

# TUGAS BESAR TEKNIK KOMPILASI

## Compiler Pengurangan dan While-Do

*Diajukan untuk memenuhi salah satu tugas Mata Kuliah Teknik Kompilasi*



**TEKKOM-11**

**Dosen : Istiqomah , S.Kom., M.Kom.**

Disusun oleh kelompok 10:

10112747 – Rahmat Syaparudin

10112728 – Abdul Aziz

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS KOMPUTER INDONESIA

2015

## DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
A. Pendahuluan.....	1
BAB II PERANCANGAN BAHASA PEMROGRAMAN.....	2
A. Perancangan struktur control yang bisa digunakan (program sumber).....	2
1. Logika matematika.....	2
B. Perancangan analisis leksikal.....	2
C. Perancangan analisis sintaksis .....	4
1. Pohon Parsing .....	4
2. Perancangan antar muka .....	5
3. Perancangan penanganan kesalahan .....	6
4. Perancangan pesan kesalahan .....	6
BAB III PEMBAHASAN.....	8
A. Program Scanner.....	8
B. Program Parser.....	9
BAB IV PENUTUP.....	14
DAFTAR PUSTAKA .....	15
KONTRIBUSI KELOMPOK .....	16

# **BAB I**

## **PENDAHULUAN**

### **A. Pendahuluan**

Teknik Kompilasi merupakan teknik dalam melakukan pembacaan suatu program yang ditulis dalam bahasa sumber, kemudian diterjemahkan ke dalam suatu bahasa lain yang disebut bahasa sasaran. Dalam melakukan proses penerjemahan tersebut, sudah barang tentu kompilator akan melaporkan adanya keanehan-keanehan atau kesalahan yang mungkin ditemukannya. Proses penerjemahan yang dilakukan oleh kompilator ini disebut proses kompilasi (compiling).

Program yang kami buat adalah compiler yang digunakan untuk melakukan parsing pengurangan dan perulangan (while-do). Program ini dibuat dengan dua aplikasi, aplikasi scanner untuk melakukan scanner terhadap token-token yang tersedia dan aplikasi parser untuk melakukan parsing dengan aturan produksi yang tersedia.

Untuk pembatasan penggunaan aplikasi pun kami batasi karena ketidakmampuan kami untuk mewujudkan aplikasi kompilator yang mendekati sempurna.

## BAB II

### PERANCANGAN BAHASA PEMROGRAMAN

#### A. Perancangan struktur control yang bisa digunakan (program sumber)

##### 1. Logika matematika

###### a. Pengurangan :

Konteks Pemrograman	Terjemahan ke bahasa German
<pre> program coba; var   a = 10;   b = 8; begin   a := a - b; end.</pre>	<pre> programm versuchen; var   a = 10;   b = 8; anfangen   a: = a - b; ende.</pre>

*Tabel 1 Logika Matematika pengurangan*

###### b. Instruksi pemrograman (while – do)

Konteks Pemrograman	Terjemahan ke bahasa German
<pre> program perulangan; var   a = 8;   b = 6; begin   while a &gt; a do     begin       a := a - b;     end; end.</pre>	<pre> programm iterationen; var   a = 8;   b = 6; anfangen   wahrend a &gt; b tun     anfangen       a: = a - b;     ende; ende.</pre>

