
TELAAH TEORITIS FINITE STATE AUTOMATA DENGAN PENGUJIAN HASIL PADA MESIN OTOMATA

WIDYASARI

Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Program Studi Teknik Informatika
Jl.Merdeka No.372 Pontianak, Kalimantan Barat
E-mail : widyasari.iftech03@gmail.com

Abstracts: Otomata and language theory cannot be separated in the world of informatics and represents compulsory subjects that must be provided. Therefore, to produce software in engineering requires a model which can be the solution of a problem later, because the model is an idea that will be applied to generate a software engineering. As one model that mean in here is otomata machine which is the model that can recognize, accept/ generate a sentence in a particular language. It is necessary to understand how to do modeling and testing so that it can be concluded on otomata engine if the string entered correct or wrong. This study will only discuss about the correctness of the theory of otomata machine. The benefits of this research are expected to provide input in order to prove the truth of a study. From the results, some conclusions that prove the truth of the review.

Keywords: Automata, FSA, DFA, NFA.

1. PENDAHULUAN

Dalam ilmu komputer, ada istilah teori bahasa dan otomata, yang masing-masing akan diuraikan sebagai berikut. Teori bahasa dan otomata merupakan komponen pertama dalam ilmu komputer, yang merupakan model dan gagasan mendasar mengenai komputasi. Jadi bukan merupakan teknik rekayasa untuk perancangan sistem komputasi, yang meliputi perangkat keras dan perangkat lunak, khususnya penerapan rancangan dari teori. Tetapi hanya sebatas model yang nantinya akan dikembangkan untuk menghasilkan teknik rekayasa tersebut. Oleh karena itu, teori harus dipelajari sebagai landasan dengan memberikan konsep dan prinsip untuk penerapannya.

Bahasa alami yang dikenal yaitu seperti bahasa Indonesia atau Inggris, namun bahasa yang dimaksud dalam konteks ini adalah merupakan himpunan string-string dari simbol-simbol untuk suatu alphabet, jadi bisa saja bahasa tersebut tidak terdiri dari string-string, yang disebut bahasa kosong, dinotasikan dengan himpunan kosong, \emptyset . Namun, bahasa kosong berbeda dengan bahasa yang terdiri dari string kosong $\{\epsilon\}$. Sebuah simbol adalah suatu entitas abstrak yang tidak didefinisikan secara formal, misalnya huruf dan digit. Sedangkan sebuah string (kata/untai) adalah suatu deretan berhingga dari simbol-simbol, misalnya 'abcd' adalah suatu string yang panjangnya 4. Sebuah string akan diapit oleh tanda petik tunggal. Untuk string kosong, yang dinyatakan dengan ϵ didefinisikan panjangnya = 0, atau $|\epsilon| = 0$ (pada beberapa buku, simbol untuk ϵ dinyatakan dengan λ).

Bahasa bisa juga disebut sebagai rangkaian simbol-simbol yang mempunyai makna.

Otomata adalah mesin abstrak yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu. Dalam prosesnya menerima *input* dan menghasilkan *output*, mesin tersebut mampu membuat keputusan dalam mentransformasikan *input* ke *output*, sehingga *input* pada mesin otomata dianggap sebagai bahasa yang harus dikenali oleh mesin yang selanjutnya membuat keputusan yang mengindikasikan apakah *input* itu diterima atau tidak. Sebuah

string *input* diterima bila mencapai state akhir/*final state*, dan sebaliknya. Contoh mesin otomata antara lain mesin Jaja/*vending machine*, kunci kombinasi dan *parser/compiler*.

Oleh karena itu, teori bahasa dan otomata merupakan langkah awal sebagai model dan gagasan untuk menghasilkan teknik rekayasa dalam perancangannya, baik berupa perangkat keras maupun perangkat lunak.

2. TINJAUAN PUSTAKA

2.1 *Finite State Automata*

Finite State Automata/otomata berhingga state, selanjutnya disebut sebagai FSA yaitu suatu model matematika dari suatu sistem yang menerima *input* dan *output* diskrit.

