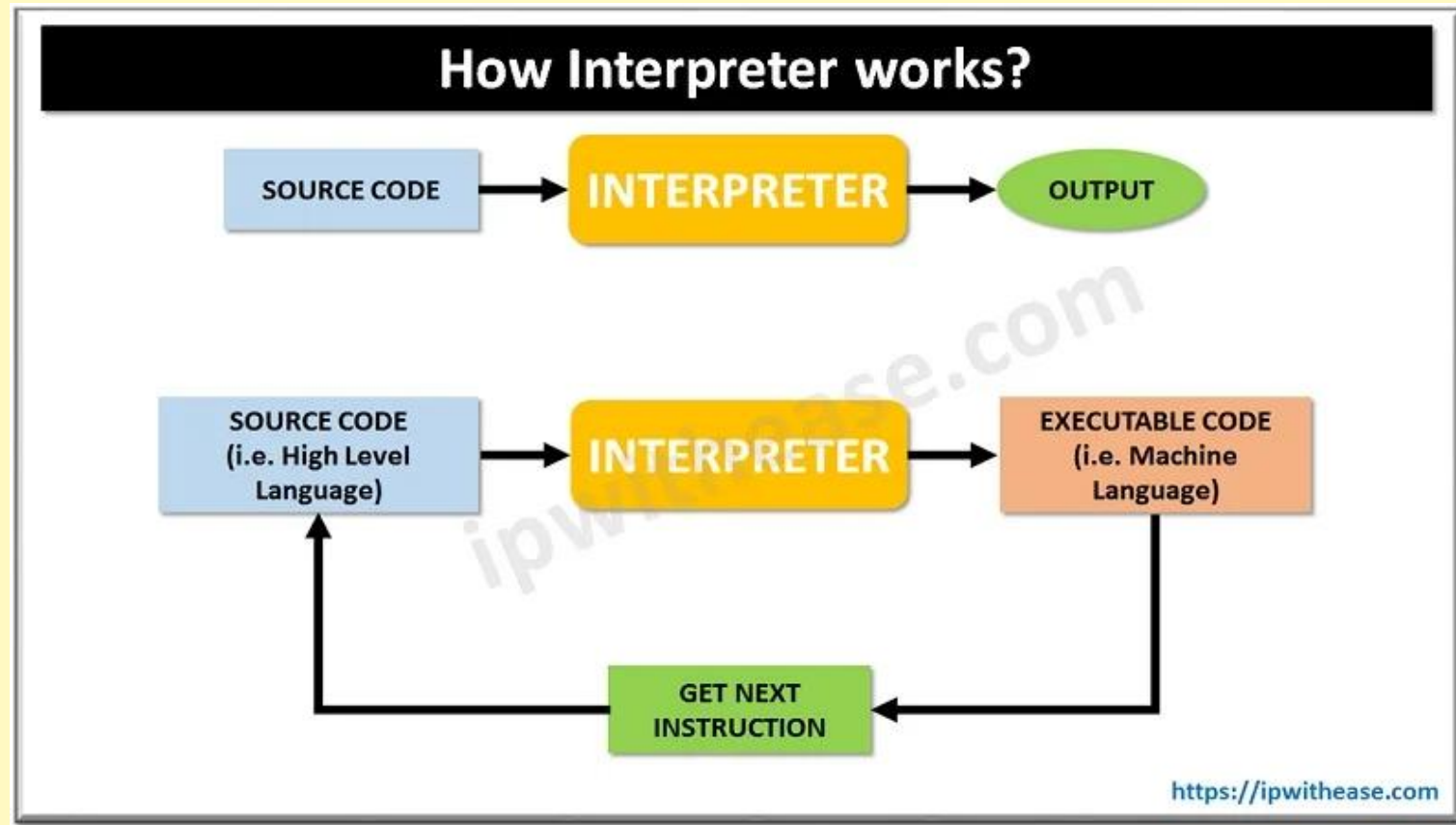
Aspek	Compiler	Interpreter	Assembler
Input	Bahasa tingkat tinggi	Bahasa tingkat tinggi	Bahasa Assembly
Output	File executable (objek/kode mesin)	Eksekusi langsung	Kode mesin
Eksekusi	Cepat (setelah kompilasi)	Lambat (eksekusi baris demi baris)	Cepat
Penggunaan	Aplikasi besar (C, C++)	Scripting, debugging (Python, PHP)	Pemrograman sistem, low-level



5. Perbandingan Compiler, Interpreter, Assembler...



5. Perbandingan Compiler, Interpreter, Assembler...



5. Perbandingan Compiler, Interpreter, Assembler...

How Assembler works?



<https://ipwithease.com>

6. Ilustrasi Proses

Kode Tingkat Tinggi (C, Java, Python)

