

# **LEXICAL ANALYZER AND PARSER**

Disusun Untuk Memenuhi Tugas Besar Teori Bahasa Automata



Disusun oleh :

1. Hilman Taris Muttaqin (1301204208)
2. Adel Naufal (1301204208)
3. Muhammad Daffa Fauzan (1301204208)

**FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY**

**2022**

# Daftar Isi

<b>CONTEXT FREE GRAMMAR</b>	<b>3</b>
<b>FINITE AUTOMATA</b>	<b>3</b>
Subjek	3
Predikat	4
Objek	4
Gabungan	5
<b>C. PARSE TABLE (LL)</b>	<b>6</b>
<b>D. IMPLEMENTASI APLIKASI</b>	<b>6</b>
Implementasi aplikasi Lexical Analyzer	6
<b>E. LANGKAH-LANGKAH PENGGUNAAN</b>	<b>11</b>
Website Lexical Analisis	11

## A. CONTEXT FREE GRAMMAR

Context free grammar yang digunakan untuk tugas besar ini adalah Bahasa Sunda.

Definisi dari grammar yang dimaksud adalah sebagai berikut:

**S**  $\rightarrow$  SB VB OB

**SB**  $\rightarrow$  (saya) abdi | (saya) urang | (ibu) indung | (tukang) emang

**VB**  $\rightarrow$  (mengendarai) tumpak | (memakai) nganggo | (memakan) dahar | (memasak) masak

**OB**  $\rightarrow$  (mobil) mobil | (cincin) ali | (roti) roti | (kain) kaen | (kecap) kecap | (bawang) bawang

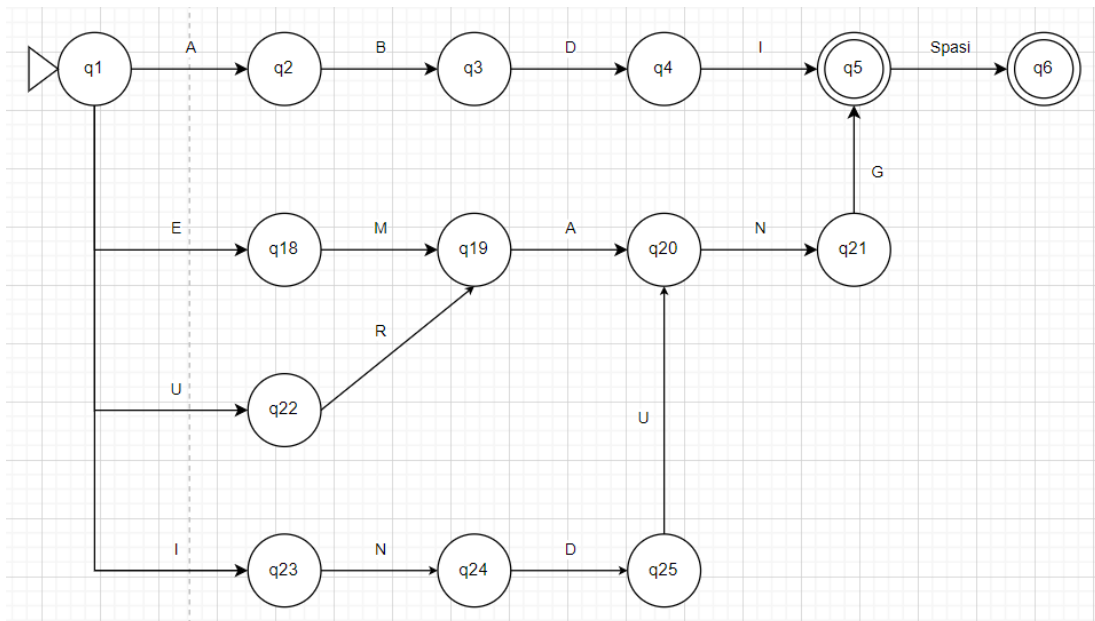
Simbol non-terminal: **S** (starting symbol), **SB** (subjek), **VB** (predikat), **OB** (objek).

Simbol terminal: kata dalam bahasa sunda yang sudah didefinisikan.

