

TEORI BAHASA DAN OTOMATA

MATERI KULIAH :

	Topik	Substansi
1	Kontrak pembelajaran, Pendahuluan	a. Ketentuan dalam Kuliah b. Pengertian Bahasa c. Pengertian Otomata
2	Pengertian Dasar dan Operasi pada string	a. Pngertian Dasar Simbol dll b. Operasi dasar string
3	Grammar dan Bahasa	a. Definisi Grammar b. Klasifikasi Grammar/bahasa c. Penentuan bahasa dari suatu grammar d. Penentuan grammar dari suatu bahasa
4, 5	Mesin Pengenal Bahasa (OTOMATA)	a. Macam-macam mesin pengenal bahasa b. Finite State Automata c. Ekuivalensi NFA-DFA
6	Ekspresi Reguler.	a. Pengertian ER b. Menentukan ER dari suatu bahasa reguler c. Membuat NFA dari ER
7	Ujian sisipan	
8, 9	Bahasa Bebas Konteks	a. Penyederhanaan tata bahasa bebas konteks b. Bentuk Normal Chomsky
10, 11	PushDown Automata (PDA)	a. Pengertian PDA b. PDA deterministik/non deterministik.
12	Mesin Turing	a. Pengertian Mesin Turing b. Penerimaan pada MT
13 - 15	Topik Khusus	Topik-topik khusus/ masalah2 yang lebih kompleks dari teori bahasa dan otomata.
16	Ujian Akhir	

Buku :

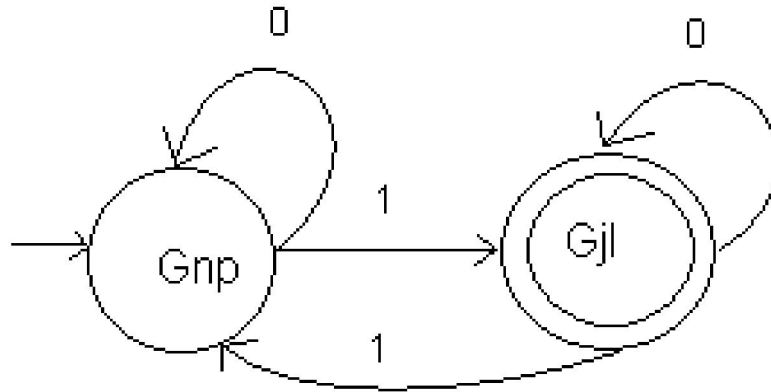
- Teori Bahasa dan Otomata, John E. Hopcroft dkk. (terjemahan, Edisi 2, 2007)
- Teori Bahasa dan Otomata, Firrar Utdirartatmo
- Introduction to Languages and The Theory of Computation, John C. Martin
- An Introduction to Formal Language and Automata, Peter Linz

Teori Bahasa

- Teori bahasa membicarakan bahasa formal (formal language), terutama untuk kepentingan perancangan kompilator (compiler) dan pemroses naskah (text processor).
- Bahasa formal adalah kumpulan kalimat. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (grammar) yang sama.
- Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda.
- Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya.
- Bahasa Natural/manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya 'bahasa formal' akan disebut 'bahasa' saja.

Otomata (Automata)

- Otomata adalah mesin abstrak yang dapat mengenali (recognize), menerima (accept), atau membangkitkan (generate) sebuah kalimat dalam bahasa tertentu.



Beberapa Pengertian Dasar :

- Simbol adalah sebuah entitas abstrak (seperti halnya pengertian titik dalam geometri). Sebuah huruf atau sebuah angka adalah contoh simbol.
- String adalah deretan terbatas (finite) simbol-simbol. Sebagai contoh, jika a, b, dan c adalah tiga buah simbol maka abcb adalah sebuah string yang dibangun dari ketiga simbol tersebut.
- Jika w adalah sebuah string maka panjang string dinyatakan sebagai $|w|$ dan didefinisikan sebagai cacahan (banyaknya) simbol yang menyusun string tersebut. Sebagai contoh, jika $w = abcb$ maka $|w| = 4$.
- String hampa adalah sebuah string dengan nol buah simbol. String hampa dinyatakan dengan simbol ϵ (atau \wedge) sehingga $|\epsilon| = 0$. String hampa dapat dipandang sebagai simbol hampa karena keduanya tersusun dari nol buah simbol.
- Alfabet adalah himpunan hingga (finite set) simbol-simbol

