

# **BAB I**

## **TEORI BAHASA DAN AUTOMATA**

<b>TUJUAN PRAKTIKUM</b>
-------------------------

1. Memahami Tentang Teori Bahasa
2. Memahami Automata dan Istilah – Istilah yang terdapat dalam Automata
3. Mengerti Tentang Operasi String dan Sifat – sifat Operasi tersebut
4. Memahami Grammar dan Klasifikasi Chomsky
5. Memahami Cara Penderivasian Kalimat dan Penentuan Bahasa

<b>TEORI PENUNJANG</b>
------------------------

### **1.1 Pendahuluan**

#### **1.1.1 Teori Bahasa**

Teori Otomata dan bahasa formal, berkaitan dalam hal pembangkitan kalimat/generation yaitu, menghasilkan *semua* kalimat dalam bahasa L berdasarkan aturan yang dimilikinya. Dan pengenalan kalimat / recognition yaitu, menentukan suatu string (kalimat) termasuk sebagai salah satu anggota himpunan L.

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 manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya ‘bahasa formal’ akan disebut ‘bahasa’ saja.

### 1.1.2 Automata

Arti menurut American Heritage Dictionary:

1. a robot
2. one that behaves in an automatic or mechanical fashion

Arti dalam dunia matematika

Berkaitan dengan teori mesin abstrak, yaitu mesin sekuensial yang menerima input, dan mengeluarkan output, dalam bentuk diskrit.

Contoh :

- ♦ Mesin Jaja / vending machine
- ♦ Kunci kombinasi
- ♦ Parser/compiler

Jika disimpulkan maka pengertian automata adalah mesin abstrak yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu.

### 1.1.3 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  $\epsilon$  (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

### 1.1.4 Operasi Dasar String

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

- Prefik 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  $\epsilon$  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  $\epsilon$  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  $\epsilon$  adalah semua Postfix( $x$ )

- ProperPostfix (atau PoperSufix) 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  $\epsilon$  adalah semua ProperPostfix( $x$ )

- Head string  $w$  adalah simbol paling depan dari string  $w$ .

Contoh :  $a$  adalah Head( $x$ )

- Tail string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan simbol paling depan dari string  $w$  tersebut.

Contoh :  $bc$  adalah 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  $\epsilon$  adalah semua 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  $\epsilon$  adalah semua 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  $\epsilon$  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  $\epsilon$  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 :  $x^+ = x | xx | xxx | \dots = x | x^2 | x^3 | \dots$

### 1.1.5 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  $\epsilon$  :  $\epsilon x = x\epsilon = x$
- 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 :  $\varepsilon \mid x^+ = x^+ \mid \varepsilon = x^+$
  - ◆ Kesamaan ke-3 :  $(x \mid y)^* = \varepsilon \mid x \mid y \mid xx \mid yy \mid xy \mid yx \mid \dots =$  semua string yang merupakan concatenation dari nol atau lebih  $x$ ,  $y$ , atau keduanya.

## 1.2 Grammar dan Bahasa

### 1.2.1 Konsep Dasar

1. Dalam pembicaraan grammar, anggota alfabet dinamakan simbol terminal atau token.
2. Kalimat adalah deretan hingga simbol-simbol terminal.
3. Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa tak hingga kalimat.
4. Simbol-simbol berikut adalah simbol terminal :
  - huruf kecil awal alfabet, misalnya : a, b, c
  - simbol operator, misalnya : +, −, dan ×
  - simbol tanda baca, misalnya : (, ), dan ;
  - string yang tercetak tebal, misalnya : **if**, **then**, dan **else**.
5. Simbol-simbol berikut adalah simbol non terminal :
  - huruf besar awal alfabet, misalnya : A, B, C
  - huruf S sebagai simbol awal
  - string yang tercetak miring, misalnya : *expr* dan *stmt*.
6. Huruf besar akhir alfabet melambangkan simbol terminal atau non terminal, misalnya : X, Y, Z.
7. Huruf kecil akhir alfabet melambangkan string yang tersusun atas simbol-simbol terminal, misalnya : x, y, z.

8. Huruf Yunani melambangkan string yang tersusun atas simbol-simbol terminal atau simbol-simbol non terminal atau campuran keduanya, misalnya :  $\alpha$ ,  $\beta$ , dan  $\gamma$ .
9. Sebuah produksi dilambangkan sebagai  $\alpha \rightarrow \beta$ , artinya : dalam sebuah derivasi dapat dilakukan penggantian simbol  $\alpha$  dengan simbol  $\beta$ .
10. Simbol  $\alpha$  dalam produksi berbentuk  $\alpha \rightarrow \beta$  disebut ruas kiri produksi sedangkan simbol  $\beta$  disebut ruas kanan produksi.
11. Derivasi adalah proses pembentukan sebuah kalimat atau sentensial. Sebuah derivasi dilambangkan sebagai :  $\alpha \Rightarrow \beta$ .
12. Sentensial adalah string yang tersusun atas simbol-simbol terminal atau simbol-simbol non terminal atau campuran keduanya.
13. Kalimat adalah string yang tersusun atas simbol-simbol terminal. Jelaslah bahwa kalimat adalah kasus khusus dari sentensial.
14. Pengertian terminal berasal dari kata *terminate* (berakhir), maksudnya derivasi berakhir jika sentensial yang dihasilkan adalah sebuah kalimat (yang tersusun atas simbol-simbol terminal itu).
15. Pengertian non terminal berasal dari kata *not terminate* (belum/tidak berakhir), maksudnya derivasi belum/tidak berakhir jika sentensial yang dihasilkan mengandung simbol non terminal.

