

LAPORAN TUGAS NLP PEKAN 6 – CONSTITUENCY PARSING

I. Deskripsi Masalah

Constituency Parsing adalah sebuah pekerjaan atau sebuah proses yang bertujuan untuk mengekstraksi sebuah kalimat agar setiap katanya dimasukkan dalam kategorikan sesuai dengan struktur sintatiknya atau aturan produksinya menurut struktur frase grammarnya. Salah satu jenis aturan produksi adalah CFG atau *Context-Free Grammar* yang memiliki tujuan untuk menunjukkan bagaimana menghasilkan suatu untai kata dalam sebuah bahasa.

II. Penyelesaian Masalah

Pada penelitian ini, dilakukan pembuatan grammar menurut dengan aturan produksi CFG menggunakan library nltk. Kemudian diberikan tiga buah kalimat yang kata-katanya berasal dari grammar yang sudah ada. Ketiga kalimat tersebut diparsing dengan parser top down, bottom up dan shift reduce. Jika sudah diparsing maka akan ketahuan kalimat mana yang memiliki ambiguitas di dalamnya. Jenis aturan produksi bukan hanya CFG, tetapi juga ada CNF. Oleh karena itu, grammar CFG tersebut diperiksa apakah sudah memenuhi syarat CNF, jika belum memenuhi maka ubah ke dalam grammar CNF.

Pada bagian dua, dilakukan pemanggilan grammar yang sudah ada dari github kethu, grammar tersebut sudah memiliki nilai probabilitasnya karena grammar tersebut termasuk dalam jenis PCFG atau *Probabilistic Context-Free Grammar*. Setelah itu coba dites parsing menggunakan kalimat yang sudah ada dari contoh tersebut menggunakan bottom up. Kemudian dilakukan lagi parsing dengan bottom up kalimat baru di luar contoh. Kedua contoh kalimat tersebut kemudian diparsing menggunakan Viterbi parser untuk menghasilkan 1 pohon parsing.

III. Analisis Hasil

1. Grammar CFG

```
grammar_1 = nltk.CFG.fromstring("""
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "makan" | "pergi" | "beli"
NP -> "Saya" | "Ayah" | "Ibu" | Det N | Det N PP
Det -> "sebuah" | "ke" | "seikat"
N -> "roti" | "Jakarta" | "Taman" | "sayur"
P -> "di" | "dengan" | "dan"
""")
```

2. 3 Kalimat, dengan kalimat 1 dan 3 mengandung ambiguitas.

Kalimat 1 = Ayah pergi ke Jakarta dengan Ibu, Kalimat 2 = Saya makan sebuah roti, Kalimat 3 = Ibu beli seikat sayur dan sebuah ayam

- Pohon hasil dengan Top Down Parser

```
> Kalimat 1
(S
  (NP Ayah)
  (VP (V pergi) (NP (Det ke) (N Jakarta) (PP (P dengan) (NP Ibu)))))
Kalimat 2
(S (NP Saya) (VP (V makan) (NP (Det sebuah) (N roti))))
Kalimat 3
(S
  (NP Ibu)
  (VP
    (V beli)
    (NP
      (Det seikat)
      (N sayur)
      (PP (P dan) (NP (Det sebuah) (N ayam)))))
  (S
    (NP Ibu)
    (VP
      (V beli)
      (NP (Det seikat) (N sayur))
      (PP (P dan) (NP (Det sebuah) (N ayam)))))
```

- Pohon hasil dengan Bottom Up Parser

```
Kalimat 1
(S
  (NP Ayah)
  (VP (V pergi) (NP (Det ke) (N Jakarta)) (PP (P dengan) (NP Ibu))))
Kalimat 2
(S (NP Saya) (VP (V makan) (NP (Det sebuah) (N roti))))
Kalimat 3
(S
  (NP Ibu)
  (VP
    (V beli)
    (NP (Det seikat) (N sayur))
    (PP (P dan) (NP (Det sebuah) (N ayam)))))
  (S
    (NP Ibu)
    (VP
      (V beli)
      (NP
        (Det seikat)
        (N sayur)
        (PP (P dan) (NP (Det sebuah) (N ayam)))))
```

