

ACHTUNG! UNDER CONSTRUCTION Information provided by this document may be outdated or just plain WRONG. Use your brain. NO REFUNDS.

Aerphanas Website

Muhammad Aviv Burhanudin

ABSTRACT

Selamat datang di blog saya! Nama saya Muhammad Aviv Burhanudin aka aerphanas, dan saya adalah seorang computerhobyist dengan minat mendalam dalam komputer. Saya memulai blog ini untuk berbagi pengalaman, pengetahuan, dan pandangan saya tentang dunia komputer, dengan latar belakang di IT, saya memiliki pengalaman yang luas dalam pemrograman dan komputer. Selain itu, saya sangat antusias tentang open source dan free software. Di sini, Anda akan menemukan artikel, tutorial, opini, proyek, dll. Saya berkomitmen untuk memberikan konten yang bermanfaat dan inspiratif, serta berbagi wawasan yang saya peroleh sepanjang perjalanan saya, jangan ragu untuk menghubungi saya jika Anda memiliki pertanyaan, komentar, atau sekadar ingin berbincang. Terima kasih telah mengunjungi blog saya, dan saya harap Anda menikmati konten yang saya sajikan!

Post Title

1. Tipe-tipe Recursion
2. Program blink arduino dengan bahasa C
3. Parsing menggunakan Definite clause grammar
4. Pengalaman saya menggunakan ACME, sam dan ed
5. bahasa program favorit saya
6. Program blink arduino dengan bahasa ada

1. Tipe-tipe Recursion

1.1. Standard Recursion

Recursion merupakan sebuah konsep dimana sebuah fungsi memanggil dirinya sendiri, recursion banyak dipakai pada bahasa fungsional bersih, namun dapat dipakai juga pada bahasa lain yang bukan bahasa fungsional bersih.

contoh recursion dalam bahasa Haskell, yang akan menghitung factorial :

```
factorial :: Int -> Int
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

dari fungsi diatas kita dapat melihat bila kondisi atau variable n adalah 0 maka akan mengembalikan angka 1 dan akan keluar dari recursion, dan jika nilai n bukan lah 0 maka fungsi ini akan memanggil dirinya sendiri dengan perubahan pada parameter n, seperti inilah bahasa yang tidak memiliki fungsi loop seperti while, for loop dapat melakukan looping, namun cara ini merupakan cara yang sangat buruk untuk melakukan loop, cara recursion seperti ini merupakan cara yang sangat tidak efisien karna setiap kali fungsi memanggil diri sendiri maka ada kemungkinan memori stack kita penuh, dan dapat terjadi Error stack

overflow.

seperti inilah ilustrasinya, jika kita memasukan angka 5 pada input fungsi factorial :

```
factorial 5 = 5 * factorial ( 5 - 1 )
factorial 4 = 4 * factorial ( 4 - 1 )
factorial 3 = 3 * factorial ( 3 - 1 )
factorial 2 = 2 * factorial ( 2 - 1 )
factorial 1 = 1 * factorial ( 1 - 1 )
factorial 0 = 1
-- then
factorial 5 = 5 * 4 * 3 * 2 * 1
```

seperti visualisasi diatas fungsi factorial akan menunggu dan memanggil dirinya sendiri sampai memenuhi nilai 0, setelah itu akan kembali lagi ke awal, bayangkan saja bila kita memasukan nilai 100 untuk inputnya, maka fungsi factorial akan memanggil 100x dirinya sendiri, dan ada kemungkinan akan terjadi stack overflow karna memori stack penuh akibat fungsi factorial

1.2. Tail Recursion

Tail recursion juga merupakan sebuah tehnik recursion tetapi cara tail recursion melakukan recursion sangat lah berbeda dari recursion biasa, dan juga lebih efisien.

seperti inilah kita melakukan sebuah tail recursion :

```
factorial :: (Eq t, Num t) => t -> t
factorial n = go n 1

go :: (Eq t, Num t) => t -> t -> t
go 1 a = a
go n a = go ( n - 1 ) ( a * n )
```

bisa dilihat kode diatas bahwa kita memerlukan 2 buah fungsi, fungsi untuk memanggil dan fungsi yang akan melakukan perhitungan, jika saya memberikan angka 5 ke dalam fungsi factorial maka seperti inilah kalkulasinya :

