

Ekspresi Reguler

Pertemuan 5

Mahasiswa mampu menerapkan konsep Ekspresi Reguler

materi

- Penerapan Ekspresi Reguler
- Notasi Ekspresi Reguler
- Hubungan Ekspresi Reguler dengan Finite State Automata
- Finite State Automata untuk suatu Tata Bahasa Reguler

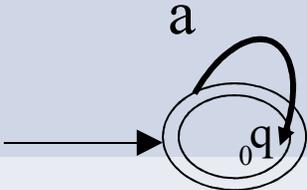
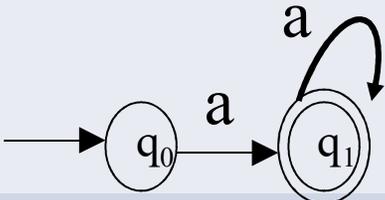
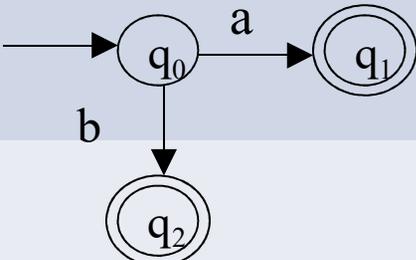
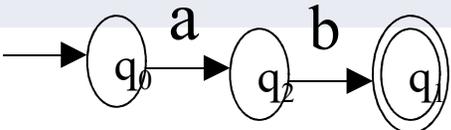
Ekpresi Reguler (ER)

Sebuah bahasa dinyatakan reguler jika terdapat FSA yang dapat menerimanya.

Penerapan ER misalnya

1. pencarian (*searching*) untai karakter (*string*) pada suatu file. Biasanya ada pada teks editor.
2. Pada kompilator ER di aplikasikan untuk melakukan analisis leksikal

Notasi Ekspresi Reguler

Notasi	ER	string	Mesin
* (asterik) Dimana string bisa muncul 0..n kali	$L(M) = a^*$	$L(M) = \{\epsilon, a, aa, aaa, \dots\}$	 <p>A DFA with a single state q_0 which is both the start and final state. There is a self-loop transition on state q_0 labeled 'a'.</p>
+ (pada posisi <i>superscript</i>) Dimana string bisa muncul 1..n kali	$L(M) = a^+$	$L(M) = \{a, aa, aaa, \dots\}$	 <p>A DFA with two states: q_0 (start state) and q_1 (final state). There is a transition from q_0 to q_1 labeled 'a', and a self-loop transition on q_1 labeled 'a'.</p>
+ atau U Berarti union	$L(M) = a \cup b$	$L(M) = \{a, b\}$	 <p>A DFA with three states: q_0 (start state), q_1 (final state), and q_2 (final state). There is a transition from q_0 to q_1 labeled 'a', and a transition from q_0 to q_2 labeled 'b'.</p>
. (titik) Berarti kontatenasi	$L(M) = a.b$	$L(M) = \{ab\}$	 <p>A DFA with three states: q_0 (start state), q_1 (final state), and q_2 (intermediate state). There is a transition from q_0 to q_2 labeled 'a', and a transition from q_2 to q_1 labeled 'b'.</p>

Contoh:

ER: a^*

Untai yang bisa dibangkitkan: $\epsilon, a, aa, aaa, \dots$

ER: ab^*

Untai yang bisa dibangkitkan: $a, ab, abb, abbb, \dots$

ER: $a^* \cup b^*$

Untai yang bisa dibangkitkan:

$a, b, aa, bb, aaa, bbb, \dots$

ER: $(a \cup b)^*$

Untai yang bisa dibangkitkan:

$a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, \dots$

ER: $a^+ d$

Himpunan untaian yang bisa dibangkitkan:

$ad, aad, aaad, \dots$

ER: 010^*

Himpunan untaian yang bisa dibangkitkan:

$01, 010, 0100, \dots$

ER: ab^*cc

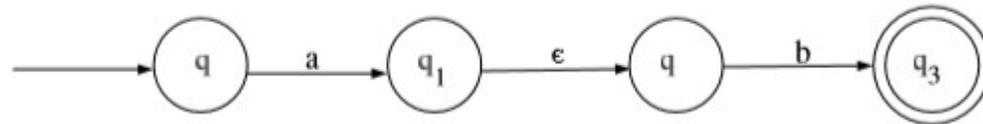
Himpunan untaian yang bisa dibangkitkan:

