

## PERTEMUAN IV

# FINITE AUTOMATA & VARIANNYA

---

Mahasiswa memahami Regular Expression dan Automaton  
sebagai alat pendefinisi serta pengenalan anggota bahasa

**DEPARTEMEN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK ELEKTRO & INFORMATIKA  
CERDAS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
2023 - 2027**

# MATERI PERTEMUAN

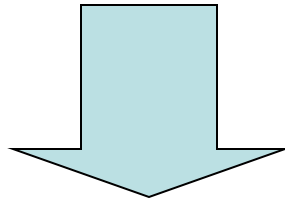
---

- **Deterministic Finite Automata** 
- **Transition Graph** 
- **Automata with Output** 
- **Tugas Mingguan IV** 

# DETERMINISTIC FINITE AUTOMATA (1)

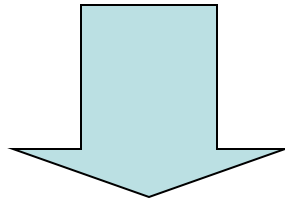
---

Pernah bermain ular dan tangga ?!



Komponen-komponen permainannya apa saja ?

$n \times n$  kotak, Dadu, Penanda, Pemain



Bagaimana karakteristik permainannya ?

# DETERMINISTIC FINITE AUTOMATA (2)

---

Pola/arrah perpindahan kotak/state dalam permainan tersebut bersifat tertentu/mutlak

Jumlah kotak, jumlah dan jenis karakter input terbatas/berhingga

Pergerakan pemain (seolah<sup>2</sup>) bersifat otomatis (ditentukan oleh hasil lemparan dadu)

**Deterministic Finite Automata ( DFA )**

# DETERMINISTIC FINITE AUTOMATA (3)

Secara definitif, DFA memiliki komponen-komponen :

1.  $S$  sebagai himpunan berhingga **state** untuk media perpindahan kendali mesin
2.  $\Sigma$  sebagai himpunan berhingga alphabet untuk **input karakter**
3.  $s_0$  adalah salah satu state dari himpunan  $S$  yang diperlakukan sebagai **start state**
4.  $s_n$  adalah salah satu state dari himpunan  $S$  yang diperlakukan sebagai **final state**  
(DFA dapat memiliki lebih dari satu final state)
5.  $\delta$  sebagai himpunan berhingga **fungsi transisi** untuk memindahkan kendali mesin

## Contoh sebuah DFA :

Himpunan State  $S = \{X, Y, Z\}$

- Himpunan alphabet  $\Sigma = \{a, b\}$
- $X \in S$  sebagai start state
- $Z \in S$  sebagai final state

- Himpunan fungsi transisi  $\delta$  didefinisikan sebagai :

1. Dari  $X$  diberi input  $a$  ke  $Y$
2. Dari  $X$  diberi input  $b$  ke  $Z$
3. Dari  $Y$  diberi input  $a$  ke  $X$
4. Dari  $Y$  diberi input  $b$  ke  $Z$
5. Dari  $Z$  diberi input  $a$  atau  $b$  ke  $Z$

# DETERMINISTIC FINITE AUTOMATA (4)

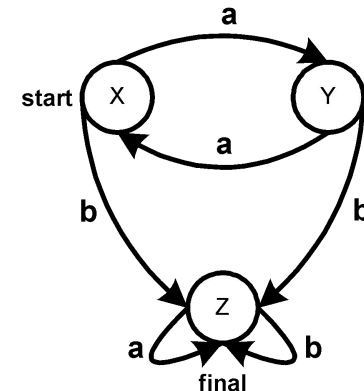
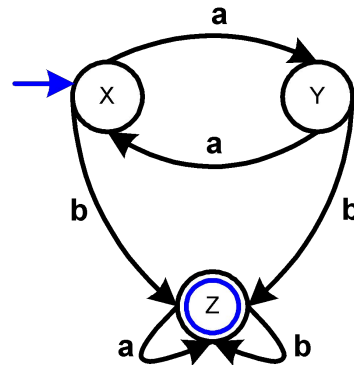
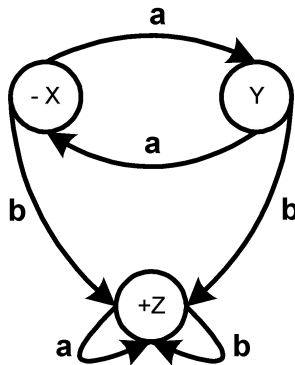
## Contoh sebuah DFA :

