

MAKALAH

FINITE STATE AUTOMATA DAN IMPLEMENTASINYA

Diajukan untuk memenuhi tugas mata kuliah Teori Bahasa dan Otomata

Dosen pengampu: Undang Syaripudin, M.Kom.



Disusun oleh:

Shabiq Ghazi Arkaan

1207050118

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UIN SUNAN GUNUNG DJATI BANDUNG

2021

KATA PENGANTAR

Puji syukur kehadiran Allah Yang Mahakuasa atas segala limpahan rahmat, iayah, taufik, dan hidayahnya sehingga saya dapat menyelesaikan penyusunan makalah ini dalam bentuk maupun isinya yang sangat sederhana. Semoga makalah ini dapat dipergunakan sebagai salah satu acuan, petunjuk maupun pedoman bagi pembaca dalam mata kuliah Teori Bahasa dan Otomata.

Saya berharap agar makalah ini membantu menambah pengetahuan dan pengalaman bagi para pembaca, sehingga saya dapat memperbaiki bentuk maupun isi makalah ini sehingga kedepannya dapat lebih baik.

Makalah ini saya akui masih banyak kekurangan karena pengalaman yang saya miliki sangat kurang. Oleh karena itu, saya harapkan kepada para pembaca untuk memberikan masukan-masukan yang bersifat membangun untuk kesempurnaan makalah ini.

Tasikmalaya, 26 Oktober 2021

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
BAB II PEMBAHASAN	3
A. Pengertian Finite State Automata	3
B. Jenis-jenis Finite State Automata	3
C. Contoh Implementasi Finite Automata.....	8
BAB III PENUTUP	14
3.1 Simpulan	14
DAFTAR PUSTAKA	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kajian ilmu komputer ada istilah teori bahasa dan automata. Teori bahasa dan automata merupakan cabang ilmu komputer yang merepresentasikan model dan ide dasar ilmu komputer. Teori bahasa dan automata merupakan model yang selanjutnya dikembangkan untuk mengembangkan teknik teknis perancangan sistem komputer yang meliputi perangkat keras dan perangkat lunak. Oleh karena itu, teori harus dipelajari sebagai landasan dengan memberikan konsep dan prinsip untuk penerapannya.

Bahasa pada teori bahasa dan otomata merupakan himpunan string-string yang terdiri dari simbol-simbol dalam alfabet. Bahasa tersebut bisa saja tidak terdiri dari string-string atau bisa disebut bahasa kosong (\emptyset). Selain itu, bahasa tersebut juga bisa terdiri dari string kosong (ϵ). Bahasa kosong (\emptyset) berbeda dengan string kosong (ϵ).

Automata adalah mesin abstrak yang dapat mengenali, menerima, atau menghasilkan kalimat dalam bahasa tertentu. Dengan menerima input dan menghasilkan output, mesin dapat mengambil keputusan dengan mengubah input menjadi output sehingga input ke mesin dipandang sebagai bahasa yang harus dikenali oleh mesin, yang kemudian membuat keputusan yang menentukan diterima atau tidaknya input tersebut. atau tidak. tidak. String input diterima ketika mencapai keadaan akhir dan sebaliknya. Contoh mesin otomatis antara lain mesin vending/haha, kunci kombinasi, dan parser/kompiler.

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

1.2 Rumusan Masalah

1. Apa itu Finite State Automata?

2. Apa perbedaan DFA dan NFA?
3. Bagaimana cara mengubah NFA ke DFA?
4. Apa contoh implementasi Finite State Automata?

1.3 Tujuan

1. Untuk mengetahui pengertian Finite State Automata.
2. Untuk mengetahui apa itu DFA dan NFA serta perbedaannya
3. Untuk mengetahui cara mengubah NFA ke DFA
4. Untuk mengetahui apa saja contoh implementasi Finite State Automata.