$acc, abcc, abbbcc, \dots$

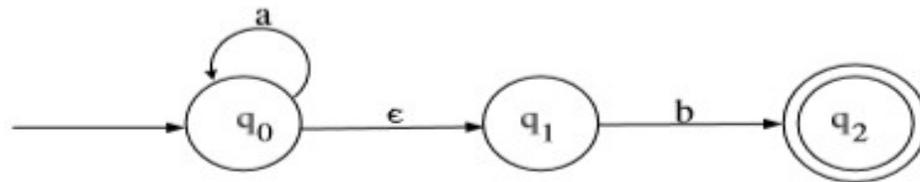
Hubungan ER dengan FSA

- Untuk setiap ekspresi regular ada satu *Non-deterministic Finite Automata* dengan transisi ϵ (NFA ϵ -move) yg ekuivalen.
- Untuk setiap *Deterministic Finite Automata* ada satu ekspresi regular dari bahasa yg diterima oleh *Deterministic Finite Automata* (DFA).

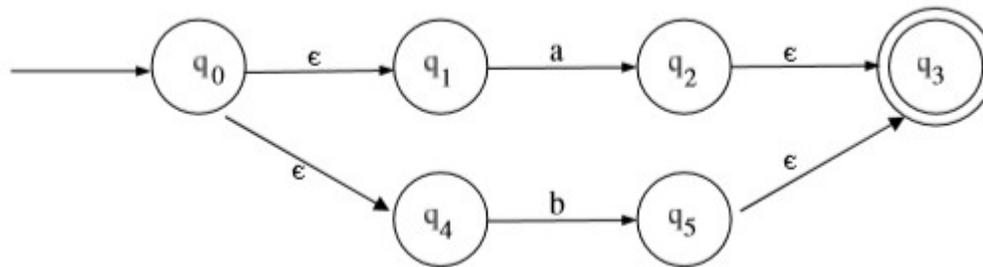
Hubungan ER dengan FSA



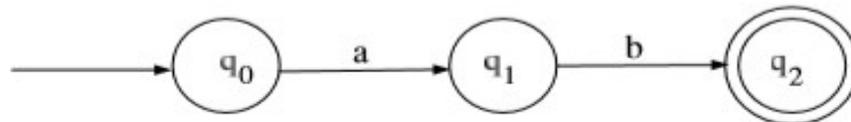
NFA ϵ – move untuk ER : ab



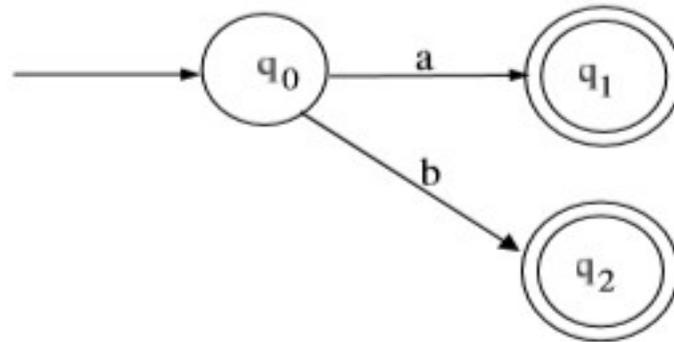