- Himpunan State  $S = \{X, Y, Z\}$
- Himpunan alphabet  $\Sigma = \{a, b\}$
  - $X \in S$  sebagai start state
  - $Z \in S$  sebagai final state
  - Himpunan fungsi transisi  $\delta$  didefinisikan sebagai :
    1. Dari X diberi input a ke Y
    2. Dari X diberi input b ke Z
    3. Dari Y diberi input a ke X
    4. Dari Y diberi input b ke Z
    5. Dari Z diberi input a atau b ke Z

## Tabel Transisi

	a	b
(start) X	Y	Z
Y	X	Z
(final) Z	Z	Z

## Labelled Directed Graph (representasi piktorial)



# TRANSITION GRAPH (1)

---

Secara definitif, TG tidak berbeda dengan DFA. Namun dari segi karakteristik, terdapat beberapa perbedaan mendasar antara TG dengan DFA. Perbedaan-perbedaan tersebut antara lain :

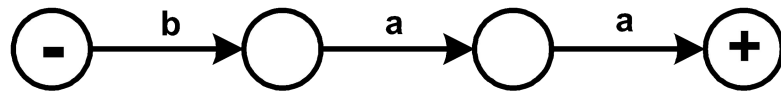
- label edge pada TG dapat berisi string (lebih dari satu karakter)
- jumlah outgoing edge dari setiap state dapat lebih dari satu dengan label edge sama (oleh karenanya TG tidak bersifat deterministic (non-deterministic). Sehingga setiap input string belum tentu memiliki path yang unik)
- TG dapat memiliki lebih dari satu Start State

# TRANSITION GRAPH (2)

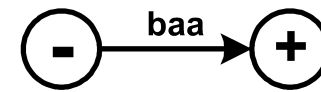


Contoh :

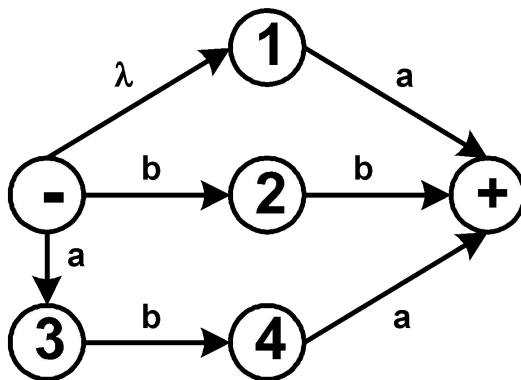
DFA :



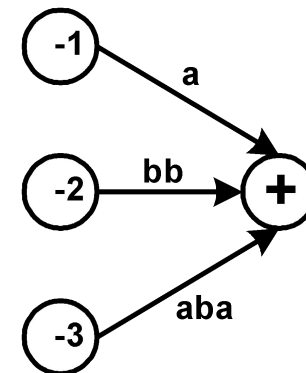
TG :



DFA :



TG :





# FINITE AUTOMATA with OUTPUT (1)

---

Sampai sejauh ini automata yang kita pelajari hanya dapat memberikan hasil/jawaban berupa "ACCEPTED" atau "REJECTED" saja. Untuk sementara, mesin kita ini hanya berfungsi sebagai **Acceptor** saja.

Tapi sekarang, kita akan mencoba membuat sebuah **Transducer**. Yaitu sebuah mesin yang dapat menghasilkan berbagai macam bentuk output (*automata with output*).

2 model *automata with output* yang tersedia, yaitu :

- **Moore Machine**
- **Mealy Machine**

# FINITE AUTOMATA with OUTPUT (2)

---

## MOORE MACHINE

Secara definitif, Moore Machine memiliki komponen-komponen :

1.  $Q$  sebagai himpunan berhingga **state** untuk media perpindahan kendali mesin.  
Label pada setiap state ditulis  $q_i/o$ , dengan  $q_i$  adalah nama state dan  $o$  adalah output yang berkorespondensi dengan state tsb;
2.  $\Sigma$  sebagai himpunan berhingga alphabet untuk **input karakter**
3.  $\Gamma$  sebagai himpunan berhingga karakter untuk **output karakter**
4.  $q_0$  adalah salah satu state dari himpunan  $S$  yang diperlakukan sebagai **start state**
5.  $\delta$  sebagai himpunan berhingga **fungsi transisi** untuk memindahkan kendali mesin

