

# LAPORAN TUGAS PEKAN 6 NLP : CONSTITUENCY PARSER

CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

## 1. Deskripsi Masalah

Akan dilakukan eksplorasi pendefinisian Grammar dengan CFG, mengurai kalimat dengan Grammar yang sudah didefinisikan, dan membangkitkan Grammar dari treebank constituency Bahasa Indonesia.

## 2. Perancangan Sistem & Analisis

### a. Bagian I

#### • Definisi CFG

```
grammar_1 = nltk.CFG.fromstring("""
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "melihat" | "memakan" | "berjalan"
NP -> "John" | "Mary" | "Bob" | Det N | Det N PP
Det -> "sebuah" | "seorang" | "seekor"
N -> "pria" | "anjing" | "kucing" | "taman" | "ayam"
P -> "di" | "oleh" | "dengan" | "milik"
""")
```

#### • Kalimat Uji

```
sent_1 = 'John melihat seekor kucing di sebuah taman'.split()
sent_2 = 'Mary memakan seekor ayam dengan John'.split()
sent_3 = 'seorang pria melihat seekor anjing milik Bob'.split()
```

#### • Top Down Parser

```
Top Down Parser
=====
(S
  (NP John)
  (VP
    (V melihat)
    (NP
      (Det seekor)
      (N kucing)
      (PP (P di) (NP (Det sebuah) (N taman))))))
(S
  (NP John)
  (VP
    (V melihat)
    (NP (Det seekor) (N kucing))
    (PP (P di) (NP (Det sebuah) (N taman))))))
=====
(S
  (NP Mary)
  (VP
    (V memakan)
    (NP (Det seekor) (N ayam) (PP (P dengan) (NP John))))))
(S
  (NP Mary)
  (VP
    (V memakan)
    (NP (Det seekor) (N ayam))
    (PP (P dengan) (NP John))))))
=====
(S
  (NP (Det seorang) (N pria))
  (VP
    (V melihat)
    (NP (Det seekor) (N anjing) (PP (P milik) (NP Bob))))))
(S
  (NP (Det seorang) (N pria))
  (VP
    (V melihat)
    (NP (Det seekor) (N anjing))
    (PP (P milik) (NP Bob))))))
```

#### • Bottom Up Parser

```
Bottom Up Parser
=====
(S
  (NP John)
  (VP
    (V melihat)
    (NP (Det seekor) (N kucing))
    (PP (P di) (NP (Det sebuah) (N taman))))))
(S
  (NP John)
  (VP
    (V melihat)
    (NP
      (Det seekor)
      (N kucing)
      (PP (P di) (NP (Det sebuah) (N taman))))))
=====
(S
  (NP Mary)
  (VP
    (V memakan)
    (NP (Det seekor) (N ayam))
    (PP (P dengan) (NP John))))))
(S
  (NP Mary)
  (VP
    (V memakan)
    (NP (Det seekor) (N ayam) (PP (P dengan) (NP John))))))
=====
(S
  (NP (Det seorang) (N pria))
  (VP
    (V melihat)
    (NP (Det seekor) (N anjing))
    (PP (P milik) (NP Bob))))))
(S
  (NP (Det seorang) (N pria))
  (VP
    (V melihat)
    (NP (Det seekor) (N anjing) (PP (P milik) (NP Bob))))))
```

Berdasarkan top-down dan bottom-up parser bahwa 3 kalimat uji tersebut ambigu. Karena terapat lebih dari 1 tree yang dihasilkan.

#### • Shift Reduce Parser

```
Shift Reduce Parser
=====
Parsing 'John melihat seekor kucing di sebuah taman'
[ * John melihat seekor kucing di sebuah taman]
S [ 'John' * melihat seekor kucing di sebuah taman]
R [ NP * melihat seekor kucing di sebuah taman]
S [ NP 'melihat' * seekor kucing di sebuah taman]
R [ NP V * seekor kucing di sebuah taman]
S [ NP V 'seekor' * kucing di sebuah taman]
R [ NP V Det * kucing di sebuah taman]
S [ NP V Det 'kucing' * di sebuah taman]
R [ NP V Det N * di sebuah taman]
S [ NP V NP * di sebuah taman]
R [ NP VP * di sebuah taman]
S [ S * di sebuah taman]
S [ S 'di' * sebuah taman]
R [ S P * sebuah taman]
S [ S P 'sebuah' * taman]
R [ S P Det * taman]
S [ S P Det 'taman' * ]
R [ S P Det N * ]
S [ S P NP * ]
R [ S PP * ]
=====
```

