



# TEKNIK KOMPILASI

**PROGRAM STUDI : TEKNIK INFORMATIKA**

**UNIVERSITAS INDRAPRASTA PGRI**

JL. Nangka No. 58 C Tanjung Barat Jagakarsa, Jakarta Selatan  
JL. Raya Tengah, Kelurahan Gedong, Pasar Rebo, Jakarta Timur

Website : <http://www.unindra.ac.id>

## Context Free Grammar (CFG)

**Context Free Grammar / Bahasa Bebas Konteks** adalah sebuah tata bahasa dimana tidak terdapat pembatasan pada hasil produksinya, Contoh Pada aturan produksi :mempunyai tujuan sama seperti halnya tata bahasa regular yaitu merupakan suatu cara untuk menunjukkan bagaimana menghasilkan suatu untai-untai dalam sebuah bahasa.

$$\alpha \rightarrow \beta$$

batasannya hanyalah ruas kiri ( $\alpha$ ) adalah sebuah simbol variabel. Sedangkan contoh aturan produksi yang termasuk CFG adalah seperti di bawah :

$$B \rightarrow CDeFg$$

$$D \rightarrow BcDe$$

[Context Free Grammar \( CFG \)](#) adalah tata bahasa yang mempunyai tujuan sama seperti halnya tata bahasa regular yaitu merupakan suatu cara untuk menunjukkan bagaimana menghasilkan suatu untai-untai dalam sebuah bahasa.

# PARSING

- Parsing bertugas untuk menentukan sintaks atau struktur dari sebuah program, karena alasan ini juga disebut analisis sintaks. Sintaks dari Bahasa pemrograman biasanya diberikan berdasarkan produksi
- Parsing dapat dilakukan dengan cara :
- Penurunan terkiri (leftmost derivation ): simbol variable yang paling kiri diturunkan (tuntas) dahulu
- Penurunan terkanan (rightmost derivation ): variable yang paling kanan diturunkan (tuntas) dahulu

Contoh : CFG

$S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid ba$

String: aabbaa

lakukan penurunan terkiri dan terkanan

- Leftmost Derivation

$S \Rightarrow aAS$

$S \Rightarrow aSbAS$

$S \Rightarrow aabAS$

$S \Rightarrow aabbaS$

$S \Rightarrow aabbaa$

- Rightmost Derivation

$S \Rightarrow aAS$

$S \Rightarrow aAa$

$S \Rightarrow aSbAa$

$S \Rightarrow aSbbaa$

$S \Rightarrow aabbaa$

## DERIVATION

Pada CFG sebuah string dapat diterima, apabila dapat dibentuk proses derivasi.

Derivasi adalah urutan pergantian nama struktur dengan pilihan sisi kanan dari produksi. Sebuah derivasi dimulai dengan nama struktur tunggal dan berakhir dengan sebuah string dari token symbol.

Contoh:

Diketahui sebuah grammar berikut :

$Exp \rightarrow (exp) \mid exp \ op \ exp \mid number$

$Op \rightarrow + \mid - \mid *$

$Number \rightarrow digit \mid digit *$

$Digit \rightarrow 0 \mid 1 \mid \dots \mid 9$

Apakah string  $(34-3) * 42$  diterima atau ditolak ?

Exp  $\rightarrow$  exp op exp  
 $\rightarrow$  exp op number  
 $\rightarrow$  exp op digit\*  
 $\rightarrow$  exp op **42**  
 $\rightarrow$  exp\* **42**  
 $\rightarrow$  (exp) \* **42**  
 $\rightarrow$  (exp op exp) \* **42**  
 $\rightarrow$  (exp op number)\* **42**  
 $\rightarrow$  (exp op digit) \* **42**  
 $\rightarrow$  (exp op **3**) \* **42**  
 $\rightarrow$  (exp - **3**) \* **42**  
 $\rightarrow$  (number - **3**) \* **42**  
 $\rightarrow$  (digit\* - **3**)\* **42**  
 $\rightarrow$  (**34** - **3**) \* **42**