# FINITE AUTOMATA with OUTPUT (3)

## Contoh :

Sebuah Moore Machine didefinisikan sbb :

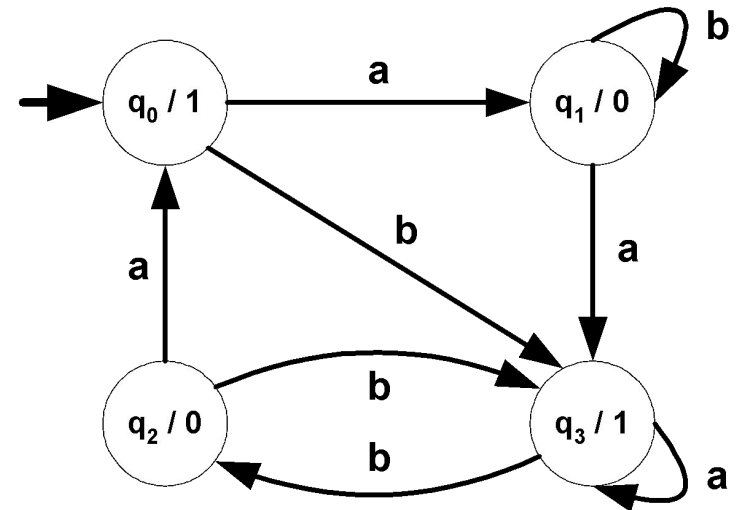
Himpunan state  $Q = \{q_0, q_1, q_2, q_3\}$  dengan  $q_0$  sebagai start state

Himpunan input karakter  $\Sigma = \{a, b\}$

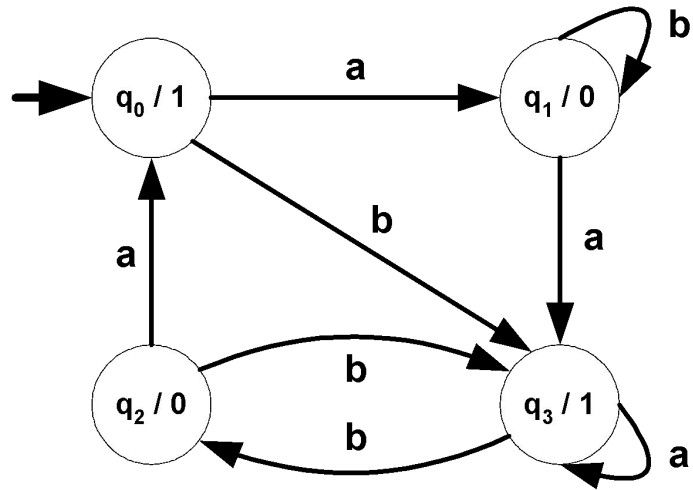
Himpunan output karakter  $\Gamma = \{0, 1\}$

Himpunan fungsi transisi didefinisikan pada tabel berikut :

State Awal	Input a	Input b	Output
- $q_0$	$q_1$	$q_3$	1
$q_1$	$q_3$	$q_1$	0
$q_2$	$q_0$	$q_3$	0
$q_3$	$q_3$	$q_2$	1



# FINITE AUTOMATA with OUTPUT (4)



Sebuah input string **abab**  
akan diproses seperti berikut :

State Awal	Input	State Tujuan	Output
- $q_0$			1
$q_0$	a	$q_1$	1 0
$q_1$	b	$q_1$	1 0 0
$q_1$	a	$q_3$	1 0 0 1
$q_3$	b	$q_2$	1 0 0 1 0

Atau proses penelusurannya dapat pula digambarkan seperti berikut :