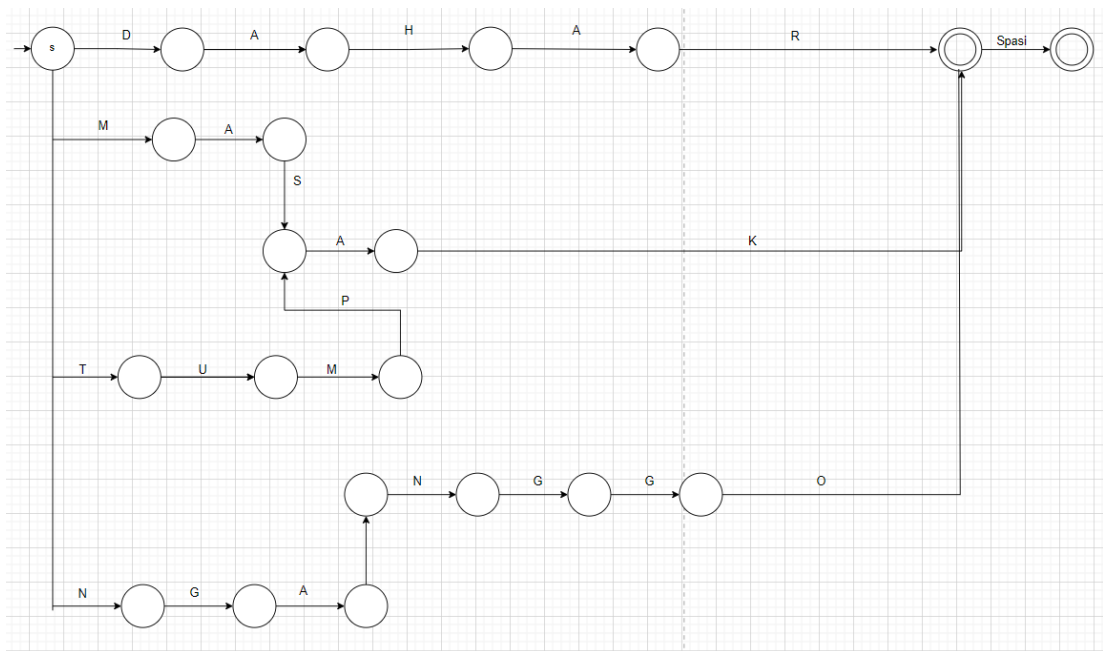
## B. FINITE AUTOMATA

Setelah membuat aturan untuk grammar yang akan digunakan nantinya, langkah selanjutnya adalah membuat finite automata sebagai rules of state yang akan digunakan pada program lexical analyzer yang akan dibuat. Bentuk model finite automata adalah sebagai berikut:

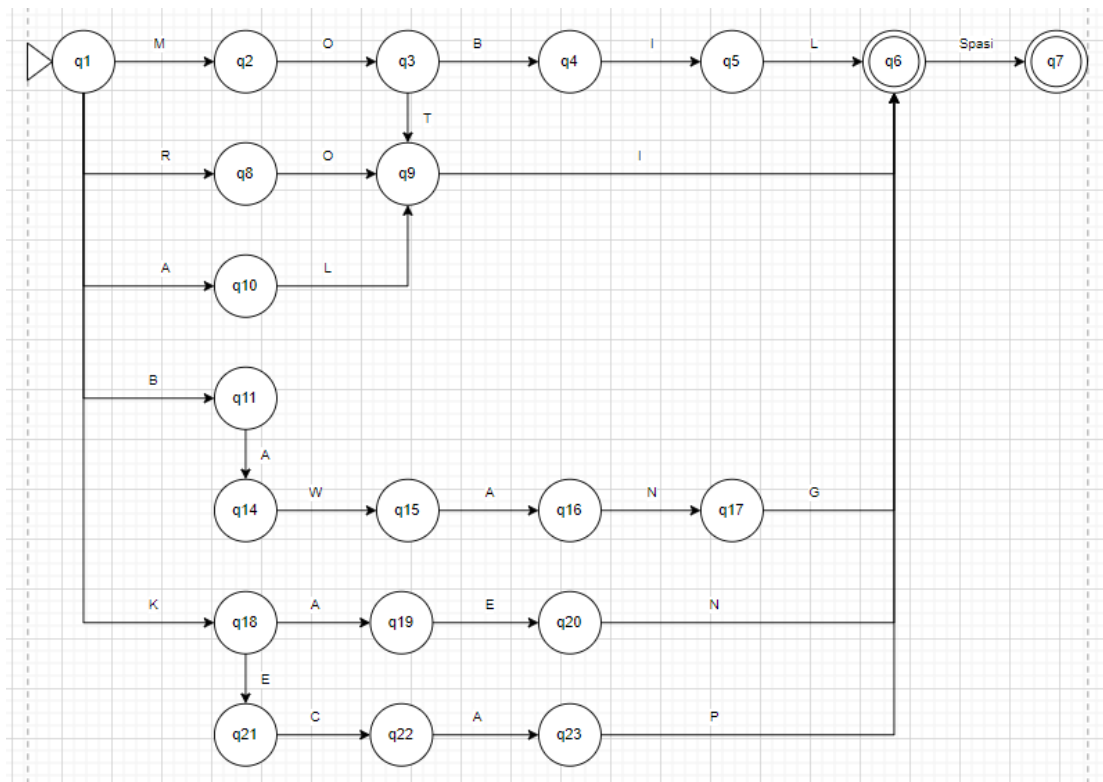
### 1. Subjek



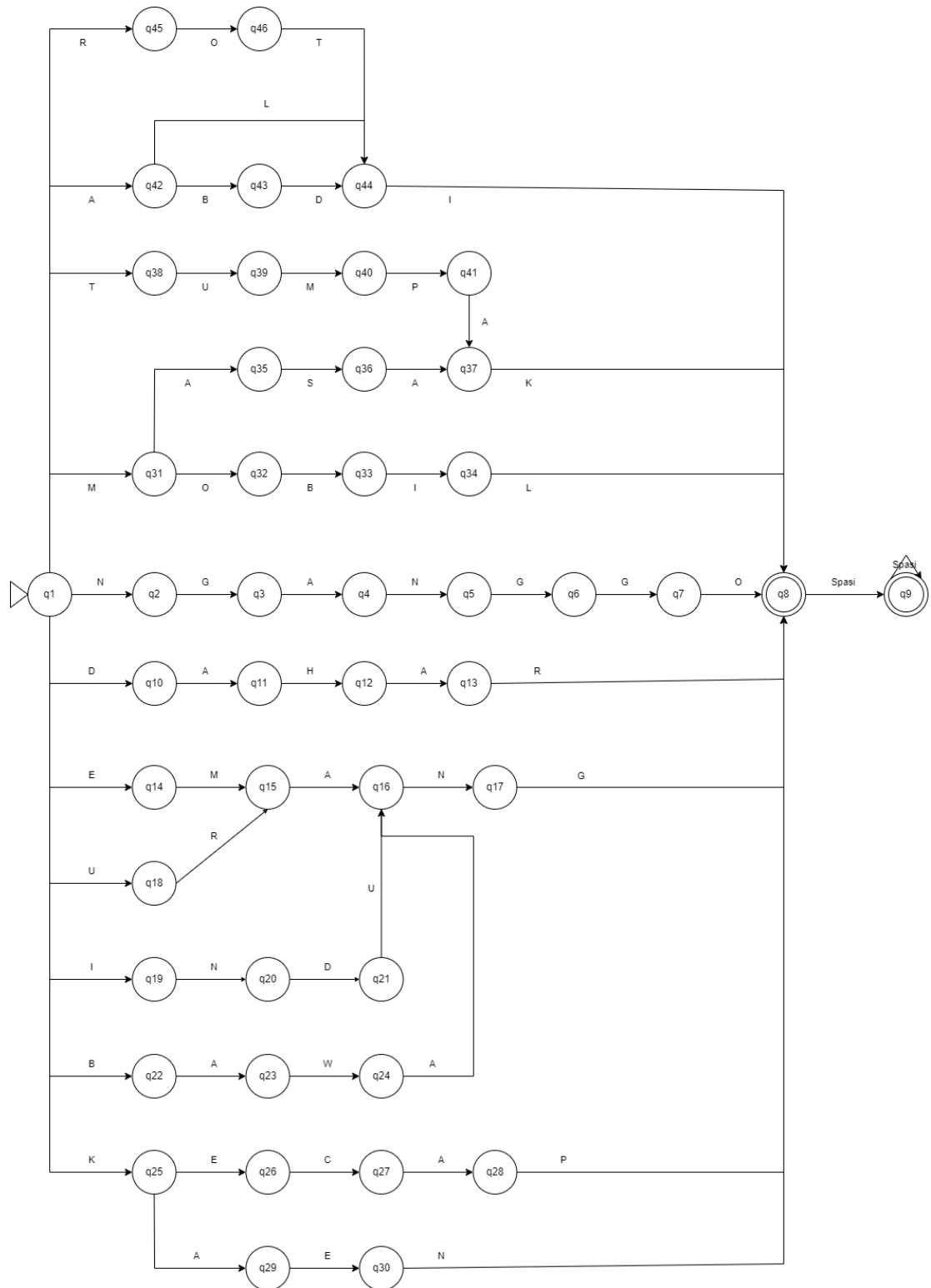
## 2. Predikat



## 3. Objek



#### 4. Gabungan



## C. PARSE TABLE (LL)

	abdi	urang	indung	emang	tumpak	nganggo	dahar	masak	mobil	ali	roti	kaen	kecap	bawang
S	NN VB OB	NN VB OB	NN VB OB	NN VB OB	error	error	error	error	NN VB OB	NN VB OB	NN VB OB	NN VB OB	NN VB OB	NN VB OB
NN	abdi	urang	indung	emang	error	error	error	error	error	error	error	error	error	error
VB	error	error	error	error	tumpak	nganggo	dahar	masak	error	error	error	error	error	error
OB	error	error	error	error	error	error	error	error	mobil	ali	roti	kaen	kecap	bawang

## D. IMPLEMENTASI APLIKASI

Source code:

<https://github.com/Hilmantm/lexical-analyzer-and-parser>

### a. Implementasi aplikasi Lexical Analyzer

Tampilan untuk website

```

index.html M X lexical_analyzer.js README.md U Preview README.md
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <!-- CSS only -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet"
9     integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor" crossorigin="anonymous">
10  <title>Lexical Analyzer</title>
11 </head>
12 <body class="container mt-5">
13
14   <div class="card">
15     <div class="card-body">
16       <form id="form">
17         <h5 class="card-title">Lexical Analyzer</h5>
18         <h6 class="card-subtitle text-muted">Bahasa Sunda</h6>
19         <p class="card-text">Input berupa bahasa sunda dengan kriteria 3 kata seperti subjek, predikat, dan objek</p>
20         <div class="mb-3">
21           <label for="input_kalimat" class="form-label">Input kalimat</label>
22           <input type="text" class="form-control" id="input_kalimat" required placeholder="ex: abdi nganggo mobil">
23         </div>
24         <button type="submit" id="btn-analyze" class="btn btn-primary">
25           Analyze
26           <div class="spinner-border text-light spinner-border-sm ms-2" id="loading" style="display: none; role="status">
27             <span class="visually-hidden">Loading...</span>
28           </div>
29         </button>
30         <div class="mb-3 mt-3">
31           <label for="result" class="form-label">Hasil</label>
32           <input type="text" class="form-control" disabled id="result">
33         </div>
34         <button type="button" id="btn-clear" class="btn btn-danger">
35           Clear
36         </button>
37       </form>
38     </div>
39   </div>
40
41   <!-- JavaScript Bundle with Popper -->
42   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
43     integrity="sha384-pprn3073KE6tl6bjs2QrFajGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2" crossorigin="anonymous"></script>
44   <script type="text/javascript" src="/lexical_analyzer.js"></script>
45 </body>
46 </html>

