

Laboratorium Ilmu Rekayasa dan Komputasi

TEKNIK INFORMATIKA
STEI ITB

IF2224 - Formal Language and Automata Theory

PASCAL-S Compiler

Milestone 1 - Lexical Analysis



IF2224

Milestone 1

Lexical Analysis

Release : Kamis, 9 Oktober 2025

Deadline : Minggu, 19 Oktober 2025 pukul 23.59 WIB

Revisi : -

Daftar Revisi

[DD-MM-YYYY] Revisi X. Keterangan Revisi.

Daftar Isi

Daftar Revisi.....	2
Daftar Isi.....	4
I. Pendahuluan.....	5
II. Teori Singkat.....	6
III. Spesifikasi Tugas Besar.....	7
IV. Teknis Pengerjaan.....	11
V. Pengumpulan.....	13
Referensi.....	14
Lampiran.....	15

I. Pendahuluan



Hello. And welcome to the Los Pollos Hermanos family. My name is Gustavo but you can call me Gus.

Selamat datang kembali ke mata kuliah IRK. Disini kami ingin memperkenalkan ke *client* kita yang terbaru yaitu Gus Fring. Gus Fring adalah pemilik restoran yang bernama Los Pollos Hermanos dan seorang yang sangat paranoid dan ingin membuat rantaian komunikasi yang sangat rahasia. Dia menemukan bahasa pemrograman yang menarik dan dieksekusi untuk memberikan output bernama Pascal. Namun, karena dia secara personal melakukan *filtering* terhadap bahasa ini sehingga terbentuklah bahasa subset dari Pascal (sebut saja Pascal-S).

Kalian dipekerjakan oleh Los Pollos Hermanos dan Gus Fring untuk membuat *compiler* dari Pascal-S. Harapannya hanya orang-orang tertentu yang memiliki *compiler* ini sehingga hanya mereka yang bisa mengetahui pesannya Gus Fring. Takut? Iya kami juga. Oleh karena itu, ikuti instruksi yang ada di bawah supaya kita bersama-sama tidak diberikan perjalanan gratis ke Belize. Without further ado, mari kita mulai *journey* pertama di TBFO and *the beginning of the end of your journey with IRK* 😢😢😢.

II. Teori Singkat

Bahasa pemrograman adalah bahasa yang selalu digunakan untuk melakukan operasi-operasi yang melibatkan komputasi. Terkait bahasa pemrograman ini, proses untuk mengeksekusi bahasa pemrograman tersebut tidaklah mudah. Terdapat banyak proses yang perlu dilakukan oleh *compiler* supaya kode yang ditulis benar dan bisa memberikan output yang sesuai dengan bahasa pemrograman yang ditulis. Tahapan pertama yang akan dikerjakan dalam tugas besar mata kuliah IF2224 Teori Bahasa Formal dan Otomata ini adalah membuat *lexical analyzer* (lexer).

Analisis leksikal (lexical analysis) adalah tahap pertama dalam proses kompilasi atau interpretasi bahasa pemrograman. Pada tahap ini, *compiler* membaca kode sumber mentah yang pada dasarnya hanyalah rangkaian karakter dan mengubahnya menjadi rangkaian satuan makna yang disebut token. Token merupakan unit terkecil yang memiliki arti dalam sebuah program, misalnya kata kunci (keyword), nama variabel (identifier), operator, angka, atau tanda baca.

Tujuan utama dari analisis leksikal adalah untuk mempermudah tahap berikutnya, yaitu parsing. Alih-alih berurusan langsung dengan karakter mentah, tahap parsing bekerja dengan kumpulan token yang sudah terstruktur, sehingga lebih mudah memahami susunan logis dari program. Pada proses ini, analisis leksikal juga membuang bagian yang tidak diperlukan seperti spasi, tab, dan komentar, serta dapat mendeteksi kesalahan seperti simbol yang tidak dikenal sejak awal.

Proses ini dilakukan oleh komponen yang disebut lexer. Lexer membaca kode sumber dari kiri ke kanan dan menggunakan pola tertentu untuk mengenali jenis token yang berbeda-beda. Ketika lexer menemukan urutan karakter yang cocok dengan salah satu pola tersebut, ia membuat sebuah token yang biasanya berisi dua hal utama: jenis token dan nilai token.