Input		a	b	a	b
State	$q_0$	$q_1$	$q_1$	$q_3$	$q_2$
Output	1	0	0	1	0

# FINITE AUTOMATA with OUTPUT (5)

---

## MEALY MACHINE

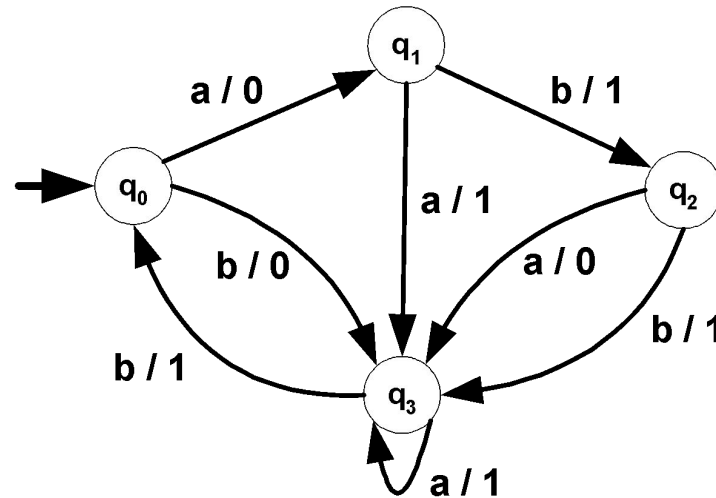
Secara definitif, Mealy Machine memiliki komponen-komponen :

1.  $Q$  sebagai himpunan berhingga **state** untuk media perpindahan kendali mesin.
2.  $\Sigma$  sebagai himpunan berhingga alphabet untuk **input karakter**
3.  $O$  sebagai himpunan berhingga karakter untuk **output karakter**
4.  $q_0$  adalah salah satu state dari himpunan  $S$  yang diperlakukan sebagai **start state**
5.  $\delta$  sebagai himpunan berhingga **fungsi transisi** untuk memindahkan kendali mesin.

Label pada setiap panah ditulis  $\epsilon/o$ , dengan  $\epsilon$  adalah input karakter dan  $o$  adalah output karakter yang berkorespondensi dengan kedua adjacent state tsb.

# FINITE AUTOMATA with OUTPUT (6)

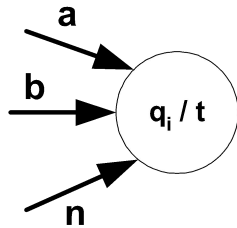
Sebuah contoh Mealy Machine dan penelusurannya untuk input **aaabb** :



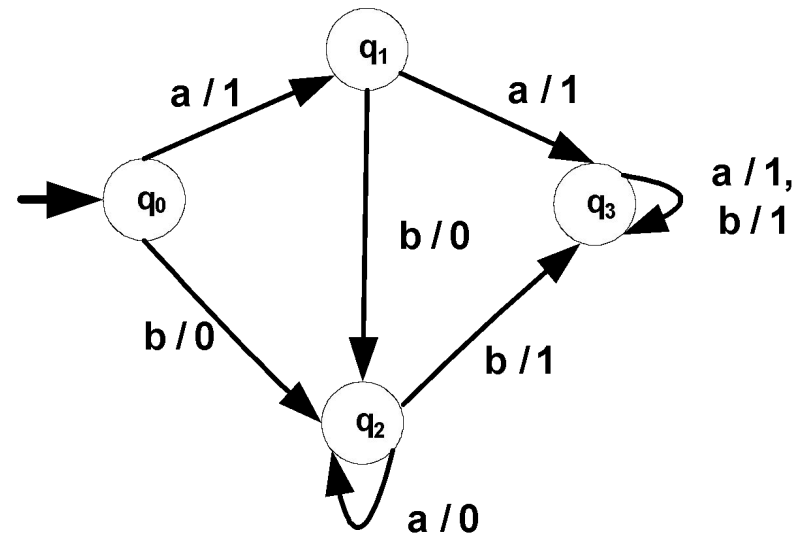
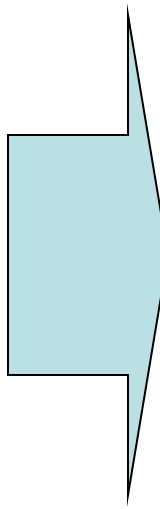
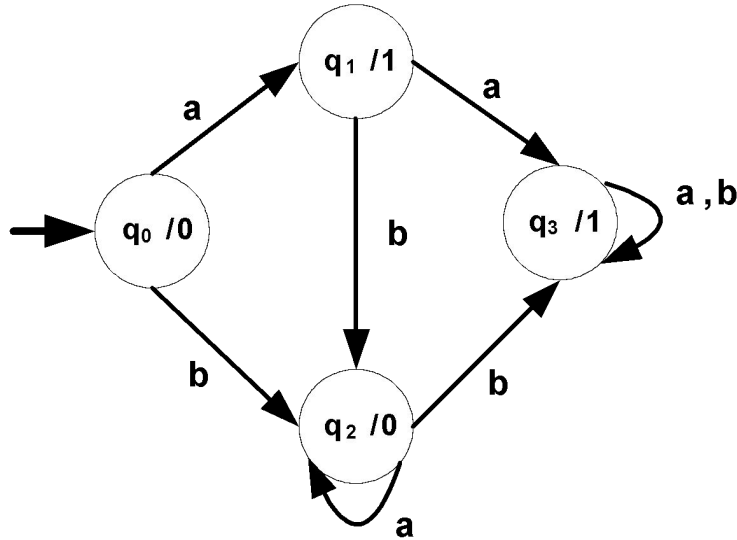
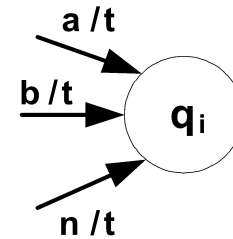
Input		a	a	a	b	b
State	$q_0$	$q_1$	$q_3$	$q_3$	$q_0$	$q_3$
Output		0	1	1	1	0

