

PERTEMUAN 3 :

TATA BAHASA (GRAMMAR) HIRARKI CHOMSKY

A. TUJUAN PEMBELAJARAN

Dalam pertemuan 3 ini akan dibahas mengenai definisi yang penting dari istilah – istilah yang melengkapi teori otomata meliputi konsep “alfabet”, “string”, dan “bahasa” serta tata bahasa (Grammar) yang ada dalam hierarki Chomsky serta mengenal apa perbedaan dalam hierarki tersebut hingga memahami CFG (Context Free Grammar). Setelah mengikuti perkuliahan mahasiswa diharapkan mampu :

- 1) Menjabarkan Struktur Tata Bahasa
- 2) Menganalisa Klasifikasi Grammar Noam Chomsky berdasarkan pembatasan produksi

B. URAIAN MATERI

1. Struktur Tata Bahasa

a. Alfabet

Sebuah alfabet (alfabet Dapat kita sebut juga sebagai simbol) adalah berhingga, himpunan simbol yang tidak kosong. Biasanya dan seterusnya, akan menyebut alfabet dengan simbol, notasi akan digunakan sebagai lambang simbol. Simbol yang umum akan digunakan di antaranya :

- (1) $= \{0,1\}$, himpunan simbol bilangan biner
- (2) $= \{a,b,c,...,z\}$ himpunan semua simbol huruf kecil
- (3) Himpunan seluruh karakter ASCII, atau himpunan seluruh karakter ASCII yang dapat dicetak
- (4) Simbol operator, misalnya : +, -, dan x

b. String

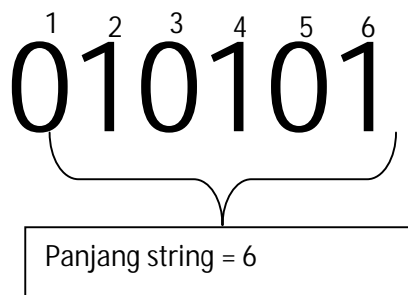
String dalam pemrograman komputer adalah sebuah deret simbol. Bedakan dengan tipe data string adalah tipe data yang digunakan untuk menyimpan barisan karakter, karena seterusnya akan kita gunakan istilah string dalam buku ini sebagai definisi deret simbol berhingga. String (String dapat kita sebut juga dengan kata jika untai string adalah kalimat, sedangkan string merupakan huruf atau simbol jika untai

string nya adalah kata) merupakan urutan simbol terbatas atau berhingga yang diambil dari beberapa simbol atau lebih dari satu simbol.

Contoh : 010101 adalah sebuah string dari simbol bilangan biner $= \{0,1\}$. String 11011 juga merupakan contoh string lainnya dari himpunan bilangan biner tersebut.

String Kosong adalah sebuah string dengan simbol yang tanpa isi atau kosong. Sting nya dilambangkan dengan ϵ , adalah sebuah string yang berasal dari sembarang simbol.

Panjang sebuah string adalah perlu diklasifikasikan untuk beberapa kondisi, yakni nomor posisi simbol dalam string. Misalnya, 010101 memiliki panjang string 6.



Angka 6 diperoleh dengan menghitung jumlah digit biner dalam string, analogi nya jika string adalah kata “bahasa” maka panjang string adalah jumlah huruf dalam kata “bahasa” yakni 6 huruf.

Umumnya pada string tersebut dapat dikatakan bahwa panjang string adalah jumlah dari simbol, pernyataan tersebut dapat diterima tapi tidak tepat betul. Melainkan hanya ada dua simbol yakni 0 dan 1 pada string 010101, tetapi ada enam (6) posisi simbol dan panjangnya adalah 6. Akan tetapi tidak jarang akan menggunakan secara general istilah jumlah dari sebuah string sebagai nomor posisi string jika memungkinkan. Notasi standar dari panjang suatu string ω adalah $|\omega|$. Contoh $|0111| = 4$ dan $|\epsilon| = 0$.