III. Spesifikasi Tugas Besar

Pada milestone ini, Anda diminta untuk membuat **tahapan pertama dari compiler**, yaitu lexer atau **lexical analyzer**. Lexer berfungsi melakukan analisis leksikal (lexical analysis) dengan menggunakan Deterministic Finite Automata (**DFA**) untuk mengenali pola karakter dalam source code.

Tahapan ini bertujuan untuk mengubah source code Pascal-S dari kumpulan karakter mentah menjadi unit-unit bermakna yang disebut *token*. Setiap token memiliki jenis (type) dan nilai (value), serta akan menjadi masukan bagi tahap berikutnya dalam proses kompilasi.

Masukan	Kode Pascal-S (.pas)
Keluaran	List Token (misalnya VAR(X), ASSIGN(=), OPERATOR(+))
Otomata	Deterministic finite automata (DFA)

Berikut merupakan daftar tipe token dan nilainya dalam bahasa Pascal-S sebagai referensi.

Daftar Token			
No	Tipe (type)	Nilai (value)	Keterangan
1	KEYWORD	program, var, begin, end, if, then, else, while, do, for, to, downto, integer, real, boolean, char, array, of, procedure, function, const, type	Kata kunci yang sudah didefinisikan oleh bahasa Pascal-S dan memiliki fungsi khusus dalam struktur program.
2	IDENTIFIER	x, y, z, sum, avg, count	Nama yang didefinisikan oleh pengguna, misalnya nama variabel, prosedur, atau fungsi.

3	ARITHMETIC_OPERATOR	+, -, *, /, div, mod	-
4	RELATIONAL_OPERATOR	=, <>, <, <=, >, >=	-
5	LOGICAL_OPERATOR	and, or, not	-
6	ASSIGN_OPERATOR	:=	Operator penugasan yang digunakan untuk memberi nilai ke variabel
7	NUMBER	22, 3, 2018	Bilangan berupa integer atau ril
8	CHAR_LITERAL	'a', 'b', 'c'	-
9	STRING_LITERAL	'tbfo', 'seru sekali'	-
10	SEMICOLON	;	-
11	COMMA	,	-
12	COLON	:	-
13	DOT	.	-
14	LPARENTHESIS	(-
15	RPARENTHESIS)	-
16	LBRACKET	[-
17	RBRACKET]	-
18	RANGE_OPERATOR	..	-
19	COMMENT_START	{, (*	-
20	COMMENT_END	}, *)	-

Program harus menggunakan DFA, dengan **file aturan DFA disimpan dalam format .txt atau .json** dan dibaca oleh program. Proses scanning dilakukan huruf demi huruf pada source code Pascal-S.

Input Command (contoh yang digunakan menggunakan Python)
Format: python [Compiler] [Kode Pascal]

```
python compiler.py program.pas
```

Contoh file aturan DFA (harus dengan format [State Awal] [Input] [State Selanjutnya] dan ada pendefinisan start state dan final state)

```
# DEFINE THE START STATE AND FINAL STATE
Start_state = State_0
Final_state = State_x, State_y, State_z

State_0 a State_1
State_1 b State_1
.
.
```

Masukan (input)

```
program Hello;

var
  a, b: integer;

begin
  a := 5;
  b := a + 10;
  writeln('Result = ', b);
end.
```

Proses (State DFA belum tentu benar)

```
p => char => State 1
r => char => State 1
.
m => char => State 1
=> char => State 2 => Gotten: KEYWORD(program)
.
H => char => State 3
.
; => char => State 0 => Gotten: IDENTIFIER(Hello) & SEMICOLON
.
.
```

Keluaran (output)

```
KEYWORD(program)
IDENTIFIER(Hello)
SEMICOLON(;)
KEYWORD(var)
IDENTIFIER(a)
COMMA(,)
IDENTIFIER(b)
COLON(:)
KEYWORD(integer)
SEMICOLON(;)
KEYWORD(begin)
IDENTIFIER(a)
ASSIGN_OPERATOR(:=)
NUMBER(5)
SEMICOLON(;)
IDENTIFIER(b)
ASSIGN_OPERATOR(:=)
IDENTIFIER(a)
ARITHMETIC_OPERATOR(+)
NUMBER(10)
SEMICOLON(;)
IDENTIFIER(writeln)
LPARENTHESIS(())
STRING_LITERAL('Result = ')
COMMA(,)
IDENTIFIER(b)
RPARENTHESIS())
SEMICOLON(;)
KEYWORD(end)
DOT(.)
```

Selain program, Anda wajib membuat **diagram transisi DFA** yang sesuai dengan aturan pada file tersebut. Diagram dapat dibuat menggunakan alat apa pun (misalnya draw.io) dan diserahkan dalam format **.pdf** beserta **tautan workspace**-nya.

