LAPORAN TUGAS BESAR TEORI BAHASA AUTOMATA
MEMBUAT LEXICAL ANALYZER DAN PARSER SEDERHANA UNTUK TEKS BAHASA ALAMI

Tugas

Disusun untuk memenuhi tugas besar Mata Kuliah Teori Bahasa Automata

Oleh:

Aditya Andar Rahim 1301194029

Irfan Ahmad Asqolani 1301190323

Muhammad Zalfa Thoriq 1301194473

PROGRAM STUDI INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2021

Daftar Isi

BAB I

PENDAHULUAN

Lexical analyzer adalah sebuah proses yang mendahului parsing sebuah rangkaian karakter, lexical analyzer menerima masukan serangkaian karakter (seperti dalam dokumen *plain-text* atau *source code*) dan menghasilkan deretan simbol yang masing-masing dinamakan token**,** proses parsing akan lebih mudah dilakukan bila inputnya sudah berupa token, sedangkan Parser adalah komponen *compiler* atau juru bahasa yang memecah data menjadi elemen yang lebih kecil untuk memudahkan terjemahan ke bahasa lain, parser mengambil input dalam bentuk urutan token atau instruksi program dan biasanya membangun struktur data dalam bentuk pohon parse atau pohon sintaksis abstrak. Untuk membuat program Lexical Analyzer dan Parser membutuhkan beberapa parameter yaitu Context Free Grammar, Finite Automata dan Parse Table LL(1). *Context Free Grammar* (CFG) atau disebut dengan Bahasa Bebas Konteks adalah sebuah tata bahasa dimana tidak terdapat pembatasan pada hasil produksinya, dan Finite State Automata adalah model matematika yang dapat menerima inputan dan mengeluarkan output. Memiliki state berhingga banyaknya dan dapat berpindah dari satu ke yang lainnya sesuai dengan inputan dan fungsi transisi. Lalu Parse Table LL(1) adalah L pertama berarti string input diproses dari kiri ke kanan. L kedua berarti derivasi akan menjadi derivasi paling kiri (variabel paling kiri diganti pada setiap langkah). 1 berarti satu simbol dalam string input digunakan untuk membantu memandu parse. Pembahasan lebih lanjut mengenai perancangan program lexical analyzer dan parser akan dijelaskan pada bab selanjutnya.

BAB II

STRATEGI & PARAMETER PROGRAM

Daftar Kata

Berikut adalah daftar kata dalam bahasa spanyol yang kami gunakan sebagai parameter program :

Bahasa Indonesia	Bahasa Spanyol
saya	me
ayah	padre
adik	hermanito
kakek	abuelo
jus	jugo
churros	churros
novel	novela
sepatu	zapatos
minum	beber
memasak	cocinero
membaca	leer
memakai	usar

Context Free Grammar

Berikut adalah context free grammar dari daftar kata yang kami pilih dan akan digunakan sebagai parameter program :

S → NN VB NN
NN → me | padre | hermanito | abuelo | jugo | churros | novela | zapatos

