

## AUTOMATA SEBAGAI MODEL PENGENAL BAHASA

**Eddy Maryanto**

Fakultas Sains dan Teknik, Universitas Jenderal Soedirman  
Purwokerto, Indonesia  
email: [eddy\\_maryanto@unsoed.ac.id](mailto:eddy_maryanto@unsoed.ac.id)

**Abstract.** *A deterministic finite automaton as well a nondeterministic finite automaton can be used to model a language recognizer. In computer software technology, language recognizer usually be an integrated part of a compiler, that is a computer program that take responsibility to translate source code into machine code. Comparing with a deterministic finite automaton, a nondeterministic finite automaton is a better model for language recognizer because it might be simpler and less in size than a deterministic one.*

**Keywords.** *Deterministic Finite Automaton, Nondeterministic Finite Automaton, Language Recognizer*

### PENDAHULUAN

#### Bahasa

Teori komputasi merupakan studi matematis proses kerja mesin komputer. Komputer merupakan mesin elektronik yang memproses data berdasarkan instruksi yang diberikan yang mana instruksi ini ditulis dengan menggunakan bahasa pemrograman tertentu. Setiap bahasa pemrograman mempunyai aturan tata bahasa (*syntax*) sendiri-sendiri seperti halnya bahasa alami.

Secara umum, bahasa merupakan himpunan dari kata (*string*) yang tersusun dari simbol-simbol yang merupakan anggota dari alfabet (*alphabet*). Sedangkan alfabet didefinisikan sebagai himpunan terhingga dari simbol-simbol. Sebagai contoh, alfabet Romawi yang sangat kita kenal yaitu  $\{a, b, c, \dots, z\}$ . Pada bahasa mesin, alfabet yang digunakan adalah alfabet biner yaitu  $\{0, 1\}$ . Bahasa merupakan alat untuk berkomunikasi dengan menggunakan kata (yang selanjutnya akan disebut sebagai *string*) yang dirangkai menjadi kalimat yang bermakna. *String* pada sebuah alfabet didefinisikan sebagai barisan terhingga simbol dari alfabet tersebut.

Sebagai contoh, *computer* merupakan sebuah *string* pada alfabet  $\{a,b,c,\dots,z\}$ , 00010001 merupakan sebuah *string* pada alfabet  $\{0,1\}$ . Sebuah *string* bisa saja tidak mempunyai simbol, *string* yang tidak mempunyai simbol disebut *string* kosong (*empty string*). Misalkan  $\Gamma$  merupakan sebuah alfabet, himpunan dari semua *string* -termasuk *string* kosong- dinyatakan dengan  $\Gamma^*$ . Secara matematis, bahasa didefinisikan sebagai sembarang himpunan dari *string* pada alfabet  $\Gamma$ , dengan kata lain bahasa merupakan sembarang himpunan bagian dari  $\Gamma^*$ . Jadi  $\emptyset$ ,  $\Gamma$ , dan  $\Gamma^*$  masing-masing merupakan bahasa.

Karena bahasa merupakan himpunan maka kita dapat menyatakan bahasa yang terhingga dengan membuat daftar *string* dari bahasa tersebut. Sebagai contoh,  $\{for, switch, int, real, while\}$  merupakan sebuah bahasa pada alfabet  $\{a,b,c,\dots,z\}$ . Sedangkan untuk menyatakan bahasa tak hingga dengan membuat syarat keanggotaan *string* dari bahasa tersebut yaitu  $\{w \in \Gamma^*; w \text{ dengan syarat } P\}$ . Sebagai contoh,  $\{w \in \{0,1\}^*; w \text{ mempunyai jumlah } 0 \text{ dan } 1 \text{ sama banyaknya}\}$  merupakan sebuah bahasa tak terhingga pada alfabet  $\{0,1\}$ .