FSA merupakan mesin otomata dari bahas regular. Suatu FSA memiliki state yang banyaknya berhingga dan dapat berpindah-pindah dari suatu state ke state lain. Perubahan state ini dinyatakan oleh fungsi transisi. FSA tidak memiliki tempat penyimpanan, sehingga kemampuan ‘mengingatnya’ terbatas, hanya bisa mengingat state yang terkini. Contoh FSA antara lain elevator, text editor, analisa leksikal, protokol komunikasi jaringan dan pengecek parity. FSA berdasar pada pendefinisian kemampuan berubah state-statenya bisa dibagi menjadi *Deterministic Finite Automata* (DFA) dan *Non-deterministic Finite Automata* (NFA).

Secara formal FSA dinyatakan oleh 5 tupel atau $M = (Q, \Sigma, \delta, S, F)$ dimana:

Q = himpunan state/kedudukan

Σ = himpunan simbol *input*/masukan/abjad

δ = fungsi transisi

S = state awal/kedudukan awal (initial state), $S \in Q$

F = himpunan state akhir, $F \cap Q$ (jumlah state akhir pada suatu FSA bisa lebih dari satu)

2.2 *Deterministic Finite Automata*

Deterministic Finite Automata/otomata berhingga deterministik, selanjutnya disebut sebagai DFA. Pada DFA, dari suatu state ada tepat satu state berikutnya untuk setiap simbol masukan yang diterima. Suatu string x dinyatakan diterima bila $\delta(S, x)$ berada pada state akhir/*final state*.

2.3 *Non-deterministic Finite Automata*

Non-deterministic Finite Automata, selanjutnya disebut sebagai NFA. Pada NFA, dari suatu state bisa terdapat 0 atau 1 atau lebih busur keluar (transisi) berlabel simbol *input* yang sama. Suatu string diterima oleh NFA bila terdapat suatu urutan transisi sehubungan dengan *input* string tersebut dari state awal menuju state akhir. Untuk NFA, maka harus mencoba semua kemungkinan yang ada sampai terdapat satu yang mencapai state akhir.

3. METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

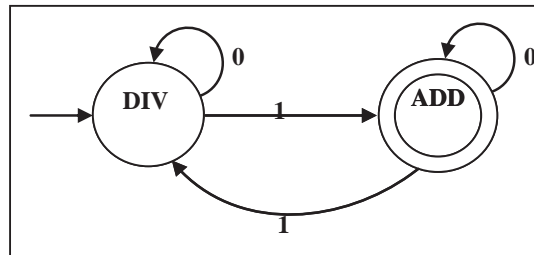
Metode pengumpulan data yang digunakan pada penelitian ini hanya berdasarkan pada *study literature* yang menyajikan teori FSA secara gamblang, karena hanya sedikit sekali buku yang membahas mengenai teori bahasa dan otomata.

3.2 Sifat Penelitian

Sifat penelitian ini hanya berupa pengujian hasil dengan menggunakan mesin otomata. Seperti diketahui bersama bahwa mesin otomata merupakan mesin abstrak, jadi penelitian ini hanya untuk membuktikan apakah suatu string yang dimasukkan dapat diterima atau ditolak oleh mesin melalui sebuah model matematika yang digambarkan dalam diagram transisi.

4. PEMBAHASAN

4.1 Mesin FSA



Gambar 1. Diagram Transisi untuk Pengek Parity Ganjil

Keterangan gambar :

-) Lingkaran menyatakan state/kedudukan; $Q = \{DIV, ADD\}$
-) Label pada lingkaran adalah nama state tersebut
-) Busur menyatakan transisi (perpindahan state/kedudukan)
-) Label pada busur adalah simbol *input*; $\Sigma = \{0, 1\}$
-) Lingkaran didahului sebuah busur tanpa label menyatakan state awal; $S = DIV$
-) Lingkaran ganda menyatakan state akhir/*final state*; $F = \{ADD\}$