# LAPORAN TUGAS PEKAN 6 NLP : CONSTITUENCY PARSER

CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

```
Parsing 'Mary memakan seekor ayam dengan John'
[ * Mary memakan seekor ayam dengan John]
S [ 'Mary' * memakan seekor ayam dengan John]
R [ NP * memakan seekor ayam dengan John]
S [ NP 'memakan' * seekor ayam dengan John]
R [ NP V * seekor ayam dengan John]
S [ NP V 'seekor' * ayam dengan John]
R [ NP V Det * ayam dengan John]
S [ NP V Det 'ayam' * dengan John]
R [ NP V Det N * dengan John]
R [ NP V NP * dengan John]
R [ NP VP * dengan John]
R [ S * dengan John]
S [ S 'dengan' * John]
R [ S P * John]
S [ S P 'John' * ]
R [ S P NP * ]
R [ S PP * ]
```

```
Parsing 'seorang pria melihat seekor anjing milik Bob'
[ * seorang pria melihat seekor anjing milik Bob]
S [ 'seorang' * pria melihat seekor anjing milik Bob]
R [ Det * pria melihat seekor anjing milik Bob]
S [ Det 'pria' * melihat seekor anjing milik Bob]
R [ Det N * melihat seekor anjing milik Bob]
R [ NP * melihat seekor anjing milik Bob]
S [ NP 'melihat' * seekor anjing milik Bob]
R [ NP V * seekor anjing milik Bob]
S [ NP V 'seekor' * anjing milik Bob]
R [ NP V Det * anjing milik Bob]
S [ NP V Det 'anjing' * milik Bob]
R [ NP V Det N * milik Bob]
R [ NP V NP * milik Bob]
R [ NP VP * milik Bob]
R [ S * milik Bob]
S [ S 'milik' * Bob]
R [ S P * Bob]
```

Pada shift-reducer, kata akan di parse secara bertahap, dari bawah ke atas, dan dari kiri ke kanan.

## • Konversi CNF

Dikarenakan grammar\_1 belum dalam bentuk CNF, maka akan dikonversi menjadi CNF. Syarat CNF adalah di sebelah kanan hanya boleh terdapat 1 terminal atau 2 non-terminal. Pada grammar\_1 terdapat 2 aturan produksi yang belum dalam bentuk CNF, yaitu “PP -> V NP PP” dan “NP -> Det N PP”. Kemudian penggantian aturan produksi yang baru, “PP -> H0 PP” dan “NP -> H1 PP”. terbentuk aturan produksi baru, yaitu “H0 -> V NP” dan “H1 -> Det N”.

```
grammar_2 = nltk.CFG.fromstring("""
S -> NP VP
VP -> V NP
PP -> P NP | H0 PP
V -> "melihat" | "memakan" | "berjalan"
NP -> "John" | "Mary" | "Bob" | Det N | H1 PP
Det -> "sebuah" | "seorang" | "seekor"
N -> "pria" | "anjing" | "kucing" | "taman" | "ayam"
P -> "di" | "oleh" | "dengan" | "milik"
H0 -> V NP
H1 -> Det N
""")
1 | print(grammar_2.is_chomsky_normal_form())
```

True

## b. Bagian II

### • Kalimat Uji

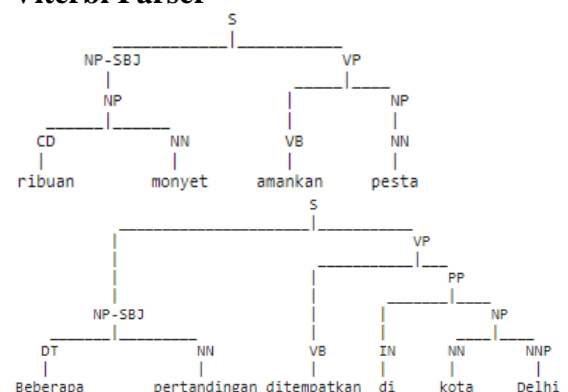
```
sent1 = 'ribuan monyet amankan pesta'.split()
sent2 = 'Beberapa pertandingan ditempatkan di kota Delhi'.split()
```

### • Bottom Up Parser

```
Bottom Up Parser
=====
(S
  (NP-SBJ (NP (CD ribuan) (NN monyet)))
  (VP (VB amankan) (NP (NN pesta))))
(S
  (NP-SBJ (NP (CD ribuan) (NN monyet)))
  (VP (VB amankan) (NP (NP (NN pesta)))))
(S
  (NP-SBJ (NP (NP (CD ribuan) (NN monyet))))
  (VP (VB amankan) (NP (NN pesta))))
(S
  (NP-SBJ (NP (NP (CD ribuan) (NN monyet))))
  (VP (VB amankan) (NP (NP (NN pesta)))))
=====
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP (VB ditempatkan) (PP (IN di) (NP (NP (NN kota)) (NNP Delhi)))))
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP
    (VB ditempatkan)
    (PP (IN di) (NP (NP (NP (NN kota)) (NNP Delhi))))))
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP
    (VB ditempatkan)
    (PP (IN di) (NP (NP (NP (NP (NN kota)) (NNP Delhi))))))
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP
    (VB ditempatkan)
    (PP (IN di) (NP (NP (NP (NP (NP (NN kota)) (NNP Delhi))))))
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP (VB ditempatkan) (PP (IN di) (NP (NP (NN kota) (NNP Delhi))))))
(S
  (NP-SBJ (DT Beberapa) (NN pertandingan))
  (VP (VB ditempatkan) (PP (IN di) (NP (NN kota) (NNP Delhi))))))
```

Berdasarkan bottom up parser, 2 kalimat uji tersebut terdapat ambigu, dikarenakan dihasilkan lebih dari 1 tree. Kalimat yang bukan berasal dari constituency treebank, memiliki tree lebih banyak dibandingkan kalimat yang berasal dari constituency treebank.

### • Viterbi Parser



Kemudian pada Viterbi parser, setiap kalimat akan dicari probability parser tree tertinggi, sehingga hanya akan ditampilkan 1 parser tree.