*Tabel 2 Logika Matematika While-Do*

#### B. Perancangan analisis leksikal

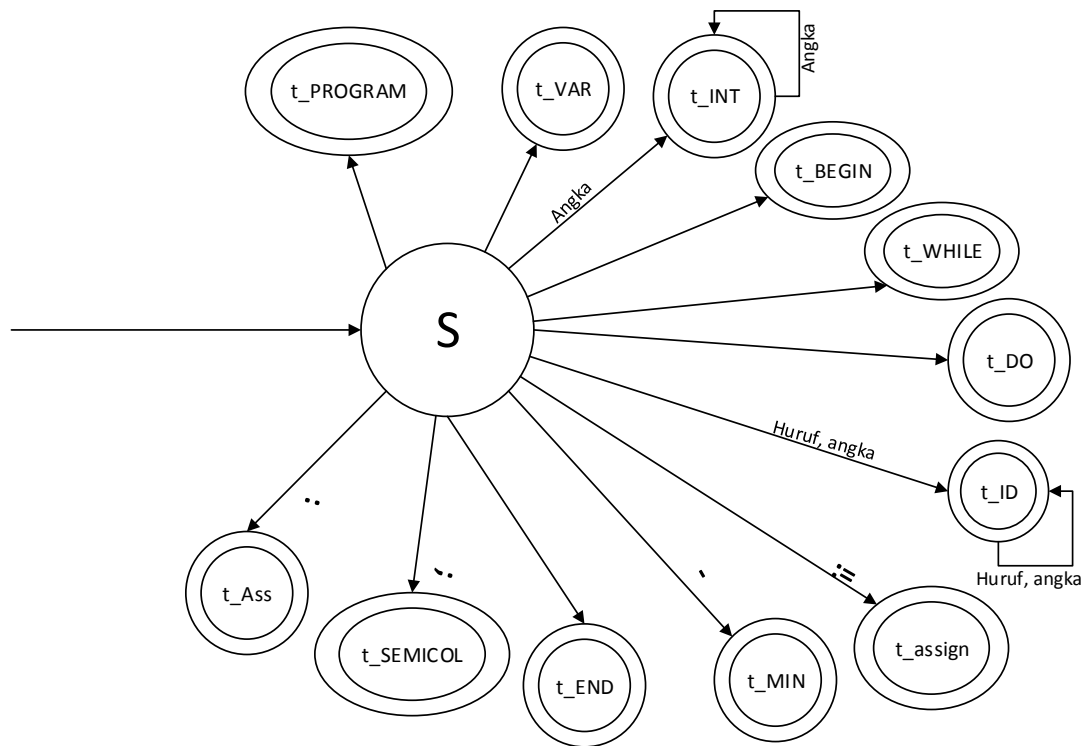
##### a. Token – token yang tersedia :

Nama Token		Arti Token
t_ID	:	Untuk huruf A..Z, a..z
t_INT	:	Untuk bilangan bulat
t_ASS	:	Untuk tanda =
t_ASSIGN	:	Untuk tanda :=

t_SEMICOL	:	Untuk tanda titik koma
t_WHILE	:	Untuk keyword While
t_BEGIN	:	Untuk keyword begin (Mulai statement)
t_END	:	Untuk keyword end
t_DO	:	Untuk keyword do
t_MIN	:	Untuk pengurangan
t_PROGRAM	:	Untuk memulai membuat program baru
t_VAR	:	Untuk mendeklarasikan variabel