Operasi Dasar String

Diberikan dua string : $x = abc$, dan $y = 123$

- Prefix string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling belakang dari string w tersebut.
Contoh : abc, ab, a, dan ϵ adalah semua Prefix(x)
- ProperPrefix string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling belakang dari string w tersebut.
Contoh : ab, a, dan ϵ adalah semua ProperPrefix(x)

- Postfix (atau Sufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling depan dari string w tersebut.
Contoh : abc , bc , c , dan ϵ adalah semua $\text{Postfix}(x)$
- ProperPostfix (atau ProperSufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling depan dari string w tersebut.
Contoh : bc , c , dan ϵ adalah semua $\text{ProperPostfix}(x)$
- Head string w adalah simbol paling depan dari string w .
Contoh : a adalah $\text{Head}(x)$
- Tail string w adalah string yang dihasilkan dari string w dengan menghilangkan simbol paling depan dari string w tersebut.
Contoh : bc adalah $\text{Tail}(x)$
- Substring string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.
Contoh : abc , ab , bc , a , b , c , dan ϵ adalah semua $\text{Substring}(x)$
- ProperSubstring string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.
Contoh : ab , bc , a , b , c , dan ϵ adalah semua $\text{Substring}(x)$
- Subsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan nol atau lebih simbol-simbol dari string w tersebut.
Contoh : abc , ab , bc , ac , a , b , c , dan ϵ adalah semua $\text{Subsequence}(x)$
- ProperSubsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan satu atau lebih simbol-simbol dari string w tersebut.
Contoh : ab , bc , ac , a , b , c , dan ϵ adalah semua $\text{Subsequence}(x)$
- Concatenation adalah penyambungan dua buah string. Operator concatenation adalah concat atau tanpa lambang apapun.
Contoh : $\text{concat}(xy) = xy = abc123$
- Alternation adalah pilihan satu di antara dua buah string. Operator alternation adalah alternate atau $|$.
Contoh : $\text{alternate}(xy) = x | y = abc \text{ atau } 123$
- Kleene Closure : $x^* = \epsilon | x | xx | xxx | \dots = \epsilon | x | x^2 | x^3 | \dots$
- Positive Closure : $x^+ = x | xx | xxx | \dots = x | x^2 | x^3 | \dots$

Beberapa Sifat Operasi

- Tidak selalu berlaku : $x = \text{Prefix}(x)\text{Postfix}(x)$
- Selalu berlaku : $x = \text{Head}(x)\text{Tail}(x)$
- Tidak selalu berlaku : $\text{Prefix}(x) = \text{Postfix}(x)$ atau $\text{Prefix}(x) \neq \text{Postfix}(x)$
- Selalu berlaku : $\text{ProperPrefix}(x) \neq \text{ProperPostfix}(x)$
- Selalu berlaku : $\text{Head}(x) \neq \text{Tail}(x)$
- Setiap $\text{Prefix}(x)$, $\text{ProperPrefix}(x)$, $\text{Postfix}(x)$, $\text{ProperPostfix}(x)$, $\text{Head}(x)$, dan $\text{Tail}(x)$ adalah $\text{Substring}(x)$, tetapi tidak sebaliknya
- Setiap $\text{Substring}(x)$ adalah $\text{Subsequence}(x)$, tetapi tidak sebaliknya

- Dua sifat aljabar concatenation :
 - ♦ Operasi concatenation bersifat asosiatif : $x(yz) = (xy)z$
 - ♦ Elemen identitas operasi concatenation adalah ϵ : $\epsilon x = x\epsilon = x$
- Tiga sifat aljabar alternation :
 - ♦ Operasi alternation bersifat komutatif : $x \mid y = y \mid x$
 - ♦ Operasi alternation bersifat asosiatif : $x \mid (y \mid z) = (x \mid y) \mid z$
 - ♦ Elemen identitas operasi alternation adalah dirinya sendiri : $x \mid x = x$
- Sifat distributif concatenation terhadap alternation : $x (y \mid z) = xy \mid xz$
- Beberapa kesamaan :
 - ♦ Kesamaan ke-1 : $(x^*)^* = x^*$
 - ♦ Kesamaan ke-2 : $\epsilon \mid x^+ = x^+ \mid \epsilon = x^+$
 - ♦ Kesamaan ke-3 : $(x \mid y)^* = \epsilon \mid x \mid y \mid xx \mid yy \mid xy \mid yx \mid \dots =$ semua string yang merupakan concatenation dari nol atau lebih x , y , atau keduanya.

