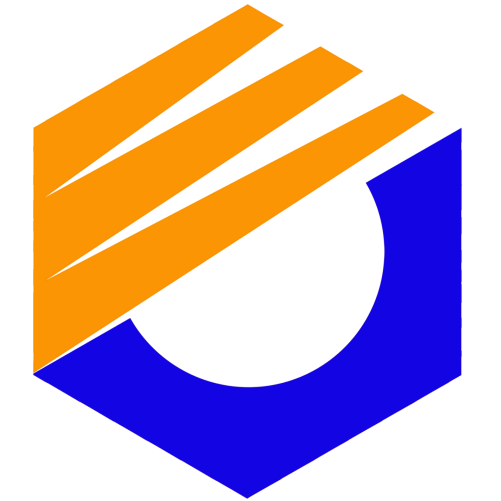
Teknik Kompilasi

*Resume Modul Praktikum*



Sadan Fitroni 151511029

2A-JTK Polban

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D3 Teknik Informatika**

**Politeknik Negeri Bandung**

**2017**

1. Resume Praktikum 1

Scanner.cpp

Dalam Modul praktikum 1 ini, harus dibuat modul get\_token () yang berfungsi untuk mengambil satu token dari source code dan menyimpannya pada array of characer yang merupakan variable gloabal.

Deklarasi Variabel

char token[50]; // String untuk menampung token

int n; // counter character token

char file[20]; // menampung nama file.

char kar; // variabel penampung karakter yang dibaca dari file

char tempKar; // menampung character saat melakukan break pada perulangan

int isBreak = 0; // jika bertemu symbol setelah numeric atau alphanumeric maka looping di break.

/\* Main Program \*/

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

{

strcpy(namafile, "example12.txt");

init\_token(namafile);

n=0;

resetToken();

get\_token();

return 0;

}

Program pertama kali akan membaca file Text, jika File text tidak ada maka program akan menampilkan : Error : Can’t open source code. Jika file ada maka iterasi token akan di nol kan, dan token akan direset, setelah itu program akan menjalankan fungsi get\_token.

Program akan membaca setiap karakter pada file yang akan dilakukan selama bukan EOF.

Setiap karakter akan dicek apakah berupa simbol, numerik, atau alfanumerik jika ya maka karakter tersebut akan disimpan ke dalam array of character, jika terjadi interupt saat looping yang berarti ada simbol setelah token numerik atau alfanumerik (isBreak =1) maka program akan membentuk token baru dan status Break akan menjadi = 0 bernilai false.

Catatan :

Jika karakter berupa spasi, tab atau enter maka karakter tidak akan dimasukkan kedalam array of character.

1. Resume Praktikum 2

Modul Praktikum 2 ini terdiri dari 2 file, file pertama scanner.c dan file kedua scanner.h

Modul ke- 2 ini merupakan modifikasi dari modul pertama sama sama membuat modul get\_token untuk mengmbil satu token dan dimasukan ke dalam Struct, yang terdiri dari :

typedef struct \_token {

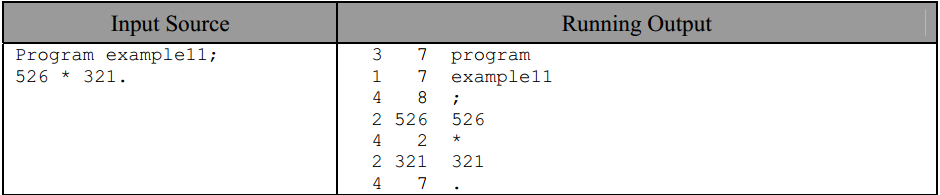
char attr : /\* attribute type \*/

char charvalue[30] : /\* string \*/

int value : /\* string value \*/

} token\_t;

Dengan tambahan kondisi karakter karakter yang dimasukkan akan dikategorikan ke dalam array of character dan membedakannya berdasarkan RWORD, SYMBOL, NUMBER, IDENTIFIER, yang sudah didefinisikan di dalam modul scanner.h. Modul get\_token () inipun melakukan perbandingan string dengan strcmp untuk menentukan nilai atribut dan token valuenya.



3 7 program

Keterangan :

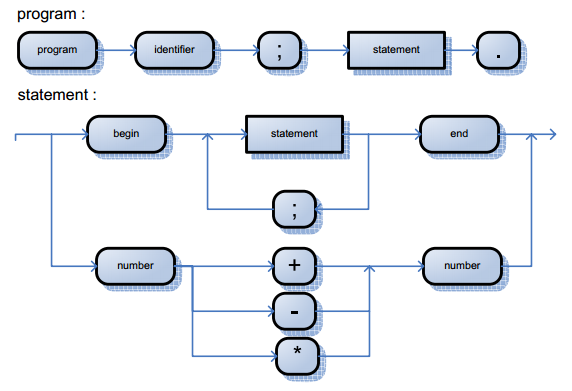
3 : token.atrr

7 : token.value

program : token.charvalue

Catatan :

Untuk Token dengan atribut number maka token.value dan token charvaluenya diisi oleh nilai nya sendiri.

