

# Kuliah 3 Finite Automata (FA)

Pendahuluan

Konsep Otomata

Sifat-sifat Otomata

Finite State Automata (FSA)

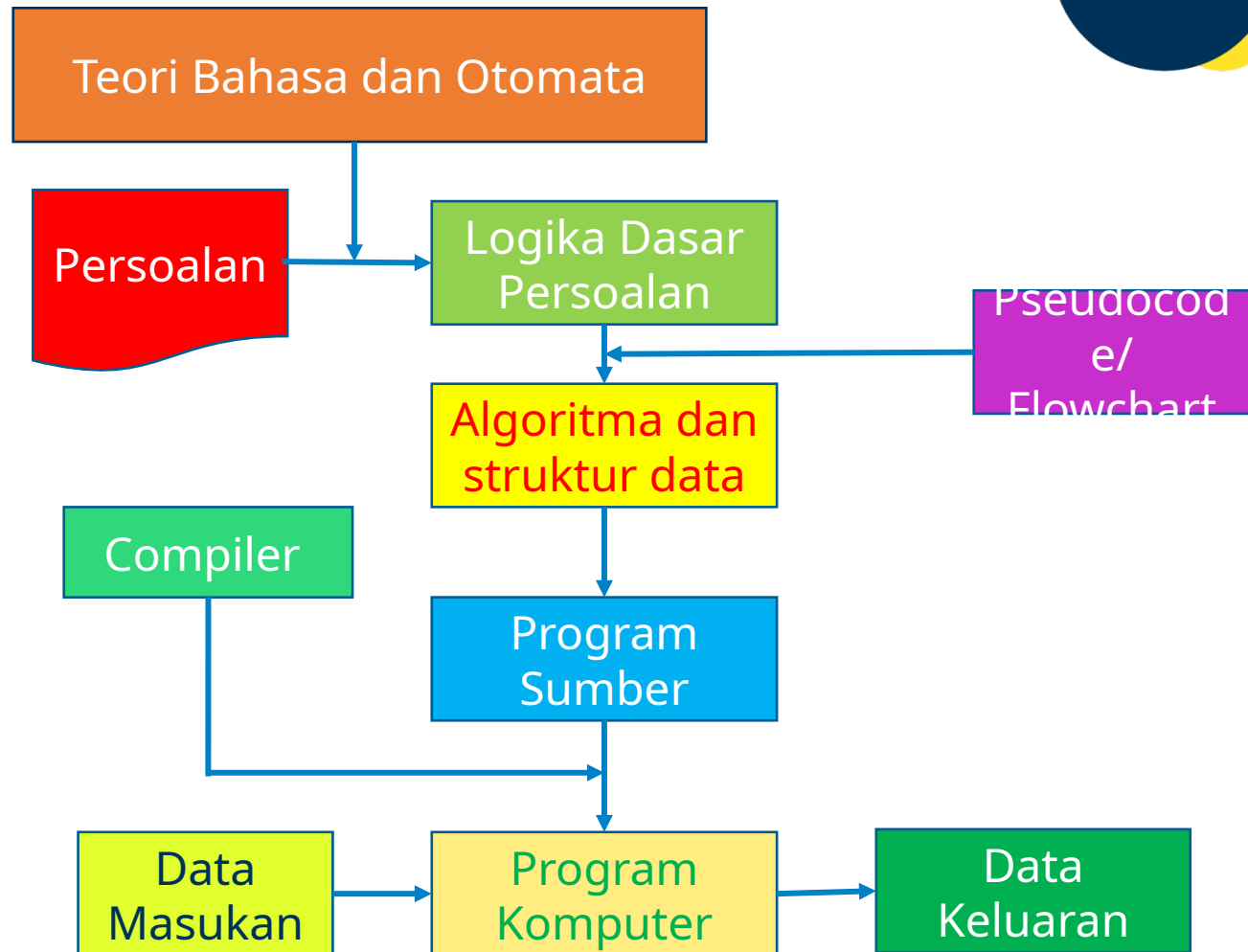
Deterministic Finite Automata (DFA)

Non Deterministic Finite Automata  
(NDFA)



