

PERTEMUAN 12

ATURAN PRODUKSI UNTUK SUATU TATA BAHASA REGULAR DAN FINITE STATE AUTOMATA UNTUK SUATU TATA BAHASA REGULAR

A. TUJUAN PEMBELAJARAN

Pembahasan pada pertemuan ini secara keseluruhan mengenai Aturan Produksi untuk Suatu Tata Bahasa Regular dan Finite State Automata untuk suatu tata bahasa Regular. Maka setelah mengikuti perkuliahan pertemuan 10, mahasiswa diharapkan mampu :

- 1) Menjabarkan Aturan Produksi untuk Suatu Tata Bahasa Regular
- 2) Membuat Finite State Automata untuk suatu tata bahasa Regular

B. URAIAN MATERI

1. Aturan Produksi untuk Suatu Tata Bahasa Regular

Batasan aturan produksi untuk bahasa regular :

Suatu tata bahasa (Grammar) didefinisikan dengan 4 Tupel yaitu : V, T, P, dan S

Di mana,

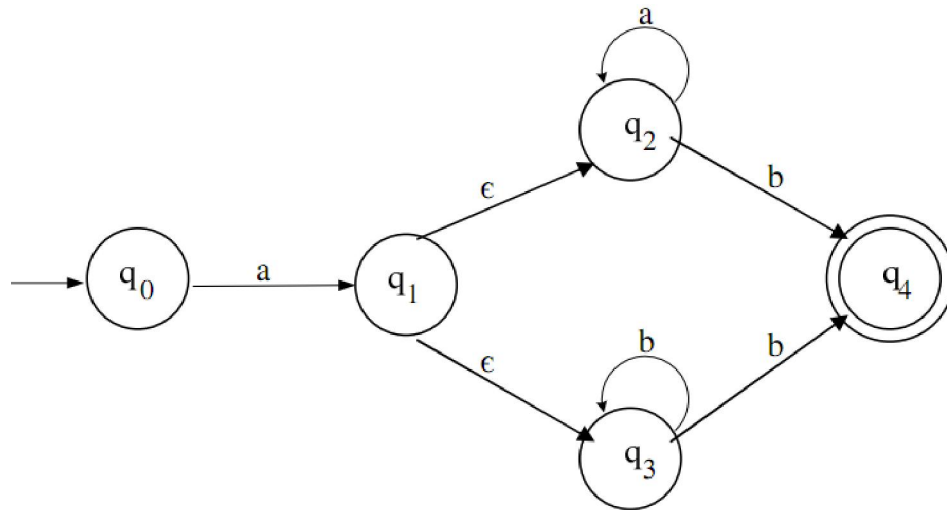
V = Himpunan simbol variabel / non terminal

T = Himpunan simbol terminal

P = Kumpulan aturan produksi

S = Simbol awal

Sebagai contoh terdapat Mesin FSA berikut



Mesin finite state automata pada gambar di atas memiliki simbol input ‘a’ dan ‘b’. Simbol ‘a’ dan ‘b’ akan menjadi simbol terminal pada aturan produksi yang akan kita bentuk.

Misalnya kita tentukan simbol awal adalah S. Kita identikkan S dengan state awal q0. Dari q0 mendapat input a menjadi q1. Bisa kita tuliskan sebagai aturan produksi :

S → aE

Maka kita gunakan sebagai E dan bukan A karena menyatakan bagian yang belum terbangkitkan mulai dari state q1.

Dari q1 mendapat transisi – (tanpa menerima input) ke q2 dan q3 . Bisa kita tuliskan :

E → A

E → B

(kita identikkan q2 sebagai A dan q3 sebagai B)

Dari q2 jika mendapat input a menuju ke state q2 itu sendiri dan jika mendapat input b menuju ke state q4 yang merupakan state akhir dan tidak menuju ke state yang lainnya sehingga dapat dituliskan menjadi :

A → aA

A → b

Dari q_3 jika mendapat input b menuju ke state q_3 itu sendiri dan jika mendapat input b juga menuju ke state q_4 yang merupakan state akhir dan tidak menuju ke state yang

lainnya sehingga dapat dituliskan menjadi :

$B \rightarrow bB$

$B \rightarrow b$

Kumpulan aturan produksi yang kita peroleh bisa kita tuliskan sebagai berikut.

$S \rightarrow aE$

$E \rightarrow A \mid B$

$A \rightarrow aA \mid b$

$B \rightarrow bB \mid b$

Secara formal tata bahasa yang diperoleh dari otomata tersebut adalah sebagai berikut.

