**Laporan Tugas Besar Mata Kuliah Teori Bahasa dan Automata**

Python While-do Parser and Lexical Analyzer

A picture containing logo, text, symbol, graphics

Description automatically generated

**Disusun Oleh :**

IF-45-08

|  |  |
| --- | --- |
| Muhammad Naufal Hawari | 1301213069 |
| Reza Muammar Widyanto | 1301210513 |
| Deaz Setyo Nugroho | 1301210248 |
|  |  |

# CONTEXT FREE GRAMMAR

Pada tugas besar ini kami membuat parser dan lexical analyzer untuk *syntax* while-do pada Bahasa pemrograman Python. Berikut adalah CFG nya:

statement à while <kondisi> : <aksi>

kondisi à <variabel> <comparator> <variabel> | True | False

aksi à <variabel> = <variabel> <operator> <variabel> | print <variabel>

variabel à x | y

comparator à is | in | == | > | >= | < | <= | !=

operator à + | - | \* | \*\* | / | //

# LEXICAL ANALYZER

## Finite Automata

Berikut adalah diagram finite automata untuk *lexical analyzer* terhadap variabel-variabel terminal.

**Gambar 1. Finite Automata untuk Lexical Analyzer**

A picture containing diagram, text, technical drawing, line

Description automatically generated

# PARSE TABLE

Berikut adalah isi dari *parse table* berdasarkan CFG yang sudah dibuat di BAB 1.

**Tabel 1. Parse Table CFG**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | while | : | True | False | = | print | x | y | is | in |
| statement | while <kondisi> : aksi | error | error | error | error | error | error | error | error | error |
| kondisi | error | error | True | False | error | error | <variabel> <comparator> <variabel> | <variabel> <comparator> <variabel> | error | error |
| aksi | error | error | error | error | error | print <variabel> | <variabel> = <variabel> <operator> <variabel> | <variabel> = <variabel> <operator> <variabel> | error | error |
| variabel | error | error | error | error | error | error | x | y | error | error |
| comparator | error | error | error | error | error | error | error | error | is | in |
| operator | error | error | error | error | error | error | error | error | error | error |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | == | | != | > | >= | < | <= | + | - | / | // | \* | \*\* |
| statement | error | | error | error | error | error | error | error | error | error | error | error | error |
| kondisi | | error | error | error | error | error | error | error | error | error | error | error | error |
| aksi | | error | error | error | error | error | error | error | error | error | error | error | error |
| variabel | | error | error | error | error | error | error | error | error | error | error | error | error |
| comparator | | == | != | > | >= | < | <= | error | error | error | error | error | error |
| operator | | error | error | error | error | error | error | + | - | / | error | \* | \*\* |

# IMPLEMENTASI

Implementasi finite automata dan parse table dilakukan menggunakan bahasa pemrograman JavaScript. Penerimaan teks input dilakukan melalui file html. Link kode program dan cara menjalankan program ada di bagian lampiran laporan ini.

|  |  |
| --- | --- |
|  |  |
| **Cuplikan Fungsi Lexical Analyzer** | **Cuplikan Fungsi Parser** |

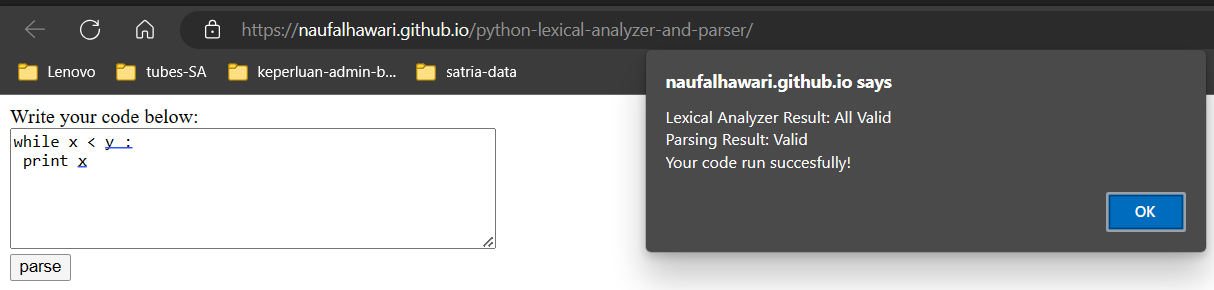
Output program terdiri atas tiga jenis. Jenis pertama adalah input sesuai dengan *grammar* yang sudah ditentukan dan masing-masing variabel terminal sudah sesuai dengan finite automata. Output jenis ini akan memberitahukan bahwa hasil parser adalah valid dan hasil lexical analyzer adalah valid. Jenis kedua adalah masing-masing variabel terminal sesuai dengan finite automata, tetapi tidak sesuai dengan grammar yang telah ditentukan. Output jenis ini akan memberikan informasi bahwa hasil lexical analyzer adalah valid, tetapi hasil parser tidak valid. Jenis ketiga adalah input yang tidak memenuhi grammar dan tidak memenuhi finite automata. Output jenis ini adalah hasil lexical analyzer dan hasil parser tidak valid.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Output Jenis Pertama** | **Output Jenis Kedua** | **Output Jenis Ketiga** |

