

# Tugas Besar Teori Bahasa dan Automata

Semester Genap 2022/2023

## Membuat Lexical Analyzer dan Parser Sederhana untuk Pengenalan Kode Pemrograman

### Deskripsi:

Pada Tugas Besar ini, mahasiswa (secara berkelompok) diminta untuk mendefinisikan sebuah Context Free Grammar (CFG) yang merepresentasikan aturan bahasa sederhana untuk bahasa pemrograman. Kemudian berdasarkan CFG yang sudah didefinisikan tersebut, mahasiswa diminta untuk membuat sebuah **program sederhana** (diperbolehkan untuk menggunakan bahasa pemrograman apapun) untuk lexical analyzer (mengidentifikasi apakah sebuah lexical/token/kata valid sesuai simbol terminal yang didefinisikan) dan parser (apakah susunan token/kata sudah memenuhi aturan pada Grammar).

Berikut adalah contoh Grammar yang didefinisikan, yang merepresentasikan satu kode bahasa pemrograman.

### Contoh 1: seleksi suatu kondisi dalam pseudocode

Kode program:

```
IF <KONDISI> THEN <AKSI> ENDIF
```

Grammar:

<statement> ::= if <kondisi> then <aksi> endif

<kondisi> ::= <variabel> <operator> <variabel>

<kondisi> ::= true | false

<aksi> ::= <variabel> = <variabel> \* <variabel>

<variabel> ::= a | b

<operator> ::= == | >= | <= | < | >

Simbol non-terminal: <statement>, <aksi>, <kondisi>, <variabel>, <operator>

Simbol terminal: true, false, a, b, ==, \*, if, then, endif, <=, >=, >, <, =

\*\* Catatan: notasi Grammar dituliskan menggunakan notasi BNF (Backus-Naur Form)

### Contoh 2: deklarasi variabel dalam bahasa GO

Kode program:

```
var a int
```

```
var nilai string = "context free grammar"
```

Grammar:

<statement> ::= var <variabel> int <int\_support>

<statement> ::= var <variabel> bool <bool\_support>

<statement> ::= var <variabel> float64 <float\_support>

<statement> ::= var <variabel> string <str\_support>

<data\_type> ::= int | bool | string | float64

Simbol non-terminal: Statement (starting symbol), <variabel>, <int\_support>, <bool\_support>, <float\_support>, <str\_support>, <int\_number>, <float\_number>, <float\_number>, <string\_value>  
 Simbol terminal: int, bool, float64, string, a, nilai, empty, 0, 1, 2, 3, 100, 1000, 2.4, 1.5, halo, b, context

Kemudian berdasarkan Grammar **contoh 1**, didefinisikan Finite Automata untuk mengenali lexical serta parse-table LL(1) sebagai berikut:

- Finite automata yang lengkap untuk mengenali semua variabel terminal. Finite Automata ini akan mengecek apakah penulisan statement valid atau tidak.
- Parse table LL(1). Contoh parse table LL (1) dari CFG pada **contoh 1** dapat dilihat pada Tabel 1.

[illegible]

Berdasarkan Grammar dan parse-table, dibuat program lexical analyzer dan parser. Program akan diberikan sejumlah kalimat uji untuk diidentifikasi apakah lexical pada kalimat uji tersebut valid dan apakah kalimat tersebut sintaksnya sesuai dengan Grammar.

**Tugas yang harus dikerjakan mahasiswa per kelompok (2-3 mahasiswa per kelompok):**

1. Buatlah Context Free Grammar yang merepresentasikan aturan/sintaks kalimat dalam sebuah bahas pemrograman tertentu (lihat Tabel Pembagian Topik dan Kelompok di akhir halaman file ini).
2. Buatlah rancangan Finite Automata untuk lexical analyzer kata/token/symbol terminal sesuai dengan yang didefinisikan pada Grammar.
3. Buatlah rancangan parse-table LL (1) sesuai Grammar yang sudah didefinisikan.
4. Buatlah program lexical analyzer dan parser sederhana sesuai rancangan Finite Automata dan parse table yang sudah dibuat.
5. Lakukan pengujian pada program lexical analyzer dan parser yang sudah dibuat.

