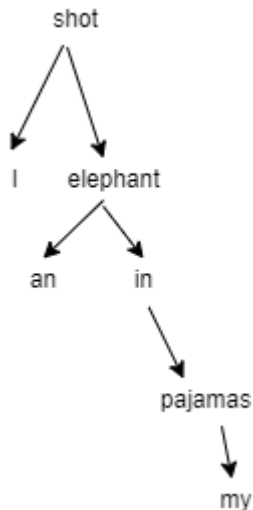


LAPORAN TUGAS NLP PEKAN 7 – DEPENDENCY PARSING

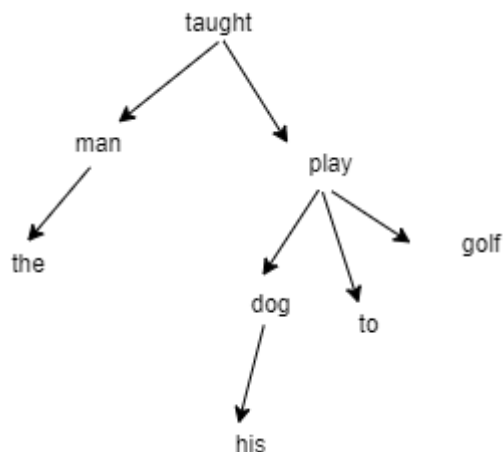
Bagian I

1. Bentuk pohon hasil parsing kalimat dengan dep_grammar_1 :
Asal kalimat 'I shot an elephant in my pajamas'



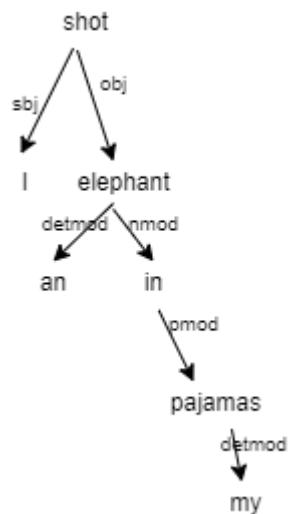
Pada tree ini, kata yang menjadi head adalah 'shot' karena pada grammar kata utama yang mendefinisikan dan yang memulai tracing kalimat adalah 'shot'. Dan kata elephant bergantung pada kata shot begitu pula seterusnya.

- Bentuk pohon hasil parsing kalimat dengan dep_grammar_2 :
Asal kalimat 'The man taught his dog to play golf'



Pada tree ini, kata yang menjadi head adalah 'taught' karena pada grammar kata utama yang mendefinisikan dan yang memulai tracing kalimat adalah 'taught'. Dan kata elephant bergantung pada kata shot begitu pula seterusnya.

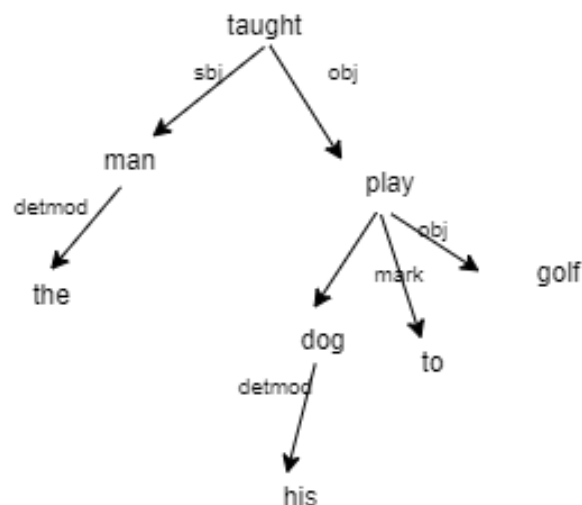
2. Bentuk pohon hasil parsing kalimat dengan dep_grammar_1 setelah diberi relasi :



Kalimat di atas adalah 'I shot an elephant in my pajamas'.

- Kata I dengan shot memiliki relasi sbj/nsubj, relasi ini menghubungkan subjek/orang yang melakukan dengan pekerjaan yang dilakukannya atau predikat.
- Kata shot dengan elephant memiliki relasi obj, relasi ini menghubungkan antara pekerjaan atau yang sedang dilakukan dengan objek dari perlakuan tersebut.
- Kata an dengan elephant memiliki relasi detmod atau det, relasi ini untuk menghubungkan sebuah kata benda dengan keterangannya atau jumlahnya.
- Kata elephant dengan kata in memiliki relasi nmod, yang menghubungkan antara kata benda dengan keterangan tempatnya.
- Kata in dengan pajamas memiliki relasi pmod, karna menunjukkan keterangan keberadaan sebuah benda.
- Kata pajamas dengan my memiliki relasi detmod atau det yang menunjukkan keterangan kepunyaan sebuah benda.

