

Buatlah fungsi seperti yang terdeskripsikan dibawah ini !

Prototype :

```
int get_token(void)
```

Spesifikasi fungsi

Mengambil satu token dari source code, dengan format text, dan menyimpannya pada ADT dengan struktur data berikut :

```
typedef struct _token {
    char attr;          /* attribute type */
    char charvalue[30]; /* string */
    int value;          /* string value */
} token_t;
```

String dari token tersimpan pada field **charvalue**, dalam format lowercase (hints: gunakan fungsi **tolower** dari C). Sesuai dengan jenis token yang ada (yaitu identifier, number, reserved word, dan symbol), maka nilai field **attr** ada 4 (empat) kemungkinan IDENTIFIER, NUMBER, RWORD, atau SYMBOL (nilai disesuaikan dengan definisi pada lampiran source code). Adapun isi dari field **value** disesuaikan dengan jenis token-nya.

Return Value :

Fungsi akan mengembalikan nilai **EOF** apabila telah sampai pada akhir file, dan akan mengembalikan nilai positif sembarang diluar itu.

Lampiran source code :

```
/*
    File name   : scanner.h
    Vers       : 0.01a
    Written by  : Setiadi Rachmat
    Date       : Fri Aug 28 10:08:03 WIT 2009
*/

typedef struct _token {
    char attr;          /* attribute type */
    char charvalue[30]; /* string */
    int value;          /* string value */
} token_t;

/* token type */

#define IDENTIFIER 1
#define NUMBER     2
#define RWORD      3
#define SYMBOL     4
```

```
/* reserved word definition */
```

```
#define BEGIN      0
#define DIV        1
#define DO         2
#define ELSE       3
#define END        4
#define IF         5
#define PROCEDURE  6
#define PROGRAM    7
#define THEN       8
#define VAR        9
#define WHILE     10
#define READ      11
#define WRITE     12
#define FORWARD  13
#define FUNCTION  14
```

```
/* symbol definition */
```

```
#define PLUS      0
#define MINUS     1  /* - */
#define TIMES     2  /* * */
#define LPAREN    3  /* ( */
#define RPAREN    4  /* ) */
#define EQL       5  /* = */
#define COMMA     6  /* , */
#define PERIOD    7  /* . */
#define SEMICOLON 8  /* ; */
#define BECOMES   9  /* := */
#define LESSTHAN  10 /* < */
#define LESSEQ    11 /* <= */
#define NOTEQL    12 /* <> */
#define GRTRTHAN  13 /* > */
#define GRTREQ    14 /* >= */
#define LBRACE    15 /* [ */
#define RBRACE    16 /* ] */
#define INTERVAL  17 /* .. */
#define COLON     18 /* : */
```

```
/*
    File name   : scanner.c
    Vers        : 0.01a
    Written by   : Setiadi Rachmat
    Date        : Fri Aug 28 10:08:03 WIT 2009

    Modified by : ..
    Date        :
*/
```

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

```
FILE * infile;
```

```
token_t token;
```

```
int get_token(void);
```

```
/* inialisasi token */
```

```

void init_token(char *name)
{
    if ((infile = fopen(name,"r") == NULL) {
        printf("Error : Can't open source code %s \n",name);
        exit(-1);
    } else return;
}

int main(int argc,char * argv[])
{
    init_token(argv[1]);
    while (get_token() != EOF)
        printf("%3d %3d %s \n",token.attr, token.value, token.charvalue);
}

int get_token(void)
{
    /* silakan dilengkapi */
}

```

Contoh :

Input Source	Running Output
<pre> program example21; var n,temp; procedure fact(n); begin if n<=5 then temp:=1 else temp := n end; begin read(n); fact(n); write(temp) end. </pre>	<pre> 3 7 program 1 7 example21 4 8 ; 3 9 var 1 9 n 4 6 , 1 6 temp 4 8 ; 3 6 procedure 1 6 fact 4 3 (1 3 n 4 4) 4 8 ; 3 0 begin 3 5 if 1 5 n 4 11 <= 2 5 5 3 8 then 1 8 temp 4 9 := 2 1 1 3 3 else 1 3 temp 4 9 := 1 9 n 3 4 end 4 8 ; 3 0 begin 3 11 read 4 3 (1 3 n 4 4) 4 8 ; 1 8 fact </pre>

Teknik Kompilasi - Praktikum 2

	4	3	(
	1	3	n
	4	4)
	4	8	;
	3	12	write
	4	3	(
	1	3	temp
	4	4)
	3	4	end
	4	7	.