BAB II

PEMBAHASAN

A. Pengertian Finite State Automata

Finite State Automata (FSA) merupakan mesin abstrak berupa sistem model matematika dengan masukan dan keluaran diskrit terdiri dari string dan label dengan output terdiri dari 0s 1s yang dapat mengenali bahasa paling sederhana (bahasa reguler) yang menangkap pola dalam data dan dapat diimplementasikan secara nyata sehingga dapat dipahami oleh logika manusia. FSA adalah model matematika yang dapat menerima input dan mengeluarkan output yang memiliki state yang berhingga banyaknya dan dapat berpindah dari satu state ke state berikutnya. Dilihat dari jenisnya yaitu FSA terdapat mesin bahasa yang dapat mengenali, menerima dan menolak yang terdapat pada mesin FSA jenis Deterministic Finite Automata (DFA) dan Non-deterministic Finite Automata (NFA). FSA memiliki sifat-sifat seperti pita masukan (input tape) berisi rangkaian simbol (string) yang berasal dari himpunan alphabet / simbol, setiap kali setelah membaca satu karakter, posisi read head akan berada pada simbol berikutnya, setiap saat FSA berada dalam status tertentu serta banyaknya status yang berlaku bagi FSA adalah berhingga.

B. Jenis-jenis Finite State Automata

Ada dua jenis Finite State Automata, yaitu:

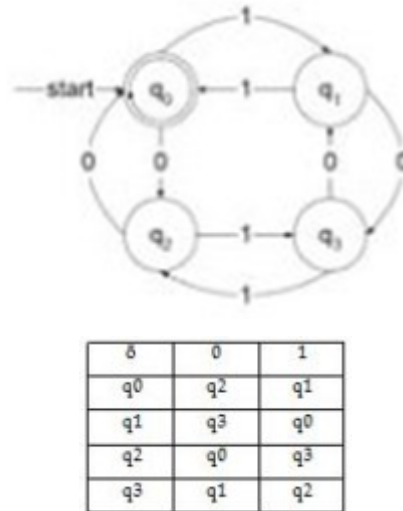
1. Deterministic Finite Automata

Deterministic Finite Automata memiliki ciri bahwa dari suatu state ada tepat satu state berikutnya untuk setiap simbol masukan yang diterima. Deterministik artinya tertentu/sudah tertentu fungsi transisinya.

Notasi matematis DFA:

- M = nama DFA
- Q = himpunan keadaan DFA
- S = himpunan alfabet
- d = fungsi transisi
- q_0 = keadaan awal

- F = keadaan akhir
- $M = (Q, S, d, q_0, F)$

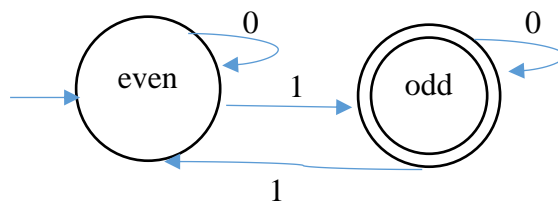


Contoh : Pengujian untuk mengecek suatu bilangan genap atau ganjil

001 : diterima, karena berakhir di odd

10010 : ditolak, karena berakhir di even

Diagram transisi-nya :



DFA nya:

$Q = \{\text{even}, \text{odd}\}$

$\Sigma = \{0,1\}$

$S = \text{even}$

$F = \{\text{odd}\}$

Fungsi transisi adalah:

$\delta(\text{even}, 001) = \text{Diterima}$

$\delta(\text{even}, 10010) = \text{Ditolak}$

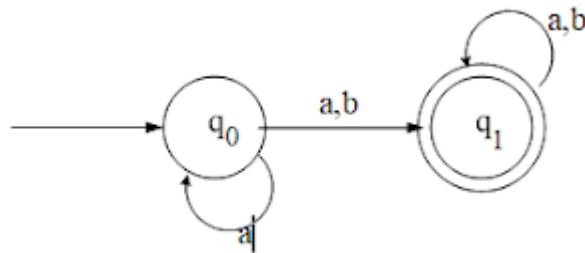
2. Non-Deterministic Finite Automata

Ciri Non-Deterministic Finite Automata adalah dari suatu state ada 0, 1 atau lebih state berikutnya untuk setiap simbol masukan yang diterima.