**UKRIDA**  
Universitas Kristen Krida Wacana

# Pendahuluan



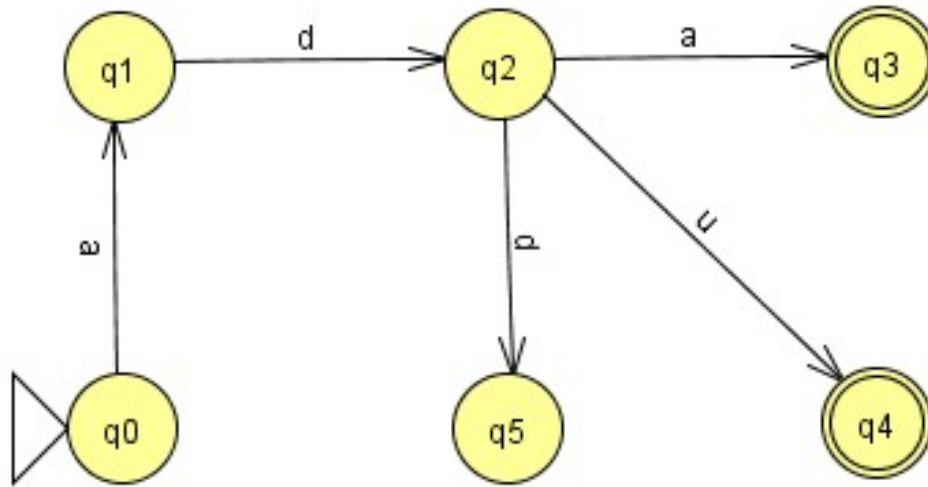
**Model Pendekatan Otomata**

# Pendahuluan

---

- Otomata merupakan suatu sistem yang terdiri dari sejumlah berhingga state, dimana state menyatakan informasi mengenai input
- Otomata dianggap sebagai mesin otomatis (bukan mesin fisik) yang merupakan suatu model matematika dari suatu sistem yang menerima input dan menghasilkan output, serta terdiri dari sejumlah berhingga state.
- Hubungan diantara bahasa dan otomata adalah bahasa dijadikan sebagai input oleh suatu mesin otomata, selanjutnya mesin otomata akan membuat keputusan yang mengindikasikan apakah input itu diterima atau ditolak.

# Pendahuluan



**Contoh Diagram FA**

Pada gambar diatas bila mesin mendapat string input berikut:

1. ada: diterima
2. adu: diterima
3. add: ditolak/tidak diterima

# Pendahuluan

---

- Sebuah string input diterima bila mencapai state akhir/final state yang digambarkan dalam lingkaran ganda.
- Mesin ini memiliki 6 state {q0, q1, q2, q3, q4, q5}.
- State awal dari mesin adalah {q0}
- State akhir/ final state adalah {q3, q4}
- Himpunan simbol input adalah {a, d, u}

## Konsep Otomata (Automata)

- Automata berasal dari bahasa Yunani automatos yang berarti sesuatu yang bekerja secara otomatis.
- Istilah automaton sebagai bentuk tunggal dan automata sebagai bentuk jamak.
- Untuk memodelkan hardware dari komputer diperkenalkan model mesin abstrak yaitu otomata.
- Otomata adalah ilmu yang mempelajari mengenai mesin abstrak, dapat juga disebut suatu model abstrak dari komputer digital yang dapat menerima input secara sekuensial dan dapat mengeluarkan output.
- Input pada mesin otomata dianggap sebagai bahasa yang harus dikenali oleh mesin.

## Konsep Otomata (Automata)

- Automata adalah mesin abstrak yang dapat mengenali (recognized), menerima (accept), atau membangkitkan (generate) sebuah kalimat dalam bahasa tertentu.
- Teori automata adalah teori tentang bahasa mesin abstrak yang:
  1. Bekerja sekuensial
  2. Menerima input
  3. Mengeluarkan output
- Pengertian mesin bukan hanya mesin elektronis/mekanis saja melainkan segala sesuatu (termasuk perangkat lunak) yang memenuhi ketiga ciri diatas.

## Konsep Otomata (Automata)

---

Penggunaan automata pada perangkat lunak terutama pada pembuatan kompiler bahasa pemrograman. Secara garis besar ada dua fungsi automata dalam hubungannya dengan bahasa, yaitu:

- a. Fungsi automata sebagai pengenalan (recognizer) string-string dari suatu bahasa, dalam hal ini bahasa sebagai masukan dari automata.
- b. Fungsi automata sebagai pembangkit (generator) string-string dari suatu bahasa, dalam hal ini bahasa sebagai keluaran dari automata.



