

# **LAPORAN TUGAS BESAR TEORI BAHASA AUTOMATA**

**IF-43-02**

**Anggota Kelompok 1 :**

1. Ananda Affan Fattahila (1301194175)
2. Armadhani Hiro Juni Permana (1301190234)
3. Kaenova Mahendra Auditama (1301190324)

## **1. Spesifikasi Program**

Program ini dapat memvalidasi sebuah kalimat sesuai dengan struktur yang tepat. Struktur Kalimat yang dianggap benar dan sesuai dengan kaidah bahasa inggris adalah sebagai berikut :

Subject -Verb - Object (SVO)

Pada Program ini terdapat batasan kata yang dapat digunakan, dimana batasnya dapat dilihat pada tabel di bawah ini :

Subject	Verb	Object
you, they, we, arts, cats, maps, dogs	love, buy, see	you, they, we, arts, cats, maps, dogs

## 2. Rancangan Context Free Grammar

Untuk membuat program sesuai spesifikasi yang kita butuhkan, kami menggunakan Context Free Grammar untuk membantu kami dalam membuat aturan-aturan yang diperlukan. Aturan-aturan tersebut seperti deskripsi di bawah ini:

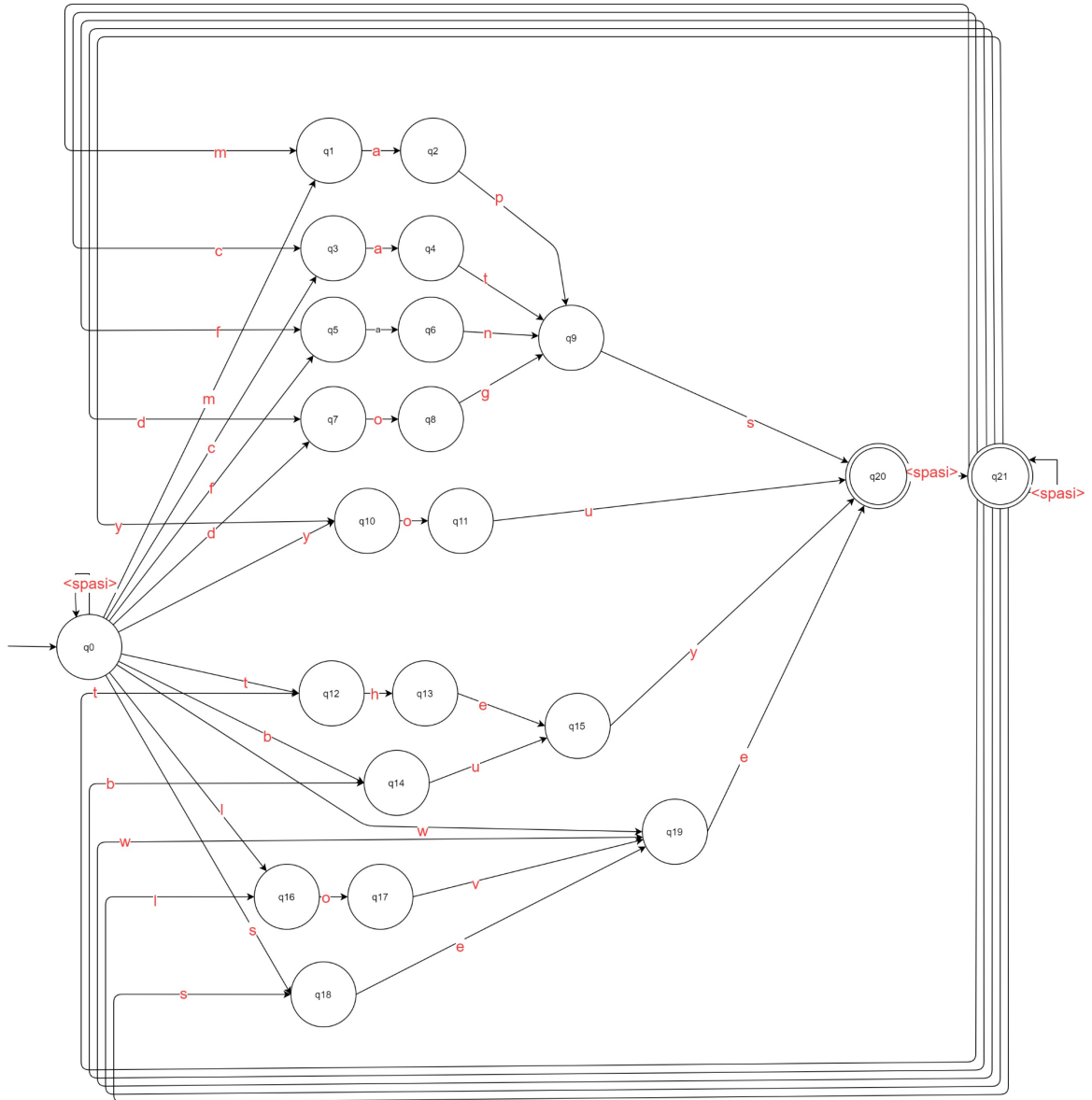
```
G = {V,T,P,S}
V = {NN,VB}
T = {you,they,we,fans,cats,maps,dogs}
S = {S}

CFG
S -> NN VB NN
NN -> you | they | we | fans | cats | maps | dogs
VB -> see | buy | love
```