```

## Logic untuk website lexical analyzer

```
index.html M  lexical_analyzer.js X  README.md U  Preview README.md

lexical_analyzer.js > ...
1  var inputKalimat = document.getElementById('input_kalimat');
2  var hasil = document.getElementById('result');
3  var clear = document.getElementById('btn-clear');
4  var loading = document.getElementById('loading');
5  var form = document.getElementById('form')
6
7  var transition_table = {};
8  var state_list = [];
9
10 var alphabet_list = [];
11 for(var i = 32; i <= 126; i++) {
12     alphabet_list.push(String.fromCharCode( i ))
13 }
14
15 // make state list for 46 state
16 for(let i = 1; i <= 46; i++) {
17     state_list.push('q' + i)
18 }
19
20 for(var state in state_list) {
21     for(alphabet in alphabet_list) {
22         transition_table[[state_list[state], alphabet_list[alphabet]]] = 'error'
23     }
24     transition_table[[state_list[state], '#']] = 'error'
25     transition_table[[state_list[state], ' ']] = 'error'
26 }
27
28 transition_table['q1', ' '] = 'q1'
29
30 // transition for new token
31 transition_table[['q9', 'n']] = 'q2'
32 transition_table[['q9', 'd']] = 'q10'
33 transition_table[['q9', 'e']] = 'q14'
34 transition_table[['q9', 'u']] = 'q18'
35 transition_table[['q9', 'i']] = 'q19'
36 transition_table[['q9', 'b']] = 'q22'
37 transition_table[['q9', 'k']] = 'q25'
38 transition_table[['q9', 'm']] = 'q31'
39 transition_table[['q9', 't']] = 'q38'
40 transition_table[['q9', 'a']] = 'q42'
41 transition_table[['q9', 'r']] = 'q45'
42
43 // transition for final state
44 transition_table[['q8', ' ']] = 'q9'
45 transition_table[['q8', '#']] = 'accept'
46 transition_table[['q9', ' ']] = 'q9'
47 transition_table[['q9', '#']] = 'accept'
48
```

index.html M

lexical\_analyzer.js X

README.md U

Preview README.md

lexical\_analyzer.js > ...

```
48
49 // transition table for: nganggo
50 transition_table[['q1', 'n']] = 'q2'
51 transition_table[['q2', 'g']] = 'q3'
52 transition_table[['q3', 'a']] = 'q4'
53 transition_table[['q4', 'n']] = 'q5'
54 transition_table[['q5', 'g']] = 'q6'
55 transition_table[['q6', 'g']] = 'q7'
56 transition_table[['q7', 'o']] = 'q8'
57
58 // transition table for: dahar
59 transition_table[['q1', 'd']] = 'q10'
60 transition_table[['q10', 'a']] = 'q11'
61 transition_table[['q11', 'h']] = 'q12'
62 transition_table[['q12', 'a']] = 'q13'
63 transition_table[['q13', 'r']] = 'q8'
64
65 // transition table for: emang
66 transition_table[['q1', 'e']] = 'q14'
67 transition_table[['q14', 'm']] = 'q15'
68 transition_table[['q15', 'a']] = 'q16'
69 transition_table[['q16', 'n']] = 'q17'
70 transition_table[['q17', 'g']] = 'q8'
71
72 // transition table for: urang
73 transition_table[['q1', 'u']] = 'q18'
74 transition_table[['q18', 'r']] = 'q15'
75
76 // transition table for: indung
77 transition_table[['q1', 'i']] = 'q19'
78 transition_table[['q19', 'n']] = 'q20'
79 transition_table[['q20', 'd']] = 'q21'
80 transition_table[['q21', 'u']] = 'q16'
81
82 // transition table for: bawang
83 transition_table[['q1', 'b']] = 'q22'
84 transition_table[['q22', 'a']] = 'q23'
85 transition_table[['q23', 'w']] = 'q24'
86 transition_table[['q24', 'a']] = 'q16'
87
88 // transition table for: kecap
89 transition_table[['q1', 'k']] = 'q25'
90 transition_table[['q25', 'e']] = 'q26'
91 transition_table[['q26', 'c']] = 'q27'
92 transition_table[['q27', 'a']] = 'q28'
93 transition_table[['q28', 'p']] = 'q8'
94
95 // transition table for: kaen
96 transition_table[['q25', 'a']] = 'q29'
```



```
index.html M lexical_analyzer.js x README.md U Preview README.md
lexical_analyzer.js > ...
93 transition_table[['q28', 'p']] = 'q8'
94
95 // transition table for: kaen
96 transition_table[['q25', 'a']] = 'q29'
97 transition_table[['q29', 'e']] = 'q30'
98 transition_table[['q30', 'n']] = 'q8'
99
100 // transition table for: mobil
101 transition_table[['q1', 'm']] = 'q31'
102 transition_table[['q31', 'o']] = 'q32'
103 transition_table[['q32', 'b']] = 'q33'
104 transition_table[['q33', 'i']] = 'q34'
105 transition_table[['q34', 'l']] = 'q8'
106
107 // transition table for: masak
108 transition_table[['q31', 'a']] = 'q35'
109 transition_table[['q35', 's']] = 'q36'
110 transition_table[['q36', 'a']] = 'q37'
111 transition_table[['q37', 'k']] = 'q8'
112
113 // transition table for: tumpak
114 transition_table[['q1', 't']] = 'q38'
115 transition_table[['q38', 'u']] = 'q39'
116 transition_table[['q39', 'm']] = 'q40'
117 transition_table[['q40', 'p']] = 'q41'
118 transition_table[['q41', 'a']] = 'q37'
119
120 // transition table for: abdi
121 transition_table[['q1', 'a']] = 'q42'
122 transition_table[['q42', 'b']] = 'q43'
123 transition_table[['q43', 'd']] = 'q44'
124 transition_table[['q44', 'i']] = 'q8'
125
126 // transition table for: ali
127 transition_table[['q42', 'l']] = 'q44'
128
129 // transition table for: roti
130 transition_table[['q1', 'r']] = 'q45'
131 transition_table[['q45', 'o']] = 'q46'
132 transition_table[['q46', 't']] = 'q44'
133
134
135
136 form.onsubmit = (event) => {
137
138     event.preventDefault()
139
140     loading.style = 'display: inline-block'
141 }
```

