

## PERTEMUAN 3

### KLASIFIKASI GRAMMAR

#### A. TUJUAN PEMBELAJARAN

Pada pertemuan ini akan dijelaskan mengenai Klasifikasi Grammar. Setelah menyelesaikan materi pada pertemuan ini, mahasiswa mampu Menjelaskan jenis-jenis grammar.

#### B. URAIAN MATERI

##### 1. Jenis-jenis Grammar

###### a. 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:  $\alpha, \beta \in (V_T \mid V_N)^*$ ,  $|\alpha| \leq |\beta|$

###### 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  $\alpha \in V_N$ ,  $\beta \in \{a, Bc\}$

Tipe sebuah grammar ditentukan dengan aturan sebagai berikut:

Suatu bahasa dikatakan sebagai bahasa tipe-i ( $i=0, 1, 2, 3$ ) jika dapat ditentukan oleh tata bahasa tipe-i tetapi tidak dapat ditentukan tata bahasa tipe ( $i+1$ ).

Contoh Analisa Penentuan Type Grammar:

- 1) Grammar G dengan  $P = \{S \rightarrow aB, B \rightarrow bB, B \rightarrow b\}$ .

Ruas kiri semua produksinya terdiri dari sebuah V maka G kemungkinan tipe CFG atau RG. Selanjutnya karena semua ruas kanannya terdiri dari sebuah V atau string VV maka G adalah RG(3).

- 2) Grammar G dengan  $P = \{S \rightarrow Ba, B \rightarrow Bb, B \rightarrow b\}$ .

Ruas kiri semua produksinya terdiri dari sebuah V maka G kemungkinan tipe CFG atau RG. Selanjutnya karena semua ruas kanannya terdiri dari sebuah V atau string VV maka G adalah RG(3).

- 3) Grammar G dengan  $P = \{S \rightarrow Ba, B \rightarrow bB, B \rightarrow b\}$ .

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

- 4) Grammar G dengan  $P = \{S \rightarrow aAb, B \rightarrow aB\}$ .

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

- 5) Grammar G dengan  $P = \{S \rightarrow aA, S \rightarrow aB, aAb \rightarrow aBCb\}$ .

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

- 6) Grammar G dengan  $P = \{aS \rightarrow ab, SAc \rightarrow bc\}$ .

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

Analisa Penentuan Type Grammar yang lain

Contoh :

Tentukanlah type grammar G jika G memiliki  $Q = \{S \rightarrow aB, B \rightarrow bB, B \rightarrow b\}$ .

Jawab:

Ruas kiri: Semua produksinya terdiri dari sebuah VN. maka G kemungkinan tipe CFG atau RG.

Ruas kanan: karena semua produksinya terdiri dari sebuah VT atau string VTVN maka G adalah RG.

#### b. Derivasi Kalimat dan Penentuan Bahasa

Tentukan bahasa dari masing-masing gramat berikut :

- 1) G dengan  $P = \{1. S \rightarrow aAa, 2. A \rightarrow aAa, 3. A \rightarrow b\}$ .

Jawab :

Derivasi kalimat terpendek:

$$S \rightarrow aAa \text{ (1)}$$

$$S \rightarrow aba \text{ (3)}$$

Derivasi kalimat umum:

$$S \rightarrow aAa \text{ (1)}$$

$$S \rightarrow aaAaa \text{ (2)}$$

...

$$S \rightarrow a^n A a^n \text{ (2)}$$

$$S \rightarrow a^n b a^n \text{ (3)}$$

Dari pola kedua kalimat disimpulkan :  $L(G) = \{ a^n b a^n \mid n \geq 1 \}$

- 2) G dengan  $P = \{1. S \rightarrow aS, 2. S \rightarrow aB, 3. B \rightarrow bC, 4. C \rightarrow aC, 5. C \rightarrow a\}$ .

Jawab :

Derivasi kalimat terpendek:

$$S \rightarrow aB \text{ (2)}$$

$$S \rightarrow abC \text{ (3)}$$

$$S \rightarrow aba \text{ (5)}$$

Derivasi kalimat umum:

$$S \rightarrow aS \text{ (1)}$$

...

$$S \rightarrow a^{n-1} S \text{ (1)}$$

$$S \rightarrow a^n B \text{ (2)}$$

$$S \rightarrow a^n bC \text{ (3)}$$

$$S \rightarrow a^n baC \text{ (4)}$$

...

$$S \rightarrow a^n ba^{m-1} C \text{ (4)}$$

$$S \rightarrow a^n ba^m \text{ (5)}$$

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

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

Jawab :