IV. Teknis Pengerjaan

- Bahasa pemrograman yang digunakan **DIBEBASKAN** asal rasionalisasinya dijelaskan (tidak diperbolehkan menggunakan PASCAL).
- Tugas Besar ini dikerjakan secara **berkelompok 4-5 orang** dan **tidak diperbolehkan lintas kelas**.
- Setiap kelompok **WAJIB** mengisi sheets pembagian kelompok. Jika melewati deadline dan Anda tidak mengisi, maka akan dialokasikan secara acak.
 - Deadline Pengisian adalah **Senin, 13 Oktober 2025 pukul 23.59 WIB**.
- Buatlah **Repository Github** dengan format **[KODE]-Tubes-IF2224**, dengan kode adalah Kode Kelompok sesuai sheets pembagian kelompok.
- Repository setidaknya mengandung:
 - Folder **src** untuk menyimpan seluruh **source code**
 - Folder **doc** untuk menyimpan **laporan** dan **gambar diagram**
 - Folder **test** untuk menyimpan **semua input/output** yang dilakukan pada bagian pengujian laporan. Untuk membedakan dengan milestone lain, maka file dipisahkan ke dalam folder untuk setiap milestone.
Sebagai contoh: **`test/milestone-1/output-1.txt`**.
 - **README** yang setidaknya mengandung:
 - Identitas Kelompok
 - Deskripsi Program
 - Requirements
 - Cara Instalasi dan Penggunaan Program
 - Pembagian Tugas
- Anda **WAJIB** untuk membuat **diagram transisi DFA** yang berisi aturan yang digunakan program. Untuk alat implementasi diagramnya dibebaskan.

- Anda **WAJIB** membuat **laporan** dengan format **Laporan-X-[KODE].pdf**, dengan **X** adalah **nomor Milestone** dan **Kode** adalah **Kode Kelompok**.
- **Laporan setidaknya berisi:**
 - Cover dengan foto kelompok menggantikan logo ITB.
 - Landasan Teori
 - Jelaskan teori-teori atau konsep yang berhubungan atau digunakan pada Milestone 1.
 - Perancangan & Implementasi
 - Penjelasan diagram yang dibuat dan digunakan untuk Milestone 1 dan dimasukkan ke dalam laporan..
 - Implementasi yang ditambahkan dan berhubungan dengan Milestone 1.
 - Pengujian
 - Setidaknya 5 pengujian kasus unik.
 - Setiap pengujian harus menunjukan hasil input/output program.
 - Bukti berupa screenshot input/output program.
 - Kesimpulan dan Saran
 - Kesimpulan dari Milestone 1.
 - Lampiran berupa:
 - Link Release Repository Github.
 - Link *workspace* diagram.
 - Pembagian Tugas (menunjukan persentase kontribusi).
 - Referensi yang digunakan selama penggerjaan Milestone 1.
- Jika ada pertanyaan lebih lanjut, silahkan ditanyakan pada [QNA](#).
- Link Pembagian Kelompok: [Sheets Pembagian Kelompok](#)
- Link QNA: [Sheets QNA](#)

V. Penilaian

Program (70 poin)			
No	Komponen	Poin	Syarat Poin Maksimal
1.	Kelengkapan token	10	Program mampu mengenali seluruh jenis token yang ditentukan dalam spesifikasi bahasa Pascal-S
2.	Kesesuaian klasifikasi token	10	Setiap token diidentifikasi dan diklasifikasikan dengan benar sesuai kategori
3.	Struktur/modularitas program	8	Kode ditulis dengan struktur modular menggunakan fungsi atau kelas yang terpisah berdasarkan tanggung jawab. Alur eksekusi program mudah diikuti dan tidak redundan
4.	Error handling	5	Program dapat menangani error dengan baik (pesan informatif dan error tidak menyebabkan crash)
5.	Penerimaan input	3	Program dapat membaca file input (.pas)
6.	Penghasilan output	2	Program menghasilkan output daftar token secara lengkap dan terformat dengan jelas ke terminal
7.	Dokumentasi/komentar	5	Kode diberi komentar yang informatif dan tidak berlebihan
8.	File aturan DFA	12	Terdapat file terpisah yang merepresentasikan DFA sesuai dengan rancangan pada laporan. Program menggunakan file tersebut untuk proses analisis leksikal
9.	Pengujian	15	Program berhasil lolos seluruh kasus uji