```
factorial 5 = go 5 1
             = go 4 5 -- go ( 5 - 1 ) ( 5 * 1 )
             = go 3 20 -- go ( 4 - 1 ) ( 4 * 5 )
             = go 2 60 -- go ( 3 - 1 ) ( 3 * 20 )
             = go 1 120 -- go ( 2 - 1 ) ( 60 * 2 )
```

dari visualisasi diatas kita dapat melihat bahwa fungsi factorial hanya memanggil fungsi go sekali sampai parameter n yang ada di go menjadi 1 dan mengembalikan variable a yaitu 120, maka dalam fungsi go parameter n merupakan parameter tujuan dan parameter a merupakan tempat kita melakukan perhitungan, dengan melakukan recursion seperti ini kita bisa menghindari stack overflow karna kita hanya memanggil 2 fungsi saja.

2. Program blink arduino dengan bahasa C

Arduino merupakan sebuah platform yang memberikan pemula kemudahan untuk belajar mengenai Mikrokontroler, sebelum adanya Arduino kita diharuskan merakit sebuah alat yang nanti akan digunakan untuk berinteraksi dengan Mikrokontroler, dan terkadang kita juga harus memiliki keahlian menggunakan solder, dengan adanya Arduino kita bisa langsung berinteraksi dengan mikrokontroler dan melakukan komputasi, Arduino memiliki sebuah alat yang bernama Arduino IDE, dengan adanya alat ini kita dapat dengan mudah membuat instruksi untuk keperluan kita, Namun dengan adanya alat ini saya merasakan banyaknya abstraksi yang diperlukan untuk membuat Mikrokontroler melakukan apa yang saya ingin lakukan dan saya merasa saya tidak mempelajari hal baru.

Setelah itu mulailah saya mencari cara melakukan pemrograman untuk pemula seperti membuat lampu yang ada di papan Arduino berkedip, Saya menemukan dalam sebuah Blog yang berjudul [0]Programming Arduino Uno in pure C, sebuah blog yang bagus saya sarankan kamu membacanya juga, di dalam blog tersebut penulis menggunakan alat yang bernama avr-gcc, avr-objcopy dan avrdude, karena saat ini saya menggunakan [1]GNU GUIX, saya hanya perlu menginstall "gcc-cross-avr-toolchain" dan "avr-dude", avrdude digunakan untuk mengupload atau menginstall program ke Mikrokontroler dan avr-gcc digunakan untuk melakukan kompilasi.

Tetapi terdapat suatu masalah, dalam blog yang saya sebutkan sebelumnya, penulis menggunakan Arduino UNO yang menggunakan Mikrokontroler Atmega328p, sedangkan saya menggunakan Arduino Leonardo yang menggunakan Mikrokontroler Atmega32u4, kedua Mikrokontroler tersebut sangatlah berbeda, karna itu saya memerlukan sebuah petunjuk, Saya mencari sebuah skema dan datasheet untuk [2]Skema Arduino Leonardo dan [3]Datasheet Atmega32u4 dan setelah melihat kedua Dokumen itu saya membuat program lalu saya upload ke Mikrokontroler, seperti inilah program yang saya buat:

```
#include <avr/io.h>
#include <util/delay.h>

#define BLINK_DELAY_MS 500

int main (void) {
    DDRC |= _BV(DDC7);

    while(1) {
        PORTC |= _BV(PORTC7);
        _delay_ms(BLINK_DELAY_MS);

        PORTC &= ~_BV(PORTC7);
        _delay_ms(BLINK_DELAY_MS);
    }
}
```

Dari kode diatas terdapat beberapa C Makro yang asing, janganlah takut karna kita memiliki Datasheet, kita bisa langsung mencari kata-kata asing tersebut di Datasheet, untuk mengetahui apa itu DDRC saya melihat dari Datasheet halaman 213 tepatnya ada di bagian 18.11.5 "Data Direction Register bit must be correctly set to enable the pin as an output.", dari kutipan tersebut kita bisa mengetahui jika kita ingin menggunakan sebuah pin sebagai output kita perlu mengkonfigurasi DDR terlebih dahulu, setelah itu saya melihat Skema Arduino Leonardo, untuk LED ada di IO13 yang berada di PortC7 jadi kita harus mengisi DDRC ke DDC7 menggunakan C Makro _BV, Makro _BV adalah sebuah makro yang akan melakukan operasi bit, dikonteks ini kita sama saja seperti memanggil kode

3. Parsing menggunakan Definite clause grammar

4. Pengalaman saya menggunakan ACME, sam dan ed

5. bahasa program favorit saya

6. Program blink arduino dengan bahasa ada