|  |
| --- |
| Simple Parser for Arithmatical Expression (Tahap 2/2\*) |
| Laporan Tugas Program TBA |
| Dosen Pengampu:  Kelompok:  Anggota Kelompok:   1. Fedy Fahron Guntara 1301160192 2. Laras Gupitasari 1301160216 3. Axel Haikal Yusuf 1301164228 4. Geyanissa Wanadyawati 1301164265 |

**2017**

[Pick the date]

# Kelompok:

|  |  |  |
| --- | --- | --- |
| **No** | **NIM** | **Nama** |
| 1. | 1301160192 | Fedy Fahron Guntara |
| 2. | 1301160216 | Laras Gupitasari |
| 3. | 1301164228 | Axel Haikal Yusuf |
| 4. | 1301164265 | Geyanissa Wanadyawati |

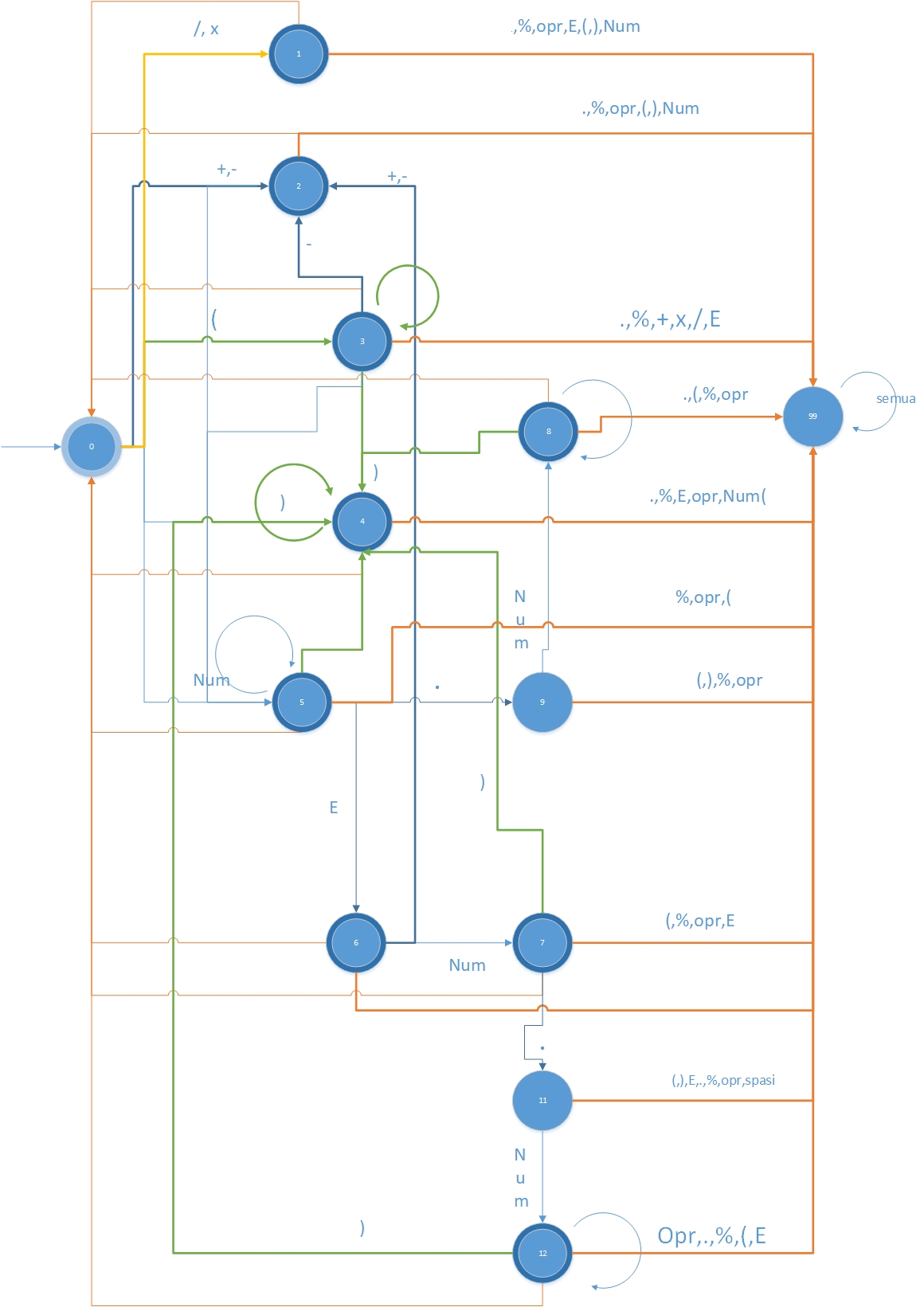
# Spesifikasi Program

Membangun program Lexical Analyzer dengan menerapkan Finite Automata untuk mengenali setiap Lexic yang dituliskan dalam Expresi. Hasil dari Tugas 1 ini adalah sequence Token Lexic berdasarkan string Formula yang dibaca.