### 1.2.2 Grammar dan Klasifikasi Chomsky

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

- $V_T$  : himpunan simbol-simbol terminal (atau himpunan token -token, atau alfabet)
- $V_N$  : himpunan simbol-simbol non terminal
- $S \in V_N$  : simbol awal (atau simbol start)
- $Q$  : himpunan produksi

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  $\alpha \in V_N, \beta \in \{V_T, V_N V_T\}$

Mengingat ketentuan simbol-simbol (hal. 3 no. 4 dan 5), ciri-ciri RG sering dituliskan sebagai :  $\alpha \in V_N, \beta \in \{a, bC\}$  atau  $\alpha \in V_N, \beta \in \{a, Bc\}$

Atau disederhanakan seperti tabel dibawah ini:

Tabel 1.1 Tabel Grammar Chomsky

Kelas	Ruas kiri	Ruas Kanan	Contoh
Regular	$\alpha \in N$	$\leq 1$ non terminal (paling kiri/kanan)	$P \rightarrow aR$ $Q \rightarrow ab$ $R \rightarrow cc$
Context Free	$\alpha \in N$	-	$P \rightarrow aQb$ $Q \rightarrow abPRS$
Context Sensitive	$\alpha \in (T \cup N)^+$	$ \alpha  \leq  \beta $	$aD \rightarrow Da$ $AD \rightarrow aCD$
Unrestricted	$\alpha \in (T \cup N)^+$	-	$CB \rightarrow DB$ $ADc \rightarrow \epsilon$

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.

### 1.2.3 Mesin Pengenal Bahasa

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

Tabel 1.2 Tabel Kelas Bahasa dan Mesin Pengenal Bahasa

Kelas Bahasa	Mesin Pengenal Bahasa
<i>Regular Grammar</i> , RG	Automata Hingga ( <i>Finite Automata</i> ), FA
<i>Context Free Grammar</i> (CFG)	Automata Pushdown ( <i>Pushdown Automata</i> ), PDA
<i>Context Sensitive Grammar</i> (CSG)	<i>Linear Bounded Automaton</i> , LBA
<i>Unrestricted Grammar</i> (UG)	Mesin Turing ( <i>Turing Machine</i> ), TM

Catatan :

1. Pengenal bahasa adalah salah satu kemampuan mesin turing.
2. LBA adalah variasi dari Mesin Turing Nondeterministik.

### 1.2.4 Contoh Analisa Penentuan Tipe Grammar

1. Grammar  $G_1$  dengan  $Q_1 = \{S \rightarrow aB, B \rightarrow bB, B \rightarrow b\}$ . 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.
2. Grammar  $G_2$  dengan  $Q_2 = \{S \rightarrow Ba, B \rightarrow Bb, B \rightarrow b\}$ . 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. Grammar  $G_3$  dengan  $Q_3 = \{S \rightarrow Ba, B \rightarrow bB, B \rightarrow b\}$ . 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.

### 1.2.5 Derivasi Kalimat dan Penentuan Bahasa

Tentukan bahasa dari masing-masing grammar berikut :

1.  $G_1$  dengan  $Q_1 = \{1. S \rightarrow aAa, 2. A \rightarrow aAa, 3. A \rightarrow b\}$ .

Jawab :

Derivasi kalimat terpendek :

$$S \Rightarrow aAa \quad (1)$$

$$\Rightarrow aba \quad (3)$$

Derivasi kalimat umum :

$$S \Rightarrow aAa \quad (1)$$

$$\Rightarrow aaAaa \quad (2)$$

...

$$\Rightarrow a^n Aa^n \quad (2)$$

$$\Rightarrow a^n ba^n \quad (3)$$

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

2.  $G_2$  dengan  $Q_2 = \{1. S \rightarrow aS, 2. S \rightarrow aB, 3. B \rightarrow bC, 4. C \rightarrow aC, 5. C \rightarrow a\}$ .

Jawab :

Derivasi kalimat terpendek :

$$S \Rightarrow aB \quad (2)$$

$$\Rightarrow abC \quad (3)$$

$$\Rightarrow aba \quad (5)$$

Derivasi kalimat umum :

$$S \Rightarrow aS \quad (1)$$

...

$$\Rightarrow a^{n-1} S \quad (1)$$

$$\Rightarrow a^n B \quad (2)$$

$$\Rightarrow a^n bC \quad (3)$$

$$\Rightarrow a^n baC \quad (4)$$

...

$$\Rightarrow a^n ba^{m-1} C \quad (4)$$

$$\Rightarrow a^n ba^m \quad (5)$$

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

### 1.2.6 Menentukan Grammar Sebuah Bahasa

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

Jawab :

$$Q_1(L_1) = \{ S \rightarrow aS \mid a \}$$

2. Tentukan sebuah grammar bebas konteks untuk bahasa :

$L_2$  : himpunan bilangan bulat non negatif ganjil

Jawab :

Langkah kunci : digit terakhir bilangan harus ganjil.

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

$$Q_2(L_2) = \{ S \rightarrow J \mid GS \mid JS, G \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8, J \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9 \}$$