# Konsep Otomata (Automata)

- Untuk mengenali string-string suatu bahasa akan dibutuhkan sebuah automaton yang memiliki komponen sebagai berikut:
  1. Pita masukan, yang menyimpan string masukan yang akan dikenali
  2. Kepala pita (tape head), untuk membaca/menulis ke pita masukan
  3. Finite state controller (FSC), yang berisi status-status dan aturan-aturan yang mengatur langkah-langkah yang dilakukan oleh automaton berdasarkan status setiap saat dan simbol masukan yang sedang dibaca oleh kepala pita
  4. Peningkat (memory), untuk tempat penyimpanan dan pemrosesan sementara automaton pengenalan, setelah membaca string masukan dan melakukan langkah-langkah pemrosesan yang diperlukan, akan mengeluarkan keputusan apakah string tersebut dikenali atau tidak.
  5. Konfigurasi adalah suatu mekanisme untuk menggambarkan keadaan suatu mesin pengenalan, yang terdiri atas:
    - Status FSC
    - Isi pita masukan dan posisi kepala pita

## Konsep Otomata (Automata)

- Mesin pengenalan bersifat deterministik bila dalam setiap konfigurasi hanya ada satu kemungkinan yang dapat dilakukan mesin, jika tidak mesin pengenalan bersifat non deterministik.
- Otomata yang lebih umum yaitu yang mampu menghasilkan string output, dikenal dengan Transducer.
- Contoh penggunaan otomata adalah:
  1. Mesin Turing
  2. Mesin karakter
  3. Kompiler
  4. Mesin Jaja (Vending Machine)
- Otomata adalah mesin abstrak yang berkaitan dengan teori mesin abstrak, yaitu mesin sekuensial yang menerima input, dan mengeluarkan output dalam bentuk diskrit.

## Sifat-sifat Otomata

---

Otomata adalah suatu mesin sekuensial (otomatis) mempunyai sifat-sifat:

- a. Kelakuan mesin bergantung pada rangkaian masukan yang diterima mesin tersebut
- b. Setiap saat mesin dapat berada pada satu status tertentu dan dapat berpindah ke status baru karena adanya perubahan input
- c. Rangkaian input (diskrit) pada mesin otomata dapat dianggap sebagai bahasa yang harus “dikenali” oleh sebuah automata. Setelah pembacaan input selesai mesin automata kemudian membuat keputusan.

# Sifat-sifat Otomata

## Jenis-jenis automata

Jenis	Pita masukan	Arah Head	Memori
Finite State	Read Only	1 arah	-
Push Down	Read Only	1 arah	Stack
Linier Bounded	Read/Write	2 arah	(bounded)
Turing Machine	Read/Write	2 arah	(unbounded)

## Sifat-sifat Otomata

---

- Finite State Automata (FSA) adalah mesin yang dapat mengenali kelas bahasa reguler dan memiliki sifat-sifat:
  1. Pita masukan (input pita) berisi rangkaian simbol (string) yang berasal dari himpunan simbol/alphabet
  2. Setiap kali setelah membaca satu karakter, posisi read head akan berada pada simbol berikutnya
  3. Setiap saat FSA berada pada status tertentu
  4. Banyaknya status yang berlaku bagi FSA adalah berhingga
- Jika mesin setelah membaca simbol terakhir dari string input berada disalah satu dari serangkaian status tertentu, maka mesin tersebut dikatakan menerima string input.

## Sifat-sifat Otomata

Untuk setiap kelas bahasa Chomsky, terdapat sebuah mesin pengenalan bahasa. Masing-masing mesin tersebut adalah:

Kelas Bahasa	Mesin pengenalan Bahasa
Unrestricted Grammar (UG)	Mesin Turing (Turing Machine), TM
Context Sensitive Grammar (CSG)	Linier Bounded Automaton, LBA
Context Free Grammar (CFG)	Pushdown Automata (PDA)
Regular Grammar (RG)	Finite Automata (FA)

Catatan:

1. Pengenal bahasa adalah salah satu kemampuan mesin turing
2. LBA adalah variasi dari mesin turing non deterministik
3. Yang akan dibahas dalam kuliah TBO adalah TM (tidak terlalu mendalam), FA dan PDA

## Finite State Automata (FSA)

Finite State Automata (FSA) merupakan mesin otomata dari bahasa reguler.

Suatu FSA memiliki state yang banyaknya berhingga dan dapat berpindah-pindah dari suatu state ke state lain.

