

LAPORAN TUGAS BESAR

LEXICAL ANALYZER DAN PARSER

Disusun untuk memenuhi tugas besar mata kuliah Teori Bahasa dan Automata

Dosen Pengampu: Ahmad Suryan, M.T.

Oleh Kelompok 8:

- | | |
|-------------------------------|--------------|
| 1. Naufal Abdurrahman Burhani | (1301204008) |
| 2. Harvian Khusnan Hafidz | (1301204303) |
| 3. Khalillullah Al Faath | (1301204376) |



Fakultas Informatika

Telkom University

2022

Daftar Isi

| | |
|---|----|
| Daftar Isi | 2 |
| 1. Tinjauan Pustaka..... | 4 |
| 1.1. Bahasa Rusia dan Aksara Kiril..... | 4 |
| 1.2. <i>Lexical Analysis</i> | 5 |
| 1.2.1. Token | 5 |
| 1.2.2. Deterministic Finite Automata (DFA) | 5 |
| 1.2.3. Context-free Grammar | 5 |
| 1.3. Parser..... | 6 |
| 1.3.1. LL(k) Parser | 6 |
| 1.3.2. LR(k) Parser..... | 6 |
| 1.3.3. LL(k) Parser vs LR(k) Parser..... | 6 |
| 2. Ketentuan Program | 6 |
| 2.1. Context-free Grammar | 6 |
| 2.2. Finite Automata..... | 7 |
| 2.2.1. FA Subject | 7 |
| 2.2.2. FA Verb | 8 |
| 2.2.3. FA Object..... | 8 |
| 2.2.4. FA Gabungan | 9 |
| 2.3. Parse table | 10 |
| 3. Implementasi Program | 10 |
| 3.1. Variabel-variabel dalam pembangunan web | 10 |
| 3.2. Enumerasi list alfabet dan daftar state..... | 11 |
| 3.3. Daftar Transition Table | 12 |
| 3.4. Lexical Analyzer | 13 |
| 3.5. Daftar symbol terminal dan nonterminal serta inisialisasi parse table..... | 13 |
| 3.6. Daftar Parser Table..... | 14 |
| 4. Hasil Program | 16 |
| 4.1. Tampilan web..... | 16 |
| 4.1.1. Lexical Analyzer | 16 |
| 4.1.2. Parser..... | 16 |

| | | |
|--------|---|----|
| 4.2. | Output program | 16 |
| 4.2.1. | Contoh input valid lexical analyzer | 16 |
| 4.2.2. | Contoh input invalid lexical Analyzer | 17 |
| 4.2.3. | Contoh input valid parser..... | 19 |
| 4.2.4. | Contoh input invalid parser..... | 21 |
| 5. | Resources dan tutorial | 23 |
| 5.1. | Resources | 23 |
| 5.2. | Tutorial | 23 |
| 6. | Bibliography | 25 |

1. Tinjauan Pustaka

1.1. Bahasa Rusia dan Aksara Kiril

Bahasa Rusia adalah bahasa yang termasuk dalam rumpun bahasa Slavia Timur, yang merupakan bahasa yang umumnya digunakan di negara-negara bekas Uni Soviet. Bahasa Rusia juga termasuk ke dalam bahasa resmi PBB (Esipova & Gradirovski, 2008).

Bahasa Rusia umumnya memakai aksara Kiril (bahasa Inggris: cyrillics, rusia: кириллица). Saat ini, alfabet Kiril yang dipakai di Rusia telah dimodifikasi sedemikian rupa dan terdiri atas sebelas huruf vokal, dua puluh konsonan, serta dua penanda keras dan lunak sehingga seluruhnya ada 33 huruf.

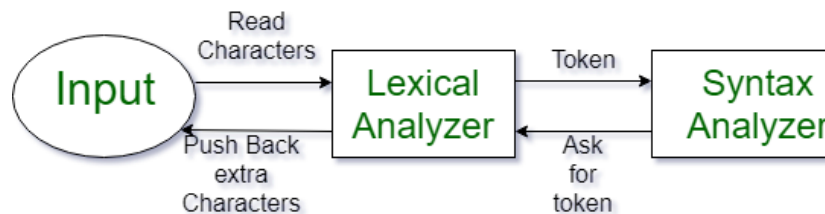
Berikut adalah aksara dalam bahasa Rusia.