lexical\_analyzer.js &gt; ...

```
130 transition_table[['q1', 'r']] = 'q45'
131 transition_table[['q45', 'o']] = 'q46'
132 transition_table[['q46', 't']] = 'q44'
133
134
135
136 form.onsubmit = (event) => {
137
138     event.preventDefault()
139
140     loading.style = 'display: inline-block'
141
142     // lexical analysis
143     var indexChar = 0;
144     var state = 'q1';
145     var currentToken = '';
146     var validation = '';
147     var inputChar = inputKalimat.value + '#';
148     console.log(inputChar);
149     while (state != 'accept') {
150         var currentChar = inputChar.charAt(indexChar)
151         currentToken += currentChar
152         state = transition_table[[state, currentChar]]
153         if(state == 'q8') {
154             console.log("valid gais")
155             validation += "valid "
156             currentToken = ''
157         }
158         if(state == 'error') {
159             console.log("error")
160             validation += "error "
161             break;
162         }
163         indexChar += 1
164     }
165
166     console.log(validation);
167     hasil.value = validation.trim();
168
169     loading.style = 'display: none'
170 }
171
172 clear.onclick = (event) => {
173     inputKalimat.value = "";
174     hasil.value = "";
175 }
176
```

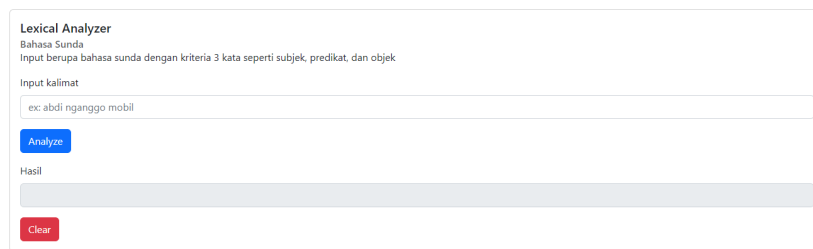
## E. LANGKAH-LANGKAH PENGGUNAAN

### a. Website Lexical Analysis

Clone atau download aplikasi dari <https://github.com/Hilmantm/lexicala-analyzer-and-parser>

Setelah itu buka file **index.html** di browser.

Tampilan akan seperti ini:



The screenshot shows a web application titled "Lexical Analyzer" for "Bahasa Sunda". It includes instructions: "Input berupa bahasa sunda dengan kriteria 3 kata seperti subjek, predikat, dan objek". There is an input field labeled "Input kalimat" containing the example text "ex: abdi nganggo mobil". Below the input field is a blue "Analyze" button. Underneath is a section labeled "Hasil" with a large, empty light-gray rectangular box for the output. At the bottom left of the form is a red "Clear" button.

Setelah berhasil dibuka, maka ketikkan inputan berupa subjek, predikat, dan objek dalam bahasa sunda.

Hasil akan muncul pada field hasil berupa valid atau error.

Contoh input:

1. abdi tumpak mobil

hasil:

valid valid valid

2. abdi numpak mobil

hasil:

valid error

program akan berhenti ketika terdeteksi kata yang error.