GRAMMAR DAN BAHASA

Konsep Dasar

- Anggota alfabet dinamakan simbol terminal.
- Kalimat adalah deretan hingga simbol-simbol terminal.
- Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa tak hingga kalimat.
- Simbol-simbol berikut adalah simbol terminal :
 - ü huruf kecil, misalnya : $a, b, c, 0, 1, ..$
 - ü simbol operator, misalnya : $+, -, \text{ dan } \times$
 - ü simbol tanda baca, misalnya : $(,), \text{ dan } ;$
 - ü string yang tercetak tebal, misalnya : **if, then, dan else.**
- Simbol-simbol berikut adalah simbol non terminal /Variabel :
 - ü huruf besar, misalnya : A, B, C
 - ü huruf S sebagai simbol awal
 - ü string yang tercetak miring, misalnya : *expr*

- Huruf Yunani melambangkan string yang tersusun atas simbol-simbol terminal atau simbol-simbol non terminal atau campuran keduanya, misalnya : α , β , dan γ .
- Sebuah produksi dilambangkan sebagai $\alpha \rightarrow \beta$, artinya : dalam sebuah derivasi dapat dilakukan penggantian simbol α dengan simbol β .
- Derivasi adalah proses pembentukan sebuah kalimat atau sentensial. Sebuah derivasi dilambangkan sebagai : $\alpha \Rightarrow \beta$.
- Sentensial adalah string yang tersusun atas simbol-simbol terminal atau simbol-simbol non terminal atau campuran keduanya.
- Kalimat adalah string yang tersusun atas simbol-simbol terminal. Kalimat adalah merupakan sentensial, sebaliknya belum tentu..

Grammar :

Grammar G didefinisikan sebagai pasangan 4 tuple : V_T , V_N , S , dan P , dan dituliskan sebagai $G(V_T, V_N, S, P)$, dimana :

V_T : himpunan simbol-simbol terminal (alfabet) \ni kamus

V_N : himpunan simbol-simbol non terminal

$S \in V_N$: simbol awal (atau simbol start)

P : himpunan produksi

Contoh :

1. G_1 : $V_T = \{I, \text{Love}, \text{Miss}, \text{You}\}$, $V_N = \{S, A, B, C\}$,
 $P = \{S \rightarrow ABC, A \rightarrow I, B \rightarrow \text{Love} \mid \text{Miss}, C \rightarrow \text{You}\}$

$$\begin{aligned} S &\Rightarrow ABC \\ &\Rightarrow \text{IloveYou} \end{aligned}$$

$$L(G_1) = \{\text{IloveYou}, \text{IMissYou}\}$$

$$2. \text{ . } G_2 : V_T = \{a\}, V_N = \{S\}, P = \{S \rightarrow aS \mid a\}$$

$$\begin{aligned} S &\Rightarrow aS \\ &\Rightarrow aaS \\ &\Rightarrow aaa \end{aligned} \quad L(G_2) = \{a^n \mid n \geq 1\}$$

$$L(G_2) = \{a, aa, aaa, aaaa, \dots\}$$

Klasifikasi Chomsky

Berdasarkan komposisi bentuk ruas kiri dan ruas kanan produksinya ($\alpha \rightarrow \beta$), Noam Chomsky mengklasifikasikan 4 tipe grammar :

1. Grammar tipe ke-0 : Unrestricted Grammar (UG)
Ciri : $\alpha, \beta \in (V_T \mid V_N)^*, |\alpha| > 0$
2. Grammar tipe ke-1 : Context Sensitive Grammar (CSG)
Ciri : $\alpha, \beta \in (V_T \mid V_N)^*, 0 < |\alpha| \leq |\beta|$
3. Grammar tipe ke-2 : Context Free Grammar (CFG)
Ciri : $\alpha \in V_N, \beta \in (V_T \mid V_N)^*$
4. Grammar tipe ke-3 : Regular Grammar (RG)
Ciri : $\alpha \in V_N, \beta \in \{V_T, V_T V_N\}$ atau $\alpha \in V_N, \beta \in \{V_T, V_N V_T\}$