# HASIL IMPLEMENTASI

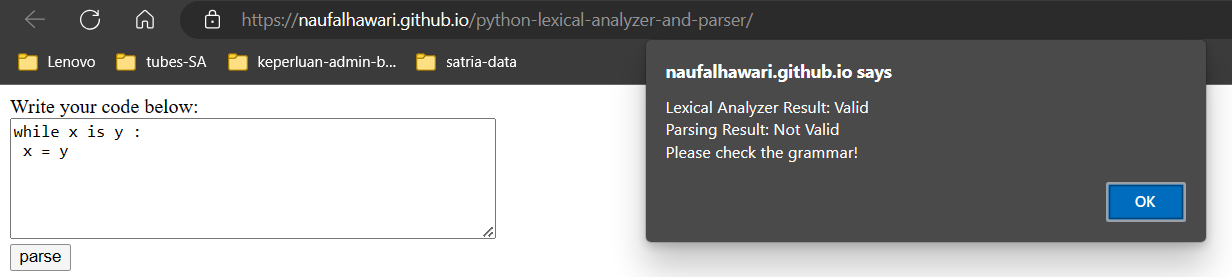
Berikut adalah contoh output jenis pertama. Kode yang dituliskan sudah sesuai dengan *grammar* yang terdefinisi dan masing-masing variabel terminal memenuhi finite automata.

**Gambar 2. Contoh Output Jenis Pertama**



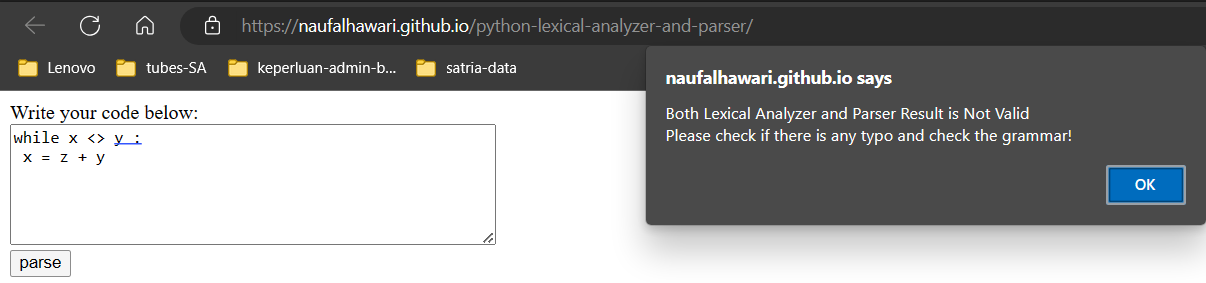
Berikut adalah contoh output jenis kedua ketika semua variabel terminal memenuhi lexical analyzer, tetapi terdapat kesalah grammar. Kesalahan grammar terdapat pada aturan aksi. Aksi seharusnya terdiri atas <variabel> = <variabel> <operator> <variabel>, tetapi aksi pada input hanya terdiri atas <variabel> = <variabel>.

**Gambar 3. Contoh Output Jenis Pertama**



Berikut adalah contoh output hasil ketiga ketika hasil grammar dan lexical analyzer tidak memenuhi karena tidak ada variabel terminal“<>” dan tidak ada variabel terminal “z”.

**Gambar 4. Contoh Output Jenis Pertama**



# LAMPIRAN

Kode program dan cara menjalankan program dapat dilihat di link berikut:

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

**\*Cara menjalankan program dapat dilihat melalui README.md**