String akan biasa digunakan dalam perkuliahan ini sehingga perlu pemahaman simbol dan string mana sajakah yang akan bersifat umum

Pangkat atau notasi eksponensial dari Simbol akan digunakan sebagai lambang himpunan panjang simbol dalam sebuah string. Jika a adalah sebuah simbol, maka a^k menjadi himpunan dari panjang string ke k yang simbolnya ada dalam string a . Contoh Diketahui sebuah $\{a\} = \{\epsilon\}$, berdasarkan dari simbol yang ada di a . Maka, ϵ adalah simbol yang panjang nya 0.

Jika $a = \{0,1\}$,

kemudian

$$a^0 = \{\epsilon\};$$

$$a^1 = \{00, 11, 10, 11\};$$

$$a^2 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

Dan seterusnya,.

Jika dilihat contoh ada pernyataan yang kelihatan membingungkan antara a^0 dan a^1 , dimana anggota dari kedua string tersebut sama yakni 0 dan 1, maka untuk kesamaan untuk setiap anggota hanya 0 dan 1 sebaiknya dipisahkan notasi berdasarkan konteks nya atau maksud dari setiap string.

Himpunan seluruh simbol dari contoh string dari a diatas akan menjadi a^* . Sebagai contoh $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ atau sama dengan cara ini :

$$a^* = \epsilon \cup a \cup a^2 \cup \dots$$

Atau terkadang akan dibutuhkan himpunan string tanpa string kosong, maka untuk himpunan string a tanpa string kosong dapat dilambangkan a^+ .

Akan menjadi :

$$a^+ = a \cup a^2 \cup \dots$$

Ekuivalen dengan

$$= \cup \{\epsilon\}.$$

String dengan Konkatenasi, adalah jika ada string x dan y kemudian xy dinyatakan sebagai konkatenasi dari x dan y . Adalah kondisi suatu string x yang diikuti dengan string lainnya y .

Contoh $x = 0011$ dan $y = 101$, maka $xy = 0011101$

Artinya pernyataan contoh di atas sebuah simbol x dan y mempunyai himpunan simbol tertentu, menjadi konkatenasi xy yakni seluruh anggota pada x diikuti dengan anggota pada y .

c. Bahasa

Bahasa adalah suatu himpunan dari seluruh string yang berasal dari , dimana berisi simbol tertentu. Menurut Noam Chomsky dalam jurnalnya , Information dan Control, vol 2 1959, “Bahasa adalah koleksi dari kalimat yang panjangnya berhingga dan dibangun dari simbol-simbol yang berhingga”.

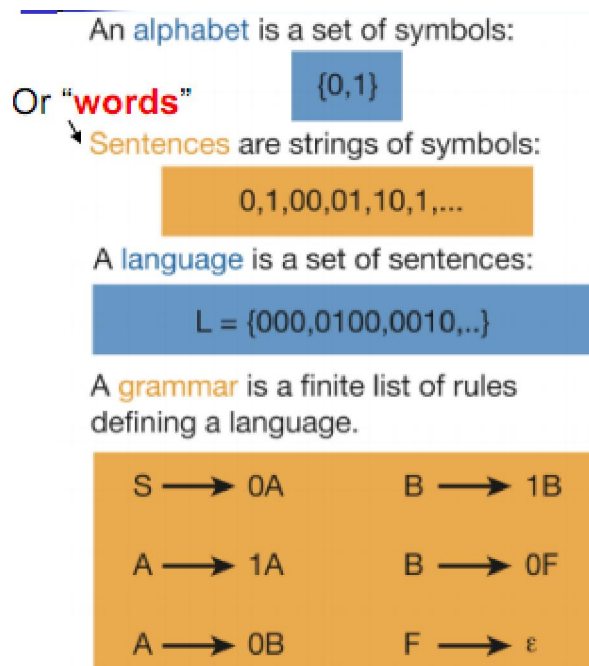


Image source: Nowak et al. Nature, vol 417, 2002

Gambar 2.1
Definisi Simbol, Bahasa dan Tata Bahasa

Jika sebuah s adalah sebuah string dan $L \subseteq \Sigma^*$

Maka L adalah sebuah bahasa dari sekumpulan string pada Σ^*

Dan setiap bahasa L akan mengandung semua simbol yang terdapat dalam Σ .