Tipe sebuah grammar (atau bahasa) ditentukan dengan aturan sebagai berikut :

A language is said to be type- i ($i = 0, 1, 2, 3$) language if it can be specified by a type- i grammar but can't be specified any type- $(i+1)$ grammar.

Contoh Analisa Penentuan Type Grammar

1. Grammar G_1 dengan $P_1 = \{S \rightarrow aB, B \rightarrow bB, B \rightarrow b\}$.

Ruas kiri semua produksinya terdiri dari sebuah V_N maka G_1 kemungkinan tipe CFG atau RG. Selanjutnya karena semua ruas kanannya terdiri dari sebuah V_T atau string $V_T V_N$ maka G_1 adalah RG(3).

2. Grammar G_2 dengan $P_2 = \{S \rightarrow Ba, B \rightarrow Bb, B \rightarrow b\}$.

Ruas kiri semua produksinya terdiri dari sebuah V_N maka G_2 kemungkinan tipe CFG atau RG. Selanjutnya karena semua ruas kanannya terdiri dari sebuah V_T atau string $V_N V_T$ maka G_2 adalah RG(3).

3. Grammar G_3 dengan $P_3 = \{S \rightarrow Ba, B \rightarrow bB, B \rightarrow b\}$.

Ruas kiri semua produksinya terdiri dari sebuah V_N maka G_3 kemungkinan tipe CFG atau RG. Selanjutnya karena ruas kanannya mengandung string $V_T V_N$ (yaitu bB) dan juga string $V_N V_T$ (Ba) maka G_3 bukan RG, dengan kata lain G_3 adalah CFG(2).

4. Grammar G_4 dengan $P_4 = \{S \rightarrow aAb, B \rightarrow aB\}$.

Ruas kiri semua produksinya terdiri dari sebuah V_N maka G_4 kemungkinan tipe CFG atau RG. Selanjutnya karena ruas kanannya mengandung string yang panjangnya lebih dari 2 (yaitu aAb) maka G_4 bukan RG, dengan kata lain G_4 adalah CFG.

5. Grammar G_5 dengan $P_5 = \{S \rightarrow aA, S \rightarrow aB, aAb \rightarrow aBCb\}$.

Ruas kirinya mengandung string yang panjangnya lebih dari 1 (yaitu aAb) maka G_5 kemungkinan tipe CSG atau UG. Selanjutnya karena semua ruas kirinya lebih pendek atau sama dengan ruas kanannya maka G_5 adalah CSG.

6. Grammar G_6 dengan $P_6 = \{aS \rightarrow ab, SAc \rightarrow bc\}$.

Ruas kirinya mengandung string yang panjangnya lebih dari 1 maka G_6 kemungkinan tipe CSG atau UG. Selanjutnya karena terdapat ruas kirinya yang lebih panjang daripada ruas kananya (yaitu SAc) maka G_6 adalah UG.

Derivasi Kalimat dan Penentuan Bahasa

Tentukan bahasa dari masing-masing gramat berikut :

1. G_1 dengan $P_1 = \{1. S \rightarrow aAa, 2. A \rightarrow aAa, 3. A \rightarrow b\}$.

Jawab :

Derivasi kalimat terpendek :

$$\begin{aligned} S &\Rightarrow aAa & (1) \\ &\Rightarrow aba & (3) \end{aligned}$$

Derivasi kalimat umum :

$$\begin{aligned} S &\Rightarrow aAa & (1) \\ &\Rightarrow aaAaa & (2) \\ &\dots \\ &\Rightarrow a^n Aa^n & (2) \\ &\Rightarrow a^n ba^n & (3) \end{aligned}$$

Dari pola kedua kalimat disimpulkan : $L_1(G_1) = \{ a^n ba^n \mid n \geq 1 \}$

2. G_2 dengan

$P_2 = \{1. S \rightarrow aS, 2. S \rightarrow aB, 3. B \rightarrow bC, 4. C \rightarrow aC, 5. C \rightarrow a\}$.