# Rancangan Finite Automata

Berikut merupakan STD, definisi formal STD FA, tabel transisi serta pseudo code hasil terjemahan untuk mesin Finite Automata yang diimplementasikan.

* STD FA



Berdasarkan gambar STD FA yang telah dibangun seperti gambar di atas, 0 merupakan inisial state (start state).

State1 akan menerima / , x dari state 0

State 2 akan menerima + , - dari state 0 dan state 6 , dan menerima - dari state 3

State 3 akan menerima ( dari state 0

State 4 akan menerima ) dari state 3, state 5, state 7, state 8 dan state 12

State 5 akan menerima Num dari state 0, state 2 dan state 3

State 6 akan menerima E dari state 5

State 7 akan menerima Num dari state 6

State 8 akan menerima Num dari state 9

State 9 akan menerima . dari state 5

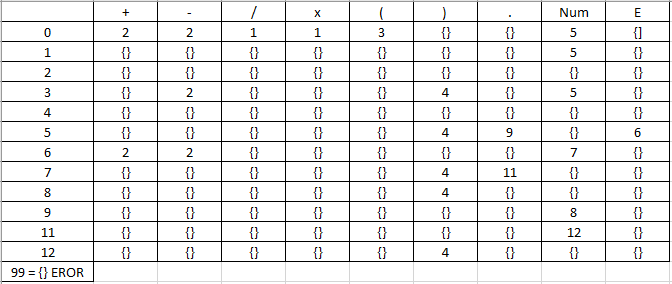
State 11 akan menerima . dari state 7

State 12 akan menerima Num dari state 11

State 0 akan menerima “ “ (spasi) dari semua state kecuali state 9, state 11, dan state 99.

State 1 merupakan final state untuk “/” dan “x”, State 2 untuk “+” dan “-“, State 3 untuk “(“, State 4 untuk “)”, State 5 untuk Num, State 6 untuk E, State 7 untuk Num, State 8 Num, State 12 untuk Num, State 99 = {}, fungsi State 99 dalam FA tersebut digunakan apabila tidak terdapat digit untuk bisa ditranslasi maka state akan langsung menuju State 99 yang berarti terjadi error.

* Tabel Transisi FA



# Cara Kerja Program

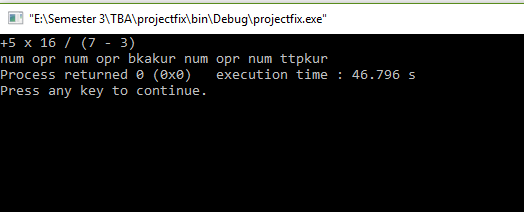
* Penginputan string proposisi lengkap dan di simpan pada variable “Inputan”
* Lakukan pengecekan sub string ( persatu karakter ) dari awal sampai mencapai sub terakhir
* Setiap sub string yang didapat dari variable “Inputan” dilakukan pengecekan yang dibedakan menjadi enam, yaitu simbol kurung atau operator aritmatika, operator aritmatika atau spasi, simbol kurung atau spasi, spasi atau simbol angka, angka atau simbol kurung, operator aritmatika atau angka.
* Setiap state akan melakukan pengecekan terhadap setiap karakter pada sub string.
* Pengoutputan number terjadi setelah kembali ke state 0 setelah pengecekan num.
* Pengoutputan operator aritmatika / dan x terjadi pada statenya yaitu state 1.
* Pengoutputan kurung terjadi di setiap state masing-masing kurung yaitu state 3 dan state 4.
* Setiap operator aritmatika dipisahkan dengan spasi.
* Pengecekan operator + dan – terjadi jika sub inputan tersebut menemukan operator + atau – yang kemudian akan dipisahkan dengan spasi.
* Program juga akan mengoutputkan num jika menemukn nilai E yang mengiringinya.
* Nilai num decimal juga akan dibaca sebagai num.
* Dan juga pengoutputan error dilakukan oleh state 99 jika sub inputan yang dieksekusi tidak sesuai dengan syarat-syarat seperti di atas.