Contoh lainnya pada bahasa pemrograman misal nya bahasa Java, setiap program diterima jika setiap string mengandung simbol dari bahasa Java. Bahasa pemrograman harus didefinisikan secara tepat, awal mulanya bahasa pemrograman memiliki kompilator khusus yang mampu mendefinisikan bahasa tersebut. Suatu bahasa pemrograman yang baik seharusnya memiliki spesifikasi sebagai berikut :

- (1).Himpunan Simbol, yakni karakter ASCII, yang sudah didesain sedemikian rupa sehingga dapat digunakan untuk menyusun program yang benar.
- (2).Himpunan semua program yang diterima secara sintaks
- (3).Arti dari semua program yang diterima sesuai sintaks

Bagaimana menyajikan sebuah bahasa? Bahasa merupakan himpunan hingga ataupun tak hingga dari kalimat. Bahasa hingga dapat dibangun dengan menyebutkan anggota nya satu persatu, sedangkan untuk bahasa tak hingga tidak dapat disebutkan anggota nya satu per satu seperti bahasa hingga. Oleh karena itu, selalu bahasa dinyatakan dalam bentuk hingga. Grammar merupakan salah satu cara untuk membangun sebuah bahasa.

Grammar terdiri dari himpunan hingga yang tak hampa dari aturan atau produksi, yang mengklasifikasikan sintaks dari bahasa. Artinya sebuah Grammar menentukan bahasa yang akan terbentuk. Studi tentang Grammar ini subbidang yang penting dalam ilmu komputer. Pada tahun 1950 bidang ini sangat ditekuni oleh Noam Chomsky yang memberikan model matematika untuk Grammar. Pada tahun 1960 konsep Grammar menjadi penting sekali bidang pemrograman, karena sintaks dari bahasa pemrograman ALGOL 60 telah menggunakan konsep Grammar formal ini.

Jadi, Grammar sebagai suatu sistem matematika untuk mendefinisikan bahasa dan alat untuk membentuk suatu struktur dari kalimat bahasa, yang dikenal struktur gramatik atau sintaks kalimat. Struktur gramatik berguna untuk menganalisa berbagai bagian kalimat dan relasi antara satu bagian kalimat dengan kalimat lainnya. Seorang pemrogram komputer memiliki tugas untuk menghasilkan program sesuai dengan aturan produksi suatu bahasa, sedangkan kompilator dari bahasa pemrograman harus mampu menghadapi masalah penentuan kalimat¹ mana yang secara sintaks benar berdasarkan aturan gramatikal yang diberikan, jika sintaks sudah benar maka kompilator menghasilkan kode objek yang selanjutnya dibangun.

Berikut ini contoh produksi rekursif yang menghasilkan himpunan tak hingga nama, karena suatu kelas sintaks I (Identifier) muncul baik pada ruas kiri maupun ruas kanan dari produksi. Jika simbol L^2 , D^3 dan I^4 , maka :

I	à	L
I	à	ID
I	à	IL
I	à	a
I	à	b
	
I	à	z
D	à	0
D	à	1
	
D	à	9

¹ Kalimat berasal dari pemrogram komputer

² Sebagai Huruf

³ Sebagai Digit

⁴ Sebagai Identifier

Tabel 2.1 Contoh produksi rekursif

Hasil dari produksi di atas akan membentuk masing – masing untai string yang terdiri dari huruf atau sebuah huruf diikuti satu huruf atau digit. Himpunan ini terjadi sebagai akibat dari penggunaan secara rekursif⁵ terhadap produksi $I \Rightarrow ID$, dan $I \Rightarrow IL$. Oleh karena itu, dasar definisi bahasa tak hingga menggunakan Grammar dengan proses yang rekursif.