Jawab :

Derivasi kalimat terpendek :

$$\begin{aligned} S &\Rightarrow aB & (2) \\ &\Rightarrow abC & (3) \\ &\Rightarrow aba & (5) \end{aligned}$$

Derivasi kalimat umum :

$$\begin{aligned} S &\Rightarrow aS & (1) \\ &\dots \\ &\Rightarrow a^{n-1}S & (1) \\ &\Rightarrow a^n B & (2) \\ &\Rightarrow a^n bC & (3) \\ &\Rightarrow a^n baC & (4) \\ &\dots \\ &\Rightarrow a^n ba^{m-1}C & (4) \\ &\Rightarrow a^n ba^m & (5) \end{aligned}$$

Dari pola kedua kalimat disimpulkan : $L_2(G_2) = \{a^n ba^m \mid n \geq 1, m \geq 1\}$

3. G_3 dengan

$P_3 = \{1. S \rightarrow aSBC, 2. S \rightarrow abC, 3. bB \rightarrow bb, 4. bC \rightarrow bc, 5. CB \rightarrow BC, 6. cC \rightarrow cc\}$.

Jawab :

Derivasi kalimat terpendek 1:
3 :

$$\begin{aligned} S &\Rightarrow abC & (2) \\ &\Rightarrow abc & (4) \end{aligned}$$

Derivasi kalimat terpendek 2 :

$$\begin{aligned} S &\Rightarrow aSBC & (1) \\ &\Rightarrow aabCBC & (2) \\ &\Rightarrow aabBCC & (5) \quad aabcBC & (4) \\ &\Rightarrow aabbCC & (3) \\ &\Rightarrow aabbcC & (4) \\ &\Rightarrow aabbcc & (6) \end{aligned}$$

Derivasi kalimat terpendek

$$\begin{aligned} S &\Rightarrow aSBC & (1) \\ &\Rightarrow aaSBCBC & (1) \\ &\Rightarrow aaabCBCBC & (2) \\ &\Rightarrow aaabBCCBC & (5) \\ &\Rightarrow aaabBCBCC & (5) \\ &\Rightarrow aaabBBCCC & (5) \\ &\Rightarrow aaabbBCCC & (3) \\ &\Rightarrow aaabbbCCC & (3) \\ &\Rightarrow aaabbbccC & (4) \\ &\Rightarrow aaabbbccC & (6) \\ &\Rightarrow aaabbbccc & (6) \end{aligned}$$

Dari pola ketiga kalimat disimpulkan : $L_3 (G_3) = \{ a^n b^n c^n \mid n \geq 1 \}$

Menentukan Grammar Sebuah Bahasa

1. Tentukan sebuah grammar regular untuk bahasa $L_1 = \{ a^n \mid n \geq 1 \}$

Jawab :

$$P_1(L_1) = \{ S \rightarrow aS \mid a \}$$

2. Tentukan sebuah grammar bebas konteks untuk bahasa :
 L_2 : himpunan bilangan bulat non negatif ganjil

Jawab :

Langkah kunci : digit terakhir bilangan harus ganjil.

$$V_t = \{ 0, 1, 2, \dots, 9 \}$$

$$V_n = \{ S, G, J \}$$

$$P = \{ S \rightarrow HT \mid JT \mid J; T \rightarrow GT \mid JT \mid J; H \rightarrow 2 \mid 4 \mid 6 \mid 8;$$

$$G \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8; J \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9 \}$$

$$P = \{ S \rightarrow GS \mid JS \mid J; G \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8; J \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9 \}$$

Buat dua buah himpunan bilangan terpisah : genap (G) dan ganjil (J)

$$P_2(L_2) = \{S \rightarrow J \mid GS \mid JS, G \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8, J \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9\}$$

3. Tentukan sebuah gramat bebas konteks untuk bahasa :

B. L_3 = himpunan semua identifier yang sah menurut bahasa pemrograman Pascal dengan batasan : terdiri dari simbol huruf kecil dan angka, panjang identifier boleh lebih dari 8 karakter

Jawab :

Langkah kunci : karakter pertama identifier harus huruf.