| Huruf | Nama huruf | Alih Aksara | Alih Aksara nama huruf |
|-------|------------|-------------|------------------------|
| А а | а | a | <i>a</i> |
| Б б | бэ | b | <i>be</i> |
| В в | вэ | v | <i>ve</i> |
| Г г | гэ | g | <i>ge</i> |
| Д д | дэ | d | <i>de</i> |
| Е е | йэ | ye | <i>ye</i> |
| Ё ё | йо | yo | <i>yo</i> |
| Ж ж | жэ | zh | <i>zhe</i> |
| З з | зэ | z | <i>ze</i> |
| И и | и | i | <i>i</i> |
| Й й | и краткое | y | <i>i kratkoye</i> |
| К к | ка | k | <i>ka</i> |
| Л л | эл | l | <i>el</i> |
| М м | эм | m | <i>em</i> |
| Н н | эн | n | <i>en</i> |
| О о | о | o | <i>o</i> |
| П п | пэ | p | <i>pe</i> |
| Р р | эр | r | <i>er</i> |
| С с | эс | s | <i>es</i> |
| Т т | тэ | t | <i>te</i> |
| У у | у | u | <i>u</i> |
| Ф ф | эф | f | <i>ef</i> |
| Х х | ха | h | <i>ha</i> |
| Ц ц | цэ | ts | <i>tse</i> |
| Ч ч | че | ch | <i>che</i> |
| Ш ш | ша | sh | <i>sha</i> |
| Щ щ | ща | sch | <i>scha</i> |

| | | | |
|-----|-----------------|----|----------------------|
| Ъ ъ | твёрдый знак | * | <i>tvyordiy znak</i> |
| Ы ы | ы | y | <i>y</i> |
| Ь ь | мягкий знак | ^* | <i>myagkiy znak</i> |
| Э э | э | e | <i>e</i> |
| Ю ю | йу | yu | <i>yu</i> |
| Я я | йа | ya | <i>ya</i> |

1.2. Lexical Analysis

Lexical Analysis adalah tahap pertama dalam compiler yang dengannya program input tingkat tinggi diubah menjadi deretan token. *Lexical Analysis* ini dapat diimplementasikan dengan menggunakan Deterministic Finite Automata (DFA).



Outputnya adalah deretan token yang kemudian akan dikirimkan ke parser untuk dilakukan analisis sintaks.

1.2.1. Token

Token adalah kumpulan karakter yang dapat diperlakukan sebagai sebuah unit dalam grammar suatu bahasa (GeeksforGeeks, 2021).

1.2.2. Deterministic Finite Automata (DFA)

DFA (Deterministic Finite Automata) adalah Finite-state Machine atau mesin keadaan terbatas yang menerima atau menolak string dari simbol dan hanya menghasilkan perhitungan unik dari otomata untuk setiap string yang di masukan (Belajar CPP, 2018).

1.2.3. Context-free Grammar

Context-free Grammar (CFG) adalah sebuah grammar yang aturan produksinya berbentuk $A \rightarrow \alpha$, di mana A adalah sebuah symbol non-terminal dan α adalah kumpulan symbol terminal dan non-terminal dengan catatan α bisa merupakan string kosong (John E. Hopcroft, et.al., 2001).

1.3. Parser

Jenis Parser ada 2:

- LL(k) parser
- LR(k) parser

1.3.1. LL(k) Parser

Alasan dinamakan LL parser adalah:

- L pertama menunjukkan string input akan dibaca dari kiri menuju ke kanan.
- L kedua menandakan prase tree yang dibangun akan menghasilkan leftmost derivation

1.3.2. LR(k) Parser

Alasan dinamakan LR parser adalah:

- L menunjukkan string input akan dibaca dari kiri menuju ke kanan.
- R menandakan prase tree yang dibangun akan menghasilkan rightmost derivation.

1.3.3. LL(k) Parser vs LR(k) Parser

- LL(k) parser dikenal dengan nama top-down parser. Dimulai dengan Start simbol pada stack dan secara berulang memecah non terminal pada stack ke dalam komponennnya sampai string simbol cocok dengan input string yang digenerate.
- LR(k) parser diklasifikasikan sebagai bottom-up parser. Dimulai dari aktivitas bersamaan untuk membentuk kejadian non terminal dari symbol-symbol pada string sampai ditemukan start simbol dari grammar yang men-generate.