Untuk pendefinisian selanjutnya akan dibutuhkan simbol untuk membentuk Grammar, yakni

1. Simbol sebagai himpunan simbol hingga yang tak hampa, anggota nya disebut simbol terminal. Anggotanya :
 - a) huruf besar, misalnya : A, B, C
 - b) huruf S sebagai simbol awal
2. Himpunan simbol non terminal dinyatakan dengan digunakan untuk mendefinisikan sintaks atau struktur bahasa. Anggotanya :
 - a) huruf kecil, misalnya : a, b, c, 0, 1, ..
 - b) simbol operator, misalnya : +, -, dan x
 - c) simbol tanda baca, misalnya : (,), dan ;

Himpunan huruf yunani sebagai lambang string yang tersusun hasil kombinasi dan . Anggota nya : α , β , γ dan sebagainya.

d. Tata Bahasa (Grammar)
Definisi Grammar 1 :

Grammar memiliki 4 tupel : $G = (\quad , \quad , S, Q)$, dengan definisi masing – masing tupel sebagai berikut :

- adalah himpunan simbol non terminal
- adalah himpunan simbol terminal
- S merupakan elemen tertentu yang diambil dari dan disebut sebagai lambang Start

⁵ Pengulangan penggunaan produksi yang ada

- Q merupakan sub himpunan berhingga yang tidak kosong dari relasi $(\alpha \rightarrow \beta)$ atau secara umum sebuah elemen (α, β) dari Q , maka dapat ditulis sebagai :
 $\alpha \rightarrow \beta$ adalah sebuah produksi⁶

Contoh sebuah Grammar untuk mendefinisikan sebuah Identifier :

$$G = \{ \Sigma, S, Q \}$$

Untuk

$$\Sigma = \{I, L, D\}$$

$$\Sigma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$S = I, \text{ dan}$$

$$Q = \{ I \rightarrow L, I \rightarrow IL, I \rightarrow ID, L \rightarrow a, L \rightarrow b, \dots, L \rightarrow z, D \rightarrow 0, D \rightarrow 1, \dots, D \rightarrow 9 \}$$

Contoh bahasa yang dapat dibentuk dari Grammar tersebut sebagai berikut :

Berdasarkan produksi Q , Start adalah I , maka :

$$I \Rightarrow IL$$

$$\Rightarrow ILL$$

$$\Rightarrow ILLL$$

$$\Rightarrow LLLL$$

$$\Rightarrow bLLL$$

$$\Rightarrow baLL$$

$$\Rightarrow bacL$$

$$\Rightarrow baca$$

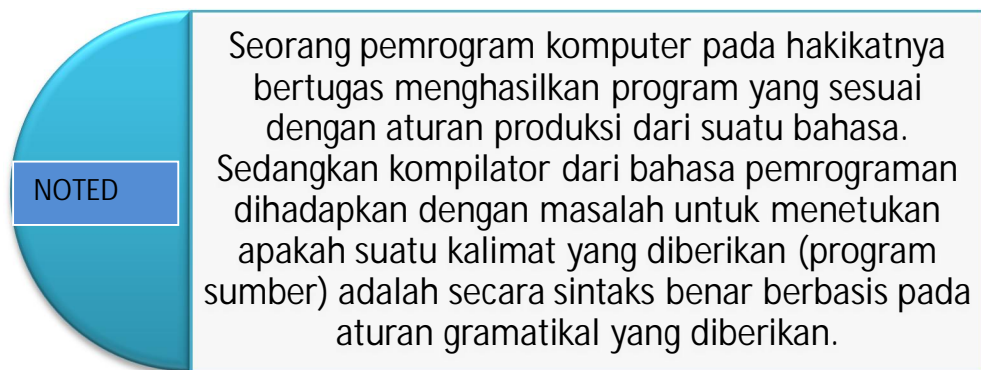
⁶ Dikenal juga dengan istilah rewriting rule

Definisi lainnya dari suatu bentuk sentensial adalah untai yang dihasilkan melalui derivasi yang berawal dari simbol non terminal S. Bahasa L dapat dibentuk dari Grammar G adalah himpunan sentensial yang semua simbolnya adalah simbol terminal.

Atau dinotasikan sebagai berikut :

$$L(G) = \{ w \mid S \xRightarrow{*} w, w \text{ anggota } \Sigma^* \}$$

Oleh karena itu, suatu bahasa adalah himpunan bagian dari himpunan semua untai terminal (kalimat) Σ^* .