**Yang harus dikumpulkan:**

1. Rancangan FA dan CFG yang dibuat dalam 1 file word [POIN maksimum: 50]
2. Program lexical analyzer dan parser yang siap di-run dan dilengkapi penjelasan bagaimana cara menjalankannya. [poin maksimum: 50]
3. 2 poin di atas dikumpulkan pada tanggal **18 Juni 2023** di LMS.

**TABEL PEMBAGIAN TOPIK DAN KELOMPOK**

Kelompok	Nama	Topik Parser	CFG disesuaikan dengan aturan bahasa pemrograman
1	MOHAMMAD BINTANG PUTERA IRAWAN	SELEKSI KONDISI IF – ELSE	C++
	FARREL MAHARDHIKA ALFAITS TSALATS		
2	INDRA WAHYU	PERULANGAN FOR	C++
	BENITO RAYMOND		
3	RAFI ANAND PUNJABI	PERULANGAN WHILE – DO	C++
	IMELDA TYAS MELATI KIYANAYU		
	FATHAN ZHAFIRI ARSHIMNY		
4	SHABRINA SALSABILA	PERULANGAN REPEAT – UNTIL	C++
	NOVIAN EARLY PRADANAPUTRA		
	DIANDRA LINTANG HANINTYA		
5	RAYHAN ADRIAN FADLILLAH	SELEKSI KONDISI IF – ELSE	Java
	MUHAMMAD FARIDZ		
	CHRISTIAN LUKITO PANGESTU		
6	A. MUH. MAULANA FAUSAN MAS`UD IMRAN	PERULANGAN FOR	Java
	ELSAM RAFI SAPUTRA		
	HANROCKY HALIM		
7	R. ADICONDRO YUSUF HENDRATMO	PERULANGAN WHILE – DO	Java
	NAUFAL HILMI MAJID		
	DHIAURIZQI RAMADHANI SANUSI		
8	FIDELA AZIFAH	PERULANGAN REPEAT – UNTIL	Java
	AURA SUKMA AULIA		
	STEVEN CHRISTIAN DARMAWAN		
9	MUHAMMAD FATIH YUMNA LAJUWIRDI LIRRAHMAN	SELEKSI KONDISI IF – ELSE	Go
	TALITHA PRAMA NINDYA		
	AKTSARRIZQY MURAD MAHRAPUTRA		
10	MUHAMMAD ALIF SIROJUDIN	PERULANGAN FOR	Go
	DIVYA ZAHRA SABHIRA		
	RIZA AUFA YUWANATA		
11	AZIS KHOIRUL UMAM	PERULANGAN WHILE – DO	Go
	MUHAMMAD RIZKY HADIYANTO		
	PRINS NAVAL NUZEREN		

12	ARGI PUTRA ZULKARNAEN	PERULANGAN REPEAT – UNTIL	Go
	SYAHBILLA PUTRI ANDINNY		
	FAUZAN ALMAS SAEPULLAH		
13	AHMAD RIZALI	SELEKSI KONDISI IF – ELSE	Python
	ANNISA RAHMA DARMAWAN		
	FAJAR MAULANA KADIR		
14	MUHAMMAD RAFI ARDHANI	PERULANGAN FOR	Python
	KARINA KHAIRUNNISA PUTRI		
	CHIKANDRA PERMATA ZAHIRA		
15	MISBAHUL FADHILLAH	PERULANGAN WHILE – DO	Python
	MUHAMMAD JASIR NABIL		
	CAKRA BUDIMAN PUTRA		
16	AKHDIYAT DEZZA PRASETYO	PERULANGAN REPEAT – UNTIL	Python
	REVA RIVANDI SALIM		
	ANGGITO SETOADI		