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

$Q$  = himpunan state

$\Sigma$  = himpunan simbol input/masukan/abjad

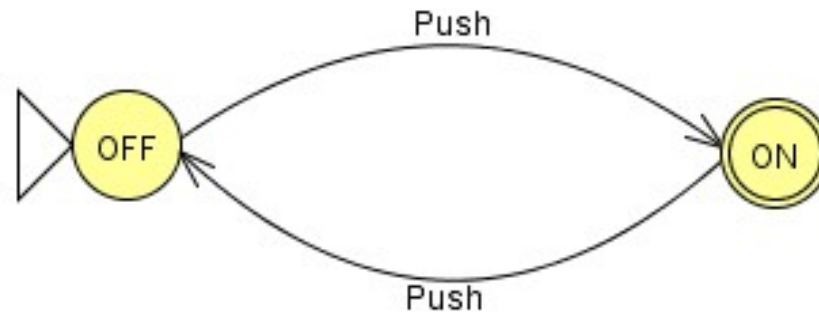
$\sigma$  = fungsi transisi

$S$  = state awal/initial state

$F$  = himpunan state akhir

## Finite State Automata (FSA)

- FSA yang memiliki tepat satu state berikutnya untuk setiap simbol masukan yang diterima disebut Deterministic Finite Automata (DFA)
- Contoh :**



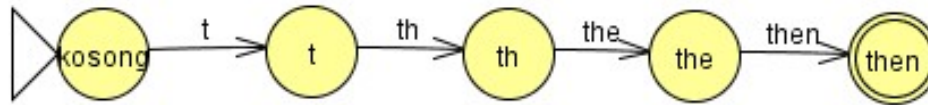
- Switch berada dalam state "off" maka setelah tombol ditekan state berubah menjadi "on".
- Jika switch berada dalam state "on" maka setelah tombol ditekan state berubah menjadi "off"



# Finite State Automata (FSA)

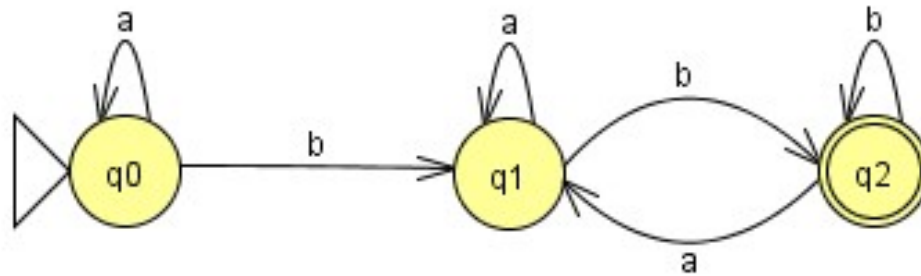
## Contoh :

- FA berikut dapat dinyatakan sebagai bagian dari lexical analyzer



- Tugas FA adalah mengenali keyword "then" sehingga diperlukan lima state masing-masing menyatakan posisi yang berbeda dalam kata "then" yang telah dicapai sejauh ini.
- Dalam gambar diatas input dinyatakan oleh huruf. Start state merupakan string kosong, dan setiap state memiliki transisi pada huruf selanjutnya dari kata then ke state yang menyatakan prefix selanjutnya yang lebih besar. State yang diberi nama "then" dimasuki ketika mengeja kata "then". Karena fungsi dari model dalam gambar diatas adalah mengenali kata then, maka state "then" dinyatakan sebagai accepting state.

# Finite State Automata (FSA)



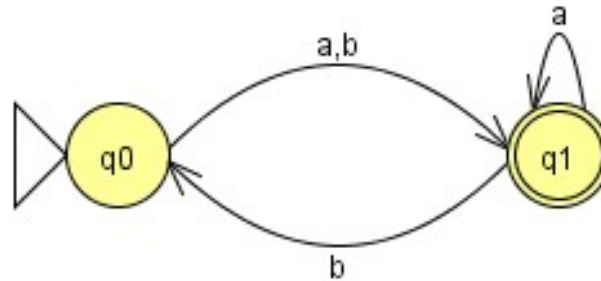
- Konfigurasi DFA diatas secara formal dinyatakan sebagai berikut:
  - $Q = \{q_0, q_1, q_2\}$
  - $\Sigma = \{a, b\}$
  - $S = q_0$
  - $F = \{q_2\}$
- Fungsi transisi yang ada adalah sebagai berikut:
  - $\sigma(q_0, a) = q_0$
  - $\sigma(q_0, b) = q_1$
  - $\sigma(q_1, a) = q_1$
  - $\sigma(q_1, b) = q_2$
  - $\sigma(q_2, a) = q_1$
  - $\sigma(q_2, b) = q_2$

# Finite State Automata (FSA)

- Biasanya fungsi-fungsi transisi disajikan dalam sebuah tabel transisi. Tabel transisi tersebut menunjukkan state-state berikutnya untuk kombinasi state-state dan input.
- Tabel transisi dari fungsi diatas adalah sebagai berikut:

$\sigma$	a	b
q0	q0	q1
q1	q1	q2
q2	q1	q2

# Finite State Automata (FSA)



Tabel transisi dari gambar diatas adalah sebagai berikut

$\sigma$	a	b
q0	q1	q1
q1	q1	q0

- Model FA diatas menerima input seperti string:
  - aaba
  - aaabb
  - baba
  - babaaba
- Sedangkan string yang ditolak FA diatas :
  - aaabbb
  - bbaab
  - babbab
  - ababab

# Finite State Automata (FSA)

---

Ada dua jenis FSA:

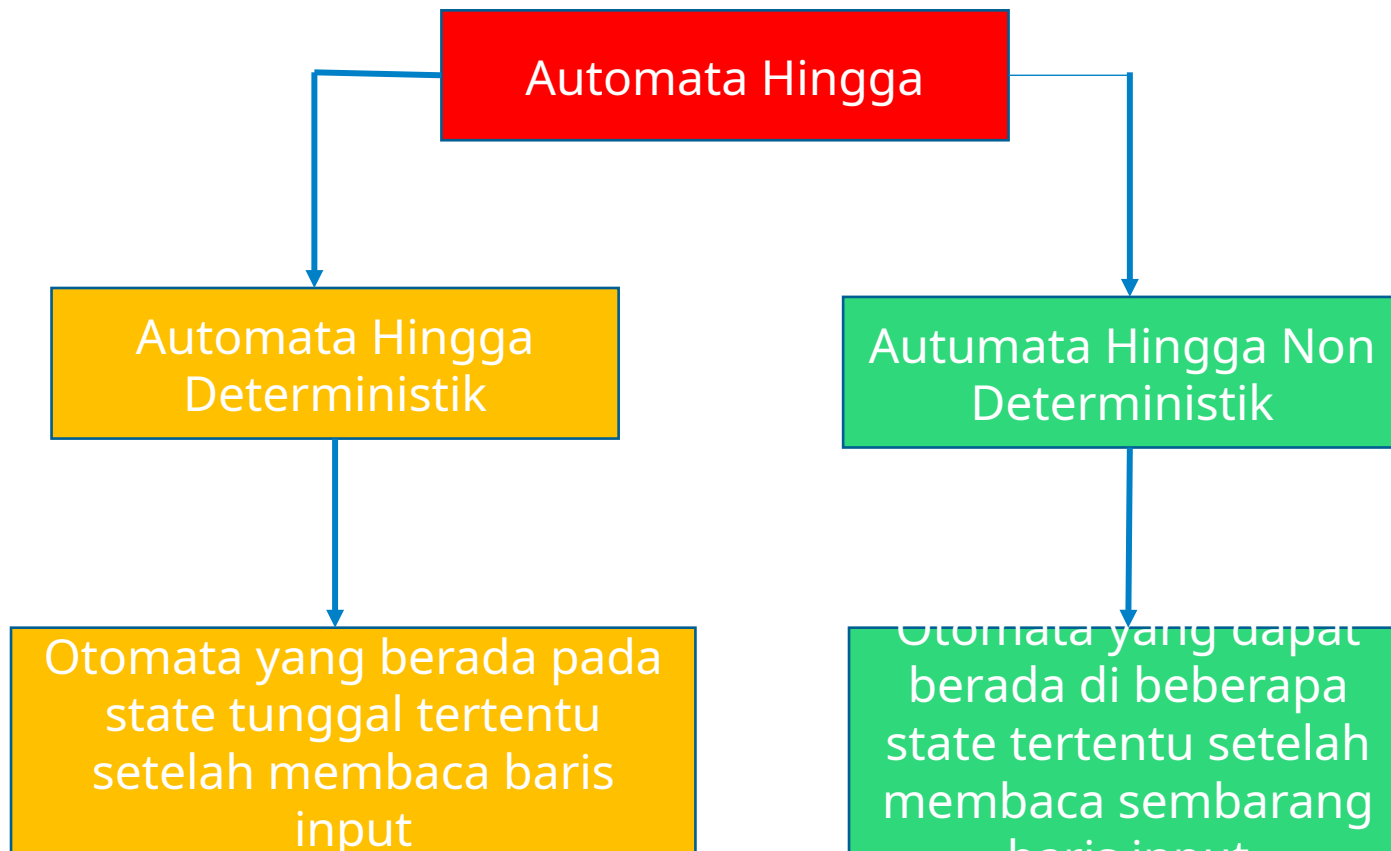
1. **Deterministic Finite Automata (DFA):** transisi state FSA akibat pembacaan sebuah simbol bersifat tertentu :  $\sigma: Q \times \Sigma \rightarrow Q$
2. **Non deterministic Finite Automata (NFA):** transisi state FSA akibat pembacaan sebuah simbol bersifat tak tentu.  $\sigma: Q \times \Sigma \rightarrow 2^Q$