Gambar 2.2 Hakikat Seorang Pemrogram

2. Klasifikasi Grammar Noam Chomsky

Berdasarkan teori dari Chomsky, Grammar diklasifikasikan menjadi 4 (empat) berdasarkan pembatasan pada produksi yakni, (1) Kelompok pertama adalah grammar yang mempunyai aturan produksi yang tidak terbatas, (2) Kelompok kedua, yang disebut Grammar Context – Sensitif, (3) Kelompok ketiga, yang disebut Grammar Context – Free, (4) Kelompok keempat, yang disebut Grammar Context – Regular.

Berikut ini akan dibahas untuk kelompok kedua, ketiga dan keempat, untuk kelompok pertama tidak dibahas

- (1) Kelompok Grammar Context – Sensitif
terdiri dari produksi dengan bentuk

$$\alpha \rightarrow \beta, \text{ dengan } |\alpha| \leq |\beta|$$

⁷ Relasi $\xRightarrow{*}$ adalah penutup transitif (transitif closure) dari relasi derivasi \Rightarrow

$|\alpha|$ menunjukkan panjang α dan $|\beta|$ menunjukkan panjang β . Artinya panjang dari suatu β selalu lebih atau sama dengan dari α , sehingga elemen β selalu tidak hampa atau kosong.

Jika suatu $\alpha = A$ dan $\beta = \gamma$ (dengan kemungkinan maupun ataupun keduanya hampa) nilai γ tidak boleh hampa. Dengan demikian setiap kondisi berikut :

$$A \rightarrow \gamma$$

Artinya adalah setiap non terminal A digantikan dengan γ diantara Q_1 dan Q_2 . Grammar Context – Sensitif membentuk Bahasa Context – Sensitif . Berikut contoh bahasa Context – Sensitif :

Bahasa atau Language $L(G_1) = \{ a^n b^n c^n \mid n \geq 1 \}$, yang dibentuk oleh Grammar $G = (\{S,B,C\}, \{a,b,c\}, S, Q)^8$. Dengan himpunan produksi Q sebagai berikut :

1. $S \rightarrow aSBC$
2. $S \rightarrow abC$
3. $bB \rightarrow bb$
4. $BC \rightarrow bc$
5. $CB \rightarrow BC$
6. $cC \rightarrow cc$

Contoh diinginkan sebuah untai terminal $aaabbccbc$ tentukan proses derivasinya. Maka dapat dilakukan proses derivasi seperti pada pembahasan bab 1 sebelumnya. Berikut proses dan langkah – langkah nya :

⁸ Dibaca kiri ke kanan adalah $\{S,B,C\}$ termasuk himpunan non terminal , $\{a,b,c\}$ adalah himpunan terminal , S adalah start atau awal , dan Q adalah himpunan produksi

S	====>	aSBC	produksi 1
	====>	aaSBCBC	produksi 1
	====>	aaabCBCBC	produksi 2
	====>	aaabBCCBC	produksi 5
	====>	aaabbcCBC	produksi 4
	====>	aaabbccBC	produksi 6
	====>	aaabbccbc	produksi 4

Tabel 2.2 Proses derivasi suatu Grammar Context Sensitif

Langkah – langkah derivasi untuk menghasilkan untai terminal aaabbccbc adalah sebagai berikut :

1. Derivasi pertama⁹ melalui produksi 1.
2. Pada derivasi kedua menggunakan produksi 1 untuk mengubah string S sehingga menjadi aaSBCBC.
3. Selanjutnya melalui produksi 2, derivasi ketiga mengubah string S sehingga menjadi aaabCBCBC.
4. Derivasi keempat menghasilkan string BC dari string CB melalui produksi 5, untai string menjadi aaabBCCBC.
5. Pada derivasi kelima melalui produksi 4, mengubah string BC menjadi bc sehingga untai string menjadi aaabbcCBC
6. Derivasi keenam melalui produksi 6 merubah string cC menjadi string cc dengan untai string yang terbentuk aaabbccBC
7. Kemudian derivasi terakhir membentuk string bc dari string BC melalui produksi 4, sehingga untai string yang terbentuk adalah aaabbccbc sesuai dengan untai string yang diinginkan.