Dari contoh gambar di atas jika mesin mendapat *input* :

a) 1101

Maka urutan state yang terjadi : DIV 1 ADD 1 DIV 0 DIV 1 ADD

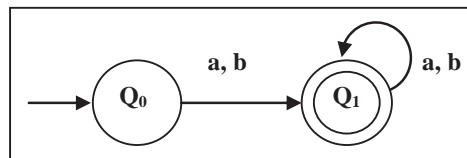
Karena berakhir pada state ADD yang merupakan state akhir, sehingga 1101 diterima oleh mesin.

b) 101

Maka urutan state yang terjadi : DIV 1 ADD 0 ADD 1 DIV

Karena berakhir pada state DIV yang bukan merupakan state akhir, sehingga 101 ditolak oleh mesin.

4.2 Mesin DFA



Gambar 2.Diagram Transisi untuk Contoh DFA 1

$$Q = \{Q_0, Q_1\}$$

$$S = Q_0$$

$$\Sigma = \{a, b\}$$

$$F = \{Q_1\}$$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

$$\delta(Q_0, a) = Q_1$$

$$\delta(Q_1, a) = Q_1$$

$$\delta(Q_0, b) = Q_1$$

$$\delta(Q_1, b) = Q_1$$

Tabel 1. Tabel Transisi untuk Gambar 2

δ	a	b
$\rightarrow Q_0$	Q_1	Q_1
* Q_1	Q_1	Q_1

Dari diagram, fungsi dan tabel transisi di atas jika mesin mendapat *input* :

a) 'aab'

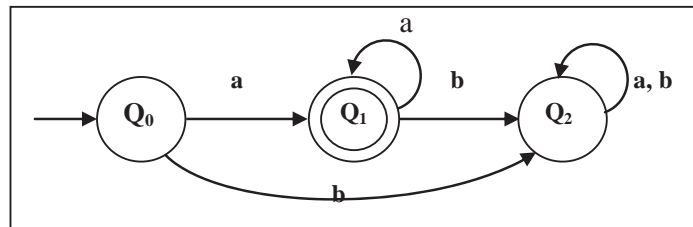
$$\text{Maka } \delta(Q_0, aab) = \delta(Q_1, ab) = \delta(Q_1, b) = Q_1$$

Karena Q_1 merupakan state akhir, sehingga 'aab' diterima oleh mesin.

b) 'abba'

$$\text{Maka } \delta(Q_0, abba) = \delta(Q_1, bba) = \delta(Q_1, ba) = \delta(Q_1, a) = Q_1$$

Karena Q_1 merupakan state akhir, sehingga 'abba' diterima oleh mesin.



Gambar 3. Diagram Transisi untuk Contoh DFA 2

$$Q = \{Q_0, Q_1, Q_2\}$$

$$S = Q_0$$

$$\Sigma = \{a, b\}$$

$$F =$$

$$\{Q_1\}$$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

$$\delta(Q_0, a) = Q_1$$

$$\delta(Q_1, a) = Q_1$$

$$\delta(Q_2, a) = Q_2$$

$$\delta(Q_0, b) = Q_2$$

$$\delta(Q_1, b) = Q_2$$

$$\delta(Q_2, b) = Q_2$$

Tabel 2. Tabel Transisi untuk Gambar 3

δ	a	b
$\rightarrow Q_0$	Q_1	Q_2
* Q_1	Q_1	Q_2
Q_2	Q_2	Q_2

Dari diagram, fungsi dan tabel transisi di atas jika mesin mendapat *input* :

a) 'aaaa'

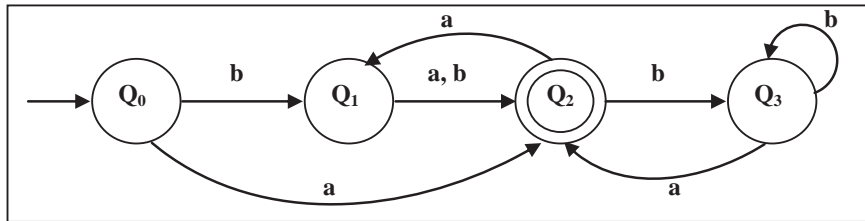
Maka $\delta(Q_0, aaaa) = \delta(Q_1, aaa) = \delta(Q_1, aa) = \delta(Q_1, a) = Q_1$

