

# Low-level Calculator

Implementasi *Naive Calculator* dengan Bahasa C

PJ: Dichi (@dichi99)

## Latar belakang

Kalian pastinya sudah pernah menggunakan kalkulator. Pada tugas kali ini, kalian akan ditugaskan untuk mengembangkan sebuah kalkulator yang bersifat **low-level** (*naive calculator*) menggunakan bahasa C.

## Tujuan

Gampangnya, tugas ini ngelatih pola pikir kalian dalam *low-level programming*, yaitu bagaimana kalian dapat memecahkan suatu masalah dengan *approach low-level*, yang kurang lebih sama kayak orkom deh.

## Spesifikasi

Gunakan kode berikut sebagai titik awal pengerjaan tugas ini.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    unsigned int bit1 : 1;
    unsigned int bit2 : 1;
    unsigned int bit3 : 1;
    unsigned int bit4 : 1;
    unsigned int bit5 : 1;
    unsigned int bit6 : 1;
    unsigned int bit7 : 1;
    unsigned int bit8 : 1;
} reg;

reg reg1;
reg reg2;
reg reg3;
reg reg4;
```

```

int main(int argc, char* argv[]){

    FILE *file = fopen("math.txt", "r");
    char c;

    if (file == NULL) {
        printf("File not found!\n");
        exit(1);
    }

    while ((c = fgetc(file)) != EOF) {
    }

    return 0;
}

```

Dari kode di atas, perhatikan hal-hal berikut ini:

- **Empat buah *register 8-bit*** yang digunakan untuk menyimpan angka dan hasil kalkulasi.
- Berkas **math.txt** yang akan dibaca oleh program berisi operasi matematika

Spesifikasi dari tugas ini adalah:

- Tugas ini dikerjakan secara individual (ya iyalah).
- Operator pada *file math.txt* dapat berupa \* (perkalian), / (pembagian integer), ^ (pangkat), + (penjumlahan), - (pengurangan), % (modulo), r (akar pangkat 2 dengan pembulatan).
- Mampu melakukan perhitungan dengan *range* 0~255 dan **dapat menangani *overflow***.

Simpel, kan?

Sekarang tantangannya adalah, implementasikan kalkulator tersebut dengan batasan:

- **Tidak boleh menggunakan operator** -, +, \*, /, %, |, ^, &.
- Penggunaan if dan konstanta nilai diperbolehkan.
- Penggunaan makro #define diperbolehkan.
- **Tidak boleh menggunakan *looping*** (*while*, *for*), kecuali dalam pembacaan berkas **math.txt**

- Penggunaan *statement* goto diperbolehkan.
- **Tidak boleh mendeklarasikan variabel apapun** (termasuk pembuatan method yang mengembalikan nilai).
- Presedensi operator adalah pangkat dan akar dikerjakan lebih dahulu, sisanya mengikuti aturan kedatangan.
- Input angka maksimal 3 digit.

Contoh isi dari **math.txt** adalah sebagai berikut :

$5+4-2/r4*5^2$

Pada contoh di atas, program akan menghasilkan angka **75** dengan langkah pengerjaan sebagai berikut (angka yang di-**bold** adalah angka hasil operasi):

1.  $5 + 4 = \mathbf{9}$
2.  $\mathbf{9} - 2 = \mathbf{7}$
3.  $\mathbf{7} / r4$ , dengan  $r4 = 2$ ,  
maka  $\mathbf{7} / 2 = \mathbf{3}$
4.  $\mathbf{3} * 5^2$ , dengan  $5^2 = 25$   
Maka  $\mathbf{3} * 25 = \mathbf{75}$

**Sebagai keringanan**, kalian dapat men-*trade* sebagian nilai kalian dengan melakukan hal-hal di luar spesifikasi sehingga kalian tidak mencapai nilai maksimum dari tugas ini. Tentunya, nilai yang di-trade akan disesuaikan dengan modifikasi yang kalian lakukan. Hal yang dapat dilakukan misalnya:

- Menambah register tambahan **reg5** dan **reg6**
- Mengerjakan hanya operator  $+$ ,  $-$ ,  $*$ ,  $/$  (tidak mengerjakan operator  $\%$ ,  $r$ , dan  $\wedge$ )
- Menggunakan loop di luar pembacaan berkas **math.txt** (*glek*, tidak disarankan)
- dll.

## Akhir Kata

Jangan mudah patah semangat!! Meskipun kalian rasa ini susah, tapi ngga impossible ko wkwk.

Selamat mengerjakan~