Bentuk pohon hasil parsing kalimat dengan dep_grammar_2 setelah diberi relasi :



Kalimat di atas adalah 'The man taught his dog to play golf'.

- Kata taught dengan man memiliki relasi sbj atau nsubj, relasi ini menghubungkan antara subjek atau pelaku dengan kata kerja atau yang dilakukannya atau bisa disebut dengan predikat.
- Kata the dengan man memiliki relasi detmod atau det, yang menunjukkan keterangan jumlah dari benda atau subjek.
- Kata taught dengan play memiliki relasi obj, relasi ini menghubungkan antara kata kerja dengan objek yang dikerjakannya.
- Kata his dengan kata dog memiliki relasi detmod, relasi ini menunjukkan keterangan kepunyaan yaitu dog dimiliki oleh his.
- Kata play dengan golf memiliki relasi obj, relasi ini menghubungkan antara kata kerja dengan apa yang dikerjakannya atau objek yang dikerjakan.
- Kata to dengan play memiliki relasi mark relasi ini menghubungkan ad verb dengan kata kerja.

3. Perbedaan antara Projective Parser dan Non-Projective Parser :

Dari root ke dependent dapat dikatakan projective jika ada path atau jalan dari head atau root tersebut ke setiap word yang ada pada kalimat, sedangkan bisa dikatakan non-projective jika ada kata yang tidak bisa saling berhubungan dan tidak ada jalan di antaranya.

4. Keterangan format CONLL :

Format CONLL adalah format yang digunakan pada teks mentah yang harus diencode seperti UTF-8 dsb.

Atribut CONLL :

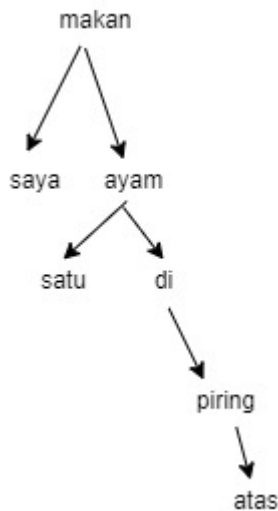
- ADV
- VERB
- PRON
- NUM
- NOUN
- PUNCT
- CONJ
- ADP
- DET
- PROPN

Bagian II

1. Grammar baru yang dibuat :

```
Dependency grammar with 7 productions
'makan' -> 'saya'
'makan' -> 'ayam'
'makan' -> 'di'
'ayam' -> 'satu'
'ayam' -> 'di'
'di' -> 'piring'
'piring' -> 'atas'
```

Grammar ini sesuai untuk di-parse dengan projective dependency parser karena semua katanya terhubung dari level paling atas hingga ke bawah.



Kalimat yang dibuat adalah 'Saya makan satu ayam di atas piring'. Jika dalam bentuk tree, kata yang berada pada ujung setiap level tetap terhubung dengan kata sesudah maupun sebelumnya. Hasil parsing dari kalimat tersebut adalah :

```
1 pdp = nltk.ProjectiveDependencyParser(dep_grammar_1)
2 sent = 'saya makan ayam di atas piring'.split()
3 trees = pdp.parse(sent)
4 for tree in trees:
5     print(tree)
```

```
(makan saya (ayam (di (piring atas))))
(makan saya ayam (di (piring atas)))
```

2. Grammar yang baru dibuat :

```
Dependency grammar with 5 productions
'membelikan' -> 'kalung'
'membelikan' -> 'ayah'
'kalung' -> 'kucing'
'kalung' -> 'sebuah'
'kucing' -> 'miliknya'
```

Grammar ini sesuai untuk diparsing dengan Non-projective parser karena tidak semua katanya dapat terhubung satu sama lain dari awal hingga akhir, seperti yang dilihat pada pohon di bawah ini. Meskipun antar katanya sudah saling terhubung dengan panah tetapi jika ditracing tidak dapat dibaca berurutan sesuai dengan kalimat yang diberikan.



Hasil parsing dari kalimat tersebut adalah sebagai berikut :

```

{0: {'address': 0,
    'ctag': 'TOP',
    'deps': defaultdict(<class 'list'>, {'ROOT': [2]}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': 'TOP',
    'word': None},
 1: {'address': 1,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'ayah'},
 2: {'address': 2,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {'': [1, 6]}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'membelikan'},
 3: {'address': 3,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {'': [4]}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'kucing'},
 4: {'address': 4,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'miliknya'},
 5: {'address': 5,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'sebuah'},
 6: {'address': 6,
    'ctag': None,
    'deps': defaultdict(<class 'list'>, {'': [3, 5]}),
    'feats': None,
    'head': None,
    'lemma': None,
    'rel': None,
    'tag': None,
    'word': 'kalung'}}

```