2. Ketentuan Program

2.1. Context-free Grammar

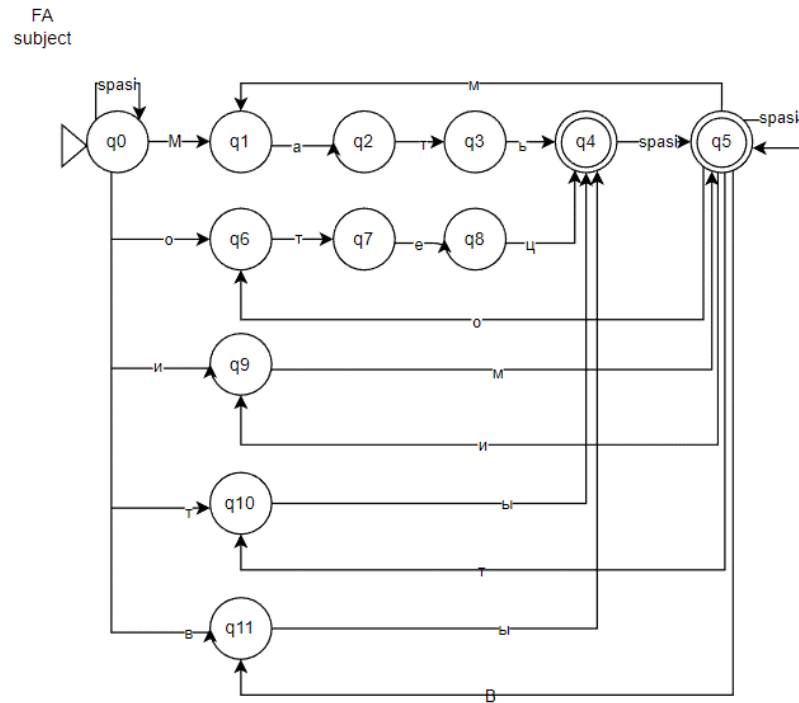
Notasi yang digunakan untuk context-free grammar adalah Backus-Naur form. Berikut ini adalah context-free grammar yang digunakan dengan notasi Backus-Naur form.

```
<S>    ::= <SB> <VB> <OB>
<SB>   ::= "мать" | "отец" | "им" | "ты" | "вы"
<VB>   ::= "готовит" | "уяить" | "тянуть" | "рисовать" | "читать"
<OB>   ::= "книга" | "комар" | "ром" | "ученик"
```

2.2. Finite Automata

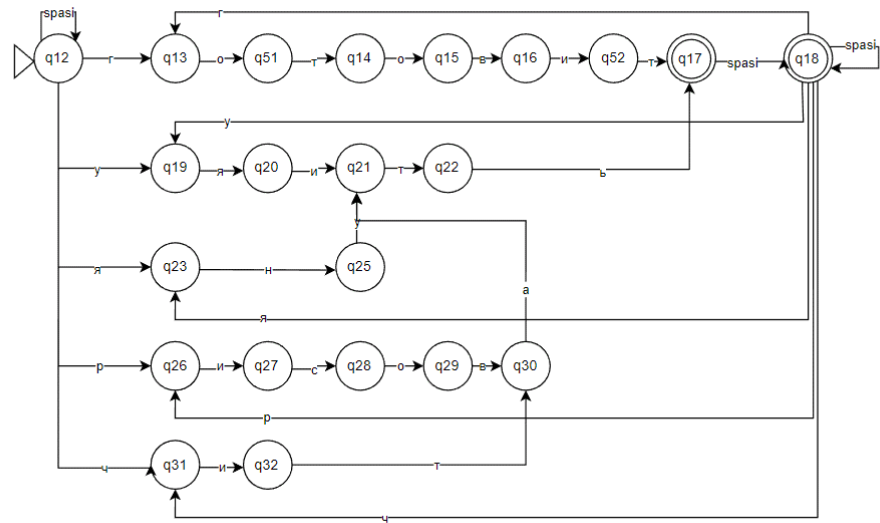
Setelah membuat aturan untuk grammar, langkah selanjutnya yang akan dilakukan adalah membangun finite automata sebagai rules state-state serta value yang akan digunakan pada program lexical analyzer yang akan dibuat. Adapun bentuk model finite automata yang dibuat adalah berikut ini.