Karena Q_1 merupakan state akhir, sehingga 'aaaa' diterima oleh mesin.

b) 'bbaba'

Maka $\delta(Q_0, bbaba) = \delta(Q_2, baba) = \delta(Q_2, aba) = \delta(Q_2, ba) = \delta(Q_2, a) = Q_2$

Karena Q_2 bukan merupakan state akhir, sehingga 'bbaba' ditolak oleh mesin.



Gambar 4. Diagram Transisi untuk Contoh DFA 3

$Q = \{Q_0, Q_1, Q_2, Q_3\}$
 $\{Q_2\}$

$S = Q_0$

$\Sigma = \{a, b\}$

$F =$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

$\delta(Q_0, a) = Q_2$ $\delta(Q_1, a) = Q_2$ $\delta(Q_2, a) = Q_1$ $\delta(Q_3, a) =$

Q_2
 $\delta(Q_0, b) = Q_1$ $\delta(Q_1, b) = Q_2$ $\delta(Q_2, b) = Q_3$ $\delta(Q_3, b) =$

Q_3

Tabel 3. Tabel Transisi untuk Gambar 4

δ	a	b
$\rightarrow Q_0$	Q_2	Q_1
Q_1	Q_2	Q_2
$* Q_2$	Q_1	Q_3
Q_3	Q_2	Q_3

Dari diagram, fungsi dan tabel transisi di atas jika mesin mendapat *input* :

a) 'aabb'

Maka $\delta(Q_0, aabb) = \delta(Q_2, abb) = \delta(Q_1, bb) = \delta(Q_2, b) = Q_3$

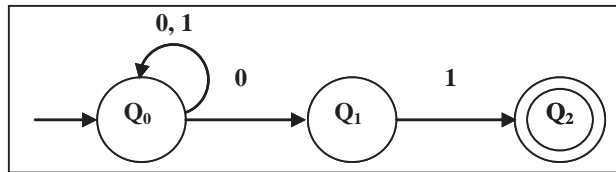
Karena Q_3 bukan merupakan state akhir, sehingga 'aabb' ditolak oleh mesin.

b) 'abbaba'

Maka $\delta(Q_0, abbaba) = \delta(Q_2, bbaba) = \delta(Q_3, baba) = \delta(Q_3, aba) = \delta(Q_2, ba) = \delta(Q_3, a) = Q_2$

Karena Q_2 merupakan state akhir, sehingga 'abbaba' diterima oleh mesin.

4.3 Mesin NFA



Gambar 5. Diagram Transisi untuk Contoh NFA 1

$$Q = \{Q_0, Q_1, Q_2\} \quad S = Q_0 \quad \Sigma = \{0, 1\} \quad F = \{Q_2\}$$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

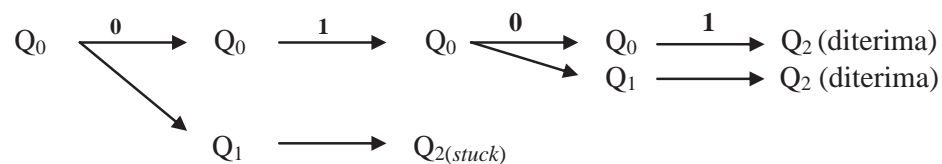
$$\begin{aligned} \delta(Q_0, 0) &= \{Q_0, Q_1\} & \delta(Q_1, 0) &= \emptyset \\ \delta(Q_0, 1) &= \{Q_0\} & \delta(Q_1, 1) &= \{Q_2\} \end{aligned}$$

Tabel 4. Tabel Transisi untuk Gambar 5

δ	0	1
$\rightarrow Q_0$	$\{Q_0, Q_1\}$	$\{Q_0\}$
Q_1	\emptyset	$\{Q_2\}$
$* Q_2$	\emptyset	\emptyset