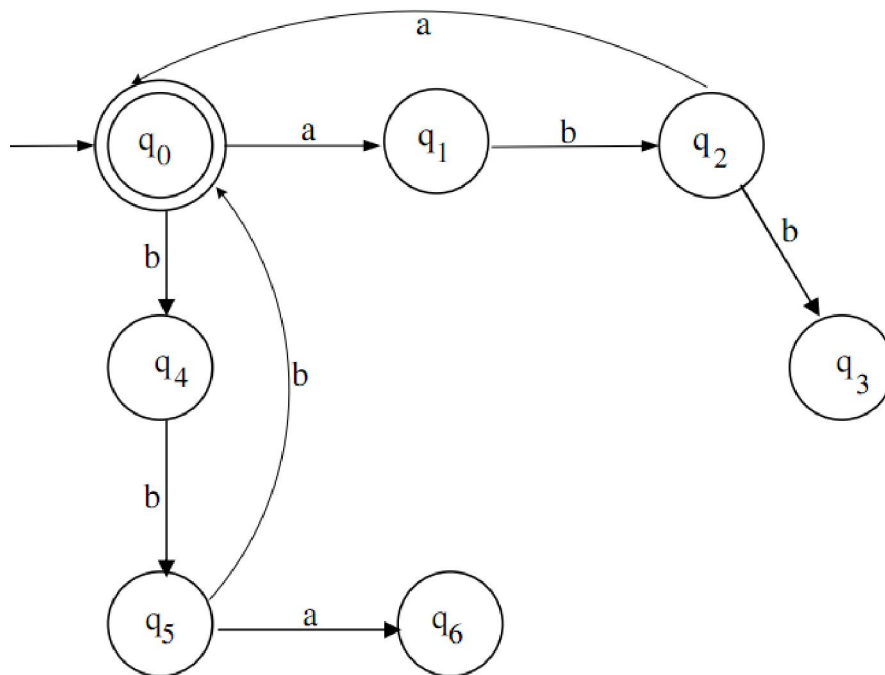
$V = \{S, E, A, B\}$

$T = \{a, b\}$

$P = \{ S \rightarrow aE, E \rightarrow A \mid B, A \rightarrow aA \mid b, B \rightarrow bB \mid b \}$

$S = S$

Contoh lainnya



Kita bisa mengkonstruksi aturan produksi untuk otomata tersebut.

$T = \{a, b\}$

$S = S$

Kita mulai dari state awal yaitu q_0 yang dalam hal ini dilambangkan dengan S .

- Bila S mendapat input a maka menuju ke q_1 yang dalam hal ini dilambangkan dengan

$A.$

$S \xrightarrow{a} A$

- Bila S mendapat input b maka menuju ke q_4 yang dalam hal ini dilambangkan dengan

$B.$

$S \xrightarrow{b} B$

- Karena q_0 dalam hal ini sebagai state akhir dan masih memiliki transisi keluar, maka untuk menandakannya sebagai state akhir kita buat :

S

Kemudian setelah itu kita lihat q1 yang tadi telah kita lambangkan sebagai A.

- Jika A mendapat input b maka menuju q2 yang dalam hal ini dilambangkan sebagai C.

A bC

Kemudian kita lihat q4 yang telah kita identikkan sebagai B.

- Jika B mendapat input b maka menuju ke q5 yang kita lambangkan sebagai D.

○ B bD

Kemudian kita lihat q2 yang telah kita lambangkan sebagai C.

- Jika C mendapat input b maka menuju ke q3 (Tetapi karena q3 tidak mempunyai transisi keluar dan bukan merupakan state akhir maka dapat kita abaikan.
- Jika C mendapat input a maka menuju ke S.

C aS

Kemudian kita lihat q5 yang telah kita lambangkan sebagai D.

- Jika D mendapat input b maka menuju ke S.
- Jika D mendapat input a maka menuju ke q6 (tetapi karena q6 bukan merupakan state akhir dan tidak ada transisi keluar dari q6 maka dapat diabaikan)

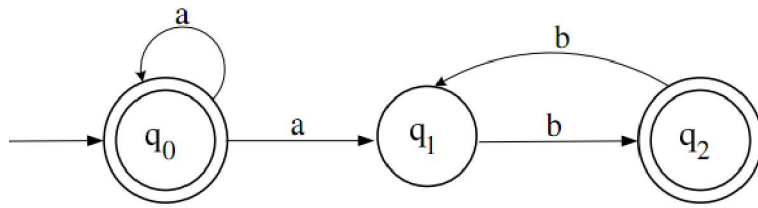
D bS

Maka Diperoleh :

$V = \{S, A, B, C, D\}$

$P = \{S \xrightarrow{a} A \mid S \xrightarrow{b} B, A \xrightarrow{b} C, B \xrightarrow{b} D, C \xrightarrow{a} S, D \xrightarrow{b} S\}$

Contoh selanjutnya :



$T = \{a, b\}$

$S = S$

Kita mulai dari state awal yaitu q_0 yang dalam hal ini dilambangkan dengan S .