[Exp  $\rightarrow$  exp op exp]  
[Exp  $\rightarrow$  number]  
[number  $\rightarrow$  digit\*]  
[digit  $\rightarrow$  42]  
[op  $\rightarrow$  \*]  
[Exp  $\rightarrow$  (exp)]  
[Exp  $\rightarrow$  exp op exp]  
[Exp  $\rightarrow$  number]  
[number  $\rightarrow$  digit]  
[digit  $\rightarrow$  3]  
[op  $\rightarrow$  -]  
[Exp  $\rightarrow$  number]  
[number  $\rightarrow$  digit\*]  
[digit  $\rightarrow$  34]

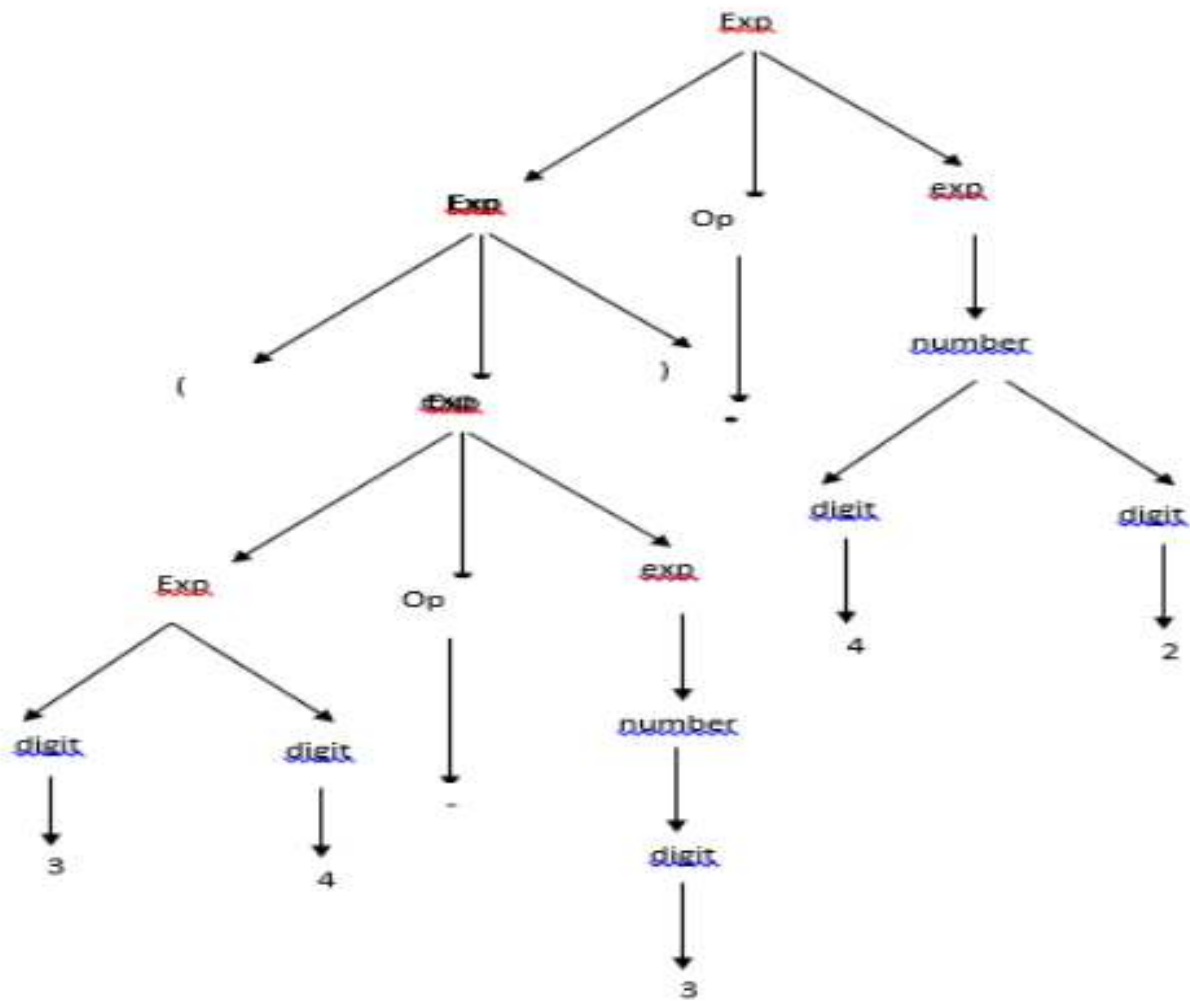
## POHON PARSING / PARSE TREE

Pohon parsing digunakan untuk menampilkan proses penurunan atau derivasi sebuah string menuju terminal menggunakan pohon.