# Finite State Automata (FSA)



DFA	NDAFA
Transisi dari suatu state ke state berikutnya untuk setiap simbol masukan. Oleh karena itu disebut deterministik.	Transisi dari suatu state dapat ke beberapa state berikutnya untuk setiap simbol masukan. Oleh karena itu disebut Non deterministik
Transisi string kosong tidak terlihat di DFA	NDAFA memungkinkan transisi string kosong
Backtracking diizinkan di DFA	NDAFA backtracking transisi tidak selalu memungkinkan
Membutuhkan lebih banyak ruang	Membutuhkan lebih sedikit ruang
String diterima oleh DFA, jika transit ke keadaan final	String diterima oleh NDAFA, jika setidaknya salah satu dari semua transisi yang mungkin berakhir dalam keadaan terakhir

# Finite State Automata (FSA)



# Deterministic Finite Automata (DFA)

Deterministic Finite Automata (DFA): dari suatu state ada tepat satu state berikutnya untuk setiap simbol masukan yang diterima.

$Q = \{q_0, q_1, q_2\}$

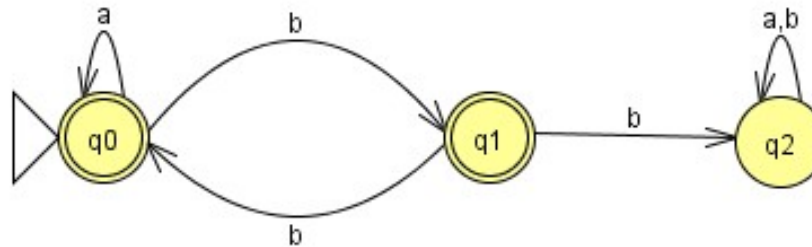
diberikan dalam tabel berikut

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

$S = q_0$

$F = \{q_0, q_1\}$

$\sigma$	a	b
q0	q0	q1
q1	q0	q2
q2	q2	q2



Kalimat yang diterima oleh DFA : a, ab, aa, ab, ba, aba, bab, abab, baba

Kalimat yang ditolak oleh DFA : bb, abb, abba

DFA ini menerima semua susunan kalimat yang tersusun dari simbol a dan b yang tidak mengandung substring bb.



# Deterministic Finite Automata (DFA)

## Contoh:

Pengujian untuk menerima bit string dengan banyaknya 0 genap, serta banyaknya 1 genap.

0011 : diterima

10010: ditolak, karena banyaknya 0 ga

Diagram transisinya:

DFAnya:

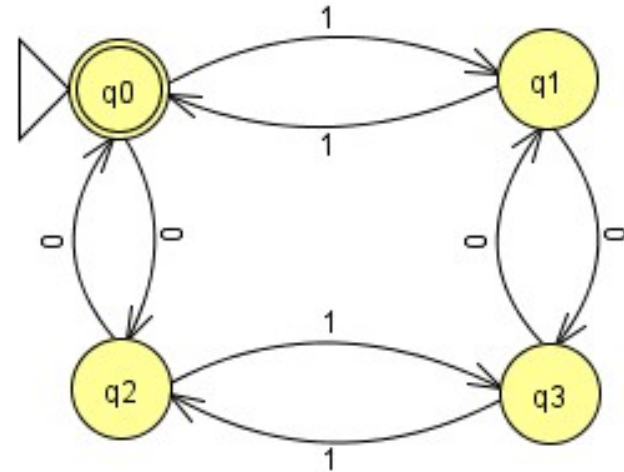
$Q = \{q_0, q_1, q_2, q_3\}$

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

$S = q_0$

$F = \{q_0\}$

Fungsi transisi



$\sigma$	0	1
q0	q2	q1
q1	q3	q0
q2	q0	q3
q3	q1	q2

# Non-deterministic Finite Automata (N DFA)

Non-deterministic finite automata (N DFA): dari suatu state ada 0, 1 atau lebih state berikutnya untuk setiap simbol ,masukan yang diterima.