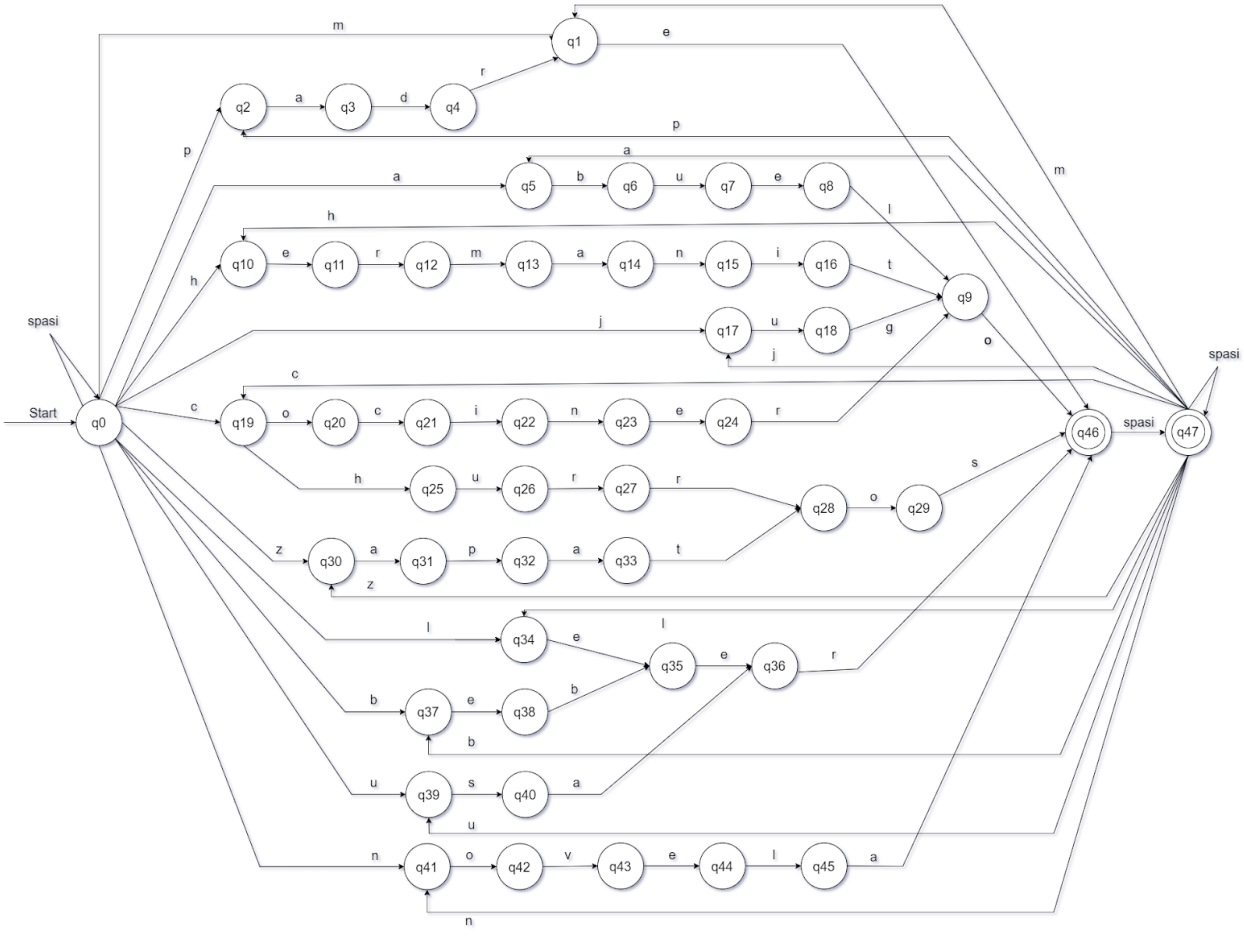
VB → beber | cocinero | leer | usar

Simbol non-terminal : S (starting symbol), NN (Noun), VB (Verb)

Simbol terminal : me, padre, hermanito, abuelo, jugo, churros, novella, zapatos, beber, cocinero, leer, usar

Finite Automata

Berikut adalah context finite automata dari context free grammar yang telah kami buat dan akan digunakan sebagai parameter program:



Parse Table LL(1)

Berikut adalah parse table LL(1) dari context free grammar yang telah kami buat dan akan digunakan sebagai parameter program :

	me	**padre**	**hermanito**	**abuelo**	**beber**	**cocinero**	**leer**	**usar**	**jugo**	**churros**	**novela**	**zapatos**	**EOS**
S	NN VB NN	NN VB NN	NN VB NN	NN VB NN	error	error	error	error	NN VB NN	NN VB NN	NN VB NN	NN VB NN	error
NN	me	padre	hermanito	abuelo	error	error	error	error	jugo	churros	novela	zapatos	error
VB	error	error	error	error	beber	cocinero	leer	usar	error	error	error	error	error

Strategi dan Cara Menjalankan Program

Penguji atau user memasukan kalimat uji, kalimat uji yang dimasukan harus sesuai dengan simbol terminal yang didefinisikan dan susunan kata harus sesuai dengan aturan pada grammar yaitu dengan struktur S-V-O (NN VB NN), kemudian program akan menguji validasi kalimat uji. Program lexical analyzer akan bekerja dengan cara mengidentifikasi apakah kata-kata pada kalimat uji sesuai dengan simbol terminal, sedangkan program parser akan bekerja dengan cara mengidentifikasi apakah kalimat uji sudah memenuhi aturan pada grammar. Jika kalimat uji dapat memenuhi aturan lexical dan parser maka kalimat uji tersebut dapat dikatakan valid atau diterima, tetapi jika kalimat uji hanya memenuhi aturan lexical saja atau tidak memenuhi aturan lexical dan parser maka kalimat uji dapat dikatakan tidak valid atau tidak di terima. Adapun Langkah-langkah untuk menjalankan program yaitu sebagai berikut :

1. Pastikan file program sudah terdownload dan python sudah terinstall.
2. Klik dua kali pada file program atau file juga bisa dibuka melalui text editor (disarankan Visual Studio Code) kemudian buka terminal pada Visual Studio Code dan untuk menjalankannya ketik python **TubesTBA.py**
3. Masukan kalimat yang ingin diuji, program akan memproses kalimat uji dan aka menghasilkan output apakah kalimat uji valid atau tidak.
4. Ketik “quit” untuk keluar dari program.