Pohon parsing yang sesuai dengan derivasi adalah pohon berlabel dimana bagian node diberikan label nama struktur dan bagian daun untuk terminal/symbol.

Contoh.

Pohon parsing untuk string  $(34-3)^*42$  adalah sebagai berikut :





## **Ambiguitas**

*Ambiguitas* terjadi bila terdapat lebih dari satu pohon penurunan yang berbeda untuk memperoleh suatu string.

Misalkan terdapat tata bahasa sebagai berikut :

$$\bullet S \rightarrow A \mid B$$

$$\bullet A \rightarrow a$$

$$\bullet B \rightarrow a$$

Untuk memperoleh untai 'a' bisa terdapat dua cara penurunan sebagai berikut :

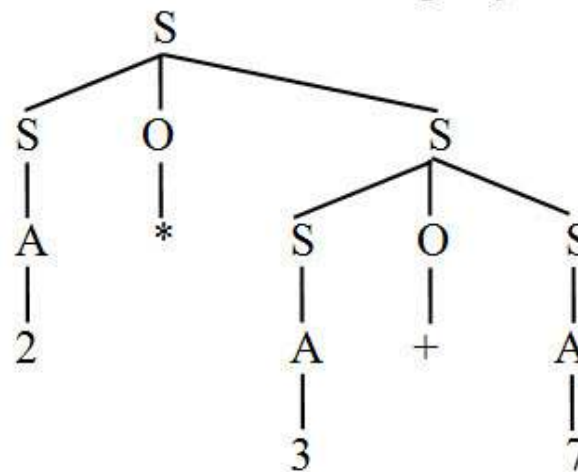
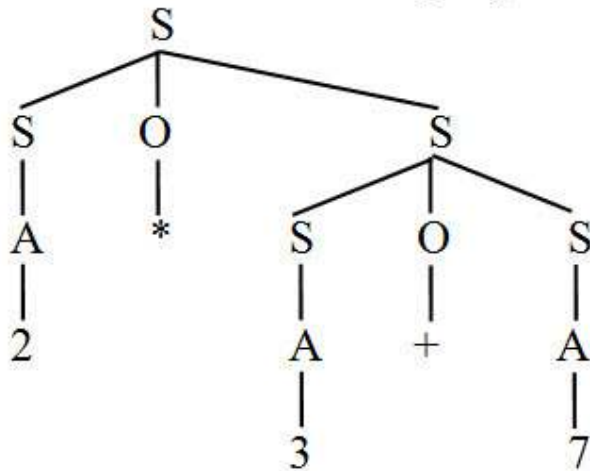
$$\bullet S \Rightarrow A \Rightarrow a$$

$$\bullet S \Rightarrow B \Rightarrow a$$

### Contoh ambiguitas lain:

Diketahui grammar  $G = \{S \rightarrow SOS|A, O \rightarrow *|+, A \rightarrow 0|1|2|\dots|9\}$

String :  $2*3+7$  mempunyai dua pohon sintaks berikut :



Sebuah string yang mempunyai lebih dari satu pohon sintaks disebut *string ambigu (ambiguous)*. Grammar yang menghasilkan paling sedikit sebuah string ambigu disebut *grammar ambigu*.

$S \rightarrow ABCb \mid BCbAB$

$A \rightarrow Aba \mid a \mid b \mid c$

$B \rightarrow CBAb \mid BCAB \mid a \mid b \mid c$

$C \rightarrow a \mid b \mid c$

Cek string berikut diterima atau ditolak

1) abcbbcb

2) bbbbbbbaac

3) aaabbbbc

2. Buatlah derivation dari grammar dibawah ini

$Exp \rightarrow (exp) \mid exp \ op \ exp \mid \textit{number}$

$Op \rightarrow + \mid - \mid *$

$Number \rightarrow \textit{digit} \mid \textit{digit} *$

$Digit \rightarrow 0 \mid 1 \mid \dots \mid 9$

Apakah string  $((40 + 20) - 6) * (5 - 4)$  diterima atau ditolak?