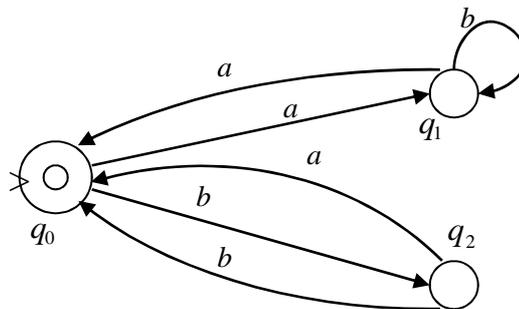
### Automata Keadaan Tentu Terhingga

Automata keadaan tentu terhingga (*Deterministic Finite Automaton*) didefinisikan sebagai kuintupel  $M = (K, \Gamma, d, s, F)$  dengan  $K$  adalah himpunan terhingga dari keadaan,  $\Gamma$  adalah sebuah alfabet,  $s$  adalah keadaan awal (untuk mana  $s \in K$ ),  $F$  adalah keadaan-keadaan akhir (untuk mana  $F \subseteq K$ ), dan  $d$  merupakan sebuah fungsi transisi dari  $K \times \Gamma$  ke  $K$ .

Sebagai contoh, misalkan  $M = (K, \Gamma, d, s, F)$  merupakan sebuah automata keadaan tentu terhingga dengan  $K = \{q_0, q_1, q_2\}$ ,  $\Gamma = \{a, b\}$ ,  $s = q_0$ ,  $F = \{q_0\}$ , dan fungsi transisi  $d$  didefinisikan dalam tabel berikut ini.

$q \in K$	$g \in \Gamma$	$d(q, g)$
$q_0$	$a$	$q_1$
$q_0$	$b$	$q_2$
$q_1$	$a$	$q_0$
$q_1$	$b$	$q_1$
$q_2$	$a$	$q_0$
$q_2$	$b$	$q_0$

Automata keadaan tentu terhingga tersebut di atas bisa digambarkan menggunakan graf berarah dan berbobot berikut ini.



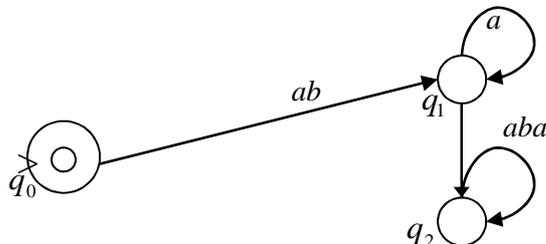
Gambar 1

**Automata Keadaan Tak Tentu Terhingga**

Automata keadaan tak tentu terhingga (*Nondeterministic Finite Automaton*) didefinisikan sebagai kuintupel  $M = (K, \Gamma, \Delta, s, F)$  dengan  $K$  adalah himpunan terhingga dari keadaan,  $\Gamma$  adalah sebuah alfabet,  $s$  adalah keadaan awal (untuk mana  $s \in K$ ),  $F$  adalah keadaan-keadaan akhir (untuk mana  $F \subseteq K$ ), dan  $\Delta$  merupakan sebuah relasi transisi  $K \times \Gamma^* \times K$ .

Sebagai contoh, misalkan  $M = (K, \Gamma, \Delta, s, F)$  merupakan sebuah automata keadaan tak tentu terhingga dengan  $K = \{q_0, q_1, q_2\}$ ,  $\Gamma = \{a, b\}$ ,  $s = q_0$ ,  $F = \{q_2\}$ , dan relasi transisi  $\Delta = \{(q_0, ab, q_1), (q_1, a, q_1), (q_1, b, q_2), (q_2, aba, q_2)\}$ .

Automata keadaan tentu terhingga tersebut di atas bisa digambarkan menggunakan graf berarah dan berbobot berikut ini.



