

C Guide

Dokumen ini dibuat sebagai *guide* secara umum untuk menjalankan bahasa C, soalnya banyak yang masih bingung dalam menjalankan program berbahasa C. Detail teknis lebih dalamnya nanti bakal kalian pelajari di mata kuliah lain (IF2130 Organisasi dan Arsitektur Komputer).

Declaration vs Definition

Di bahasa C, ada istilah declaration dan definition.

Declaration itu intinya mendeklarasikan identifier + signature-nya (tipe). Bisa buat variabel, tipe/struct, maupun fungsi. Kayak gini contohnya.

```
// tipe
typedef int EType;
typedef struct {
    EType *buffer;
    int nEff;
    int capacity;
} ListDin;

// variabel
char word;

// fungsi
int add(int, int);
```

Definition itu inisiasi/implementasi identifier tersebut. Misalnya:

```
// variabel
word = 'a';

// fungsi
int add(int a, int b) {
    return a + b;
}
```

Dalam bahasa C, duplicate declaration diperbolehkan. Tetapi **duplicate definition tidak diperbolehkan**.

Standarnya, declaration ditaro di header file (.h), dan definition ditaro di c file (.c). Dua-duanya sama-sama bahasa C kok, konvensi aja. Mungkin kalian ada pertanyaan, kalo sama-sama bahasa C ngapain dipisah? Kenapa ga include .c aja? Bakal kita bahas di [Include vs Linking](#).

Include vs Linking

Include itu “nyalin” kode dari file 1 ke file yang sekarang. Jadi misalnya ada file-file ini:

```
tes.h
```

```
int tesAja(int);
```

```
tes.c
```

```
int tesAja(int awo) {  
    return awo + 1;  
}
```

```
main.c
```

```
#include "tes.h"  
  
int main() {  
    tesAja(1);  
    return 0;  
}
```

Sebenarnya, yang dilihat komputer, main.c itu kayak gini:

```
main.c
```

```
int tesAja(int);  
  
int main() {  
    tesAja(1);  
    return 0;  
}
```

Nah dari main.c itu, kan belum ada definisi fungsi tesAja(). Gimana cara biar dia tau definisinya? Pake linking. Jadi ketika dicompile, selain memanggil main.c, tes.c juga harus dipanggil.

Termasuk yang sering muncul juga di praktikum, kalian include <math.h>. Nah makannya pas dicompile harus pake -lm untuk link ke library math.

Konvensinya yaitu yang diinclude itu header yang isinya deklarasi, yang di-link itu file .c yang isinya definisi.

Yaudah kenapa ga include .c aja?

Include .c (yang isinya definisi) bakal jadi masalah kalo misalnya ada lebih dari 1 file yang meng-include file yang sama. Kalo gini, pas kalian compile bakal kena duplicate definition error. Makannya ada konvensi kayak di atas. Dan konvensi kayak gini juga kita aplikasikan di praktikum.

How to Run C using Terminal

Buat bisa nge-run kode C, pertama harus di-compile dulu jadi executable. Baru kita jalanin executable-nya kek jalanin program biasanya.

Compile

Untuk compile C kita pake [GCC](#). GCC ini udah bawaan kalo kita pake OS Linux distribution (termasuk Ubuntu yang dipake di praktikum). Sebenarnya parameternya banyak, tapi kira-kira yang kalian butuhin sih ini aja.

```
$ gcc \  
    -o <nama file executable>  
    [<file .c/library yang ingin di-link>]
```

Kalo kalian gagal compile, itu namanya compile error. Buat debug-nya, perhatikan error messagenya baik-baik. Pastiin juga kalian udah nge-include dan nge-link dengan benar.

Run Executable

Tinggal panggil nama file executablenya aja di terminal.

```
$ ./<nama file executable>
```

Done! Program jalan seperti biasa, kayak pas kalian pake Python.

Kalo habis jalanin kalian ketemu error, kita sebut itu runtime error. Runtime error ini kemungkinan besar terjadi karena kesalahan logic di kode kalian. Jadi, misalnya, kalian ketemu segmentation fault error (salah satu runtime error), coba cek lagi kode kalian, ada yang ngakses memory haram apa ngga.

Tips

- Mulai belajar pakai cara pakai terminal/CLI

- Jangan bergantung sama GUI, apalagi GUI Code Runner VSCode buat nge-run kode C. Ini C bukan Python ya temen-temen, kalian liat sendiri kan buat ngecompilanya aja bisa secustom itu.
- Jangan pake online C compiler. Paksa diri kalian buat pake terminal biar kalian juga belajar.
- Sangat direkomendasikan untuk belajar command line untuk Linux distribution OS. Silakan pelajari mandiri command-command yang umum digunakan, seperti [ls](#) dan [cd](#).
 - Kalo pake Windows kalian bisa setup WSL. Kalo pake MacOS sebenarnya command-nya mirip-mirip kok sama Linux.
 - Ini juga bakal ngebantu banget untuk tugas besar kalian.
- Rajin-rajinlah baca error message. Karena kadang error message-nya sudah jelas, tapi masih ditanyakan.
- Kurang-kurangnya lah pake cara haram seperti:
 - Karena gak mau belajar cara linking yang bener, jadi pas praprak kode definisi ADT-nya dicopas semua ke file driver buat ngetes ❌
- PASTIIN KALIAN UDAH SAVE FILE/CHANGESNYA SEBELOM COMPILE 😊 Cek lagi itu VSCode praktikum autosave apa kagak.

Referensi

[GCC Manual](#)