2.2.1. FA Subject



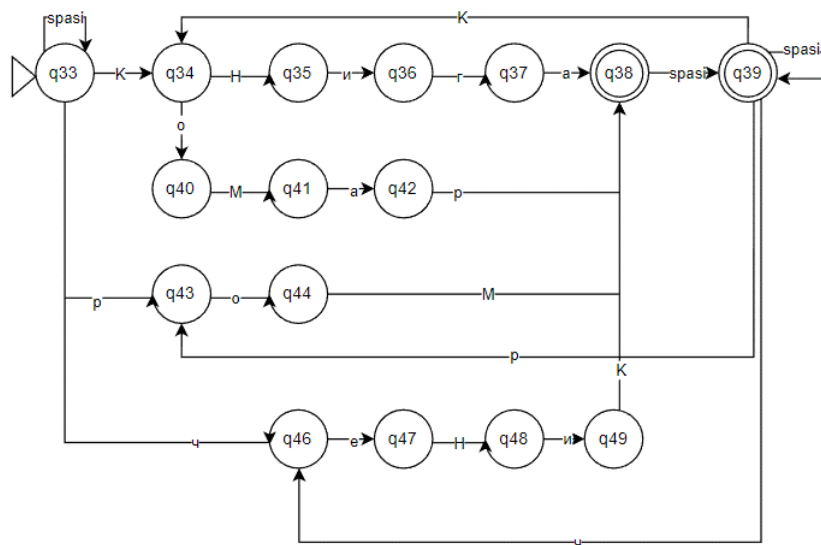
2.2.2. FA Verb

FA verb

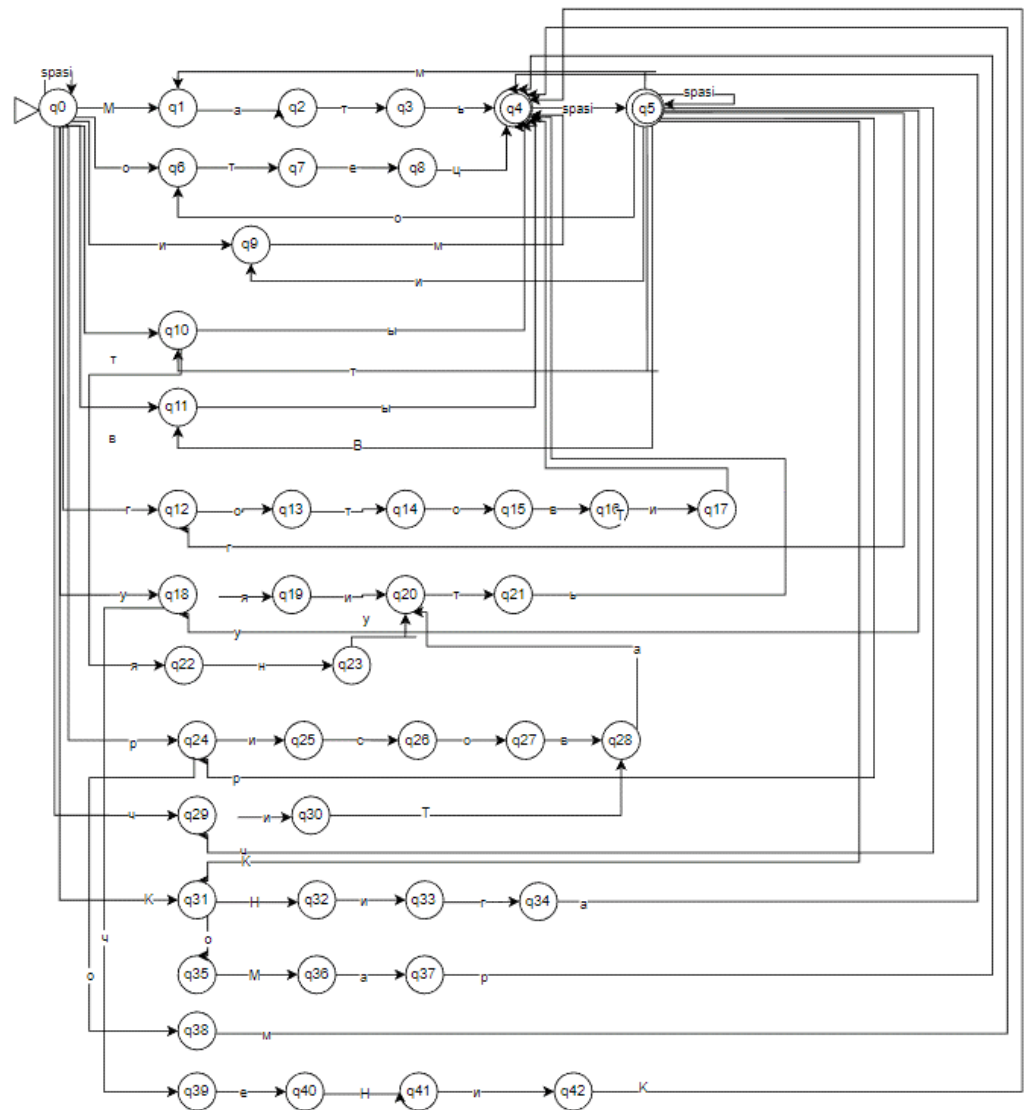


2.2.3. FA Object

FA object



2.2.4. FA Gabungan



2.3. Parse table

| | мать | отец | им | ты | вы | готовит | уяить | рисовать | читать | книга | комар | ром | ученик | EOS |
|----|----------------|----------------|----------------|----------------|----------------|---------|-------|----------|--------|-------|-------|-------|--------|-------|
| S | SB VB OB | SB VB OB | SB VB OB | SB VB OB | SB VB OB | error | error | error | error | error | error | error | error | error |
| SB | мать | отец | им | ты | вы | error | error | error | error | error | error | error | error | error |
| VB | error | error | error | error | error | готовит | уяить | рисовать | читать | error | error | error | error | error |
| OB | error | error | error | error | error | error | error | error | error | книга | комар | ром | ученик | error |

3. Implementasi Program

Program yang dibuat adalah program berbasis web dengan menggunakan bahasa Javascript. Untuk pembangunan websitenya menggunakan HTML dan CSS.

3.1. Variabel-variabel dalam pembangunan web

```
var inputKalimat = document.getElementById('input_kalimat')  
;  
var hasil = document.getElementById('result');  
var clear = document.getElementById('btn-clear');  
var loading = document.getElementById('loading');  
var form = document.getElementById('form')
```

3.2. Enumerasi list alfabet dan daftar state

Karena aksara yang digunakan bukan aksara latin maka perlu mengambil Unicodenya. Untuk aksara Kiril dalam Unicode terdiri dari decimal 1024 sampai 1279. Untuk handle huruf latin maka ditambahkan ke dalam list dengan Unicode dari 65—122.