Buat dua himpunan bilangan terpisah : huruf (H) dan angka (A)

$S \rightarrow HT \mid H; T \rightarrow HT \mid AT \mid H \mid A; H \rightarrow a \mid .. \mid z; A \rightarrow 0 \mid .. \mid 9$

$$P_3(L_3) = \{S \rightarrow H \mid HT, T \rightarrow AT \mid HT \mid H \mid A, \\ H \rightarrow a \mid b \mid c \mid \dots, A \rightarrow 0 \mid 1 \mid 2 \mid \dots\}$$

4. Tentukan gramat bebas konteks untuk bahasa

$$L_4(G_4) = \{a^n b^m \mid n, m \geq 1, n \neq m\}$$

Jawab :

Langkah kunci : sulit untuk mendefinisikan $L_4(G_4)$ secara langsung. Jalan keluarnya adalah dengan mengingat bahwa $x \neq y$ berarti $x > y$ atau $x < y$.

$L_4 = L_A \cup L_B$, $L_A = \{a^n b^m \mid n > m \geq 1\}$, $L_B = \{a^n b^m \mid 1 \leq n < m\}$.

$P_A(L_A) = \{A \rightarrow aA \mid aC, C \rightarrow aCb \mid ab\}$, $Q(L_B) = \{B \rightarrow Bb \mid Db, D \rightarrow aDb \mid ab\}$

$P_4(L_4) = \{S \rightarrow A \mid B, A \rightarrow aA \mid aC, C \rightarrow aCb \mid ab, B \rightarrow Bb \mid Db, D \rightarrow aDb \mid ab\}$

5. Tentukan sebuah grammar bebas konteks untuk bahasa :

$L_5 =$ bilangan bulat non negatif genap. Jika bilangan tersebut terdiri dari dua digit atau lebih maka nol tidak boleh muncul sebagai digit pertama.

Jawab :

Langkah kunci : Digit terakhir bilangan harus genap. Digit pertama tidak boleh nol. Buat tiga himpunan terpisah : bilangan genap tanpa nol (G), bilangan genap dengan nol (N), serta bilangan ganjil (J).

$P_5(L_5) = \{S \rightarrow N \mid GA \mid JA, A \rightarrow N \mid NA \mid JA, G \rightarrow 2 \mid 4 \mid 6 \mid 8, N \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8, J \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9\}$

C.Mesin Pengenal Bahasa

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

Kelas Bahasa	Mesin Pengenal Bahasa
Unrestricted Grammar (UG)	Mesin Turing (Turing Machine), TM

Context Sensitive Grammar (CSG)	Linear Bounded Automata, LBA
Context Free Grammar (CFG)	Pushdown Automata, PDA
Regular Grammar, RG	Finite State Automata, FSA

FINITE STATE AUTOMATA (FSA)

- FSA didefinisikan sebagai pasangan 5 tuple : $(Q, \Sigma, \delta, S, F)$.

Q : himpunan hingga state

Σ : himpunan hingga simbol input (alfabet)

δ : fungsi transisi, menggambarkan transisi state FSA akibat pembacaan simbol input.

Fungsi transisi ini biasanya diberikan dalam bentuk tabel.