# FINITE AUTOMATA with OUTPUT (7)

Proses konversi dari Moore Machine ke bentuk Mealy Machine dapat dilakukan dengan sangat mudah.

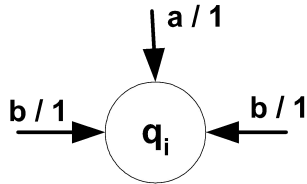


menjadi

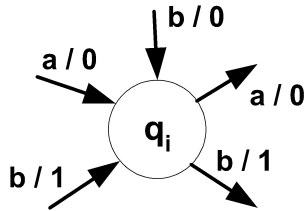
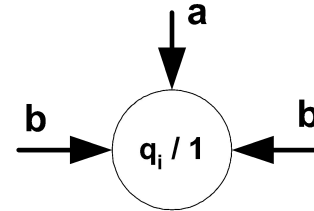


# FINITE AUTOMATA with OUTPUT (8)

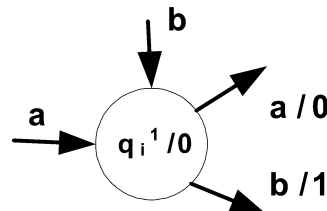
Tetapi sebaliknya, konversi dari Mealy Machine ke bentuk Moore Machine agak sedikit 'ribet'.



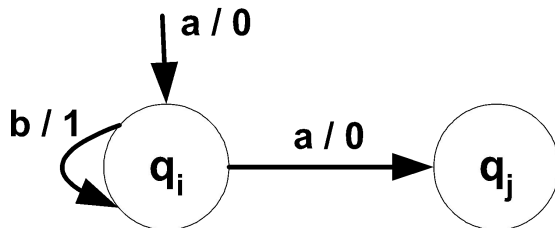
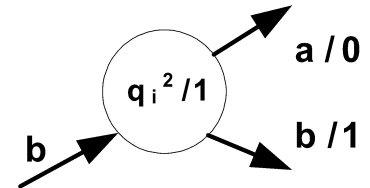
menjadi