BAB III

HASIL PENGUJIAN

Program Lexical Analyzer

- Hasil pengujian kata yang valid, kata-kata yang diuji: me cocinero churros

```
Kalimat Uji: me cocinero churros

---- Lexical Analyzer ----
current token:  me , valid
current token:   cocinero , valid
current token:   churros , valid
semua token diinput:  me cocinero churros , valid

--------- Parser ---------
top =  S
symbol =  me
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'NN']

top =  NN
symbol =  me
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'me']

top =  me
symbol =  me
top adalah simbol terminal
isi stack: ['#', 'NN', 'VB']

top =  VB
symbol =  cocinero
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'cocinero']

top =  cocinero
symbol =  cocinero
top adalah simbol terminal
isi stack: ['#', 'NN']
```

```
top =  NN
symbol =  churros
top adalah simbol non-terminal
isi stack: ['#', 'churros']

top =  churros
symbol =  churros
top adalah simbol terminal
isi stack: ['#']
isi stack: []

Kesimpulan:
Lexical Analyzer: Valid
Parser: Valid
Kalimat Uji: me cocinero churros , diterima karena sesuai dengan simbol terminal dan aturan pada grammar
```

- Hasil pengujian kata yang tidak valid, kata-kata yang diuji: abueli user zapatus

```
Kalimat Uji: abueli user zapatus

---- Lexical Analyzer ----
error

Kesimpulan:
Lexical Analyzer: Not Valid
Error, Kalimat Uji: abueli user zapatus , tidak diterima karena tidak sesuai dengan simbol terminal yang didefinisikan
```

Kata-kata diatas tidak diterima atau tidak valid karena terdapat salah ketik (typo) saat penulisan sehingga kata-kata tersebut tidak sesuai dengan simbol terminal.

Program Parser

- Hasil pengujian kalimat uji yang valid
    1. Kalimat yang diuji: me beber jugo

```
Kalimat Uji: me beber jugo

---- Lexical Analyzer ----
current token:  me , valid
current token:   beber , valid
current token:   jugo , valid
semua token diinput:  me beber jugo , valid

--------- Parser ---------
top =  S
symbol =  me
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'NN']

top =  NN
symbol =  me
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'me']
```

```
top =  me
symbol =  me
top adalah simbol terminal
isi stack: ['#', 'NN', 'VB']

top =  VB
symbol =  beber
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'beber']

top =  beber
symbol =  beber
top adalah simbol terminal
isi stack: ['#', 'NN']

top =  NN
symbol =  jugo
top adalah simbol non-terminal
isi stack: ['#', 'jugo']

top =  jugo
symbol =  jugo
top adalah simbol terminal
isi stack: ['#']
isi stack: []

Kesimpulan:
Lexical Analyzer: Valid
Parser: Valid
Kalimat Uji: me beber jugo , diterima karena sesuai dengan simbol terminal dan aturan pada grammar
```

2. Kalimat yang diuji: padre cocinero churros

```
Kalimat Uji: padre cocinero churros

---- Lexical Analyzer ----
current token:  padre , valid
current token:   cocinero , valid
current token:   churros , valid
semua token diinput:  padre cocinero churros , valid

--------- Parser ---------
top =  S
symbol =  padre
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'NN']

top =  NN
symbol =  padre
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'padre']

top =  padre
symbol =  padre
top adalah simbol terminal
isi stack: ['#', 'NN', 'VB']
```

```
top =  VB
symbol =  cocinero
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'cocinero']

top =  cocinero
symbol =  cocinero
top adalah simbol terminal
isi stack: ['#', 'NN']

top =  NN
symbol =  churros
top adalah simbol non-terminal
isi stack: ['#', 'churros']

top =  churros
symbol =  churros
top adalah simbol terminal
isi stack: ['#']
isi stack: []

Kesimpulan:
Lexical Analyzer: Valid
Parser: Valid
Kalimat Uji: padre cocinero churros , diterima karena sesuai dengan simbol terminal dan aturan pada grammar
```