Dilakukan juga pembentukan state yang terdiri dari “q” yang diikuti angka dari 0—24.

```
var alfabet = [];  
for (let i = 1024; i <= 1279; i++){  
  let text = String.fromCharCode(i);  
  alfabet.push(text);  
}  
  
for (let i = 65; i <= 122; i++){  
  let text = String.fromCharCode(i);  
  alfabet.push(text);  
}  
  
// const alfabet = /^[u0400-\u04FF]+$/;  
// console.log('Пруθim:', alfabet.test('Пруθim'));  
// console.log('Hello:', alfabet.test('Hello'));  
  
var stateList = [];  
  
let N = 42;  
for (let i = 0; i <= N; i++){  
  var strAngka = i.toString();  
  var str = "q";  
  var namaState = str.concat(strAngka);  
  stateList.push(namaState)  
}
```

3.3. Daftar Transition Table

Menginisialisasi untuk setiap alfabet sebagai error. Setelah itu dibuat transition table sesuai dengan FA gabungan.

```
for(var state in stateList) {
    for(daftarAlfabet in alfabet) {
        transitionTable[[stateList[state], daftarAlfabet]] = 'ERROR'
    }
    transitionTable[[stateList[state], '#']] = 'ERROR'
    transitionTable[[stateList[state], ' ']] = 'ERROR'
}
```

```
// initial state
transitionTable[["q0", " "]] = "q0";

// final state
transitionTable[["q4", " "]] = "q5";
transitionTable[["q4", "#"]] = "ACCEPT";
transitionTable[["q5", " "]] = "q5";
transitionTable[["q5", "#"]] = "ACCEPT";

//Subject мать
transitionTable[["q0", "м"]] = "q1"
transitionTable[["q1", "а"]] = "q2"
transitionTable[["q2", "т"]] = "q3"
transitionTable[["q3", "б"]] = "q4"

transitionTable[["q5", "м"]] = "q1"

//subject омега
transitionTable[["q0", "о"]] = "q6"
transitionTable[["q6", "т"]] = "q7"
transitionTable[["q7", "е"]] = "q8"
transitionTable[["q8", "у"]] = "q4"

transitionTable[["q5", "у"]] = "q6"
```