NFA ϵ – move untuk ER : $a^* b$



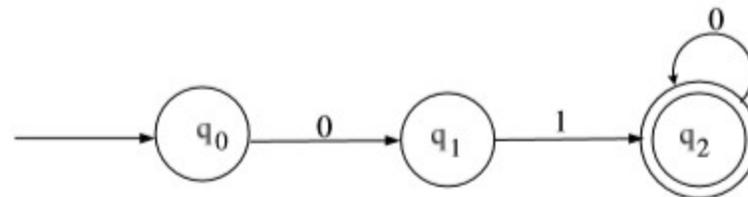
NFA ϵ – move untuk ER : $a \cup b$



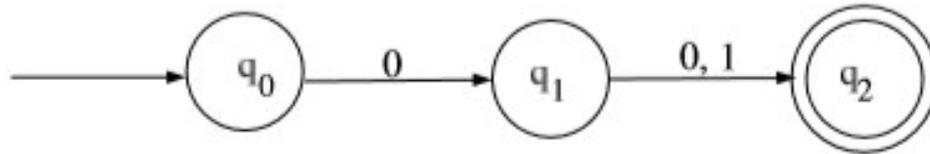
NFA ϵ – move untuk ER : a.b



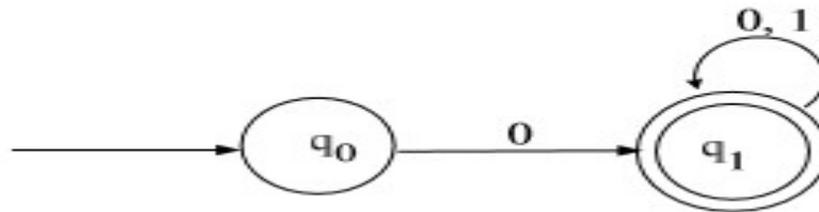
NFA untuk ER : a U b



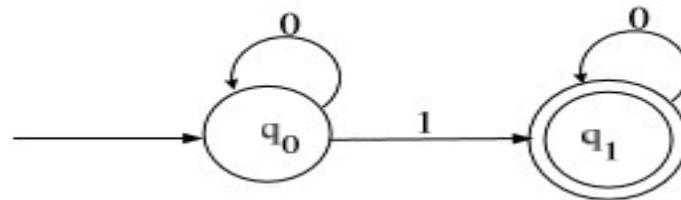
NFA untuk ER : 010*



NFA untuk ER : $0(1 \cup 0)$

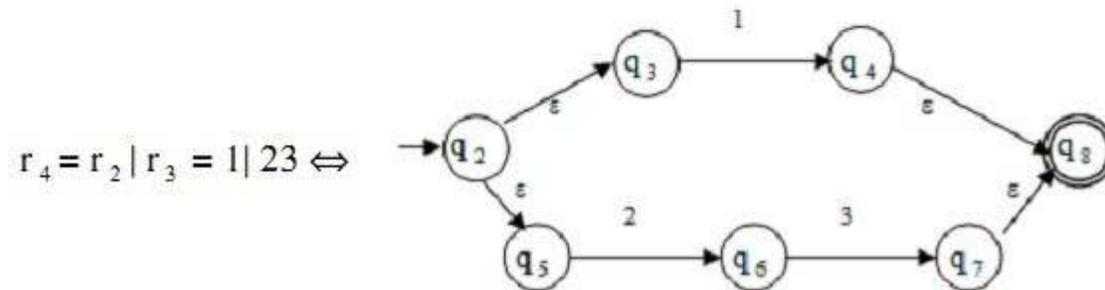
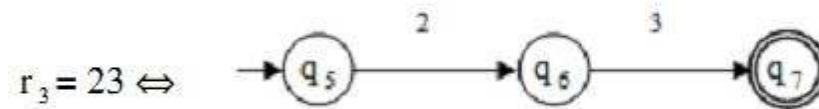
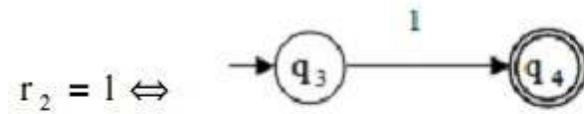
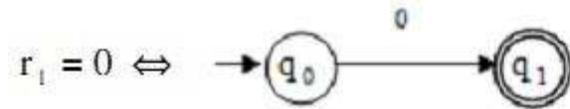