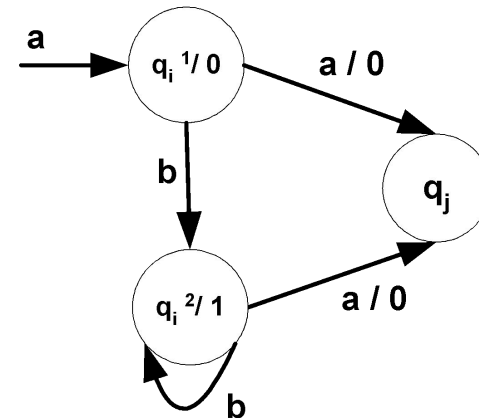
menjadi



dan



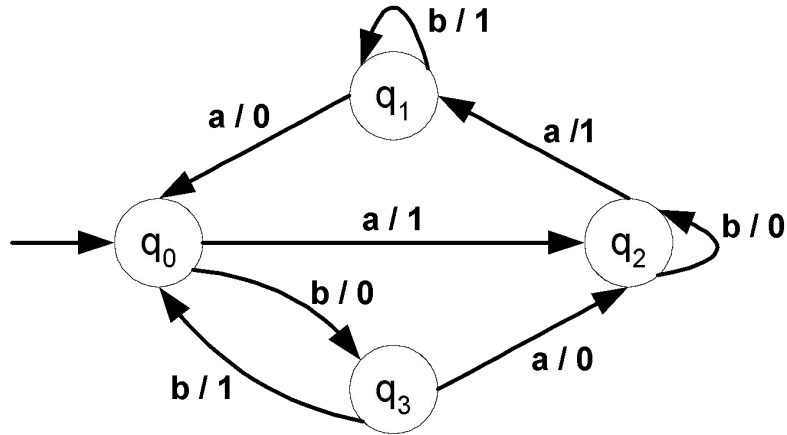
menjadi





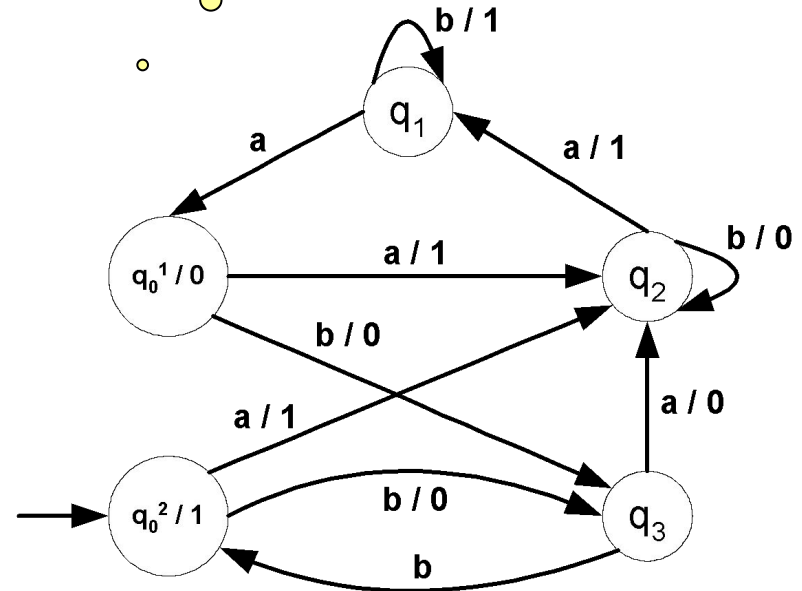
# FINITE AUTOMATA with OUTPUT (9)

Mealy Machine berikut dapat dikonversikan menjadi Moore Machine :



Gambarkan dulu semua state & edge yg tidak terlibat dg proses konversi saat itu

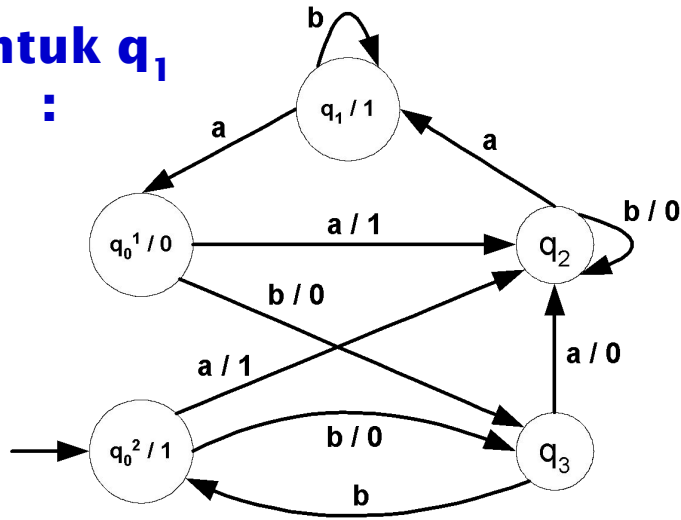
Untuk  $q_0$  :



