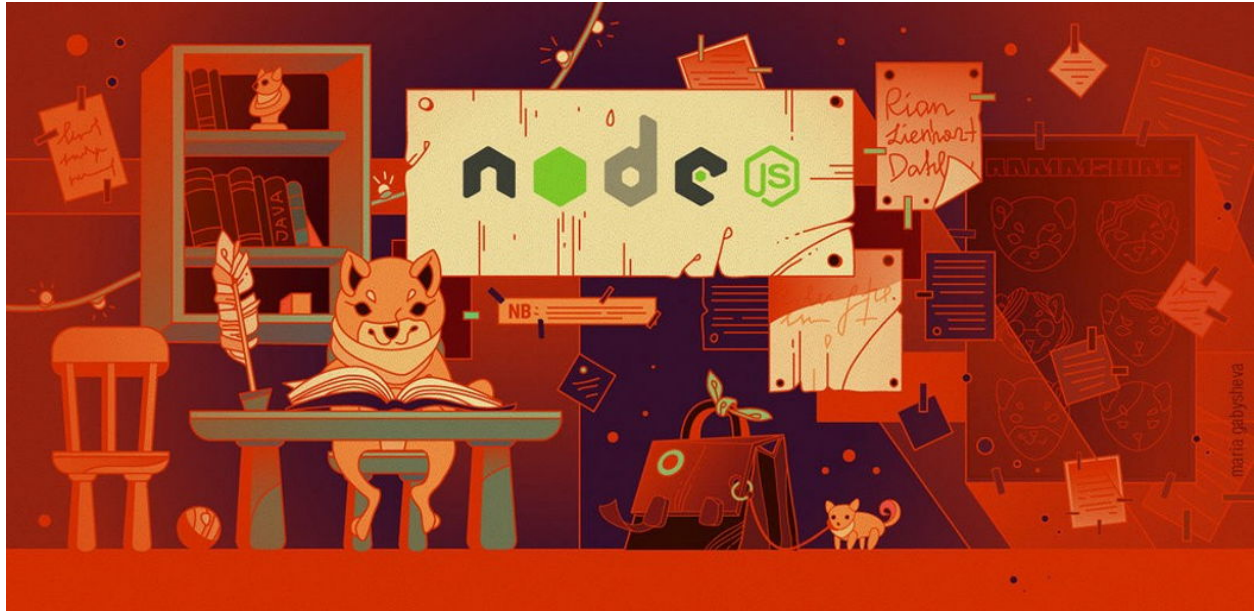


# Tugas Besar IF2124 Teori Bahasa Formal dan Otomata

## *Parser Bahasa JavaScript (Node.js)*

Tanggal Rilis : Jumat, 4 November 2022

Tanggal Pengumpulan : Jumat, 25 November 2022 pukul 23.59 WIB



LINK QnA :

[https://docs.google.com/spreadsheets/d/1wdoJPKM\\_Q4bYNAW7EXH1M5vGi5FSL18DvNo970oaWxQ/e/dit?usp=sharing](https://docs.google.com/spreadsheets/d/1wdoJPKM_Q4bYNAW7EXH1M5vGi5FSL18DvNo970oaWxQ/e/dit?usp=sharing)

### Deskripsi Permasalahan

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<sup>e</sup>, CNF<sup>+e</sup>, 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 atau pada [pranala ini](#)

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 perlu kalian masukkan 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*

## Pengerjaan

Pengerjaan dilakukan secara berkelompok, setiap kelompok terdiri dari maksimal 3 mahasiswa dan tidak boleh lintas kelas, mahasiswa diperkenankan membentuk kelompok masing masing dan mengisi data kelompok di sheets berikut :

<https://docs.google.com/spreadsheets/d/1oDp9Jc13WonMi5ly6CsbTPUgZzXcZlfTJdzMltzsQ1o/edit?usp=sharing>

## 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.
  - a. Menerima **input berupa file eksternal** berisi string yang merupakan kode sebuah program JavaScript (Node.js)
  - b. Melakukan **evaluasi sintaks** dengan CFG
  - c. Melakukan **evaluasi nama-nama variabel dan operasi** yang ada dengan FA
  - d. Memberikan keluaran hasil evaluasi program antara **"Accepted"** jika input diterima atau **"Syntax Error"** jika input tidak diterima
  - e. **(BONUS)** Memberi tahu letak dan detail kesalahan syntax jika ada
6. Buatlah Laporan terkait pengerjaan tugas kalian, laporan wajib memiliki :
  - a. **Teori dasar** FA, CFG, serta penjelasan per-syntax-an JavaScript yang perlu diperhatikan selama pembuatan CFG dan FA.

- b. **Hasil** FA dan CFG yang dibuat.
- c. **Implementasi dan Pengujian**
  - i. Spesifikasi teknis program, termasuk di dalamnya struktur data, fungsi dan prosedur (header fungsi dan prosedur saja, tidak perlu source code), antarmuka, dan lain-lain yang dianggap perlu.
  - ii. *Screenshot* yang memperlihatkan contoh dari berbagai kasus yang muncul (buatlah minimal 3 kemungkinan tipe kasus uji yang mungkin) dan analisis hasilnya
- d. Link ke **repository github**.
- e. Pembagian **Tugas**.

## Contoh Input dan Output Program

Diberikan 2 file .js.

### 1. inputAcc.js

```
// inputAcc.js
function do_something(x) {
  // This is a sample comment
  if (x == 0) {
    return 0;
  } else if (x + 4 == 1) {
    if (true) {
      return 3;
    } else {
      return 2;
    }
  } else if (x == 32) {
    return 4;
  } else {
    return "Momen";
  }
}
```

### 2. inputReject.js

```
// inputReject.js
function do_something(x) {
  // This is a sample multiline comment
  if (x == 0) {
    return 0;
  } else if x + 4 == 1 {
    if (true) {
      return 3;
    } else {
```

```

    return 2;
}
} else if (x == 32) {
    return 4;
} else {
    return "Momen";
}
}
}

```

3. Contoh interaksi program python untuk file "inputAcc.js" pada nomor 1

```
python parserprogram.py "inputAcc.js"
```

**Accepted**

4. Contoh interaksi program python untuk file "inputReject.js" pada nomor 2

```
python parserprogram.py "inputReject.js"
```

**Syntax Error**

**(Output tambahan jika mengerjakan bonus)**  
**Terjadi kesalahan ekspresi pada line 5 : "if x + 4 == 1"**

## Pengumpulan

- Waktu pengerjaan tugas adalah 21 Hari, batas akhir pengumpulan ialah tanggal **25 November jam 23.59 WIB**.
- Pengumpulan dilakukan dengan **melalui form** <https://forms.gle/fVRxWGW9Km2ozHdq9>, buatlah **repository github yang PRIVATE untuk pengerjaan kalian, lalu set ke public paling lambat tepat 24 jam setelah deadline**.
- Kecurangan dan plagiarisme akan mengakibatkan **sanksi ke semua peserta terkait**.
- Demo akan dilakukan setelah pengumpulan. Metode dan peraturan demo akan diberitahukan nanti.