NFA untuk ER : $0(1 \cup 0)^*$

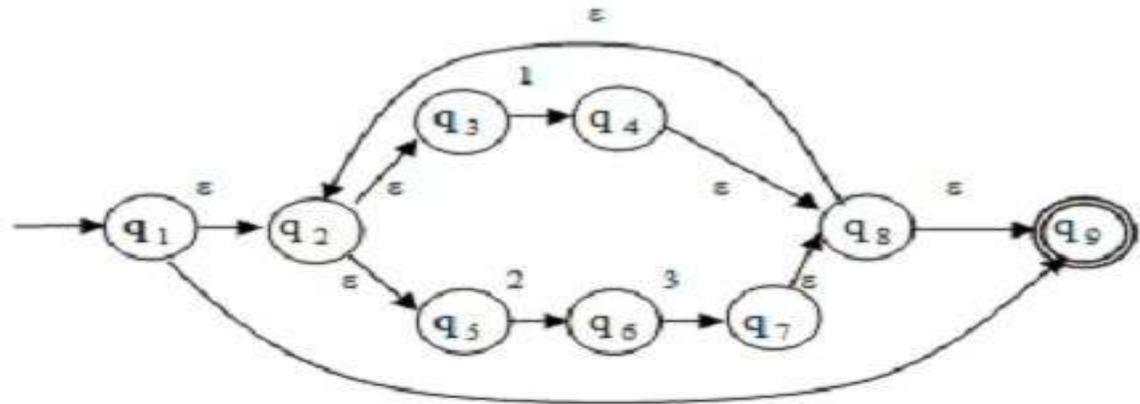


NFA untuk ER : 0^*10^*

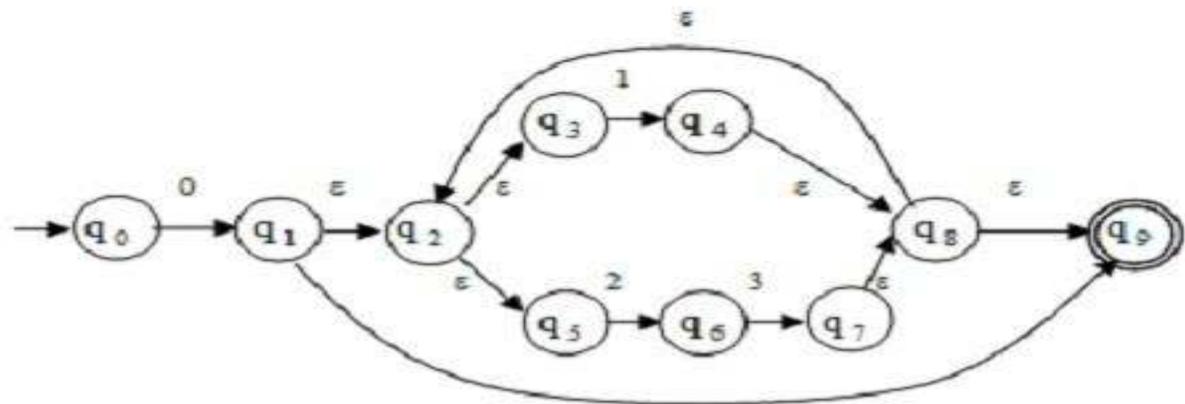
NFA untuk ekspresi regular $r = 0(1|23)^*$



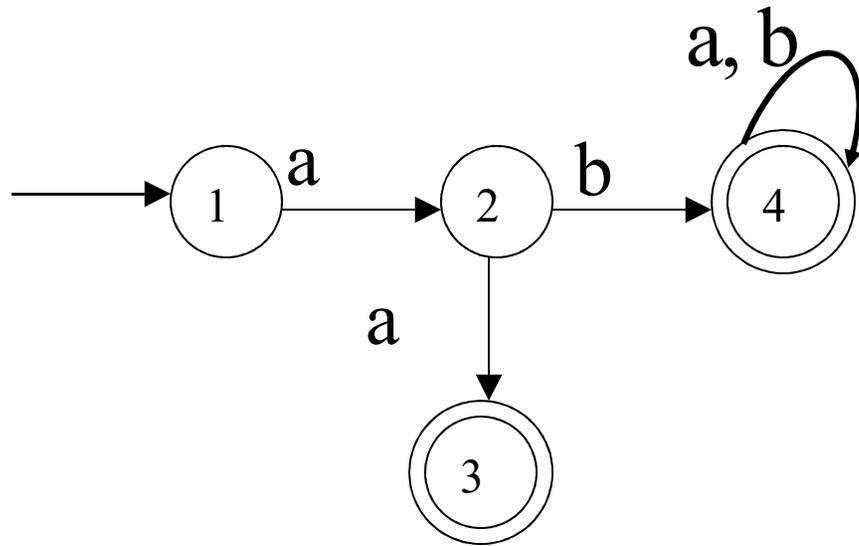
$$r_5 = r_4^* = (1|23)^* \Leftrightarrow$$



$$r = r_5 = 0(1|23)^* \Leftrightarrow$$



Finite State Automata untuk suatu ER

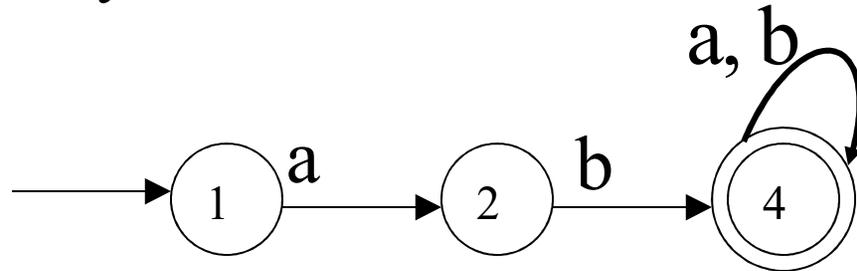


Mesin FSA M5

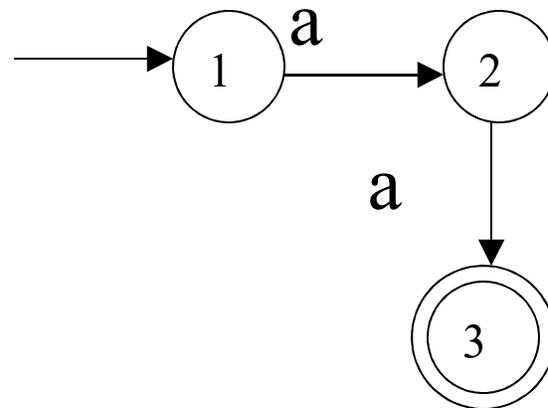
Tentukan ER FSA M5??