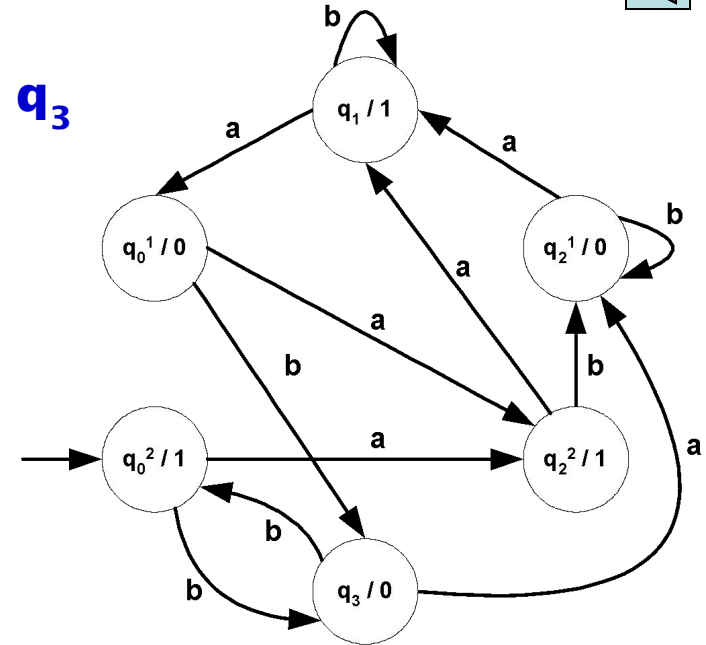
# FINITE AUTOMATA with OUTPUT (10)



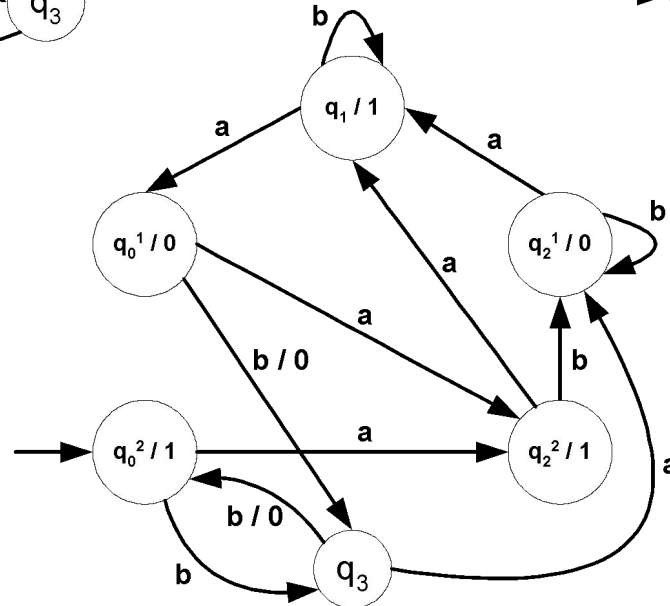
Untuk  $q_1$   
:



Untuk  $q_3$   
:



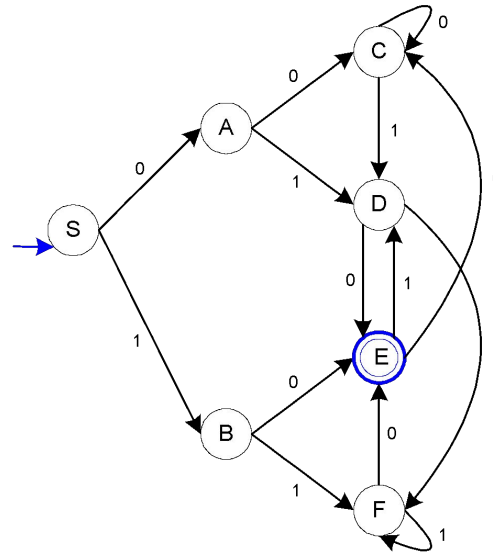
Untuk  $q_2$  :



# TUGAS MINGGUAN IV



- 1, Amatilah lingkungan hidup anda sehari-hari. Tentukan sebuah obyek (misalnya, sistem lift, sistem traffic light, sistem perpanjangan STKB, dll) yang anda dapat gambarkan/modelkan dengan Automata. **(no groups tells the same stories & no groups adopts my examples above)**



2. Lakukan penelusuran input-input string berikut pada FSD di atas :
  - a. 10
  - b. 010
  - c. 110
  - d. 110010
  - e. 0010
  - f. 001