*Tabel 3 Token-token yang tersedia*

b. Diagram State



*Gambar 1 Diagram State*

### C. Perancangan analisis sintaksis

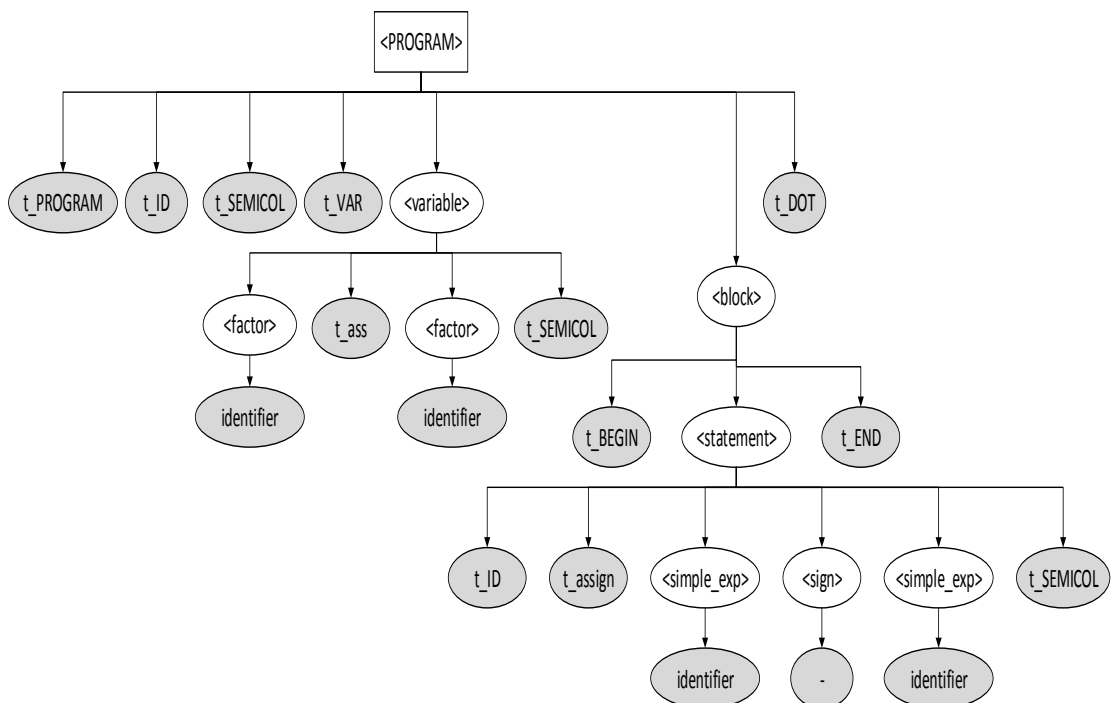
Aturan Produksi dalam notasi BNF

<program>	::=	t_PROGRAM t_ID t_SEMICOL t_VAR <variable> <block> t_DOT
<variable>	::=	{<factor> t_ASS <factor> t_SEMICOL}
<block>	::=	t_BEGIN <statement> t_END
<statement>	::=	t_ID t_ASSIGN <simple_exp> <sign> <simple_exp> t_SEMICOL   t_WHILE <simple_exp> t_DO <block> t_SEMICOL   <simple_exp> t_SEMICOL
<simple_exp>	::=	<factor>   <factor> <operator> <factor>
<factor>	::=	identifier   integer
<operator>	::=	>
<sign>	::=	-   +

