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 Finite Automata

### 2.1.1 Deskripsi Singkat

Finite State Automata (FSA) adalah suatu mesin abstrak yang diguanakan untuk merepresentasikan pernyelesaian suatu persoalan dari suatu sistem diskrit. Sebagai sebuah mesin maka FSA akan bekerja jika diberi suatu masukan. Hasil proses dari suatu nilai kebenaran diterima atau tidaknya masukan yang diberikan. FSA memiliki state yang banyaknya berhingga, jika diberikan suatu symbol input maka dapat terjadi suatu perpindahan dari sebuah state ke state lainnya. Perubahan state tersebut dinyatakan oleh suatu symbol transisi. Mekanisme FSA tidak memiliki memori sehingga selalu mendasarkan prosesnya pada posisi state “saat ini”. Misalnya pada mekanisme kontrol sebuah lift, selalu didasari pada posisi lift saat itu pada suatu lantai, pergerakan ke atas atau ke bawah dan sekumpulan permintaan yang belum terpenuhi

Finite State Automata merupakan suatu tool yang berguna untuk merancang sistem nyata.

## 2.2 Context Free Grammar

### 2.2.1 Deskripsi Singkat

CFG atau Context Free Grammar adalah tata bahasa formal di mana setiap aturan produksi adalah dalam bentuk A → B di mana A adalah pemproduksi, dan B adalah hasil produksi. Batasannya hanyalah ruas kiri adalah sebuah simbol variabel. Dan pada ruas kanan bisa berupa terminal, symbol, variable ataupun ɛ, Contoh aturan produksi yang termasuk CFG adalah seperti berikut ini:

* X → bY | Za
* Y → aY | b
* Z → bZ | ε

CFG adalah tata bahasa yang mempunyai tujuan sama seperti halnya tata bahasa regular yaitu merupakan suatu cara untuk menunjukkan bagaimana menghasilkan suatu untai-untai dalam sebuah bahasa.

### 2.2.2 Chomsky Normal Form

Bentuk normal Chomsky atau Chomsky Normal Forn (CNF) merupakan salah satu bentuk normal yang sangat berguna untuk Context Free Grammar. Bentuk normal Chomsky dapat dibuat dari sebuah tata bahasa bebas konteks yang telah mengalami penyederhanaan yaitu penghilangan produksi useless, unit, dan ε. Dengan kata lain, suatu tata bahasa bebas konteks dapat dibuat menjadi bentuk normal Chomsky dengan syarat tata bahasa bebas kontesk tersebut:

* Tidak memiliki produksi useless
* Tidak memiliki produksi unit
* Tidak memiliki produksi ε

Bentuk normal Chomsky (Chomsky Normal Form, CNF) adalah Context Free Grammar (CFG) dengan setiap produksinya berbentuk :

A → BC atau A → a.

Langkah-langkah pembentukan bentuk normal Chomsky secara umum sebagai berikut:

* Biarkan aturan produksi yang sudah dalam bentuk normal Chomsky
* Lakukan penggantian aturan produksi yang ruas kanannya memuat simbol terminal dan panjang ruas kanan > 1
* Lakukan penggantian aturan produksi yang ruas kanannya memuat > 2 simbol variabel
* Penggantian-penggantian tersebut bisa dilakukan berkali-kali sampai akhirnya semua aturan produksi dalam bentuk normal Chomsky
* Selama dilakukan penggantian, kemungkinan kita akan memperoleh aturan-aturan produksi baru, dan juga memunculkan simbol-simbol variabel baru

### 2.2.3 Algoritma Cocke–Younger–Kasami

Algoritma Cocke-Younger-Kasami (CYK) merupakan algoritma parsing dan keanggotaan (membership) untuk tata Bahasa bebas konteks. Algortima ini diciptakan oleh J. Cocke, DH. Younger, dan T. Kasami. Syarat untuk penggunaan algortima ini adalah tata bahasa harus berada dalam bentuk normal Chomsky Obyektif dari algortima ini adalah untuk menunjukkan apakah suatu string dapatdiperoleh dari suatu tata bahasa.