Derivasi kalimat terpendek 1:

$$S \rightarrow abC \text{ (2)}$$

$$S \rightarrow abc \text{ (4)}$$

Derivasi kalimat terpendek 2:

$$S \rightarrow aSBC \text{ (1)}$$

$$\rightarrow aabCBC \text{ (2)}$$

$$\rightarrow aabBCC \text{ (5)}$$

$$\rightarrow aabbCC \text{ (3)}$$

$$\rightarrow aabbccC \text{ (4)}$$

→ aabbcc (6)

Deriviasi kalimat terpendek 3:

$S \rightarrow aSBC$  (1)

→ aaSBCB (1)

→ aaabCBCBC (2)

→ aaabBCCBC (5)

→ aaabBBCCC (5)

→ aaabbBCCC (3)

→ aaabbbCCC (3)

→ aaabbbccC (4)

→ aaabbbccC (6)

→ aaabbbccc (6)

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

### c. Menentukan Grammar Sebuah Bahasa

- 1) Tentukan sebuah grammar regular untuk bahasa  $L = \{ a^n \mid n \geq 1 \}$

Jawab :

$$P(L) = \{ S \rightarrow aS \mid a \}$$

- 2) Tentukan sebuah grammar bebas konteks untuk bahasa:

$L$ : himpunan bilangan bulat non negatif ganjil.

Jawab:

Langkah kunci: digit terakhir harus bilangan ganjil.

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

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

- 3) Tentukan sebuah grammar bebas konteks untuk bahasa:

L: 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.

Buatlah dua buah himpunan bilangan terpisah: huruf (H) dan angka (A).

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

- 4) Tentukan grammar bebas konteks untuk bahasa  $L(G) = \{a^n b^m \mid n, m \geq 1, n \neq m\}$

Jawab:

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

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

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

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

- 5) Tentukan sebuah grammar bebas konteks untuk bahasa:

L: 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).

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

## d. Mesin Pengenal Bahasa

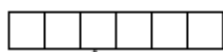
Untuk setiap kelas bahasa Chomsky, terdapat sebuah mesin pengenal bahasa. Masing-masing mesin tersebut adalah :

**Tabel 3.1** Mesin Pengenal Bahasa menurut Chomsky

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

## e. Mesin Turing

Ilustrasi TM sebagai sebuah 'mesin':



Pita TM. Terbatas di kiri. Setiap sel berisi sebuah karakter dari kalimat yang akan dikenali. Di kanan kalimat terdapat tak hingga simbol hampa.



Head : membaca dan menulisi sel pita TM, bisa bergerak ke kiri atau ke kanan Finite State

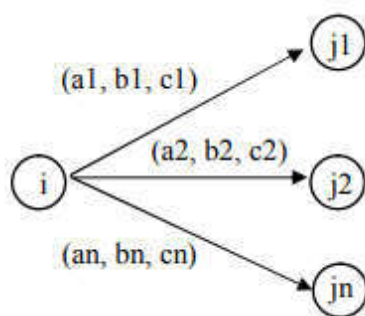


FSC : otak dari TM, diimplementasikan dari algoritma pengenalan Control (FSC) kalimat.

Ilustrasi TM sebagai sebuah graf berarah :

- 1) Sebagaimana graf, TM terdiri dari beberapa node dan beberapa edge. Dari satu node mungkin terdapat satu atau lebih edge yang menuju node lainnya atau dirinya sendiri.

- 2) Sebuah node menyatakan sebuah stata (state). Dua stata penting adalah stata awal S (start) dan stata penerima H (halt). Sesaat sebelum proses pengenalan sebuah kalimat, TM berada pada stata S. Jika kalimat tersebut dikenali maka, setelah selesai membaca kalimat tersebut, TM akan akan berhenti pada stata H.
- 3) Sebuah edge mempunyai 'bobot' yang dinotasikan sebagai triple : (a, b, d). a adalah karakter acuan bagi karakter dalam sel pita TM yang sedang dibaca head. Jika yang dibaca head adalah karakter a maka a akan di-overwrite dengan karakter b dan head akan berpindah satu sel ke arah d (kanan atau kiri).
- 4) Kondisi crash akan terjadi jika ditemui keadaan sebagai berikut :



TM sedang berada pada stata i. Jika TM sedang membaca simbol  $a_x \neq a_1 \neq a_2 \neq \dots \neq a_n$  maka TM tidak mungkin beranjak dari stata i. Jadi pada kasus ini penelusuran (tracing) TM berakhir pada stata i.

Contoh :

Rancanglah sebuah mesin turing pengenalan bahasa  $L = \{a^n b^n \mid n \geq 0\}$ .