Non-Deterministic Finite Automata:

- Otomata berhingga yang tidak pasti untuk setiap pasangan state input, bisa memiliki 0 (nol) atau lebih pilihan untuk state berikutnya.
- Untuk setiap state tidak selalu tepat ada satu state berikutnya untuk setiap simbol input yang ada.
- Dari suatu state bisa terdapat 0,1 atau lebih busur keluar (transisi) berlabel simbol input yang sama.
- Untuk NFA harus dicoba semua kemungkinan yang ada sampai terdapat satu yang mencapai state akhir.
- Suatu string x dinyatakan diterima oleh bahasa NFA, $M = (Q, \Sigma, \delta, S, F)$ bila $\{x \mid \delta(S, x) \text{ memuat sebuah state di dalam } F\}$

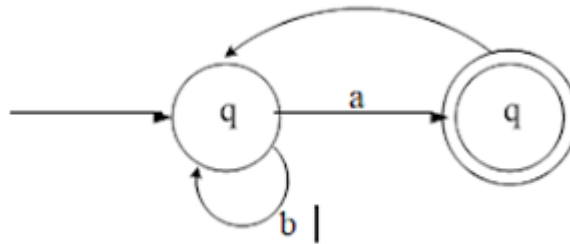
Contoh NFA:



Perhatikan gambar di atas, bila state q_0 mendapat input 'a' bisa berpindah ke state q_0 atau q_1 , yang secara formal dinyatakan: $\delta(q_0, a) = \{q_0, q_1\}$. Maka otomata ini disebut non-deterministik (tidak pasti arahnya). Bisa kita lihat tabel transisinya seperti di bawah ini.

Δ	A	B
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_1\}$

Contoh lainnya:



Kita bisa melihat tabel transisinya di bawah ini :

Δ	A	B
q0	{q1}	{q0}
q1	{q0}	\emptyset

Perbedaan DFA dan NFA

DFA	NFA
DFA tidak dapat menggunakan transisi string kosong (empty string)	NFA dapat menggunakan transisi string kosong (empty string)
DFA dipahami sebagai sebuah mesin	NFA dipahami sebagai beberapa mesin kecil yang melakukan komputasi di waktu bersamaan
DFA untuk state selanjutnya bisa ditetapkan dengan jelas	NFA untuk state selanjutnya mempunyai banyak kemungkinan
DFA lebih sulit dibuat	NFA lebih mudah dibuat
Waktu yang dibutuhkan untuk mengeksekusi string input lebih sedikit	Waktu yang dibutuhkan untuk mengeksekusi string input lebih banyak
Semua DFA merupakan NFA	Tidak semua NFA adalah DFA

DFA membutuhkan lebih banyak ruang (space)	NFA membutuhkan lebih sedikit ruang (space)
--	---

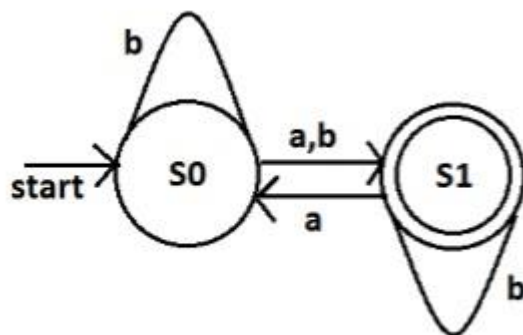
Konversi NFA menjadi DFA:

Dari suatu mesin Non Deterministic Finite Automata (NFA) dapat dikonversi atau dibuat menjadi suatu mesin Deterministic Finite Automata (DFA) yang memiliki kemampuan menerima Bahasa yang sama (ekuivalen).