3.4. Lexical Analyzer

Dibuat lexical analyzer mengikut pada tata cara analisis leksikal pada poin 1.

```
form.onsubmit = (event) => {  
  
    event.preventDefault()  
  
    loading.style = 'display: inline-block'  
  
    // lexical analysis  
    var indexChar = 0;  
    var state = 'q0';  
    var currentToken = '';  
    var validation = '';  
    var inputChar = inputKalimat.value + '#';  
    console.log(inputChar);  
    while (state != 'ACCEPT') {  
        var currentChar = inputChar.charAt(indexChar)  
        currentToken += currentChar  
        state = transitionTable[[state, currentChar]]  
        if(state == 'q4') {  
            console.log("valid gais")  
            validation += "valid "  
            currentToken = ''  
        }  
        if(state == 'ERROR') {  
            console.log("ERROR")  
            validation += "ERROR "  
            break;  
        }  
        indexChar++;  
    }  
}
```

3.5. Daftar symbol terminal dan nonterminal serta inisialisasi parse table

```
var nonTerminals = ['S', 'SB', 'VB', 'OB'];  
var terminals = ['мать', 'отец', 'им', 'ты', 'вы', 'готовит',  
    'уяить', 'рисовать',  
    'читать', 'книга', 'комар', 'ром',  
    'ученик' ];  
  
var parse_table = [];
```

3.6. Daftar Parser Table

```
//S
parse_table[['S', 'мать']] = ['SB', 'VB', 'OB']
parse_table[['S', 'отец']] = ['SB', 'VB', 'OB']
parse_table[['S', 'им']] = ['SB', 'VB', 'OB']
parse_table[['S', 'ты']] = ['SB', 'VB', 'OB']
parse_table[['S', 'вы']] = ['SB', 'VB', 'OB']
parse_table[['S', 'готовит']] = ['error']
parse_table[['S', 'уяить']] = ['error']
parse_table[['S', 'рисовать']] = ['error']
parse_table[['S', 'читать']] = ['error']
parse_table[['S', 'книга']] = ['error']
parse_table[['S', 'комар']] = ['error']
parse_table[['S', 'ром']] = ['error']
parse_table[['S', 'ученик']] = ['error']
parse_table[['S', 'EOS']] = ['error']

//SB
parse_table[['SB', 'мать']] = ['мать']
parse_table[['SB', 'отец']] = ['отец']
parse_table[['SB', 'им']] = ['им']
parse_table[['SB', 'ты']] = ['ты']
parse_table[['SB', 'вы']] = ['вы']
parse_table[['SB', 'готовит']] = ['error']
parse_table[['SB', 'уяить']] = ['error']
parse_table[['SB', 'рисовать']] = ['error']
parse_table[['SB', 'читать']] = ['error']
parse_table[['SB', 'книга']] = ['error']
parse_table[['SB', 'комар']] = ['error']
parse_table[['SB', 'ром']] = ['error']
parse_table[['SB', 'ученик']] = ['error']
parse_table[['SB', 'EOS']] = ['error']
```

```
//VB
parse_table[["VB", 'мать']] = ["error"]
parse_table[["VB", 'отец']] = ["error"]
parse_table[["VB", 'им']] = ["error"]
parse_table[["VB", 'ты']] = ["error"]
parse_table[["VB", 'вы']] = ["error"]
parse_table[["VB", 'готовит']] = ['готовит']
parse_table[["VB", 'уяить']] = ['уяить']
parse_table[["VB", 'рисовать']] = ['рисовать']
parse_table[["VB", 'читать']] = ['читать']
parse_table[["VB", 'книга']] = ['erroe']
parse_table[["VB", 'комар']] = ['error']
parse_table[["VB", 'ром']] = ['error']
parse_table[["VB", 'ученик']] = ['error']
parse_table[["VB", 'EOS']] = ['error']

//OB
parse_table[["OB", 'мать']] = ["error"]
parse_table[["OB", 'отец']] = ["error"]
parse_table[["OB", 'им']] = ["error"]
parse_table[["OB", 'ты']] = ["error"]
parse_table[["OB", 'вы']] = ["error"]
parse_table[["OB", 'готовит']] = ['error']
parse_table[["OB", 'уяить']] = ['error']
parse_table[["OB", 'рисовать']] = ['error']
parse_table[["OB", 'читать']] = ['error']
parse_table[["OB", 'книга']] = ['книга']
parse_table[["OB", 'комар']] = ['комар']
parse_table[["OB", 'ром']] = ['ром']
parse_table[["OB", 'ученик']] = ['ученик']
parse_table[["OB", 'EOS']] = ['error']
```