Tabel 4 Aturan produksi dalam notasi BNF

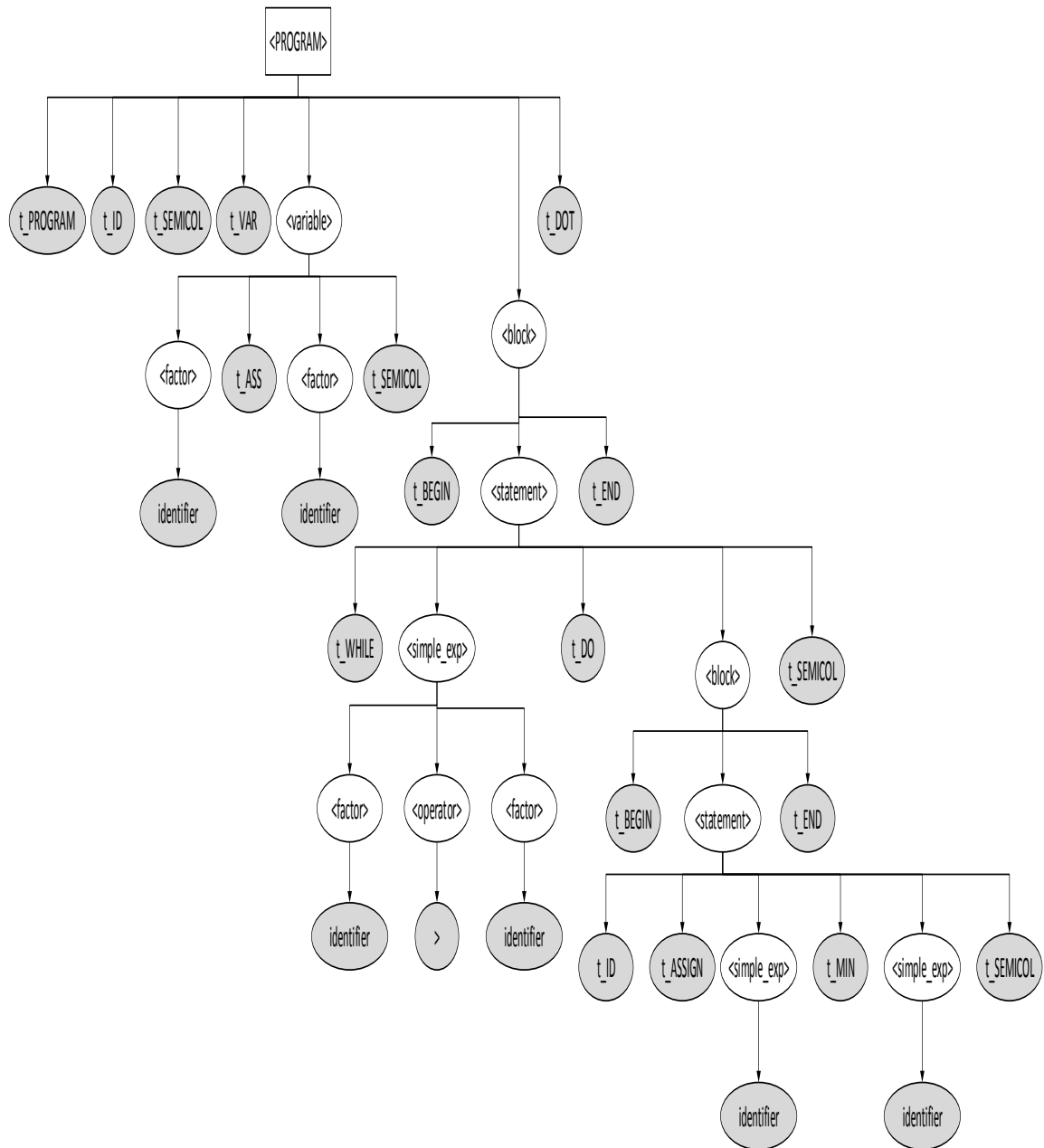
#### 1. Pohon Parsing

##### a. Pohon Parsing Pengurangan



Gambar 2 Pohon parsing pengurangan

## b. Pohon Parsing while - do



Gambar 3 pohon parsring while – do

## 2. Perancangan antar muka

- Flow of Control Check

Setiap statement harus diakhiri dengan tanda “;” (titik koma). Misalnya pada statement ini :

```

programm versuchen;
var
  a = 10;
  b = 8
anfangen
  a: = a - b;
ende.

```

maka akan muncul pesan kesalahan, karena tidak ada tanda titik koma pada statement baris keempat.

### 3. Perancangan penanganan kesalahan

#### ➤ Kesalahan leksikal

Terjadi kesalahan dalam mengeja keyword. Maka penanganannya dengan memberikan pesan dengan kode #Ex008 berisi “#Ex008 : Tidak ada keyword atau terjadi kesalahan”.

#### ➤ Kesalahan sintaks

Terjadi kesalahan dalam penulisan sintaks, seperti kurangnya tanda semicolon (;). Maka penanganannya dengan memberikan pesan dengan kode #Ex001 berisi “#Ex001 : Tanda ; tidak ditemukan”.

#### ➤ Kesalahan semantic

Terjadi salah tipe data. Maka penanganannya dengan memberikan pesan dengan kode #Ex004 berisi “#Ex004 : Identifier atau Integer tidak ditemukan”.

### 4. Perancangan pesan kesalahan

No	Deskripsi	Kode Error	Pesan Error
1	Apabila statement tidak diakhiri dengan titik koma (;)	#Ex001	#Ex001 : zu unterzeichnen ; nicht gefunden "Tanda ; tidak ditemukan"
2	Apabila statement yang diawali BEGIN tidak ditutup dengan END	#Ex002	#Ex002 : anfangen startet, aber am Ende nicht gefunden "BEGIN dimulai, tetapi END tidak ditemukan"



3	Apabila END ditemukan tetapi tidak dimulai dengan BEGIN	#Ex003	#Ex003 : anfangen nicht gefunden “BEGIN tidak ditemukan”
4	Apabila Identifier atau Integer tidak didefinisikan	#Ex004	#Ex004 : kennzeichnung oder ganze zahl nicht gefunden “Identifier atau Integer tidak ditemukan”
5	Apabila variable belum didefinisikan sebagai integer atau identifier (tipe data)	#Ex005	#Ex005 : variable wert oder art nicht definiert “Nilai atau tipe Variable belum didefinisikan”
6	Apabila := tidak dapat ditemukan	#Ex006	#Ex006 : nichts := “Tidak ada :=”
7	Apabila di akhir program tidak ditemukan titik (.)	#Ex007	#Ex007 : das ende des programms wurde nicht gefunden “Akhir program tidak ditemukan”
8	Apabila keyword tidak ada atau kesalahan pengetikan keyword	#Ex008	#Ex008 : kein stichwort oder fehler auftritt “Tidak ada keyword atau terjadi kesalahan”
9	Apabila tidak ada operator pembandingan (>)	#Ex009	#Ex009 : operator nicht gefunden “Operator tidak ditemukan”
10	Apabila tidak ada perhitungan matematika (-)	#Ex010	#Ex010 : zeichen marken nicht gefunden “Tanda Sign tidak ditemukan”