Berikut contoh konversi dari Non Deterministic Finite Automata menjadi Deterministic Finite Automata:

- Tabel transisi dan diagram state DFA:

State	a	b
S0	S1	{S0,S1}
S1*	S0	S1

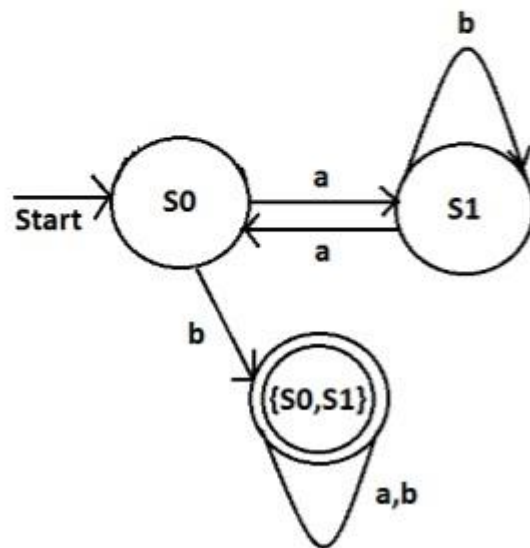


Dari gambar di atas, dapat dilihat ada suatu state yang diberi inputan menuju ke beberapa state, yaitu S0 yang diberi inputan b bisa menuju ke S0 dan S1

- Lalu dikonversi menjadi DFA, dengan cara membuat state baru berupa gabungan dari S0 dan S1.
- Lalu untuk state {S0,S1} jika diberi inputan a maka hasilnya adalah gabungan dari hasil S0 dan S1 yang diberi inputan a yaitu {S0,S1}

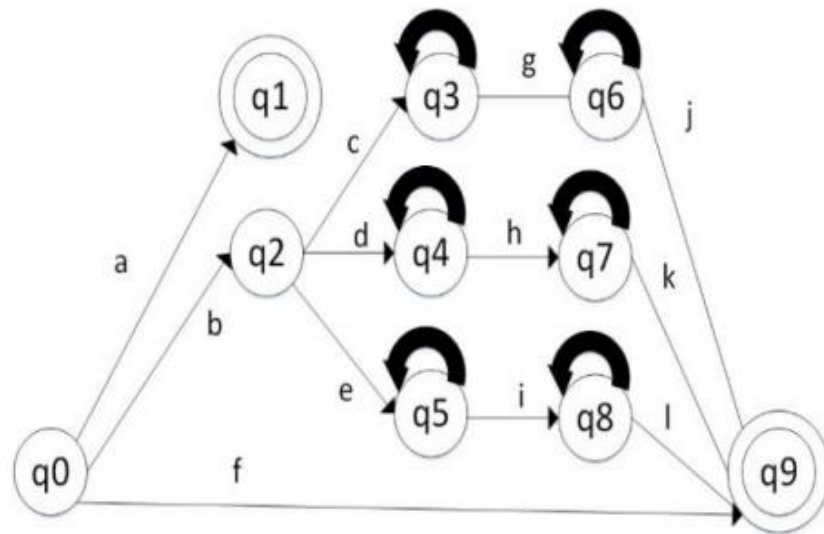
- Lalu untuk state $\{S0, S1\}$ jika diberi inputan b maka hasilnya adalah gabungan dari hasil S0 dan S1 yang diberi inputan b yaitu $\{S0, S1\}$
- Berikut table transisi dan diagram transisi untuk DFA nya :

State	a	b
S0	S1	$\{S0, S1\}$
S1*	S0	S1
$\{S0, S1\}^*$	$\{S0, S1\}$	$\{S0, S1\}$



C. Contoh Implementasi Finite Automata

1. Program Registrasi Workout Plan pada Pusat Kebugaran



Keterangan:

q0 = Status Awal

q1 = Mengikuti hanya sesi Latihan harian

q2 = Registrasi/Konsultasi sebagai member bulanan

q3 = Terdaftar sebagai Member dengan kategori Tubuh

q4 = Terdaftar sebagai Member dengan kategori Tubuh Endomorph

q5 = Terdaftar sebagai Member dengan kategori Tubuh Mesomorph

q6 = Mendapatkan Bulking Workout Plan Program

q7 = Mendapatkan Fatloss/Weightloss Workout Plan Program

q8 = Mendapatkan Kombinasi Workout Plan Program

q9 = Sesi Latihan

a = Member harian

b = Registrasi pendaftaran sebagai member

c = Sistem memferifikasi Tubuh masuk dalam kategori Ectomorph

d = Sistem memferifikasi Tubuh masuk dalam kategori Endomorph

e = Sistem memferifikasi Tubuh masuk dalam kategori Mesomorph

f = Member bulanan tanpa workout plan

g = Member mengambil Bulking workout Plan Program

h = Member mengambil Fatloss/weightloss workout Plan Program

i = Member mengambil Kombinasi workout Plan Program

j = Member melakukan sesi program Bulking Workout Plan Program Bersama Personal Trainer

k = Member melakukan sesi program Fatloss/weightloss Workout Plan Program Bersama Personal Trainer

l = Member melakukan sesi program Kombinasi Workout Plan Program Bersama Personal Trainer