⁹ produksi pertama dengan start adalah S, maka dapat dipilih antara produksi 1 atau produksi 2 dimana kedua produksi ini memiliki S pada posisi sebelah kiri (left most)

Pada Grammar selanjutnya akan ditambahkan batasan terhadap produksi untuk menghasilkan Grammar Context – Free.

(2) Kelompok Grammar Context – Free terdiri dari produksi berikut :

1. $\alpha \rightarrow \beta$, dengan $|\alpha| \leq |\beta|$ dan
2. Anggota α berasal dari anggota β (simbol non terminal).

Artinya pada Grammar Context – Free seluruh produksinya dibatasi dengan kondisi pada ruas kiri hanya terdiri dari satu (1) simbol non terminal. Pada pernyataan ekspresi dalam sederet statement dari sebuah program tidak mampu ditampung menggunakan Grammar context free ini, karena tidak mempunyai kemampuan untuk menentukan bahwa sebuah variabel tertentu sudah dideklarasikan. Akan tetapi, Grammar ini sudah dapat merumuskan sebagian besar sintaks dalam bahasa pemrograman.

Grammar context free merupakan pembentuk Bahasa context free. Dapat dilihat contoh Grammar ini pada pembahasan produksi rekursif sebelumnya. Berikut ini contoh Grammar sebagai pembentuk Bahasa context free :

$L(G_2) = \{ a^n c a^n \mid n \geq 1 \}$ berasal dari Grammar $G_2 = (\{S, C\}, \{a, b\}, S, Q)^{10}$, dengan produksi sebagai berikut :

1. $S \rightarrow aCa$
2. $C \rightarrow aCa$
3. $C \rightarrow b$

Jika diinginkan proses derivasi untuk menghasilkan untai terminal $aaabaaa$ atau $aaabaaa$, maka berikut ini derivasi dan langkah – langkah nya :

S	\Rightarrow	aCa	produksi 1
	\Rightarrow	aaCaa	produksi 2
	\Rightarrow	aaaCaaa	produksi 2
	\Rightarrow	aaabaaa	produksi 3

Tabel 2.3 Proses Derivasi dari Grammar Context Free

¹⁰ Dibaca kiri ke kanan adalah $\{S, C\}$ termasuk himpunan non terminal, $\{a, b\}$ adalah himpunan terminal, S adalah start atau awal, dan Q adalah himpunan produksi

Langkah – langkah detail proses derivasi adalah sebagai berikut :

1. Derivasi pertama melalui produksi start yakni produksi 1 yang mengandung S.
2. Pada derivasi kedua melalui produksi 2 mengubah string C menjadi aCa, sehingga untai string menjadi aaCaa
3. Selanjutnya derivasi ketiga string C diubah menjadi aCa, melalui produksi 2 sehingga menjadi untai string aaaCaaa
4. Derivasi keempat melalui produksi 3 mengubah string C menjadi string b, sehingga terbentuk untai terminal aaabaaa

(3) Kelompok Grammar Regular adalah tipe klasifikasi terakhir dari hierarki Chomsky. Terdiri dari pembatasan produksi seperti berikut :

1. $\alpha \rightarrow \beta$, dengan $|\alpha| \leq |\beta|$,
2. α anggota dari Σ^* , dan
3. β merupakan string hanya boleh bentuk¹¹ aB atau a saja, dengan a anggota Σ dan B anggota Σ^*

Melalui Grammar Regular ini akan dapat dapat dibentuk Bahasa Regular dengan contoh bahasa yang akan dapat dibentuk berikut ini :

$L(G_3) = \{ a^n b a^m \mid n, m \geq 1 \}$ berasal dari Grammar :

$G_3 = (\{S, A, B, C\}, \{a, b\}, S, Q)$

Untuk Produksi sebagai berikut :

1. $S \rightarrow aS$
2. $S \rightarrow aB$
3. $B \rightarrow bC$
4. $C \rightarrow aC$
5. $C \rightarrow a$

Jika diinginkan untai terminal $a^n b a^m$ atau aaabaa, maka proses derivasi sebagai berikut :

S	\Rightarrow	aS	produksi 1
	\Rightarrow	aaS	produksi 1
	\Rightarrow	aaaS	produksi 1

¹¹ Artinya ruas kanan hanya boleh Σ^* atau Σ saja.