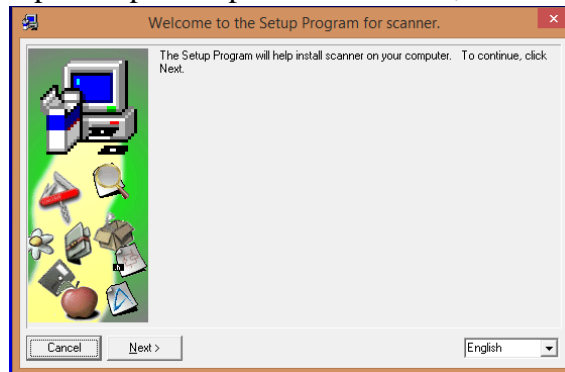
*Tabel 5 Perancangan pesan kesalahan*

## BAB III

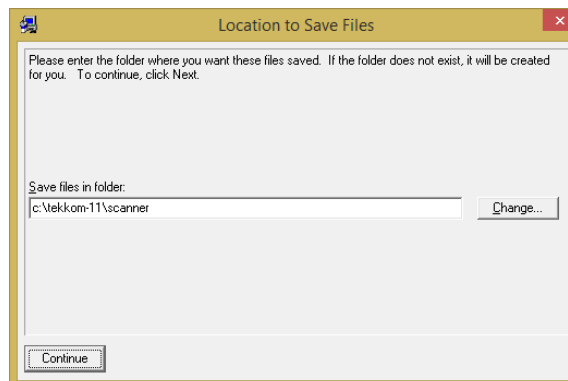
### PEMBAHASAN

#### A. Program Scanner

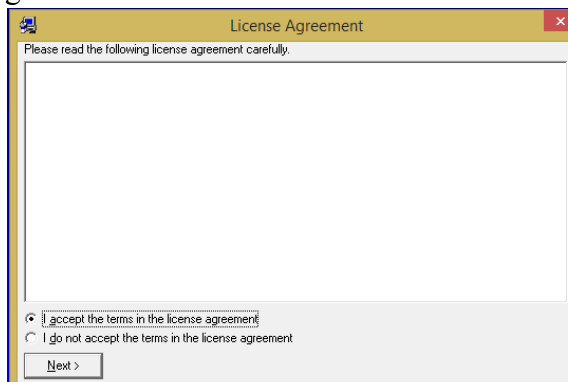
1. Cara Instal Aplikasi
  - a. Ekstrak file 'scanner-setup.rar'
  - b. Buka foldernya lalu instal setup.exe
  - c. Maka akan terdapat tampilan seperti di bawah ini, klik next



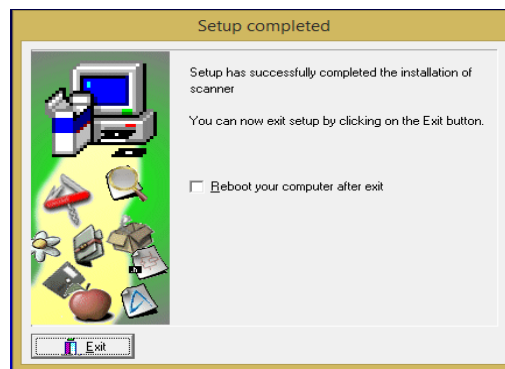
- d. Lalu klik continue



- e. Lalu pilih "I agree" dan klik next



f. Lalu klik exit

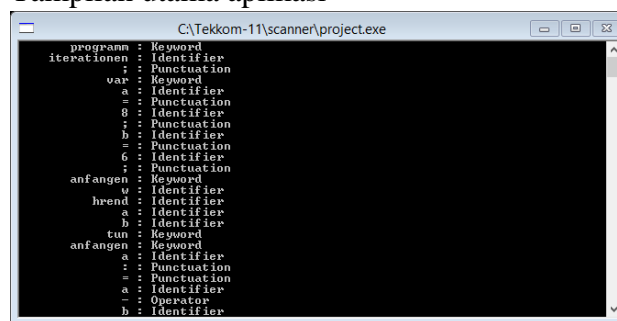


## 2. Cara menggunakan aplikasi

- a. Buka Windows Explorer lalu buka drive C:/tekkom-11/scanner
- b. Buka file “pilih soource dari sini, copas ke file program.txt.txt” untuk memilih source code yang ingin di scan
- c. Copy paste kan salah satu kode di dalam folder tersebut ke dalam file “program.txt”
- d. Lalu jalankan aplikasi “project.exe”
- e. Aplikasi akan menampilkan hasil scanner dari source code yang dipilih sebelumnya pada langkah b
- f. Hasil scanner akan tersimpan dalam file “result.txt”
- g. Ulangi langkah ke b untuk melakukan scan untuk source code lainnya.