4. Hasil Program

4.1. Tampilan web

4.1.1. Lexical Analyzer

| Kelompok Russia | |
|---|--|
| Kata-Kata inputan Subjek yang dapat diterima : | |
| <div>Lexical Analyzer русский язык Bahasa Rusia</div> <p>Input kata atau kalimat dalam aksara russia</p> <p>Input kalimat</p> <div>ex: ученик</div> <div>Analyze</div> <p>Hasil</p> <div></div> <div>Clear</div> | |

4.1.2. Parser

| |
|---|
| <div>Parser русский язык Bahasa Rusia</div> <p>Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek</p> <p>Input kalimat</p> <div>ex:мать готовит книга</div> <div>Analyze</div> <p>Hasil</p> <div></div> <div>Clear</div> |
|---|

4.2. Output program

4.2.1. Contoh input valid lexical analyzer

Dilakukan pengecekan dengan input sebagai berikut:

- Мать: mengeluarkan output “valid” karena ada pada grammar.
- Уяить: mengeluarkan output “valid” karena ada pada grammar.

- Комар: mengeluarkan output “valid” karena ada pada grammar.
- Комар готовит: mengeluarkan output “valid valid” karena kedua kata ada pada grammar

Kelompok Russia

"им (mereka)", "ты(kamu)", "вы(kalian)" Kata-Kata inputan Verb y

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

valid

Clear

Kelompok Russia

"ка)", "ты(kamu)", "вы(kalian)" Kata-Kata inputan Verb yang dapat

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

valid

Clear

Kelompok Russia

"ку)", "комар (nyamuk)", "ром (rum)", "ученик (murid)"

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

valid

Clear

Kelompok Russia

ng dapat diterima : "книга (buku)", "комар (nyamuk)", "ром (rum)"

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

valid valid

Clear

4.2.2. Contoh input invalid lexical Analyzer

Dilakukan pengecekan input sebagai berikut.

- saya: mengeluarkan output “error” karena tidak ada pada grammar
- makan: mengeluarkan output “error” karena tidak ada pada grammar

- roti: mengeluarkan output “error” karena tidak ada pada grammar
- saya makan roti: mengeluarkan output “error” karena ketiganya tidak ada pada grammar

Kelompok Russia

kan)", "тянуть (menarik)", "рисовать (menggambar)", "читать (me

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

ERROR

Clear

Kelompok Russia

“, "отец (ayah)", "им (mereka)", "ты(kami)", "вы(kalian)" Kata-Kat

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

ERROR

Clear

Kelompok Russia

dapat diterima : "книга (buku)", "комар (nyamuk)", "ром (rum)", "

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

ERROR

Clear

Kelompok Russia

мать (ibu)", "отец (ayah)", "им (mereka)", "ты(kami)", "вы(kalian)

Lexical Analyzer
русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara russia

Input kalimat

Analyze

Hasil

ERROR

Clear

4.2.3. Contoh input valid parser

Dilakukan pengecekan input мать готовит комар:

- Мать: ada dalam grammar dan merupakan subjek.
- Готовит: ada dalam grammar dan merupakan verb.
- Комар: ada dalam grammar dan merupakan objek.
- Kalimat terdiri atas subjek, verb, dan objek, maka ia diterima.

Parser
русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

Analyze

Hasil

Input string мать готовит комар diterima sesuai grammar

Clear

Dilakukan pengecekan input отец рисовать ром:

- отец: ada dalam grammar dan merupakan subjek.
- рисовать: ada dalam grammar dan merupakan verb.
- ром: ada dalam grammar dan merupakan objek.
- Kalimat terdiri atas subjek, verb, dan objek, maka ia diterima.

Parser
русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

Analyze

Hasil

Input string отец рисовать ром diterima sesuai grammar

Clear

Dilakukan pengecekan input им готовит комар

- им: ada dalam grammar dan merupakan subjek.
- готовит: ada dalam grammar dan merupakan verb.
- комар: ada dalam grammar dan merupakan objek.
- Kalimat terdiri atas subjek, verb, dan objek, maka ia diterima.