3. Kalimat yang diuji: hermanito leer novela

```
Kalimat Uji: hermanito leer novela

---- Lexical Analyzer ----
current token:  hermanito , valid
current token:   leer , valid
current token:   novela , valid
semua token diinput:  hermanito leer novela , valid

--------- Parser ---------
top =  S
symbol =  hermanito
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'NN']

top =  NN
symbol =  hermanito
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'hermanito']

top =  hermanito
symbol =  hermanito
top adalah simbol terminal
isi stack: ['#', 'NN', 'VB']

top =  VB
symbol =  leer
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'leer']
```

```
top =  leer
symbol =  leer
top adalah simbol terminal
isi stack: ['#', 'NN']

top =  NN
symbol =  novela
top adalah simbol non-terminal
isi stack: ['#', 'novela']

top =  novela
symbol =  novela
top adalah simbol terminal
isi stack: ['#']
isi stack: []

Kesimpulan:
Lexical Analyzer: Valid
Parser: Valid
Kalimat Uji: hermanito leer novela , diterima karena sesuai dengan simbol terminal dan aturan pada grammar
```

- Hasil pengujian kalimat uji yang tidak valid
    1. Kalimat yang diuji: abuelo zapatos usar

```
Kalimat Uji: abuelo zapatos usar

---- Lexical Analyzer ----
current token:  abuelo , valid
current token:   zapatos , valid
current token:   usar , valid
semua token diinput:  abuelo zapatos usar , valid

--------- Parser ---------
top =  S
symbol =  abuelo
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'NN']

top =  NN
symbol =  abuelo
top adalah simbol non-terminal
isi stack: ['#', 'NN', 'VB', 'abuelo']

top =  abuelo
symbol =  abuelo
top adalah simbol terminal
isi stack: ['#', 'NN', 'VB']

top =  VB
symbol =  zapatos
top adalah simbol non-terminal
error

Kesimpulan:
Lexical Analyzer: Valid
Parser: Not Valid
Error, Kalimat Uji: abuelo zapatos usar , tidak diterima karena tidak sesuai dengan aturan pada grammar
```

2. Kalimat yang diuji: beber padre jugo

```
Kalimat Uji: beber padre jugo

---- Lexical Analyzer ----
current token:  beber , valid
current token:   padre , valid
current token:   jugo , valid
semua token diinput:  beber padre jugo , valid

--------- Parser ---------
top =  S
symbol =  beber
top adalah simbol non-terminal
error

Kesimpulan:
Lexical Analyzer: Valid
Parser: Not Valid
Error, Kalimat Uji: beber padre jugo , tidak diterima karena tidak sesuai dengan aturan pada grammar
```

3. Kalimat yang diuji: Leer me novela

```
Kalimat Uji: leer me novela

---- Lexical Analyzer ----
current token:  leer , valid
current token:   me , valid
current token:   novela , valid
semua token diinput:  leer me novela , valid

--------- Parser ---------
top =  S
symbol =  leer
top adalah simbol non-terminal
error

Kesimpulan:
Lexical Analyzer: Valid
Parser: Not Valid
Error, Kalimat Uji: leer me novela , tidak diterima karena tidak sesuai dengan aturan pada grammar
```

Kata-kata pada ketiga kalimat uji diatas sudah valid namun untuk penulisan kalimatnya tidak sesuai dengan grammar yang ditentukan sehingga ketiga kalimat diatas tidak diterima atau tidak valid.

DAFTAR PUSTAKA

Mulyawan,Rifqi. 2019. “Mengenal Apa Itu Pengertian Parse: Apa Itu Parser? Tujuan dan Cara Kerjanya”. https://rifqimulyawan.com/blog/pengertian-parse/ diakses pada 30 mei 2021

Pidi, Diana. 2016. “PARSING (BOTTOM UP)”. https://dianapidi.blogspot.com/2016/12/parsing-bottom-up.html diakses pada 30 mei 2021

Jflap. “Build LL(1) Parse Table”. www.jflap.org/tutorial/grammar/LL/index.html diakses pada 30 mei 2021

Wikipedia. 2019. “Analisis leksikal”. https://id.wikipedia.org/wiki/Analisis_leksikal diakses pada 30 mei 2021