IF2124 Teori Bahasa Formal dan Otomata

**IMPLEMENTASI PARSER BAHASA JAVASCRIPT (Node.js)**

**Laporan Tugas Besar**

Disusun untuk memenuhi tugas mata kuliah Teori Bahasa Formal dan Otomata

pada Semester 1 (satu) Tahun Akademik 2022/2023.



**Oleh**

**Akmal Mahardika P. 13521070**

**Razzan Daksana Yoni 13521087**

**Muhamad Aji Wibisono 13521171**

**Kelompok TubesWangiWangiÃÃÃHHH<333**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2022**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc119588090)

[BAB I 1](#_Toc119588091)

[DESKRIPSI MASALAH 1](#_Toc119588092)

[BAB II 4](#_Toc119588093)

[TEORI SINGKAT 4](#_Toc119588094)

[2.1 Context Free Grammar 4](#_Toc119588095)

[BAB III 5](#_Toc119588096)

[IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM BAHASA PYTHON 5](#_Toc119588097)

[3.1 Parser CFG ke CYK 5](#_Toc119588098)

[BAB IV 6](#_Toc119588099)

[EKSPERIMEN 6](#_Toc119588100)

[4.1. test.js 6](#_Toc119588101)

[BAB V 7](#_Toc119588102)

[KESIMPULAN, SARAN, DAN REFLEKSI 7](#_Toc119588103)

[5.1 Kesimpulan 7](#_Toc119588104)

[5.2 Saran 7](#_Toc119588105)

[5.3 Refleksi 7](#_Toc119588106)

[DAFTAR REFERENSI 8](#_Toc119588107)

[LAMPIRAN 9](#_Toc119588108)

# BAB I

# DESKRIPSI MASALAH

Dalam proses pembuatan program dari sebuah bahasa menjadi instruksi yang dapat dieksekusi oleh mesin, terdapat pemeriksaan sintaks bahasa atau parsing yang dibuat oleh programmer untuk memastikan program dapat dieksekusi tanpa menghasilkan error. Parsing ini bertujuan untuk memastikan instruksi yang dibuat oleh programmer mengikuti aturan yang sudah ditentukan oleh bahasa tersebut. Baik bahasa berjenis interpreter maupun compiler, keduanya pasti melakukan pemeriksaan sintaks. Perbedaannya terletak pada apa yang dilakukan setelah proses pemeriksaan (kompilasi/compile) tersebut selesai dilakukan.

Dibutuhkan grammar bahasa dan algoritma parser untuk melakukan parsing. Sudah sangat banyak grammar dan algoritma yang dikembangkan untuk menghasilkan compiler dengan performa yang tinggi. Terdapat CFG, CNF-e, CNF+e, 2NF, 2LF, dll untuk grammar yang dapat digunakan, dan terdapat LL(0), LL(1), CYK, Earley’s Algorithm, LALR, GLR, Shift-reduce, SLR, LR(1), dll untuk algoritma yang dapat digunakan untuk melakukan parsing.

Pada tugas besar ini, implementasikan parser untuk JavaScript (Node.js) untuk beberapa statement dan sintaks bawaan JavaScript. Gunakanlah konsep CFG untuk pengerjaan parser yang mengevaluasi syntax program. Untuk nama variabel dan operasi (+, -, >, dll) dalam program, gunakanlah FA.

Algoritma yang dipakai dibebaskan, namun tim asisten menyarankan menggunakan algoritma CYK (Cocke-Younger-Kasami). Algoritma CYK harus menggunakan grammar CNF (Chomsky Normal Form) sebagai grammar masukannya. Oleh karena itu, jika ingin menggunakan CYK buatlah terlebih dahulu grammar dalam CFG (Context Free Grammar), kemudian konversikan grammar CFG tersebut ke grammar CNF.

Berikut adalah daftar kata kunci bawaan JavaScript yang harus terdaftar dalam grammar. Rincian mengenai implementasi dan contohnya dapat dilihat dari Internet:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| break | const | case | catch | continue |
| default | delete | else | false | finally |
| For | function | if | let | null |
| return | switch | throw | try | true |
| Var | while |  |  |  |

Hal-hal ini tidak dimasukkan ke dalam grammar atau diimplementasikan:

1. Semantik dari objek (mis. walaupun objek Foo belum pernah didefinisikan, bar = Foo.attr atau Foo.method() diperbolehkan)
2. Arti semantik dari method (mis. jumlah parameternya)
3. Regex dalam bentuk apapun, seperti r-string (mis. r’123’)
4. Syntactic sugar
5. Karakter-karakter di luar cakupan ASCII.
6. Indentasi
7. Untuk End of statement dibebaskan memakai titik koma atau newline

Spesifikasi Program

1. Tugas dikerjakan menggunakan bahasa pemrograman Python.
2. Program menggunakan Antarmuka berbasis CLI (Command Line Interface).
3. Mahasiswa akan menggunakan konsep CFG dan FA untuk membuat sebuah parser bahasa pemrograman (python).
4. Program akan menerima suatu teks file atau string yang merupakan kode dari sebuah program, lalu program akan memanfaatkan CFG untuk mengevaluasi kebenaran syntax dari kode yang dijadikan input tadi. Program juga menggunakan FA untuk mengecek nama variabel dan ekspresi (contoh : FA akan menolak ‘123dasda’ sebagai nama variabel, karena dalam JavaScript nama variabel tidak diperbolehkan didahului angka, contoh lain : ekspresi ‘55+’ salah karena penempatan operand + harus dijepit di antara 2 variabel valid lainnya)
5. Alur umum dari program yang akan dibangun adalah sebagai berikut.
   1. Menerima input berupa file eksternal berisi string yang merupakan kode sebuah program JavaScript (Node.js)
   2. Melakukan evaluasi sintaks dengan CFG
   3. Melakukan evaluasinama-nama variabel dan operasi yang ada dengan FA
   4. Memberikan keluaran hasil evaluasi program antara “Accepted” jika input diterima atau “Syntax Error” jika input tidak diterima

# BAB II

# TEORI SINGKAT

## 2.1 Context Free Grammar

# BAB III

# IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM BAHASA PYTHON

## 3.1 Parser CFG ke CYK

# BAB IV

# EKSPERIMEN

## 4.1. test.js

# BAB V

# KESIMPULAN, SARAN, DAN REFLEKSI

## 5.1 Kesimpulan

## 5.2 Saran

## 5.3 Refleksi

# DAFTAR REFERENSI

# LAMPIRAN

**Repository Github**

Berikut adalah link repository GitHub untuk program penulis.

**//Belom soalnya gatau format repo**