Langkah FSA ke ER

1. Bagi mesin menjadi mesin yang hanya memiliki satu state akhir



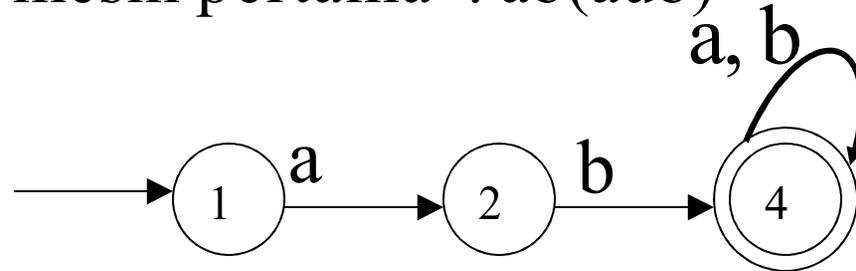
Dan



Langkah FSA ke ER

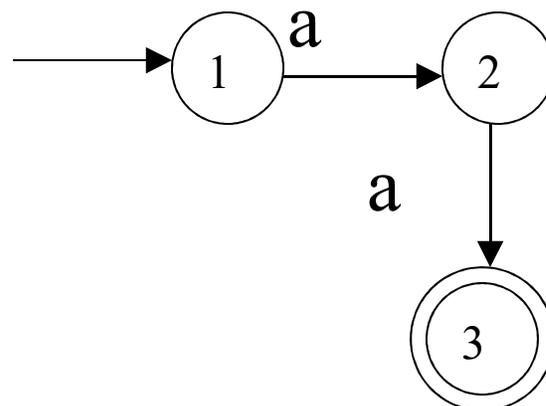
2. Bentuk ER

mesin pertama : $ab(aub)^*$



Dan

mesin kedua : (ab)



Langkah FSA ke ER

2. Gabungkan

mesin pertama : $ab(aub)^*$

mesin kedua : (aa)

Menjadi

$$\begin{aligned} L(M5) &= ab(aub)^* \cup (aa) \\ &= a(b(aub)^* \cup (a)) \end{aligned}$$

- Tata Bahasa Regular (FSA) bisa kita konstruksikan dengan :
 - Ekspresi Regular
 - Aturan Produksi

Aturan Produksi Bahasa Regular

- Sebuah otomata berhingga menspesifikasikan sebuah bahasa sebagai himpunan semua untai yang menggerakkannya dari *state* awal ke salah satu dari *state* yang diterimanya (himpunan *state* akhir).

Batasan Aturan Produksi untuk Bahasa Regular

$\alpha \rightarrow \beta$ (dibaca α menghasilkan β)

α adalah sebuah simbol variabel.

β maksimal memiliki sebuah simbol variabel yang terletak di paling kanan bila ada.

- α atau β bisa berupa simbol terminal atau non-terminal/variabel.
- Simbol variabel/non-terminal adalah simbol yang masih bisa diturunkan. Biasanya dinyatakan dengan huruf besar, misal: A, B, C.

- Suatu tata bahasa (grammar) didefinisikan dengan 4 tupel ($G = \{V, T, P, S\}$) dimana:

V = himpunan simbol variabel/non terminal.

- T = himpunan simbol terminal.
- P = kumpulan aturan produksi
- S = simbol awal

Finite State Automata untuk Suatu Tata Bahasa Regular

Bila sebelumnya dari suatu diagram transisi Finite State Automata kita bisa membuat aturan – aturan produksinya, sebaliknya kita juga bisa mengkonstruksi diagram transisi finite state automata untuk suatu tata bahasa regular yang diketahui aturan – aturan produksinya.

Misalkan terdapat tata bahasa regular dengan aturan produksi.

$$S \rightarrow aB \mid bA \mid \epsilon$$

$$A \rightarrow abaS \quad B \rightarrow babS$$

Maka diagram transisinya dapat digambar sebagai berikut.