Jawab :

L tersebut terdiri dari 2 kelompok kalimat yaitu  $\varepsilon$  dan non- $\varepsilon$ . Kelompok non- $\varepsilon$  adalah : ab, aabb, aaabbb, dan seterusnya. Untuk dapat menerima kalimat  $\varepsilon$  TM harus mempunyai edge dari S ke H dengan bobot ( $\varepsilon$ ,  $\varepsilon$ , R). TM menerima kalimat-kalimat : ab, aabb, aaabbb, dan seterusnya, dengan algoritma sebagai berikut :

- a) Mulai dari S, head membaca simbol a.
- b) Head membaca simbol a. Tandai simbol a yang sudah dibaca tersebut, head bergerak ke kanan mencari simbol b pasangannya.
- c) Head membaca simbol b. Tandai simbol b yang sudah dibaca tersebut, head bergerak ke kiri mencari simbol a baru yang belum dibaca/ditandai.



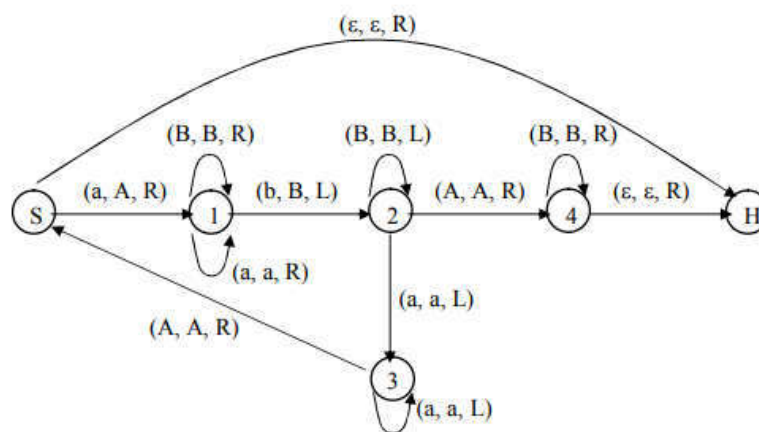
- d) Ulangi langkah 2 dan 3.
- e) Head sampai ke H hanya jika semua simbol a dan simbol b dalam kalimat  $a_n b_n$  selesai dibaca.

Algoritma di atas lebih diperinci lagi sebagai berikut :

- (1) Mulai dari S, head membaca simbol a.
- (2) Overwrite a tersebut dengan suatu simbol (misalkan A) untuk menandakan bahwa a tersebut sudah dibaca. Selanjutnya head harus bergerak ke kanan untuk mencari sebuah b sebagai pasangan a yang sudah dibaca tersebut.
  - (a) Jika yang ditemukan adalah simbol a maka a tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain a dioverwrite dengan a juga dan head bergerak ke kanan.
  - (b) Jika TM pernah membaca simbol b ada kemungkinan ditemukan simbol B. Simbol B tersebut harus dilewati (tidak boleh dioverwrite), artinya B diover-write dengan B juga dan head bergerak ke kanan.
  - (c) Head membaca simbol b, maka b tersebut harus dioverwrite dengan simbol lain (misalnya B) untuk menandakan bahwa b tersebut (sebagai pasangan dari a) telah dibaca, dan head bergerak ke kiri untuk mencari simbol A.
  - (d) Jika ditemukan B maka B tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain B dioverwrite dengan B juga dan head bergerak ke kiri.
  - (e) Jika ditemukan a maka a tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain a dioverwrite dengan a juga dan head bergerak ke kiri.
  - (f) Head membaca simbol A, maka A tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain A dioverwrite dengan A juga dan head bergerak ke kanan.
  - (g) Head membaca simbol a, ulangi langkah 2 dan 3.

- (h) (Setelah langkah 3) head membaca simbol A, maka A tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain A dioverwrite dengan A juga dan head bergerak ke kanan.
- (i) Head membaca simbol B, maka B tersebut harus dilewati (tidak boleh dioverwrite), dengan kata lain B dioverwrite dengan A juga dan head bergerak ke kanan.
- (j) Head membaca simbol  $\varepsilon$ , maka  $\varepsilon$  dioverwrite dengan  $\varepsilon$  dan head bergerak ke kanan menuju stata H.