## 3. Tampilan Aplikasi

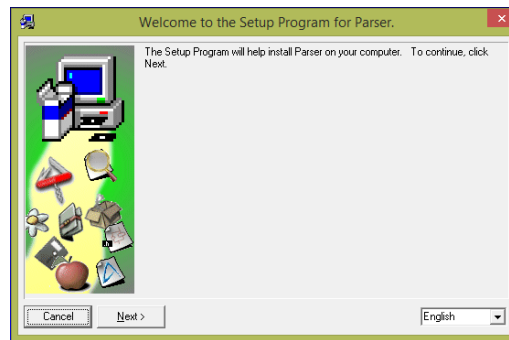
- a. Tampilan utama aplikasi



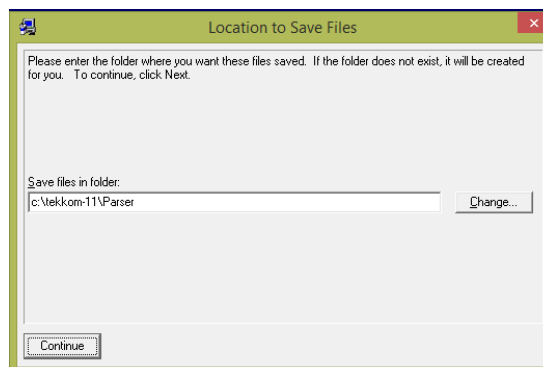
## B. Program Parser

### 1. Cara Instalasi Aplikasi

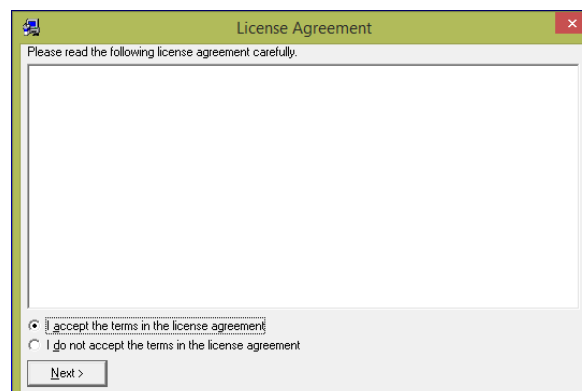
- a. Ekstrak file ‘parser-setup.rar’
- b. Buka foldernya lalu instal setup.exe
- c. Maka akan terdapat tampilan seperti di bawah ini, klik next



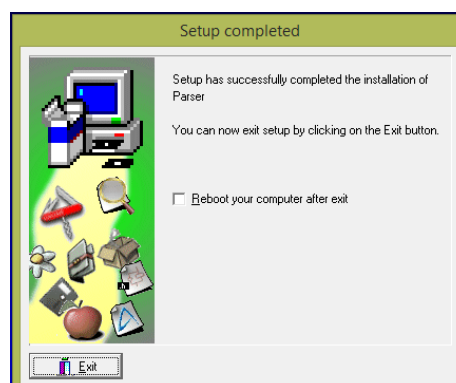
d. Lalu klik continue



e. Lalu pilih "I agree" dan klik next

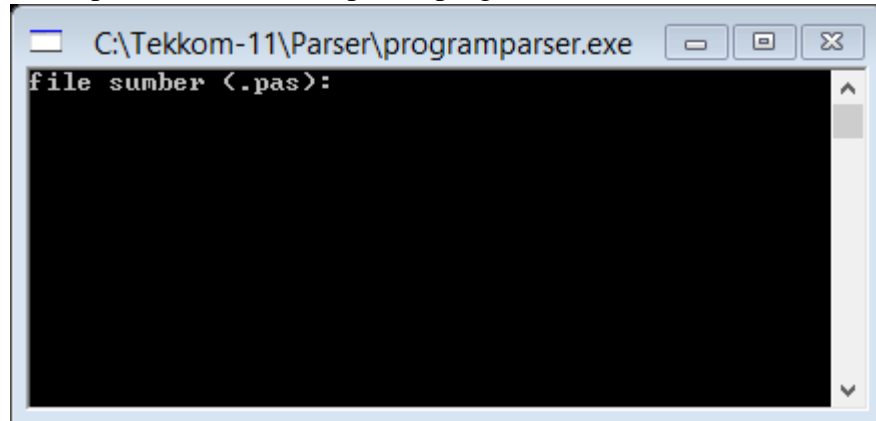


f. Lalu klik exit



## 2. Cara menggunakan aplikasi

- a. Buka Windows Explorer lalu buka drive C:/tekkom-11/Parser
- b. Lalu jalankan aplikasi “programparser.exe”
- c. Lalu aplikasi akan menampilkan program tersebut.