### 3. Rancangan Finite Automata

Dari kata-kata terminal yang ada pada Context Free Grammar, kita merancang sebuah mesin finite automata dengan diagram transisi dan tabel seperti di bawah:

#### 1. Diagram Transisi



## 2. Tabel Transisi

		current state																						
		q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17	q18	q19	q20	q21	
symbol read	<spasi>	q0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q21	q21	
	a	-	q2	-	q4	-	q6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	b	q14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q14	
	c	q3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q3	
	d	q7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q7	
	e	-	-	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-	q19	q20	-	-	
	f	q5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q5	
	g	-	-	-	-	-	-	-	-	q9	-	-	-	-	-	-	-	-	-	-	-	-	-	
	h	-	-	-	-	-	-	-	-	-	-	-	-	q13	-	-	-	-	-	-	-	-	-	
	i	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	k	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	l	q16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q16	
	m	q1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q1	
	n	-	-	-	-	-	-	q9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	o	-	-	-	-	-	-	q8	-	-	q11	-	-	-	-	-	q17	-	-	-	-	-	-	
	p	-	-	q9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	q	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	r	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	s	q18	-	-	-	-	-	-	-	q20	-	-	-	-	-	-	-	-	-	-	-	-	q18	
	t	q12	-	-	-	q9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q12	
	u	-	-	-	-	-	-	-	-	-	-	q20	-	-	q15	-	-	-	-	-	-	-	-	
	v	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q19	-	-	-	-	-	
	w	q19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q19	
	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	y	q10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q20	-	-	-	-	-	q10	
	z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ACC	ACC	
- = Error																								
ACC = Accept																								

## 4. Rancangan Parse-Table LL (1)

Dari Context Free Grammar yang telah kami rancang dapat di buat menjadi sebuah Parser-Table LL (1) sebagai berikut :

{Parser-Table} **[7 Juni 2021]**

## 5. Rancangan Program

Ada beberapa program yang harus kami buat untuk mendeteksi bahasa yang akan kita periksa dengan benar. Beberapa program diantaranya ialah:

a. Lexical Analyzer

Program ini dikode dengan menggunakan Bahasa Python.

```
# Lexical Analyzer
# Tugas Besar Teori Bahasa Automata
# Code based from Ade Romadhony, School of Computing - Telkom University
import string

# inputting string
print("Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love")
kalimat = input("Masukkan kalimat/kata yang ingin diperiksa: ")
input_string = kalimat.lower() + "#"

# initialization
alphabet_list = list(string.ascii_lowercase)
state_list = [
    'q0', 'q1', 'q2', 'q3', 'q4', 'q5', 'q6', 'q7', 'q8', 'q9', 'q10', 'q11', 'q12',
    'q13', 'q14', 'q15', 'q16', 'q17', 'q18', 'q19', 'q20', 'q21'
]

transition_table = {}

for i in state_list:
    for alphabet in alphabet_list:
        transition_table[(i, alphabet)] = "ERROR"
    transition_table[(i, "#")] = "ERROR"
    transition_table[(i, " ")] = "ERROR"

# CFG
# s -> NN VB NN
# NN -> you | they | we | fans | cats | maps | dogs
# VB -> see | buy | love

# For starting node (q0)
transition_table[("q0", " ")] = "q0"

# Finish state
transition_table[("q20", "#")] = "ACCEPT"
transition_table[("q20", " ")] = "q21"

transition_table[("q21", "#")] = "ACCEPT"
transition_table[("q21", " ")] = "q21"

# string "you"
transition_table[("q21", "y")] = "q10"
transition_table[("q0", "y")] = "q10"
transition_table[("q10", "o")] = "q11"
transition_table[("q11", "u")] = "q20"

# string "they"
transition_table[("q21", "t")] = "q12"
transition_table[("q0", "t")] = "q12"
```

```

transition_table[("q12", "h")] = "q13"
transition_table[("q13", "e")] = "q15"
transition_table[("q15", "y")] = "q20"

# string "we"
transition_table[("q21", "w")] = "q19"
transition_table[("q0", "w")] = "q19"
transition_table[("q19", "e")] = "q20"

# string "fans"
transition_table[("q21", "f")] = "q5"
transition_table[("q0", "f")] = "q5"
transition_table[("q5", "a")] = "q6"
transition_table[("q6", "n")] = "q9"
transition_table[("q9", "s")] = "q20"

# string "cats"
transition_table[("q21", "c")] = "q3"
transition_table[("q0", "c")] = "q3"
transition_table[("q3", "a")] = "q4"
transition_table[("q4", "t")] = "q9"
transition_table[("q9", "s")] = "q20"

# string "maps"
transition_table[("q21", "m")] = "q1"
transition_table[("q0", "m")] = "q1"
transition_table[("q1", "a")] = "q2"
transition_table[("q2", "p")] = "q9"
transition_table[("q9", "s")] = "q20"

# string "dogs"
transition_table[("q21", "d")] = "q7"
transition_table[("q0", "d")] = "q7"
transition_table[("q7", "o")] = "q8"
transition_table[("q8", "g")] = "q9"
transition_table[("q9", "s")] = "q20"

# string "see"
transition_table[("q21", "s")] = "q18"
transition_table[("q0", "s")] = "q18"
transition_table[("q18", "e")] = "q19"
transition_table[("q19", "e")] = "q20"

# string "love"
transition_table[("q21", "l")] = "q16"
transition_table[("q0", "l")] = "q16"
transition_table[("q16", "o")] = "q17"
transition_table[("q17", "v")] = "q19"
transition_table[("q19", "e")] = "q20"

# string "buy"
transition_table[("q0", "b")] = "q14"
transition_table[("q21", "b")] = "q14"
transition_table[("q14", "u")] = "q15"
transition_table[("q15", "y")] = "q20"

# Lexical Analysis
idx_char = 0
state = "q0"