- Bila S mendapat input a maka menuju ke q_1 yang dalam hal ini dilambangkan dengan A .

$S \xrightarrow{a} A$

- Bila S mendapat input a maka menuju ke S sendiri

$S \xrightarrow{a} S$

- Karena S adalah state awal dan bukan satu-satunya state akhir maka tidak perlu ditambahkan S

Kemudian kita lihat q_1 yang dalam hal ini dilambangkan dengan A .

- Jika A mendapat input b maka menuju ke state q_2 yang dalam hal ini dilambangkan dengan B .

$A \xrightarrow{b} B$

Kemudian kita lihat q_2 yang dalam hal ini dilambangkan dengan B

- Jika B mendapat input b maka menuju ke state q_1 yang dalam hal ini dilambangkan dengan A .

$B \xrightarrow{b} A$

- Jika B mendapat input a maka menuju ke state q3, tetapi karena q3 bukan merupakan state akhir dan tidak mempunyai transisi keluar maka dapat diabaikan.
- Karena B merupakan state akhir dan mempunyai transisi keluar maka untuk menandainya ditulis.

B

Maka diperoleh :

$$V = \{S, A, B\}$$

$$P = \{S \rightarrow aA \mid aS, A \rightarrow bB, B \rightarrow bA \mid \}$$

2. 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.

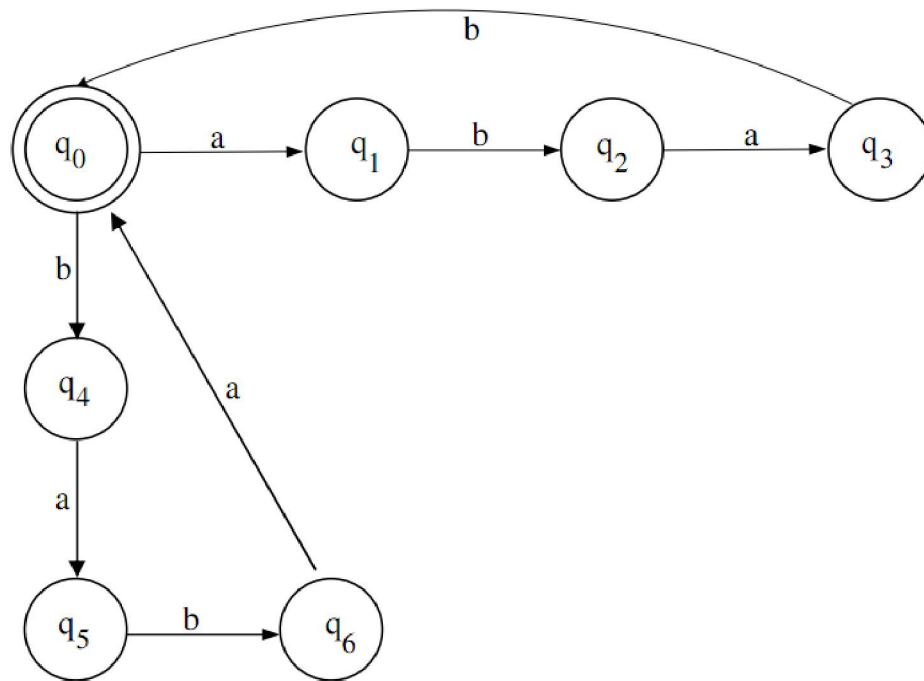
Misalkan terdapat tata bahasa regular dengan aturan produksi.

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

$$A \rightarrow abaS$$

$$B \rightarrow babS$$

Maka diagram transisi nya dapat digambarkan sebagai berikut:



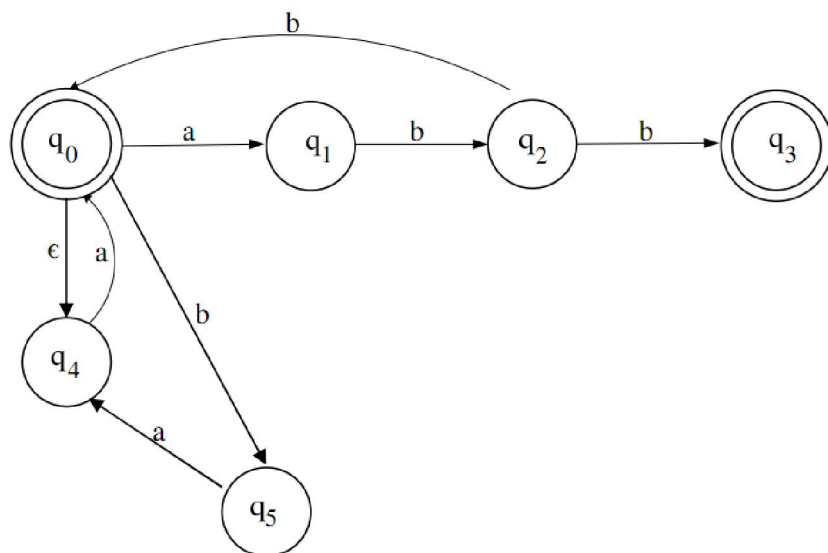
Contoh lainnya :

S $abA \mid B \mid baB \mid$

A $bS \mid b$

B aS

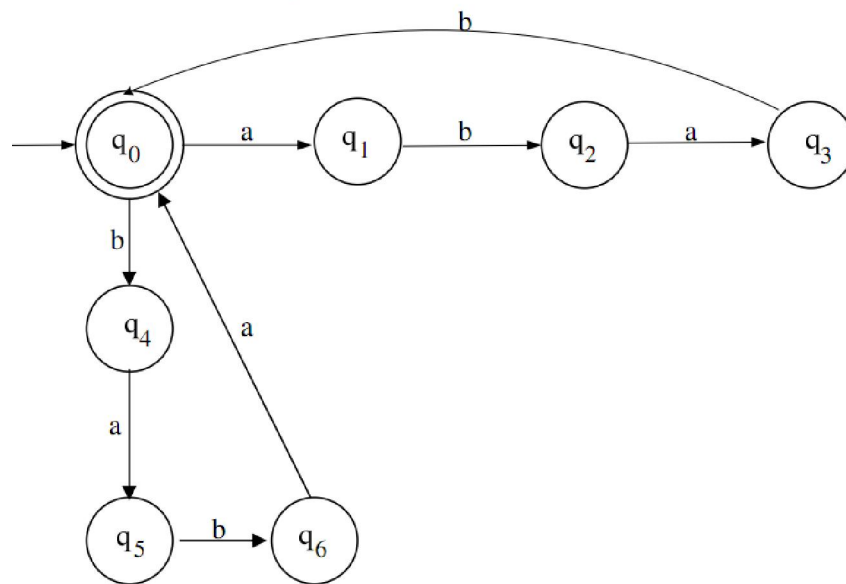
Maka dapat digambarkan sebagai berikut :



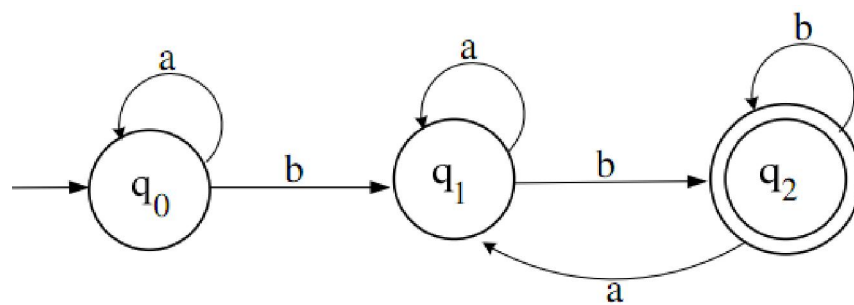
C. SOAL LATIHAN 11 DAN TUGAS

Konstruksi kan tata bahasa regular untuk otomata berikut :

a)



b)



b. Berdasarkan ER berikut, maka tentukan minimal 10 input string yang dapat diterima

- $b^*(ab)^*a^*$
- $(a^* + b^*)(a^* \cup b^*)(a^* \cup b^*)$
- $a^*(baa^*)^*b^*$

D. DAFTAR PUSTAKA

Hopcroft, John. E., etc. 2001. Second edition. Introduction to Automata Theory, Languages, and Computation. US America : Pearson

Martin, John C. 2010. Fourth Edition. Introduction to Language and The Theory of Computation. United State America : McGraw-Hill

Modul Teori bahasa Automata. (www.ibbi.ac.id/ibbiacid/bahan/teori-bahasa-dan-otomata). Diakses pada tanggal 20 November 2015

Santosa, Kussigit. Modul Teori Bahasa Otomata. Universitas Pamulang Fakultas Teknik Program Studi Teknik Informatika. Pamulang