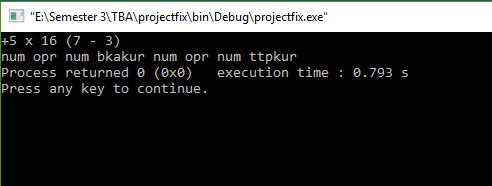
# Pengujian Program

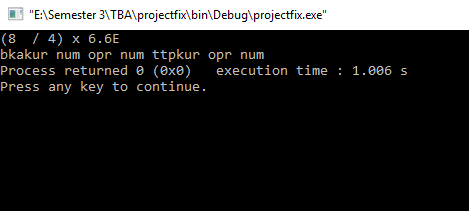
Tuliskan hasil uji terhadap program yang dibangun. Berikan contoh beberapa input string arithmetic expression, kemudian tuliskan output yang dihasilkan program. Jelaskan hasil output tersebut.

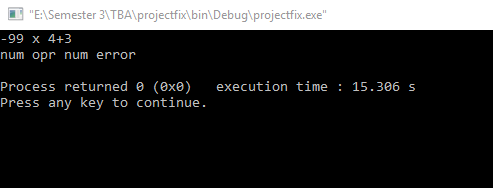
|  |  |  |
| --- | --- | --- |
| **Input** | **Output Program** | **Keterangan** |
| +5 x 16 / (7 – 3) | Num opr num opr bkakur num opr num ttpkur | Sesuai |
| (8 / 4) x 6.6E | bkakur num opr num ttpkur | Sesuai |
| -99 x 4+3 | um opr num error | Sesuai |
| +34E-4.4 + 55 x (-3 + 5 - (-4 + 5) x (3 / 5)) | num opr num opr bkakur num opr num opr bkakur num opr num ttpkur opr bkakur num opr num ttpkur ttpkur | Sesuai |