```

```

current_token = ""
while state != "ACCEPT":
    current_char = input_string[idx_char]
    current_token += current_char
    print(state, current_char)
    state = transition_table[(state, current_char)]
    if state == "q20":
        print("current token: {} is valid".format(current_token))
        current_token = ""
    if state == "ERROR":
        print("error")
        break
    idx_char += 1

# Conclusion
if state == "ACCEPT":
    print("semua token yang di input: {} valid".format(input_string))

```

Contoh input/output:

```

kaeno@PCRumah MINGW64 /j/My Drive/Semester 4/Teori Bahasa dan Automata/Tubes/TBA_Tubes (main)
$ python lexical_analyzer.py
Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love
Masukkan kalimat/kata yang ingin diperiksa: you we fans
q0 y
q10 o
q11 u
current token: you is valid
q20
q21 w
q19 e
current token: we is valid
q20
q21 f
q5 a
q6 n
q9 s
current token: fans is valid
q20 #
semua token yang di input: you we fans# valid

```

```

kaeno@PCRumah MINGW64 /j/My Drive/Semester 4/Teori Bahasa dan Automata/Tubes/TBA_Tubes (main)
$ python lexical_analyzer.py
Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love
Masukkan kalimat/kata yang ingin diperiksa: me
q0 m
q1 e
error

```

```

kaeno@PCRumah MINGW64 /j/My Drive/Semester 4/Teori Bahasa dan Automata/Tubes/TBA_Tubes (main)
$ python lexical_analyzer.py
Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love
Masukkan kalimat/kata yang ingin diperiksa: her
q0 h
error

```

```

kaeno@PCRumah MINGW64 /j/My Drive/Semester 4/Teori Bahasa dan Automata/Tubes/TBA_Tubes (main)
$ python lexical_analyzer.py
Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love
Masukkan kalimat/kata yang ingin diperiksa: dams
q0 d
q7 a
error

```

Petunjuk Penggunaan:

1. Lakukan instalasi bahasa pemrograman python
2. Download source code dari link di bawah ini:

[https://github.com/kaenova/TBA\\_Tubes/releases/download/v0.1/lexical\\_analyzer.py](https://github.com/kaenova/TBA_Tubes/releases/download/v0.1/lexical_analyzer.py)

3. Jalankan program tersebut dengan menggunakan python
4. Pada tahap seperti pada gambar di bawah ini, lakukan pengisian kata-kata yang ingin diperiksa oleh lexical analyzer yang telah dibuat oleh kami

```
kaeno@PCRumah MINGW64 /j/My Drive/Semester 4/Teori Bahasa dan Automata/Tubes/TBA_Tubes (main)
$ python lexical_analyzer.py
Terminal words: you | they | we | fans | cats | maps | dogs | see | buy | love
Masukkan kalimat/kata yang ingin diperiksa: |
```

5. Jika output yang diberikan program tersebut terdapat 'error' maka kata atau kalimat yang anda masukkan tidak valid dengan lexical analyzer yang telah kami rancang.

Lalu jalankan program tersebut

b. Parser

{masih dalam proses} [7 Juni 2021]

## 6. Cara Kerja Program

{Penjelasan initial State, Final state dan case error di dalam program}



## 7. Pengujian Program

**Note : Sementara gambaran input dan output kami letakan di atas dahulu (pada point 5) [7 Juni 2021]**

Input	Output Program	Keterangan
{Kalimat}	{screenshot/copy hasil output}	{Valid atau Tidak Valid}

## 8. Dokumentasi Program / Dokumentasi Hasil Output