Laporan & Diagram (30 poin)			
No	Komponen	Poin	Syarat Poin Maksimal
1.	Penjelasan konsep	10	Penjelasan mengenai konsep lexer yang mencakup pengertian, fungsi, dan peran lexer dalam proses kompilasi, serta kaitannya dengan token dan DFA.
2.	Perancangan program	5	Penjelasan mengenai rancangan program yang mencakup teknologi yang digunakan, struktur program, fungsi/kelas utama, dan alur kerja program. Terdapat pula justifikasi atas pilihan implementasi
3.	Hasil implementasi	2	Dokumentasi mengenai hasil implementasi, dapat berupa tangkapan layar dengan penjelasan
4.	Test case	3	Dokumentasi mengenai pengujian mandiri (minimal 5 dan mencakup berbagai kasus)
5.	Diagram DFA	10	Diagram DFA digambar dengan benar, lengkap, sesuai aturan Pascal-S yang diimplementasikan

VI. Pengumpulan

- **Deliverables:**
 - Laporan
 - Diagram
 - Release Repository
- Pada setiap Milestone, Anda **WAJIB** mengumpulkan **Link menuju Release pada Repository Github.**
 - Gunakan format tag berupa **v0.X.Y** dengan **X** adalah **nomor milestone** dan **Y** adalah **nomor pengumpulan**.
 - Sebagai contoh, untuk milestone 1, release pertama, dan pengumpulan pertama adalah **v0.1.1**. Untuk pengumpulan selanjutnya karena revisi, gunakan tag **v0.1.2** dan seterusnya sesuai nomor pengumpulan. Untuk milestone selanjutnya, pengumpulan dimulai kembali dari satu yaitu **v0.2.1**, **v0.2.2**, dan seterusnya.
- **Laporan** dikumpulkan dengan cara **meletakkan** hasil **PDF** dalam **folder doc** dengan format **Laporan-X-[KODE].pdf**, dimana **X** adalah **nomor milestone**. Pastikan isi laporan sudah sesuai dengan [teknis penggerjaan](#).
- **Diagram** dikumpulkan dengan cara **meletakkan** gambar diagram **dalam folder doc** di repository dengan format '**Diagram-X-[KODE]**', dimana **X** adalah **nomor milestone**.
- Link Pengumpulan: [Pengumpulan Milestone](#)
- Deadline milestone: **Minggu, 19 Oktober 2025 pukul 23.59 WIB**

Referensi

"PASCAL-S: A Subset and its implementation"

<http://pascal.hansotten.com/uploads/pascals/PASCAL-S%20A%20subset%20and%20its%20Implementation%20012.pdf>

"Compiler Design - Lexical Analysis"

https://www.tutorialspoint.com/compiler_design/compiler_design_lexical_analysis.htm

"Introduction to Automata Theory, Languages, and Computation 3rd Edition"

https://cdn-edunex.itb.ac.id/29161-Formal-Language-Theory-and-Automata/1629640939613_Introduction-to-Automata-Theory,-Languages,-and-Computation-Editon-3.pdf

Lampiran

Laman Dokumen Terpusat: [!\[\]\(3cf084882489248c66b41ee5d191c91e_img.jpg\) Dokumen Terpusat - Tubes IF2224 TBFO](#)

Laman Pembagian Kelompok: [!\[\]\(dfc59eaff22f8544bedb238cca58d143_img.jpg\) Pembagian Kelompok - Tubes IF2224 TBFO](#)

Laman QnA Tugas Besar: [!\[\]\(26388bf82a9d28864e0ddb284e508cab_img.jpg\) QNA - Tubes IF2224 TBFO](#)

From Pengumpulan: [Form Pengumpulan - Tubes IF2224 TBFO](#)