$S \in Q$: state AWAL

$F \subset Q$: himpunan state AKHIR

Contoh : FSA untuk mengecek parity ganjil

$Q = \{Gnp, Gjl\}$

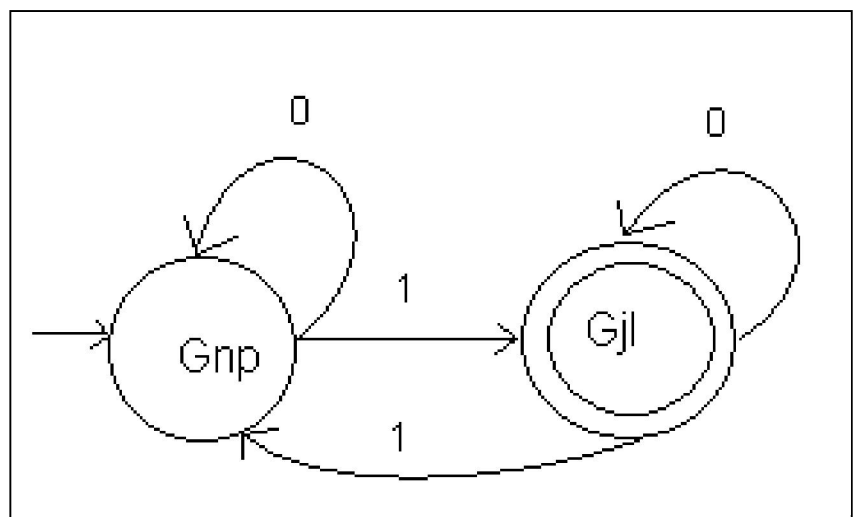
diagram transisi

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

tabel transisi

	0	1
Gnp	Gnp	Gjl
Gjl	Gjl	Gnp

$S = Gnp, F = \{Gjl\}$



- Ada dua jenis FSA :

- Deterministic finite automata (DFA)

- Non deterministik finite automata.(NFA)

- DFA : transisi state FSA akibat pembacaan sebuah simbol bersifat tertentu.

$$: Q \times \rightarrow Q$$

- NFA : transisi state FSA akibat pembacaan sebuah simbol bersifat tak tentu.

$$: Q \times \rightarrow 2^Q$$

DFA :

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

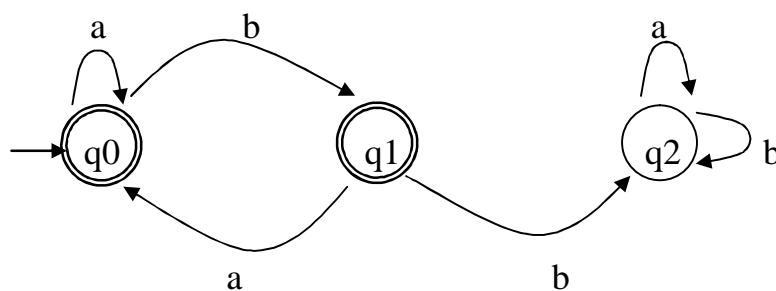
diberikan dalam tabel berikut :

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

$$S = q_0$$

$$F = \{q_0, q_1\}$$

	a	b
q0	q0	q1
q1	q0	q2
q2	q2	q2



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

Kalimat yang ditolak oleh DFA : bb, abb, abba

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

Contoh :

Telusurilah, apakah kalimat-kalimat berikut diterima DFA di atas :

abababaa \in diterima

aaaabab \in diterima

aaabbaba \notin ditolak

Jawab :

i) $(q_0, abababaa) \Rightarrow (q_0, bababaa) \Rightarrow (q_1, ababaa) \Rightarrow$
 $(q_0, babaa) \Rightarrow (q_1, abaa) \Rightarrow (q_0, baa) \Rightarrow (q_1, aa)$

\Rightarrow

$(q_0, a) \Rightarrow q_0$

Tracing berakhir di q_0 (state AKHIR) \Rightarrow kalimat abababaa diterima

ii) $(q_0, aaaabab) \Rightarrow (q_0, aaabab) \Rightarrow (q_0, aabab) \Rightarrow$
 $(q_0, abab) \Rightarrow (q_0, bab) \Rightarrow (q_1, ab) \Rightarrow (q_0, b) \Rightarrow$

q_1

Tracing berakhir di q_1 (state AKHIR) \Rightarrow kalimat
aaaababa diterima

iii) $(q_0, aaabbaba) \Rightarrow (q_0, aabbaba) \Rightarrow (q_0, abbaba) \Rightarrow$
 $(q_0, bbaba) \Rightarrow (q_1, baba) \Rightarrow (q_2, aba) \Rightarrow (q_2, ba)$
 $\Rightarrow (q_2, a) \Rightarrow q_2$

Tracing berakhir di q_2 (bukan state AKHIR) \Rightarrow kalimat
aaabbaba ditolak

Kesimpulan :

sebuah kalimat diterima oleh DFA di atas jika tracingnya
berakhir di salah satu state AKHIR.