Algoritma CYK menggunakan tabel dua dimensi untuk menyimpan hasil keputusan permasalahan yang lebih kecil terlebih dahulu. Sisi Program Dinamis dari algoritma ini terletak pada pembangunan array dua dimensi atau tabel saat memparsing sebuah untai, kemudian ketika parsing untai dilakukan dalam iterasi selanjutnya, algoritma ini akan memanfaatkan array atau tabel yang telah dibangun sebelumnya. Dari tabel yang telah terbentuk, untai yang diparsing dapat diketahui apakah valid, dalam artian CFG tersebut dapat memproduksi untai tersebut melalui aturan-aturan yang ada. Berikut ini adalah persyaratan yang dibentuk dengan mengapikasikan CYK:

* + Input: untai dengan n simbol
  + Output: valid/ tak valid
  + Struktur data: tabel n x n
  + Baris dengan indeks 0 sampai n-1 (atau 1-n dengan modifikasi)
  + Kolom dengan indeks 1 sampai n
  + Sel [i,j] simbol yang yang termasuk dalam untai input

Setiap kali variabel produksi dapat dikembalikan, maka simbol dalam untai berkurang satu. Apabila tidak ada hasil produksi, maka sel diberikan penanda khusus, atau dikosongkan. Apabila untai masih berisi simbol setelah iterasi selesai sampai pada sel batas, maka untai tersebut tidak valid.

CYK merupakan algoritma yang cukup efisien dalam hal mengenali CFG apapun dan merupakan algoritma dasar yang diperkenalkan dalam pemrograman compiler suatu bahasa pemrograman.

## 2.3 Javascript

### 2.3.1 Deskripsi Singkat

JavaScript adalah bahasa pemrograman yang digunakan developer untuk membuat halaman web yang interaktif. Dari menyegarkan umpan media sosial hingga menampilkan animasi dan peta interaktif, fungsi JavaScript dapat meningkatkan pengalaman pengguna situs web. Sebagai bahasa skrip sisi klien, JavaScript adalah salah satu teknologi inti dari World Wide Web. Secara historis, halaman web statis, mirip dengan halaman dalam buku. Halaman statis terutama menampilkan informasi dalam tata letak tetap dan tidak melakukan semua yang kita harapkan saat ini dari situs web modern. JavaScript muncul sebagai teknologi sisi peramban untuk menjadikan aplikasi web lebih dinamis. Dengan JavaScript, browser dapat merespons interaksi pengguna dan mengubah tata letak konten di halaman web.’

### 2.3.2 Node.js

Node.js adalah runtime environment untuk JavaScript yang bersifat open-source dan cross-platform. Dengan Node.js kita dapat menjalankan kode JavaScript di mana pun, tidak hanya terbatas pada lingkungan browser. Node.js menjalankan V8 JavaScript engine (yang juga merupakan inti dari Google Chrome) di luar browser. Ini memungkinkan Node.js memiliki performa yang tinggi. Node.js juga menyediakan banyak library/module JavaScript yang membantu menyederhanakan pengembangan aplikasi web. Node.js dirancang untuk aplikasi dengan proses I/O yang intensif seperti network server atau backend API. Pemrograman dengan multithreading relatif lebih berat dan sulit untuk dilakukan.

### 2.3.2 Syntax Javascript

Sintaks JavaScript adalah seperangkat aturan yang menentukan program JavaScript yang terstruktur dengan benar. JavaScript terdiri dari pernyataan JavaScript yang ditempatkan di dalam tag HTML <script> </script> di halaman web, atau di dalam file JavaScript eksternal yang memiliki ekstensi .js. Contoh berikut menunjukkan bagaimana pernyataan JavaScript terlihat:

|  |
| --- |
| var x = 5;  var y = 10;  var sum = x + y;  document.write(sum); // Prints variable value |

JavaScript sensitif dengan penulisan huruf besar-kecil. Ini berarti bahwa variabel, kata kunci bahasa, nama fungsi, dan pengidentifikasi lainnya harus selalu diketik dengan huruf besar. Sebagai contoh, variabel myVar harus diketik myVar bukan MyVar atau myvar. Demikian pula, nama metode getElementById() harus diketik dengan case yang tepat bukan sebagai getElementByID().

# BAB III

# IMPLEMENTASI PUSTAKA DAN PROGRAM DALAM BAHASA PYTHON

## 3.1 Pembuatan Grammar

## 3.2 Parser CFG ke CNF

## 3.3 Pengaplikasian Algoritma CYK

# BAB IV

# EKSPERIMEN

## 4.1. test.js

# DAFTAR REFERENSI

<https://repository.dinus.ac.id/docs/ajar/file_2013-03-18_071001_dr.r._heru_tjahjana_s.si_m.si__5883745141.pdf>

<http://web.if.unila.ac.id/ilmukomputer/cnf-chomsky-normal-form/>

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2007-2008/Makalah2008/MakalahIF2251-2008-079.pdf>

<https://aws.amazon.com/id/what-is/javascript/>

<https://www.dicoding.com/blog/apa-itu-node-js/>

<https://www.webhozz.com/code/sintaks-javascript/>

# LAMPIRAN

**Repository Github**

Berikut adalah link repository GitHub untuk program penulis.

**Pembagian Tugas**