	\Rightarrow	aaaaB	produksi 2
	\Rightarrow	aaaabC	produksi 3
	\Rightarrow	aaaabaC	produksi 4
	\Rightarrow	aaaabaaC	produksi 4
	\Rightarrow	aaaabaaaC	produksi 4
	\Rightarrow	aaaabaaaa	produksi 5

Tabel 2.4 Proses Derivasi Grammar Regular

Langkah – langkah derivasi :

1. Proses Derivasi pertama sampai ketiga melalui produksi 1 dengan mengubah string S menjadi aS, terbentuk untai string aaaS
2. Pada derivasi keempat melalui produksi 2 mengubah string S menjadi aB, untai string menjadi aaaaB
3. Derivasi kelima mengubah string B menjadi bC melalui produksi 3, sehingga untai string yang terbentuk aaaabC
4. Pada derivasi keenam sampai dengan ketujuh melalui produksi 4, mengubah string C menjadi aC, sehingga membentuk untai string aaaabaaaC
5. Derivasi kedelapan mengubah C menjadi a melalui produksi 5, sehingga menjadi untai terminal aaaabaaaa atau

Berdasarkan keempat klasifikasi Grammar tersebut yakni kelompok dengan produksi yang tidak terbatas, kelompok Context – sensitive, Context – Free dan Regular, berturut – turut definisikan sebagai Grammar T_0 , T_1 , T_2 , dan T_3 . Maka Bahasa yang dapat dibentuk adalah Language $L(T_i)$ dengan Grammar T_i , dapat menjadi $L(T_3)$ himpunan bagian dari $L(T_2)$, kemudian $L(T_2)$ himpunan bagian dari $L(T_1)$ dan $L(T_1)$ himpunan bagian dari $L(T_0)$. Inilah yang mengapa disebut sebagai hierarki Chomsky.

¹² Sentensial form merupakan suatu proses derivasi dimulai dari simbol start menghasilkan suatu produksi string yang memiliki aturan spesial.

dipakai untuk membentuk tepat pada bagian kanan. Bahasa yang akan dihasilkan adalah L2, maka L1 tidak regular dan L(T3) subset L(T2).

C. SOAL LATIHAN 2 DAN TUGAS

- 1) Terdapat sebuah Grammar dengan ketentuan untuk simbol terminal $\{a,b\}$ dan produksi sebagai berikut :

$S \rightarrow a$

$S \rightarrow Sa$

$S \rightarrow b$

$S \rightarrow bS$

Jelaskan bagaimana bentuk umum untai terminal yang akan dapat dibentuk oleh Grammar tersebut.

- 2) Diketahui suatu Produksi dengan Grammar Context Free berikut

$$S \rightarrow aSa \mid bSb \mid aAb \mid bAa \quad A \rightarrow aAa \mid bAb \mid a \mid b \mid \Lambda$$

Catatan : simbol ini adalah sebagai token atau string hampa / kosong/ terminal.

Tentukan proses derivasi untuk menghasilkan untai terminal :

- a. aaba
 - b. bbaabbb
 - c. aabab
 - d. bbababa
 - e. abaaba
- 3) Tentukan untai terminal apa saja yang dapat terbentuk berdasarkan produksi di atas, sertakan proses derivasi nya dan minimal 5 untai terminal

$$\begin{aligned} S &\rightarrow AB \mid ABC \\ A &\rightarrow BA \mid BC \mid \Lambda \mid a \\ B &\rightarrow AC \mid CB \mid \Lambda \mid b \\ C &\rightarrow BC \mid AB \mid A \mid c \end{aligned}$$

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