**TUGAS II IF2220 TEORI BAHASA FORMAL DAN AUTOMATA**

**APLIKASI CFG DAN PDA PADA PENGENALAN EKSPRESI MATEMATIKA**

**TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2018**

**BAB I**

**DESKRIPSI PERSOALAN**

Persoalan pada tugas ini adalah pembuatan kalkulator menggunakan *Context-Free Grammar* (CFG) dan *Pushdown Automata* (PDA). Kalkulator didefinisikan sebagai alat untuk menghitung. Dengan menggunakan *CFG* atau *PDA*, program dapat memeriksa kebenaran sintaks. Bila benar, maka program akan menampilkan hasil perhitungan, dan bila sintaks salah akan menampilkan SYNTAX ERROR.

    Ada beberapa batasan masalah, yakni :

1. Terminal hanya terdiri dari simbol aritmatika biasa (+, -, \*, /), perpangkatan (^), tanda negatif (-), angka (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), desimal (.), dan tanda kurung (()). Operator hanya terdiri dari simbol aritmatika biasa, tidak mengandung huruf-huruf (e, pi, dan lain-lain). Tidak ada spasi antar token.
2. Program dibuat dalam bahasa C atau Pascal dengan paradigma prosedural
3. Algoritma pemeriksaan sintaks bebas, namun tidak boleh menggunakan *library* terkait CFG/PDA.
4. Dilarang menggunakan algoritma Shunting-Yard untuk kalkulasi.
5. Tuliskan SYNTAX ERROR bilamana ada kesalahan sintaks, dan tuliskan MATH ERROR bila ada kesalahan dalam perhitungan (semisal pembagian dengan nol atau akar negatif)

**BAB 2**

**DESKRIPSI UMUM SOLUSI**

Solusi yang digunakan secara umum adalah :

1. Melakukan pengecekan sintaks dengan PDA.
2. Bila sintaks salah, maka program akan mencetak “SYNTAX ERROR”. Bila benar, maka program akan melakukan kalkulasi menggunakan *Recursive CFG Parser*
3. Bila tidak ada *Math Error,* maka program akan menampilkan hasil perhitungan. Bila terdapat *error* , program akan menampilkan MATH ERROR. Maka, digunakan satu PDA untuk melakukan evaluasi sintaks, dan untuk melakukan perhitungan menggunakan metode *recursive descent*

**BAB 3**

**CFG DAN PDA**

**3.1. Formalitas PDA Pengecekan Sintaks**

Definisi formal dari sebuah PDA adalah :

**PDA = (Q, Σ,Γ,δ, q0 , Z0 , F)**

Q = Daftar *states*

Σ = Daftar alfabet input

Γ = Daftar alfabet *stack*

δ = Fungsi transisi

q0 = *Start state*

Z0  = *Start symbol* ( Z0 ⊆ Γ )

F = *Final states* (F ⊆ Q)

    Fungsi transisi diberikan oleh :

**δ(q, a, Z) ->  (p,α)**

    Untuk pengecekan sintaks, maka PDA yang digunakan memiliki definisi :

**PDA = (Q, Σ,Γ,δ, q0 , Z0 , F)**

Q = {q0, q1,q2}

    Σ = {0,1,2,3,4,5,6,7,8,9, “(“ , “)” , ”+” , ”-” , ”\*” , ”/” , ”^” , ”.”}

    Γ = {“(“,O,X,Y,Z}

    Q0 = qo

    Zo  = Z

F = {q1}

Untuk fungsi transisi, dimisalkan bahwa Σ dapat dibagi dalam beberapa subset :

    Number = {1,2,3,4,5,6,7,8,9}

    Opr = {”+”, ”-” , ”\*” , ”/” , ”^”}

    Berikut adalah daftar fungsi transisi :

    δ(q0, a, Z) = (q0, XZ), a ∈ Number

δ(q0, “(“ , Z) = (q1, XZ )

δ(q0, “(“ , ( ) = (q1, X( )

δ(q1, ε , X) = (q0, (X )

**3.2. Formalitas CFG**

    Definisi formal CFG diberikan oleh :

CFG = (N, T, P, S)

N = himpunan simbol non-terminal

T = himpunan simbol terminal,  N∩T = ø

P =Himpunan aturan produksi

S = Simbol *start*

**3.3 PDA yang Digunakan**



ϵ, X | ϵ

ϵ, ( | ϵ

), X | ϵ

A, Z0 | XZ0

(, Z0 | (X Z0

(, ( | (X(

-. Z0 | Y Z0

A, ( | X(

-, ( | Y(

A, Y | X

A, X | X

Opr, X | O

A, O | X

(, O | (X

(, ( | (X Z0

0, Z0 | 0 Z0

0, X | X

0, ( | 0(

., 0 | .

., X | .

0, Opr | 0

Opr, 0 | O

X, D | D

Opr, D | O



ϵ, X | ϵ

`

**3.4 CFG yang Digunakan**

E 🡪 E + T | E - T | T

T 🡪 T \* P | T / P | P

P 🡪 P ^ F

F -> (E) | n

**BAB IV**

**SOURCE CODE**

**BAB V**

**KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

CFG dan PDA dapat digunakan untuk membuat kalkulator sederhana. Berdasarkan CFG dan PDA yang kita gunakan di atas, kelompok kami menggunakan PDA tersebut untuk melakukan validasi dan CFG untuk melakukan perhitungan. Selain menggunakan CFG dan PDA, masih banyak metode-metode lain yang dapat digunakan untuk melakukan perhitungan pada kalkulator, contohnya adalah metode *Shunting Yard, Earley Algorithm, CYK,* dan lain-lain. Metode CFG dan PDA bukanlah metode yang termudah untuk menyelesaikan perhitungan kalkulator ini, melainkan metode *Shunting Yard* lah yang termudah menurut kami*.*

**5.2 Saran**

Batasan masalah pada tugas besar kali ini adalah kalkulator hanya bisa melakukan perhitungan dengan operasi +, -, \*, /, ^, dan ( ). Untuk pengembangan lebih lanjut, dapat ditambahkan fungsi-fungsi lainnya seperti trigonometri, kalkulator berbasis-n, dan lain-lain agar lebih menyerupai kalkulator matematika pada umumnya.