1. Resume Praktikum

Pada Modul ke-3 ini, harus dibuat parser untuk siyntax graph yang terdeskripsikan diatas, dan ada 4 file yang harus ada untuk menyelesaikan parser adalah, scanner.c , compiler.c, scanner.h dan compiler.h

Pada lampiran source code compiler.h, didefinisikan isnumber, isprogram, isend dll, modul program sudah didefinisikan di file compiler.c tinggal dibuatkkan modul statement () untuk mengecek ada/tidaknya error pada program. Statement pada modul tiga berisi isbegin dan isnumber. Modul ini sesuai dengan parse tree yang diberikan. Dan output pada program ini adalah, Jika programnya sesuai maka akan menampilkan :

Congratulations : No error is found

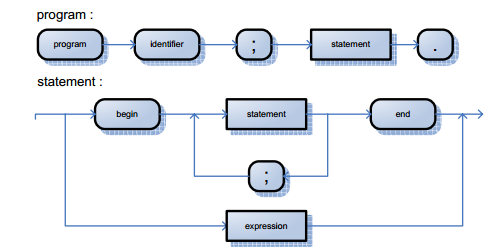
namun, jika ada error maka program akan menampilkan beberapa case diantaranya :

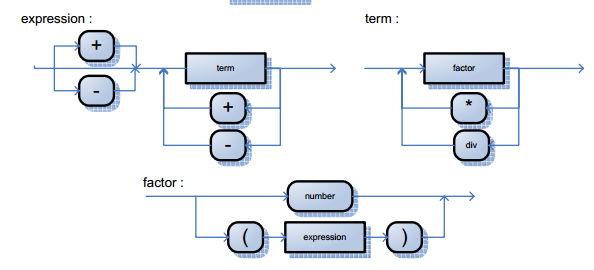
* Error : Reserved Word 'program' is expected
* Error : Reserved Word 'program' is expected
* Error : Reserved Word 'program' is expected

Dll,

Modul praktikum ke tiga ini, menggunakan operasi aritmetika sederhana diantaranya : Plus, Minus, dan Kali.

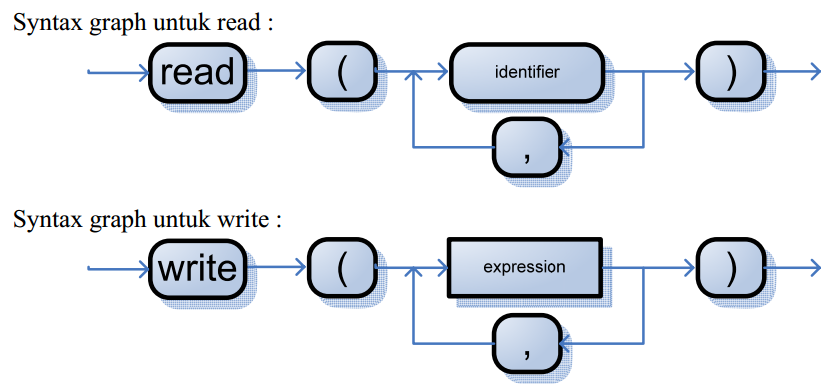
1. Resume Praktikum 4





Modul Praktikum 4 ini, hampir sama dengan modul ke 3, pada modul praktikum 3 adalah operasi aritmetika sederhana, Modul 4 ini adalah modifikasinya jadi operasi aritmetika nya lebih kompleks. Modul 4 ini pun masih menggunakan statement untuk mengecek ada/tidaknya error pada program, Namun sesuai dengan parse tree nya isbegin diawal tidak dikondisikan dengan isnumber, namun dengan memanggil modul expression (), yang didalamnya ada term(), dan di dalam term() ada factor() dan didalam factor() ada isnumber() untuk mengecek apakah token.attr nya NUMBER, jika bukan NUMBER maka dicek kembali apakah itu tanda “(“ atau “)”, untuk melakukan operasi matematis.

1. Resume Praktikum 5-6



Pada Modul 5-6 ini, harus membuat parser sintax graph seperti yang terdeklarasikan di diatas dan harus menambahkan sintax di bagian statement yaitu read dan write sehingga, dapat melakukan parsing pada testing code diatas.

Syntax graph untuk read disini adalah untuk mengecek simbol simbol yang telah didefinisikan sebelumnya. Mulai dari read lalu dicek dengan identifiernya. Begitupun berikutnya, apakah sama dengan simbol yang telah dideklarasikan. Sedangkan Syntax graph untik write adalah untuk menuliskan hasil dari proses untuk menuliskan expression apa saja yang ada.

Pada Modul statement (), setelah mengambil token akan ada kondisi untuk memastikan bahwa token yang ditunjuk adalah read atau write. Jika token yang dibaca adalah read , maka token akan masuk kedalam lparen kemudian membaca identifier sampai bertemu dengan rparen proses ini akan terus diulangi selama menemukan comma.

Jika token yang dibaca adalah write, maka akan masuk ke modul get\_token() selanjutnya masukke lparen, masuk ke modul expression(), sampai bertemu dengan rparen. Proses ini akan terus diulangi selama menemukan comma.