Berikut adalah contoh NFA ( $Q, \Sigma, \sigma, S, F$ ) dimana:

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

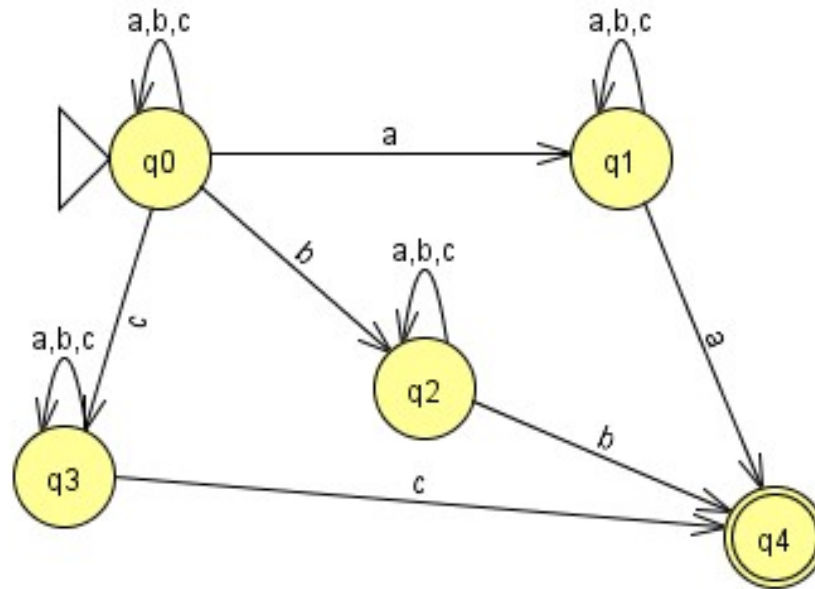
$S = q_0$

$F = \{q_4\}$

$\sigma$  diberikan dalam tabel berikut:

$\sigma$	a	b	c
q0	{q0, q1}	{q0, q2}	{q0, q3}
q1	{q1, q4}	{q1}	{q1}
q2	{q2}	{q2, q4}	{q2}
q3	{q3}	{q3}	{q3, q4}
q4	$\emptyset$	$\emptyset$	$\emptyset$

# Non-deterministic Finite Automata (NFA)



- Kalimat yang diterima NFA diatas : aa, bb, cc, aaa, abb, bcc
- Kalimat yang tidak diterima NFA diatas : a, b, c, ab, ba, ac, bc

# Non-deterministic Finite Automata (Ndfa)

Sebuah Kalimat diterima NFA jika:

- Salah satu tracingnya berakhir di state akhir atau
- Himpunan state setelah membaca string tersebut mengandung state akhir
- Perbedaan dengan DFA: fungsi transisi dapat memiliki 0 atau lebih fungsi transisi.
- $G = ( \{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_2, q_4\} )$

$\sigma$	0	1
$q_0$	$\{q_0, q_3\}$	$\{q_0, q_1\}$
$q_1$	$\epsilon$	$\{q_2\}$
$q_2$	$\{q_2\}$	$\{q_2\}$
$q_3$	$\{q_4\}$	$\epsilon$
$q_4$	$\{q_4\}$	$\{q_4\}$

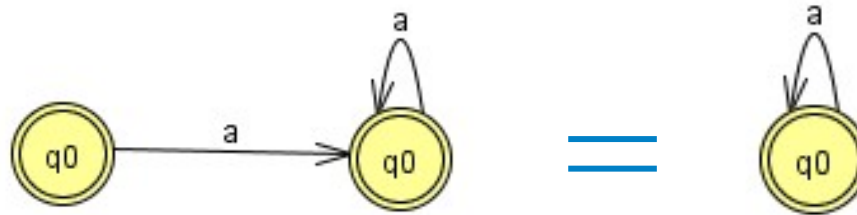
- String diterima NFA bila terdapat suatu urutan transisi berdasar input, dari state awal ke state akhir.
- Harus mencoba semua kemungkinan

## Non-deterministic Finite Automata (N DFA)

Definisi : dua buah FSA disebut ekuivalen apabila kedua FSA tersebut menerima bahasa yang sama

Contoh:

FSA yang menerima bahasa  $\{a^n \mid n \geq 0\}$



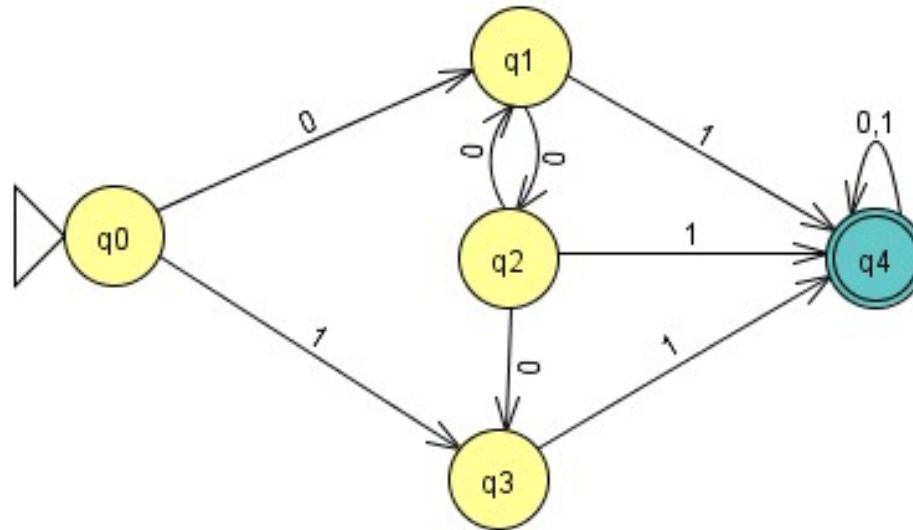
## Non-deterministic Finite Automata (Ndfa)