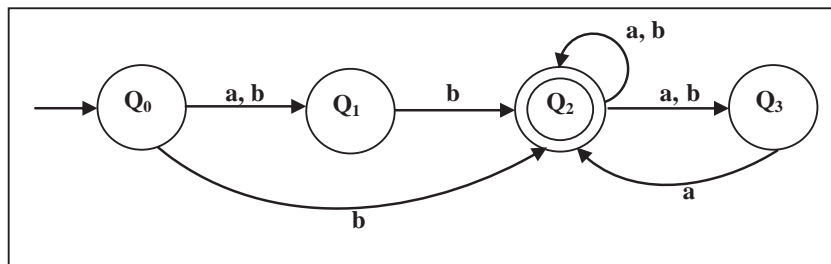
Dari diagram, fungsi dan tabel transisi di atas jika mesin mendapat *input* :

a) '0101'



Tabel 5. Tabel Transisi untuk Contoh NFA 2

δ	a	b
$\rightarrow Q_0$	{ Q ₁ }	{ Q ₁ , Q ₂ }
Q ₁	\emptyset	{ Q ₂ }
* Q ₂	{ Q ₂ , Q ₃ }	{ Q ₂ , Q ₃ }
Q ₃	{ Q ₂ }	\emptyset



Gambar 6. Diagram Transisi untuk Tabel 5

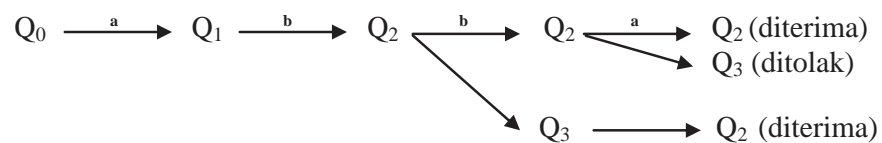
$$Q = \{Q_0, Q_1, Q_2, Q_3\} \quad S = Q_0 \quad \Sigma = \{a, b\} \quad F = \{Q_2\}$$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

$$\begin{aligned} \delta(Q_0, a) &= \{Q_1\} & \delta(Q_1, a) &= \emptyset & \delta(Q_2, a) &= \{Q_2, Q_3\} & \delta(Q_3, a) &= \{Q_2\} \\ \delta(Q_0, b) &= \{Q_1, Q_2\} & \delta(Q_1, b) &= \{Q_2\} & \delta(Q_2, b) &= \{Q_2, Q_3\} & \delta(Q_3, b) &= \emptyset \end{aligned}$$

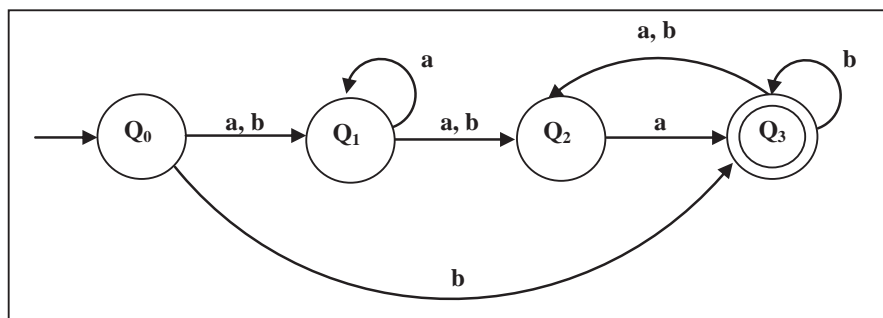
Dari tabel, diagram dan fungsi transisi di atas jika mesin mendapat *input* :

a) 'abba'



Tabel 6. Tabel Transisi untuk untuk Contoh NFA 3

δ	a	b
$\rightarrow Q_0$	{ Q ₁ }	{ Q ₁ , Q ₃ }
Q ₁	{ Q ₁ , Q ₂ }	{ Q ₂ }
Q ₂	{ Q ₃ }	\emptyset
* Q ₃	{ Q ₂ }	{ Q ₂ , Q ₃ }