|

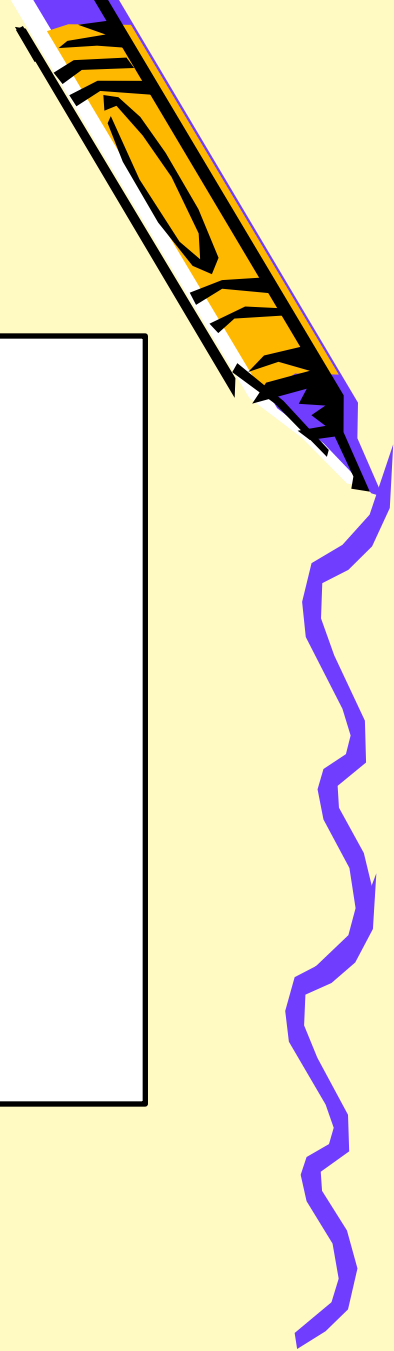
| -- Compiler --> Kode Mesin (Executable)

|

| -- Interpreter --> Eksekusi Langsung

|

| -- Assembler (dari Assembly) --> Kode Mesin

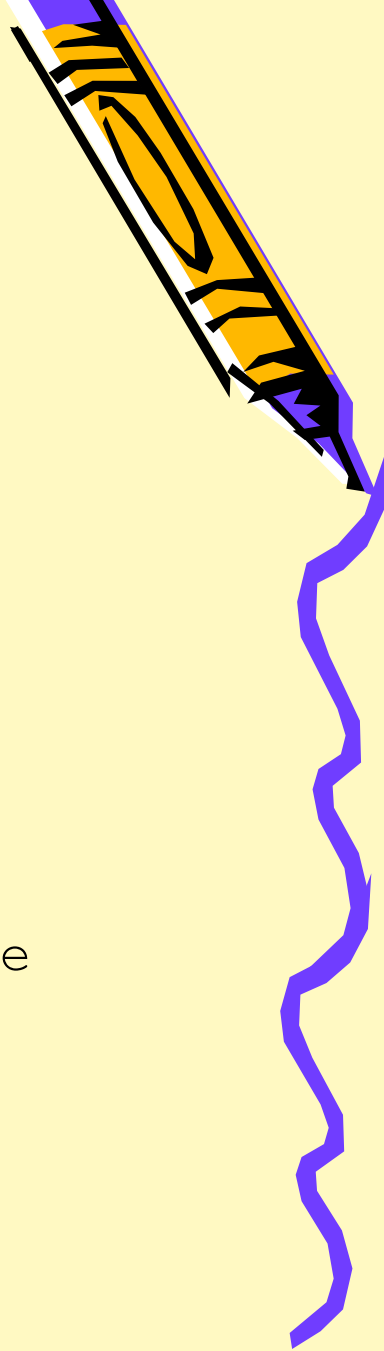


7. Contoh Singkat

- Source Code C (compiler):

```
#include <stdio.h>
int main() {
    printf("Hello World\n");
    return 0;
}
```

Diterjemahkan dengan `gcc hello.c -o hello` → menghasilkan `hello.exe`

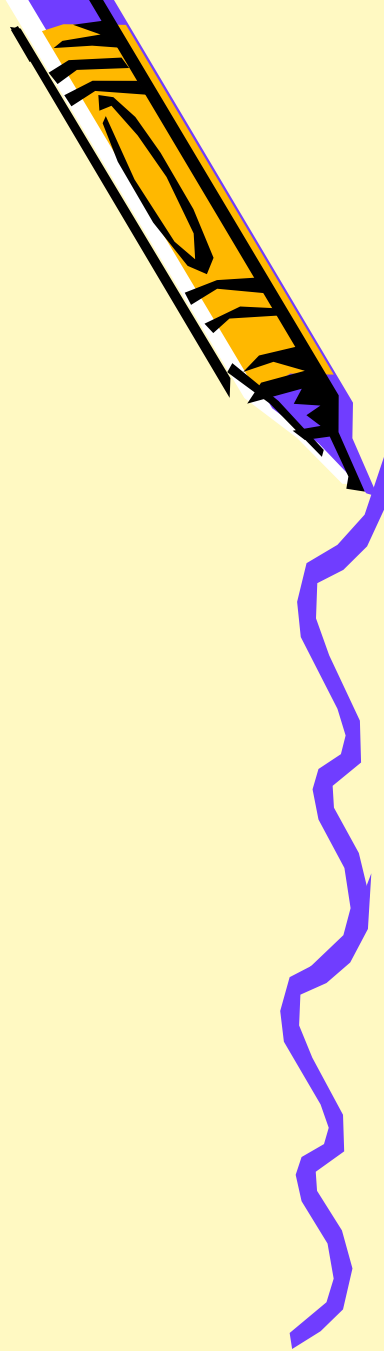


7. Contoh Singkat

Python (interpreter):

```
print("Hello World")
```

Jalankan langsung dengan `python hello.py`



7. Contoh Singkat

- **Assembly (assembler):**

```
section .data  
msg db "Hello World",0Ah
```

```
section .text  
global _start  
_start:  
    mov edx, 12  
    mov ecx, msg  
    mov ebx, 1  
    mov eax, 4  
    int 0x80  
    mov eax, 1  
    int 0x80
```

Diterjemahkan dengan `nasm -f elf hello.asm` →
menghasilkan kode mesin.