Parser

русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

им готовит комар

Analyze

Hasil

Input string им готовит комар diterima sesuai grammar

Clear

4.2.4. Contoh input invalid parser

Dilakukan pengecekan inputan мать мать мать:
Keluaran error, karena kata “мать мать мать” tidak sesuai dengan grammar

Parser
русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

Analyze

Hasil

error, tidak diterima karena tidak sesuai grammar

Clear

Dilakukan pengecekan inputan готовит готовит готовит:

Keluaran error, karena kata “готовит готовит готовит” tidak sesuai dengan grammar

Parser

русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

готовит готовит готовит

Analyze

Hasil

error, tidak diterima karena tidak sesuai grammar

Clear

Dilakukan pengecekan inputan рисовать рисовать рисовать:
Keluaran error, karena kata “рисовать рисовать рисовать” tidak sesuai dengan grammar

Parser

русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

рисовать рисовать рисовать

Analyze

Hasil

error, tidak diterima karena tidak sesuai grammar

Clear

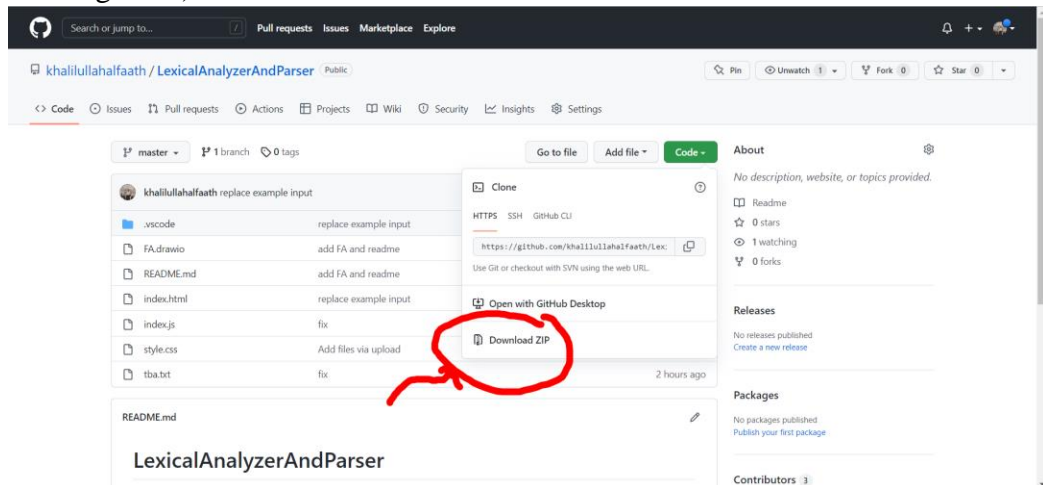
5. Resources dan tutorial

5.1. Resources

Link Github: <https://github.com/khalilullahalfaath/LexicalAnalyzerAndParser>

5.2. Tutorial

1. Download file dari link github pada resources (Jika sudah ada file zip, lompat ke Langkah 3)



2. Ekstrak file .zip dengan winrar atau apapun itu
3. Buka folder hasil ekstrak tadi dan buka file index.html

| Name | Date modified | Type | Size |
|-----------|------------------|-----------------------|--------|
| .vscode | 08/06/2022 17:06 | File folder | |
| FA.drawio | 08/06/2022 01:23 | DRAWIO File | 359 KB |
| index | 08/06/2022 01:23 | Chrome HTML Do... | 3 KB |
| index | 08/06/2022 01:23 | JavaScript Source ... | 6 KB |
| README | 08/06/2022 01:23 | Markdown Source ... | 1 KB |
| style | 08/06/2022 01:23 | CSS Source File | 1 KB |
| tba | 08/06/2022 01:23 | Text Document | 1 KB |

4. Maka tampilan web akan muncul

Kelompok Russia

Kata-Kata inputan Subjek yang dapat diterima : "мать (ibu)" , "отец (ayah)" , "им (mereka)" , "ты (kamu)" , "вы(kalian)" Kata-Kata inputan Verb yang dapat diterima : "готовит"

Lexical Analyzer

русский язык
Bahasa Rusia

Input kata atau kalimat dalam aksara Rusia

Input kalimat

еще мать

Analyze

Hasil

Clear

Parser

русский язык
Bahasa Rusia

Input kalimat dalam bahasa Rusia; kalimat yang diterima terdiri atas subjek predikat objek

Input kalimat

им готовит комп

Analyze

Hasil

Input string им готовит комп diterima sesuai grammar

Clear

5. Masukkan input sesuai yang diinginkan, contoh input yang diterima ada di teks berjalan pada web
6. Contoh inputan valid dan tidak valid ada pada laporan bab 4 (Output program)

6. Bibliography

Esipova, S., & Gradirovski, N. (2008, August 1). *Russian Language Enjoying a Boost in Post-Soviet States*. Retrieved from Gallup.com: <https://news.gallup.com/poll/109228/russian-language-enjoying-boost-postsoviet-states.aspx>