Skema graf Mesin Turing di atas adalah :



**Gambar 3.1** Contoh Skema Graf Mesin Turing

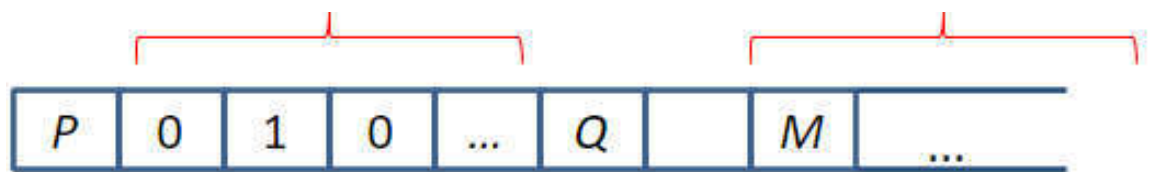
Contoh : Lakukan tracing dengan mesin turing di atas untuk kalimat-kalimat : aabb, aab. Jawab :

- i)  $(S, aabb) \rightarrow (1, Aabb) \rightarrow (1, Aabb) \rightarrow (2, AaBb) \rightarrow (3, AaBb) \rightarrow (S, AaBb) \rightarrow (1, AABb) \rightarrow (1, AABb) \rightarrow (2, AABBB) \rightarrow (2, AABBB) \rightarrow (4, AABBB) \rightarrow (4, AABBB\varepsilon) \rightarrow (H, AABBB\varepsilon\varepsilon)$
- ii)  $(S, aab) \rightarrow (1, Aab) \rightarrow (1, Aab) \rightarrow (2, AaB) \rightarrow (3, AaB) \rightarrow (S, AaB) \rightarrow (1, AAB) \rightarrow (1, AAB\varepsilon) \rightarrow \text{crash, karena dari node 1 tidak ada edge dengan bobot komponen pertamanya hampa } (\varepsilon)$

## f. Mesin Turing Penyalin Simbol

Pada contoh mesin Turing pengenalan  $0^n1^n$  terlihat bahwa operasi CARI X (pencarian simbol tertentu di dalam pita) merupakan operasi yang mendasar. Operasi penanganan simbol yang lainnya adalah penyalinan simbol.

Kita akan merancang mesin Turing yang dapat menyalin string biner yang terletak di antara simbol P dan Q ke suatu posisi sel baru yang ditandai dengan simbol M.



Mesin bekerja dari status awal  $q_1$  dengan head pada posisi simbol P. Algoritma penyalinan simbol adalah sebagai berikut:

- 1) Simbol yang akan disalin ditandai dengan simbol khusus
- 2) Gerakkan head ke kanan hingga dijumpai posisi penyalinan yang tepat.
- 3) Tuliskan simbol yang disalin pada posisi tersebut.
- 4) Gerakkan head ke kiri sampai dijumpai simbol khusus di atas.
- 5) Ganti simbol khusus dengan simbol semula.

Fungsi pergerakan Mesin Turing:

$\delta(q_1, P) = (q_2, P, R)$  Longkap simbol P

$\delta(q_2, 0) = (q_3, \alpha, R)$  Tandai 0

$\delta(q_3, *) = (q_3, *, R)$  Longkap ke kanan simbol-simbol selain M

$\delta(q_3, M) = (q_4, 0, R)$  Ganti 'M' dengan simbol 0 yang sedang disalin

$\delta(q_4, *) = (q_5, M, L)$  Tulis 'M' pada posisi baru

$\delta(q_5, *) = (q_5, *, L)$  Longkap ke kiri simbol selain  $\alpha$

$\delta(q_5, \alpha) = (q_2, 0, R)$  kembalikan  $\alpha$  menjadi 0

$\delta(q_2, 1) = (q_6, \beta, R)$  Tandai 1

$\delta(q_6, *) = (q_6, *, R)$  Longkap ke kanan simbol selain M

$\delta(q_6, M) = (q_7, 1, R)$  Ganti M dengan simbol 1 yang sedang disalin

$\delta(q_7, *) = (q_8, M, L)$  Tulis M pada posisi baru

$\delta(q_8, *) = (q_8, *, L)$  Longkap ke kiri simbol selain  $\beta$

$\delta(q_8, \beta) = (q_2, 1, R)$  Kembalikan  $\beta$  menjadi 1

$\delta(q_2, Q) = (q_9, Q, R)$  Q menandakan akhir data

### C. SOAL LATIHAN/TUGAS

1. Jelaskan jenis-jenis grammar dengan bahasa Anda sendiri!
2. Berikan contoh jenis bahasa mesin!
3. Buatlah sebuah string menggunakan penyelesaian Parsing dengan Derivasi.
4. Diketahui  $G = (\{A, B, S\}, \{a, b\}, S, P)$  dengan aturan produksi :

$S \rightarrow AB$

$A \rightarrow a \mid aA$

$B \rightarrow b \mid bB$

Untuk menghasilkan string 'aaaabbb'

### D. REFERENSI

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

## GLOSARIUM

***String*** dalam pemrograman komputer adalah sebuah deret simbol. Tipe data string adalah tipe data yang digunakan untuk menyimpan barisan karakter.

***Overwrite*** adalah istilah sebagai tindakan menimpa, mengganti data yang sudah ada sebelumnya.

***identifier*** adalah suatu nama yang digunakan untuk menyatakan variabel, konstanta, class, method.