NFA-nya:

$Q = \{q_0, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$

$\Sigma = \{a, b, c, d, e, f, g, h, I, j, k, l\}$

$S = q_0$

$F = \{q_1, q_9\}$

Fungsi transisinya:

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_2$

$\delta(q_0, f) = q_9$

$\delta(q_2, c) = q_3$

$\delta(q_2, d) = q_4$

$\delta(q_2, e) = q_5$

$\delta(q_3, g) = q_6$

$\delta(q_4, h) = q_7$

$\delta(q_5, i) = q_8$

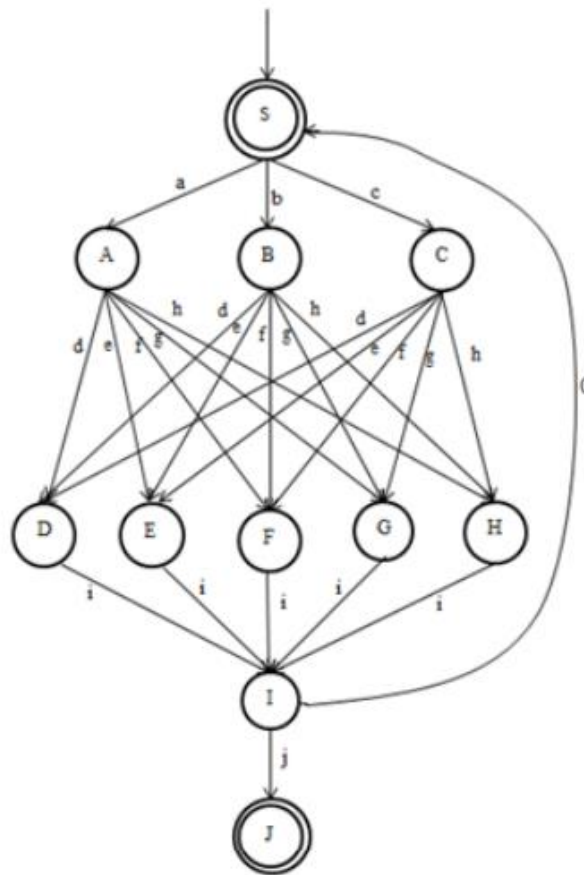
$\delta(q_6, j) = q_9$

$\delta(q_7, k) = q_9$

$\delta(q_8, l) = q_9$

Program Registrasi Workout Plan pada Pusat Kebugaran ini adalah program yang menggunakan konsep Non-Deterministic Finite Automata (NFA), yang mana NFA disini digunakan untuk membantu seseorang menentukan pilihan program workout plan yang tepat sesuai dengan kondisi tubuh.

2. Mesin Pembuat Es Krim Otomatis



Keterangan:

S = Status awal

A = Rasa coklat

B = Rasa vanilla

C = Rasa stroberi

D = Topping kacang

E = Topping Oreo

F = Topping Matcha

G = Topping Milo

H = Topping Coco-crunch

I = Pesanan dikonfirmasi

J = Es krim selesai dibuat

a = memilih rasa coklat
b = memilih rasa vanilla
c = memilih rasa stroberi
d = memilih topping kacang
e = memilih topping oreo
f = memilih topping matcha
g = memilih topping milo
h = memilih topping coco-crunch
i = mengkonfirmasi pesanan
j = membuat es krim