Gambar 2

## PEMBAHASAN

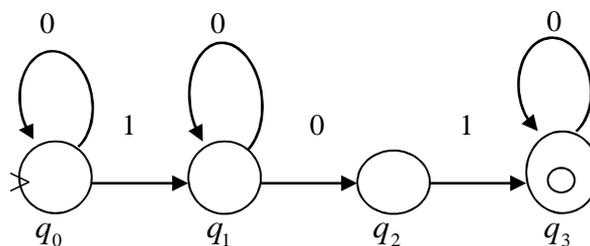
Suatu bahasa dapat disajikan dengan cara membuat daftar dari semua *string* yang ada dalam bahasa tersebut sebagai contoh  $L = \{for, while, main, \dots\}$  merupakan sebuah bahasa pada alfabet  $\{a, b, c, \dots, z\}$ . Selain itu sebuah bahasa juga dapat disajikan dengan menggunakan cara penyajian himpunan lainnya yaitu dengan menggunakan syarat keanggotaan sebagai contoh sebuah bahasa  $L = \{w \in \{0,1\}^* : w \text{ mempunyai dua buah } 1 \text{ yang tidak berurutan}\}$ .

Berdasarkan definisi bahasa, kita dapatkan kenyataan bahwa bahasa-bahasa pada sebuah alfabet dapat diklasifikasikan menjadi tiga kelompok yaitu: (1) tercacah terhingga, (2) tercacah tak terhingga, dan (3) tak tercacah tak terhingga. Sebagai contoh, bahasa yang disebutkan pada bagian terdahulu yaitu  $L = \{w \in \{0,1\}^* : w \text{ mempunyai dua buah } 1 \text{ yang tidak berurutan}\}$  termasuk sebuah bahasa yang tercacah tak terhingga. Bahasa ini dapat disajikan dengan spesifikasi terhingga dengan menggunakan ekspresi reguler yaitu  $L = 0^*10^*010^*$ .

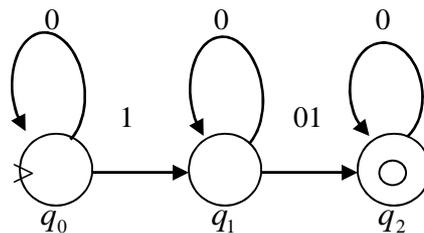
Jika kita cermati dan bandingkan antara cara penyajian bahasa dengan menggunakan notasi himpunan dan ekspresi reguler maka dapat disimpulkan bahwa penyajian bahasa dengan menggunakan ekspresi reguler lebih efisien.

Selain itu penyajian dengan menggunakan ekspresi reguler akan memudahkan kita untuk menentukan automata yang dapat menerima bahasa tersebut. Sebagai contoh, bahasa yang diberikan pada contoh terdahulu yang sudah disajikan dengan spesifikasi terhingga yaitu  $L = 0^*10^*010^*$  menyediakan informasi yang sangat memudahkan untuk penentuan automata yang dapat mengenali bahasa tersebut.

Berdasarkan spesifikasi tersebut dapat ditentukan automata terhingga tertentu maupun terhingga tak tertentu sebagaimana tersaji pada gambar berikut.



Gambar 3 (a)



Gambar 3 (b)

Jika kita perhatikan Gambar 3(a) dan 3(b) maka dapat kita lihat bahwa automata tak tentu terhingga mempunyai struktur yang lebih ringkas, hal ini dikarenakan automata tak tentu terhingga dapat membaca lebih dari satu simbol pada sekali baca, sedangkan automata tentu terhingga hanya diperbolehkan membaca sebuah simbol pada sekali baca.

## KESIMPULAN

Kesimpulan yang bisa diperoleh adalah :

1. Automata keadaan tentu terhingga maupun automata keadaan tak tentu terhingga dapat digunakan untuk merepresentasikan pengenalan bahasa, namun automata keadaan tak tentu terhingga merupakan model yang lebih baik karena selalu mempunyai ukuran yang kecil.

2. Jika diberikan sebuah bahasa maka dapat ditentukan automata yang dapat mengenali bahasa tersebut. Sebaliknya, jika diketahui sebuah automata maka dapat ditentukan bahasa yang dikenali oleh automata tersebut.

#### **DAFTAR PUSTAKA**

Durbin, J.R., *Modern Algebra : An Introduction*, Fourth Edition, John Wiley & Sons Inc., New York, USA, 2000.

Lewis, H.R., dan Papadimitriou C.H., *Elements of The Theory of Computation*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA, 1981.

Rosen, K.H., *Discrete Mathematics and Its Applications*, Fourth Edition, McGraw-Hill Book Inc., New York, USA, 1999.