NFA :

Berikut ini sebuah contoh NFA (Q, Σ, S, F). dimana :

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ diberikan dalam tabel berikut :

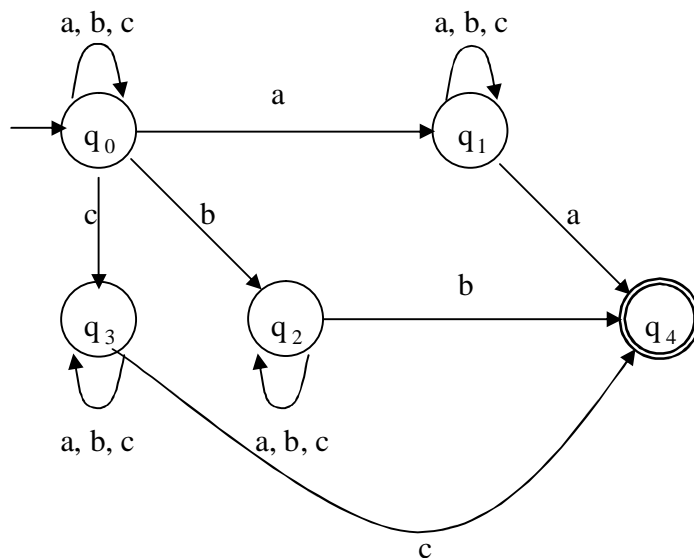
$\Sigma = \{a, b, c\}$

$S = q_0$

$F = \{q_4\}$

	a	b	c
q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_3\}$
q_1	$\{q_1, q_4\}$	$\{q_1\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2, q_4\}$	$\{q_2\}$
q_3	$\{q_3\}$	$\{q_3\}$	$\{q_3, q_4\}$
q_4	\emptyset	\emptyset	\emptyset

Ilustrasi graf untuk NFA adalah sebagai berikut :



kalimat yang diterima NFA di atas : aa, bb, cc, aaa, abb, bcc, cbb

kalimat yang tidak diterima NFA di atas : a, b, c, ab, ba, ac, bc

Sebuah kalimat di terima NFA jika :

- salah satu tracing-nya berakhir di state AKHIR, atau
- himpunan state setelah membaca string tersebut mengandung state AKHIR

Contoh :

Telusurilah, apakah kalimat-kalimat berikut diterima NFA di atas :
ab, abc, aabc, aabb

Jawab :

$$1. (q_0, ab) \Rightarrow (q_0, b) \cup (q_1, b) \Rightarrow \{q_0, q_2\} \cup \{q_1\} = \{q_0, q_1, q_2\}$$

Himpunan state TIDAK mengandung state AKHIR \Rightarrow kalimat
ab tidak diterima

$$2. (q_0, abc) \Rightarrow (q_0, bc) \cup (q_1, bc) \Rightarrow \{ (q_0, c) \cup (q_2, c) \} \cup (q_1, c)$$

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

Himpunan state TIDAK mengandung state AKHIR \Rightarrow kalimat
abc tidak diterima

$$3. (q_0, aabc) \Rightarrow (q_0, abc) \cup (q_1, abc) \Rightarrow \{ (q_0, bc) \cup (q_1, bc) \} \cup (q_1, bc) \Rightarrow \{ \{ (q_0, c) \cup (q_2, c) \} \cup (q_1, c) \} \cup (q_1, c) \Rightarrow \{ \{ \{ q_0, q_3 \} \cup \{ q_2 \} \} \cup \{ q_1 \} \} \cup \{ q_1 \} = \{ q_0, q_1, q_2, q_3 \}$$

Himpunan state TIDAK mengandung state AKHIR \Rightarrow kalimat
aabc tidak diterima

$$4. (q_0, aabb) \Rightarrow (q_0, abb) \cup (q_1, abb) \Rightarrow \{ (q_0, bb) \cup (q_1, bb) \} \cup (q_1, bb)$$

$$\Rightarrow \{ \{ (q_0, b) \cup (q_2, b) \} \cup (q_1, b) \} \cup (q_1, b)$$

$$\Rightarrow \{ \{ \{ q_0, q_2 \} \cup \{ q_2, q_4 \} \} \cup \{ q_1 \} \} \cup \{ q_1 \} = \{ q_0, q_1, q_2, q_4 \}$$

Himpunan state mengandung state AKHIR \Rightarrow kalimat aabb diterima