Gambar 7. Diagram Transisi untuk Tabel 6

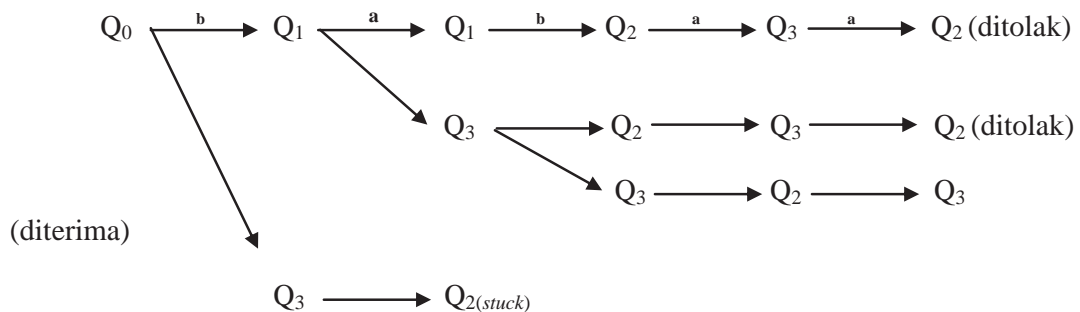
$Q = \{Q_0, Q_1, Q_2, Q_3\}$ $S = Q_0$ $\Sigma = \{a, b\}$ $F = \{Q_3\}$

Dari diagram transisi di atas, maka didapat fungsi transisi yaitu :

$\delta(Q_0, a) = \{Q_1\}$ $\delta(Q_1, a) = \{Q_1, Q_3\}$ $\delta(Q_2, a) = \{Q_3\}$ $\delta(Q_3, a) = \{Q_2\}$
 $\delta(Q_0, b) = \{Q_1, Q_3\}$ $\delta(Q_1, b) = \{Q_2\}$ $\delta(Q_2, b) = \emptyset$ $\delta(Q_3, b) = \{Q_2, Q_3\}$

Dari tabel, diagram dan fungsi transisi di atas jika mesin mendapat *input* :

a) 'babaa'



5. KESIMPULAN

Dari pembahasan-pembahasan yang telah dilakukan sebelumnya, dapat diambil beberapa simpulan:

- 1) Dapat diketahui hasil fungsi transisi, tabel transisi dan memutuskan apakah *input* string yang dimasukkan itu diterima atau ditolak oleh mesin jika diberikan diagram transisinya saja, dan hal itu dapat berlaku sebaliknya jika yang diberikan tabel transisinya saja.
- 2) Hasil pengujian dengan menggunakan mesin otomatis akan menghasilkan keputusan apakah *input* string yang dimasukkan diterima atau ditolak oleh mesin.
- 3) Jika dari suatu state ada tepat satu state berikutnya untuk setiap simbol *input* yang diterima, maka dapat digolongkan menjadi DFA.
- 4) Suatu string x dinyatakan diterima oleh mesin DFA jika $\delta(S, x)$ berada pada state akhir.
- 5) Jika dari suatu state bisa terdapat 0 atau 1 atau lebih busur keluar (transisi) berlabel simbol *input* yang sama, maka akan digolongkan menjadi NFA.
- 6) Suatu string diterima oleh mesin NFA jika terdapat suatu urutan transisi sehubungan dengan *input* string tersebut dari state awal menuju state akhir, untuk itu harus mencoba semua kemungkinan yang ada sampai terdapat minimal satu yang mencapai state akhir.

DAFTAR RUJUKAN

- Utdirartatmo, Furrar. 2001. *Teori Bahasa Dan Otomata*. J & J Learning. Yogyakarta.
- Hopcroft, John E.; Motwani, Rajeev.; Ullman, Jeffrey D. 2001. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.
- Martin, John C. 1997. *Introduction To Languages And The Theory Of Computation*. TheMcGraw-Hill Companies, Inc.
- Lewis, Harry R.; Papadimitriou, Christos H. 1981. *Elements Of The Theory Of Computation*. Prentice-Hall, Inc.