- Pohon hasil dengan Shift Reduce Parser

Kalimat 1
 Parsing 'Ayah pergi ke Jakarta dengan Ibu'
 [* Ayah pergi ke Jakarta dengan Ibu]
 S ['Ayah' * pergi ke Jakarta dengan Ibu]
 R [NP * pergi ke Jakarta dengan Ibu]
 S [NP 'pergi' * ke Jakarta dengan Ibu]
 R [NP V * ke Jakarta dengan Ibu]
 S [NP V 'ke' * Jakarta dengan Ibu]
 R [NP V Det * Jakarta dengan Ibu]
 S [NP V Det 'Jakarta' * dengan Ibu]
 R [NP V Det N * dengan Ibu]
 R [NP V NP * dengan Ibu]
 R [NP VP * dengan Ibu]
 R [S * dengan Ibu]
 S [S 'dengan' * Ibu]
 R [S P * Ibu]
 S [S P 'Ibu' *]
 R [S P NP *]
 R [S PP *]

Kalimat 2
 Parsing 'Saya makan sebuah roti'
 [* Saya makan sebuah roti]
 S ['Saya' * makan sebuah roti]
 R [NP * makan sebuah roti]
 S [NP 'makan' * sebuah roti]
 R [NP V * sebuah roti]
 S [NP V 'sebuah' * roti]
 R [NP V Det * roti]
 S [NP V Det 'roti' *]
 R [NP V Det N *]
 R [NP V NP *]
 R [NP VP *]
 R [S *]
 (S (NP Saya) (VP (V makan) (NP (Det sebuah) (N roti))

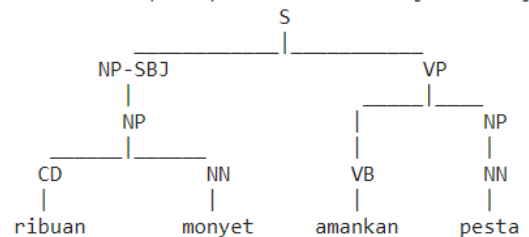
Kalimat 3
 Parsing 'Ibu beli seikat sayur dan sebuah ayam'
 [* Ibu beli seikat sayur dan sebuah ayam]
 S ['Ibu' * beli seikat sayur dan sebuah ayam]
 R [NP * beli seikat sayur dan sebuah ayam]
 S [NP 'beli' * seikat sayur dan sebuah ayam]
 R [NP V * seikat sayur dan sebuah ayam]
 S [NP V 'seikat' * sayur dan sebuah ayam]
 R [NP V Det * sayur dan sebuah ayam]
 S [NP V Det 'sayur' * dan sebuah ayam]
 R [NP V Det N * dan sebuah ayam]
 R [NP V NP * dan sebuah ayam]
 R [NP VP * dan sebuah ayam]
 R [S * dan sebuah ayam]
 S [S 'dan' * sebuah ayam]
 R [S P * sebuah ayam]
 S [S P 'sebuah' * ayam]
 R [S P Det * ayam]
 S [S P Det 'ayam' *]
 R [S P Det N *]
 R [S P NP *]
 R [S PP *]

- Grammar yang didefinisikan tersebut tidak memenuhi syarat CNF.
- Grammar diubah agar memenuhi syarat CNF secara manual. Syarat CNF adalah pada ruas kanan atau hasil produksi harus dalam bentuk 1 terminal atau 2 variabel. Sedangkan, pada grammar yang ada masih terdapat hasil produksi yang terdapat 3 variabel, maka dari itu hasil produksi tersebut dihapus agar grammar memenuhi syarat CNF.

```
grammar_CNF = nltk.CFG.fromstring("""
S -> NP VP
VP -> V NP
PP -> P NP
V -> "makan" | "pergi" | "beli"
NP -> "Saya" | "Ayah" | "Ibu" | Det N
Det -> "sebuah" | "ke"
N -> "roti" | "sayur" | "jakarta" | "taman" | "ayam"
P -> "di" | "dengan" | "dan"
""")
```

Bagian 2;

- Hasil parsing kalimat yang ada pada contoh ('ribuan monyet amankan pesta') menggunakan bottom up parser memiliki hasil dua kemungkinan pohon akar. Sedangkan, yang menggunakan metode Viterbi parsing hanya menghasilkan satu pohon akar seperti berikut ini.



- Hasil parsing kalimat di luar contoh ('sebagian hewan India ditempatkan pemerintah di negara bagian') menggunakan bottom up parser memiliki banyak kemungkinan pohon akar. Sedangkan, yang menggunakan metode Viterbi parsing hanya memiliki satu pohon akar seperti berikut ini.



Perbedaan jumlah pohon akar pada metode bottom up parser kedua kalimat tersebut, serta perbedaan jumlah level pada pohon akar metode Viterbi parser disebabkan oleh jumlah kata yang berbeda. Contoh kalimat 1 hanya terdiri dari empat kata, sedangkan contoh kalimat 2 terdiri dari 8 kata. Hal tersebut bisa menjadi faktor karena banyaknya kemungkinan hasil produksi pada saat parsing.