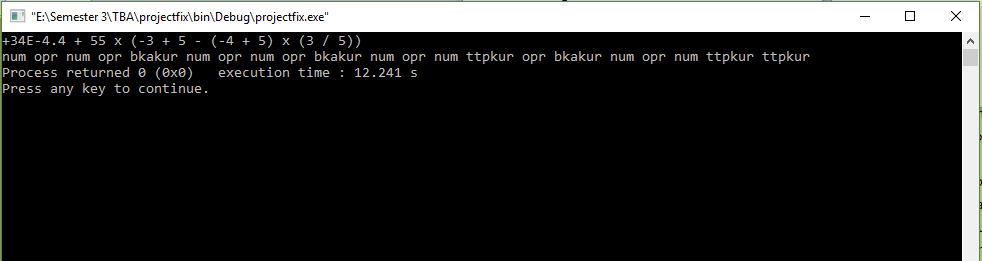
# Screenshot Program







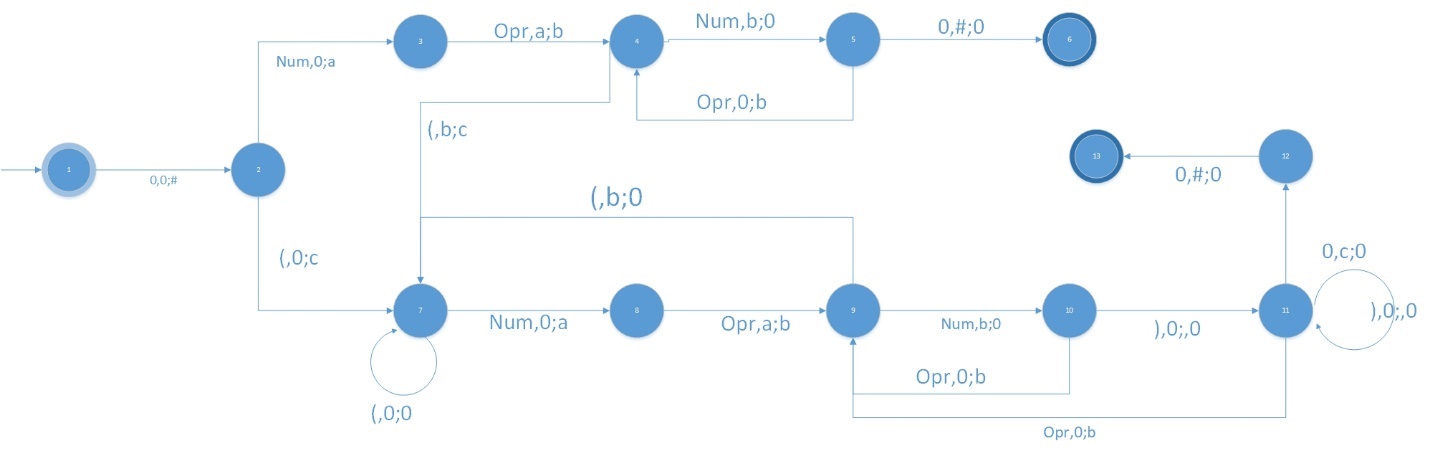




# Rancangan Pushdown Automata

Berikut merupakan Rancangan PDA, tabel transisi serta pseudo code hasil terjemahan yang diimplementasikan.

* PDA



Berdasarkan gambar transisi pada PDA yang telah dibangun seperti gambar di atas,

State 1 merupakan inisial state (start state) yang akan menerima input kosong, melakukan pop kosong, kemudian melakukan push #.

State 2 menerima input Num, melakukan pop kosong, melakukan push a (Num) ke State 3

State 2 menerima input ( , melakukan pop kosong, melakukan push c ( ‘ ( ‘ ) ke State 7

State 3 menerima input opr, melakukan pop a, melakukan push b (opr) ke State 4

State 4 menerima input Num, melakukan pop b, melakukan push kosong ke State 5

State 4 menerima input ( , melakukan pop b, melakukan push c ( ‘ ( ‘ ) ke State 7

State 5 menerima input opr, melakukan pop kosong, melakukan push b (opr) ke State 4

State 5 menerima input kosong, melakukan pop # , melakukan push kosong ke State 6

State 7 menerima input ( , melakukan pop kosong , melakukan push kosong ke State 7

State 7 menerima input Num , melakukan pop kosong , melakukan push a (Num) ke State 8

State 8 menerima input opr, melakukan pop a, memlakukan push b (opr) ke State 9

State 9 menerima input Num, melakukan pop b, melakukan push kosong ke State 10

State 9 menerima input ( , melakukan pop b, melakukan push kosong ke State 7

State 10 menerima input opr, melakukan pop kosong, melakukan push b (opr) ke State 9

State 10 menerima input ) , melakukan pop kosong, melakukan push kosong ke State 11

State 11 menerima input opr, melakukan pop kosong, melakukan push b (opr) ke State 9

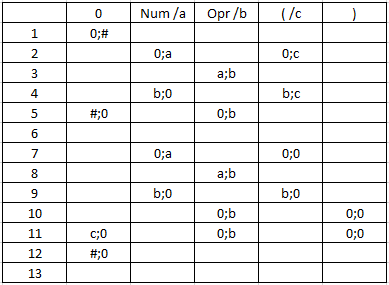
State 11 menerima input ) , melakukan pop kosong, melakukan push kosong ke State 11

State 11 menerima input kosong, melakukan pop c, melakukan push kosong ke State 12

State 12 menerima input kosong, melakukan pop #, melakukan push kosong ke State 13

State 6 dan State 13 merupakan Final State.

* Tabel Transisi PDA



# Cara Kerja Program

* Stack akan melakukan push dan pop sesuai dengan inputan dan state yang menerimanya
* Mengecek apabila stack kosong maka output valid
* atau apabila stack tidak kosong maka output tidak valid

# Pengujian Program

|  |  |  |
| --- | --- | --- |
| **Input** | **Output Program** | **Keterangan** |
| 6E8 x (5E-5 + 6 x (5 x 9)) | Num opr bkakur num opr num opr bkakur num opr num ttpkur ttpkur | VALID |
| 4 x (4 x 8 | Num opr bkakur num opr num | TIDAK VALID |
| 8E4 x (6 x (4 x 6) | Num opr bkakur num opr bkakur num opr num ttpkur | TIDAK VALID |
| 4E5 x (4 x (5 x 5)) | Num opr bkakur num opr bkakur num opr num ttpkur ttpkur | VALID |

# Screenshot Program

# 

# 

# 

# 