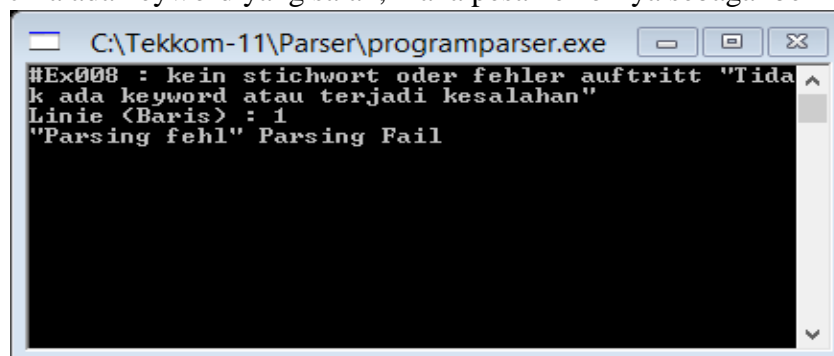
- d. Lalu ketik “**while.pas**” untuk perulangan atau “**sub.pas**” untuk pengurangan lalu enter.
- e. Jika parsing berhasil maka akan menampilkan pesan



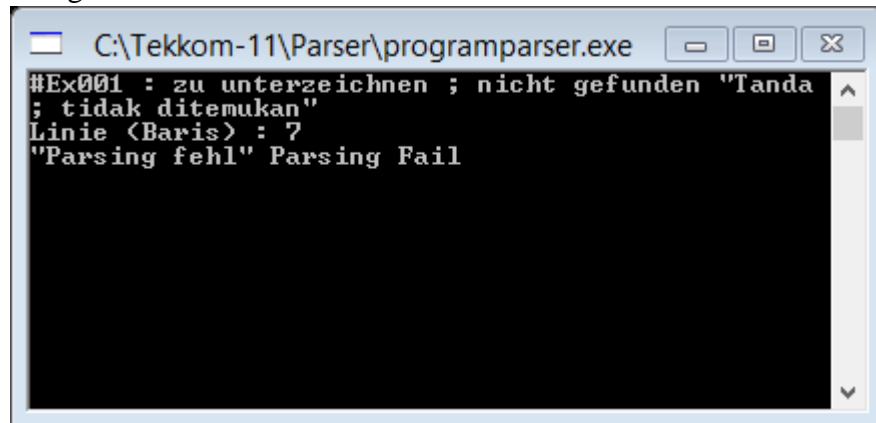
- f. Untuk mencoba error yang terdapat di aplikasi ini maka kita perlu sedikit mengedit file “**while.pas**” untuk perulangan atau “**sub.pas**” untuk pengurangan. Bisa menggunakan notepad sebagai sarana editor.

## 3. Percobaan untuk menampilkan pesan error (file sub.pas)

- a. Jika ada keyword yang salah, maka pesan errornya sebagai berikut:



- b. Jika akhir statement tidak diakhiri titik koma, maka pesan errornya sebagai berikut:

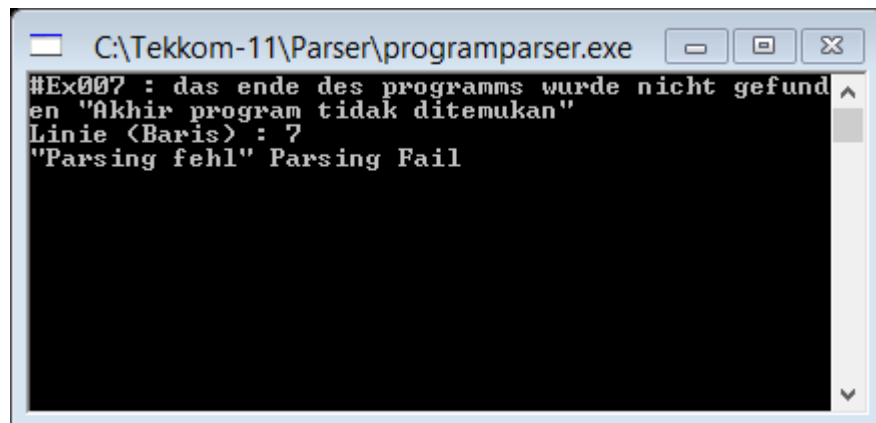


```

C:\Tekkom-11\Parser\programparser.exe
#Ex001 : zu unterzeichnen ; nicht gefunden "Tanda
; tidak ditemukan"
Linie <Baris> : 7
"Parsing fehl" Parsing Fail

```

- c. Jika akhir program tidak ditemukan, maka pesan errornya sebagai berikut:

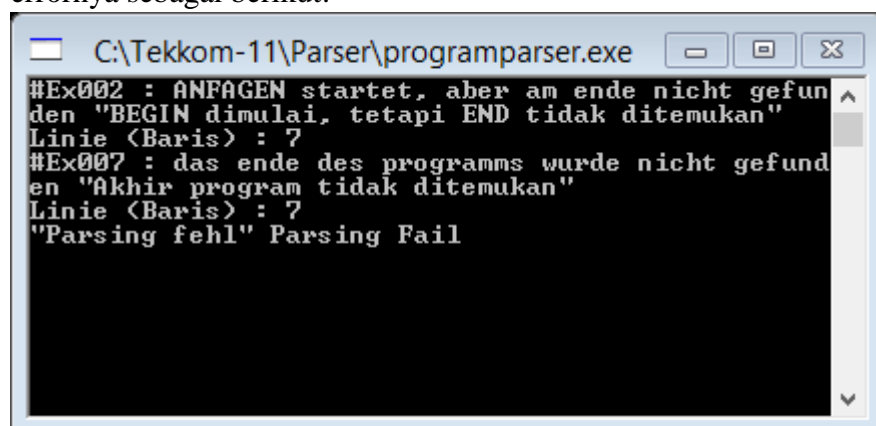


```

C:\Tekkom-11\Parser\programparser.exe
#Ex007 : das ende des programms wurde nicht gefund
en "Akhir program tidak ditemukan"
Linie <Baris> : 7
"Parsing fehl" Parsing Fail

```

- d. Jika anfangen dimulai, tetapi ende tidak ditemukan, maka pesan errornya sebagai berikut:

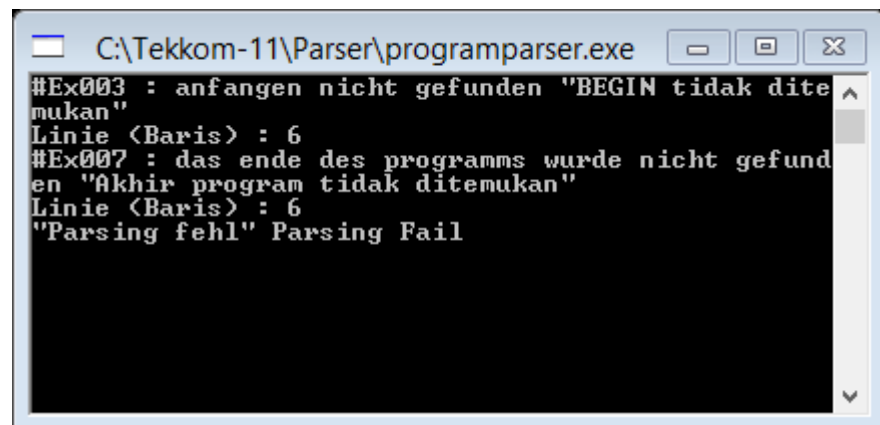


```

C:\Tekkom-11\Parser\programparser.exe
#Ex002 : ANFAGEN startet, aber am ende nicht gefun
den "BEGIN dimulai, tetapi END tidak ditemukan"
Linie <Baris> : 7
#Ex007 : das ende des programms wurde nicht gefund
en "Akhir program tidak ditemukan"
Linie <Baris> : 7
"Parsing fehl" Parsing Fail

```

- e. Jika ende ditemukan, tetapi tidak dimulai dengan anfangen, maka pesan errornya sebagai berikut:



```
C:\Tekkom-11\Parser\programparser.exe
#Ex003 : anfangen nicht gefunden "BEGIN tidak ditemukan"
Linie <Baris> : 6
#Ex007 : das ende des programms wurde nicht gefunden "Akhir program tidak ditemukan"
Linie <Baris> : 6
"Parsing fehl" Parsing Fail
```

## **BAB IV**

### **PENUTUP**

Mengembangkan sebuah kompilator dengan bahasa tertentu (penulis menggunakan bahasa jerman) merupakan pekerjaan yang tidak sederhana. Karena suatu bahasa memiliki grammar (aturan bahasa) masing-masing yang menyulitkan penulis untuk memahaminya. Orang-orang menginginkan bahasa pemrograman yang sederhana, tetapi dengan kesederhanaan yang diinginkan tersebut akan terdapat kekurangan dalam mewujudkannya. Oleh karena itu, dalam pembuatan kompilator ini penulis memberikan batasan yng bisa digunakan user dalam penggunaan aplikasi tersebut.



## **DAFTAR PUSTAKA**

Utdirartatmo, Firrar. 2005. *Teknik Kompilasi*. Yogyakarta: Graha Ilmu.

### KONTRIBUSI KELOMPOK

<b>NIM</b>	<b>Nama</b>	<b>Kontribusi</b>
10112747	Rahmat Syaparudin	Program
10112728	Abdul Aziz	Laporan