NFA-nya:

$$Q = \{S, A, B, C, D, E, F, G, H, I, J\}$$

$$\Sigma = \{a, b, c, d, e, f, g, h, i, j\}$$

$$S = A$$

$$F = \{A, J\}$$

Fungsi transisi:

$$\delta(S, a) = A$$

$$\delta(S, b) = B$$

$$\delta(S, c) = C$$

$$\delta(A, d) = D$$

$$\delta(A, e) = E$$

$$\delta(A, f) = F$$

$$\delta(A, g) = G$$

$$\delta(A, h) = H$$

$$\delta(B, d) = D$$

$$\delta(B, e) = E$$

$$\delta(B, f) = F$$

$$\delta(B, g) = G$$

$$\delta(B,h) = H$$

$$\delta(C,d) = D$$

$$\delta(C,e) = E$$

$$\delta(C,f) = F$$

$$\delta(C,g) = G$$

$$\delta(C,h) = H$$

$$\delta(D,i) = I$$

$$\delta(E,i) = I$$

$$\delta(F,i) = I$$

$$\delta(G,i) = I$$

$$\delta(H,i) = I$$

$$\delta(I,j) = J$$

Mesin Pembuat Es Krim Otomatis menggunakan konsep Finite State Automata yang mana ini menggunakan konsep NFA sebagai rancangan programnya. Mesin ini dapat membaca inputan user yang membentuk pola pembuatan es krim dengan 3 rasa dan 5 topping.

Finite State Automata juga sangat bias diterapkan di bidang lainnya.

BAB III

PENUTUP

3.1 Simpulan

FSA adalah model matematika yang dapat menerima input dan mengeluarkan output yang memiliki state yang berhingga banyaknya dan dapat berpindah dari satu state ke state berikutnya.

FSA ada dua jenis, yaitu Deterministic Finite Automata dan Non-Deterministic Finite Automata. Perbedaan keduanya dapat dilihat dari statenya yang mana DFA setiap statenya tidak dapat menerima string kosong, sementara NFA dapat menerima string kosong, string satu, ataupun lebih.

NFA bisa dikonversi menjadi DFA dengan cara mencari suatu state yang menuju ke beberapa state dengan inputan yang sama, atau tidak menuju satu state pun. kemudian dibuat state baru untuk menampung inputan dari state tersebut

FSA dapat diimplementasikan di banyak tempat, salahsatu yang kami bahas di makalah ini adalah Program Registrasi Workout Plan pada Pusat Kebugaran dan Mesin Pembuat Es Krim Otomatis.

DAFTAR PUSTAKA

- Widyasari. 2011. *Telaah Teoritis Finite State Automata dengan Pengujian Hasil pada Mesin Otomata*. Jurnal Ilmiah SISFOTENIKA Vol. 1, No. 1, Januari 2011.
- Aziz,Faruq; Said, Fadillah; Sudrajat, Adjat. 2020. *Penerapan Konsep Finite State Automata Dalam Proses Pendaftaran Kelas Kursus Bahasa Inggris Pada Tempat Kursus*. MATICS: Jurnal Ilmu Komputer dan Teknologi Informasi Vol. 12, No. 2(2020), pp 93-98.
- Aldiana, Nikko. 2020. *Finite State Automata (FSA) dan Non Finite State Automata*. Diambil dari <https://nikkoald.wordpress.com/2020/04/25/finite-state-automata-fsa-non-finite-state-automata/> (diakses pada 26 Oktober 2021 pada 17.26 WIB).
- Rivanie, Tri; Adilah M, Tika; Alkhalifi, Yuris. 2020. *Implementasi Finite State Automata dalam Proses Registrasi Workout Plan pada Pusat Kebugaran*. MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi Volume 12, No. 1 (2020), pp 94-98.
- Fida. 2020. *Perbedaan DFA dan NFA*. Diambil dari <https://fida.ump.ac.id/perbedaan-dfa-dan-nfa/> (diakses pada 27 Oktober 2021 pada 11.31 WIB)
- Kaunang, Fergie Joanda. 2019. *Penerapan Konsep Finite State Automata (FSA) pada Mesin Pembuat Ice cream Otomatis*. Jurnal TeIKa, Volume 9, Nomor 2, Oktober 2019.