**Definisi** : dua buah state dari FSA disebut indistinguishable ( tidak dapat dibedakan) apabila:  
 $\sigma(q, w) \in F$  sedangkan  $\sigma(p, w) \notin F$  dan  
 $\sigma(q, w) \notin F$  sedangkan  $\sigma(p, w) \in F$  untuk semua  
 $w \in \Sigma^*$

**Definisi** : Dua buah state dari FSA disebut distinguishable (dapat dibedakan) bila terdapat  $w \in \Sigma^*$  sedemikian hingga :  
 $\sigma(q, w) \in F$  sedangkan  $\sigma(p, w) \notin F$  dan  $\sigma(q, w) \notin F$   
sedangkan  $\sigma(p, w) \in F$  untuk semua  $w \in \Sigma^*$

# Non-deterministic Finite Automata (Ndfa)

- Prosedur menentukan pasangan status indistinguishable
  - Hapus semua state tak dapat dicapai dari state awal.
  - Catat semua pasangan state (p,q) yang distinguishable, yaitu  $\{(p, q) \mid p \in F \wedge q \notin F\}$
  - Untuk setiap pasangan (p, q) sisanya, untuk setiap  $a \in \Sigma$ , tentukan  $\sigma(p, a)$  dan  $\sigma(q, a)$
- Contoh:



# Non-deterministic Finite Automata (Ndfa)

Hapus state yang tidak tercapai → tidak ada

Pasangan distinguishable (q0, q4), (q1, q4), (q2, q4), (q3, q4)

Pasangan sisanya (q0,q1), (q0, q2), (q0, q3), q1, q2), (q1, q3), (q2, q3)

Pasangan n	State 1		State 2		Hasil
	0	1	0	1	
(q0, q1)	Q1	Q3	Q2	Q4	Distinguishable
(q0, q2)	Q1	Q3	Q1	Q4	Distinguishable
(q1, q2)	Q2	Q4	Q1	Q4	InDistinguishable
(q0, q3)	Q1	Q3	Q2	Q4	Distinguishable
(q1, q3)	Q2	Q4	Q2	Q4	InDistinguishable
(q2, q3)	Q1	Q4	Q2	Q4	InDistinguishable

Catatan:

Jumlah pasangan seluruhnya :

$$C\binom{5}{2} = \frac{5!}{2!3!} = 10$$



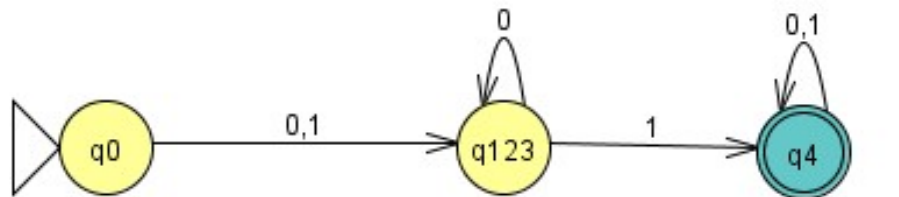
# Non-deterministic Finite Automata (N DFA)

Prosedur Reduksi DFA:

1. Tentukan pasangan status indistinguishable
2. Gabungkan setiap group indistinguishable state ke dalam satu state dengan relasi pembentukan group secara berantai: jika p dan q indistinguishable dan jika q dan r indistinguishable maka p dan r indistinguishable, dan p, q, serta r indistinguishable semua berada dalam satu group.
3. Sesuaikan transisi dari ke state-state gabungan.

**Contoh:**

4. Pasangan status indistinguishable ( $q_1, q_2$ ), ( $q_1, q_3$ ), dan ( $q_2, q_3$ ).
5.  $q_1, q_2, q_3$  ketiganya dapat digabung dalam satu state  $q_1$ .
6. Menyes



# Thank You

[ukrida.ac.id](http://ukrida.ac.id)



**UKRIDA**  
Universitas Kristen Krida Wacana