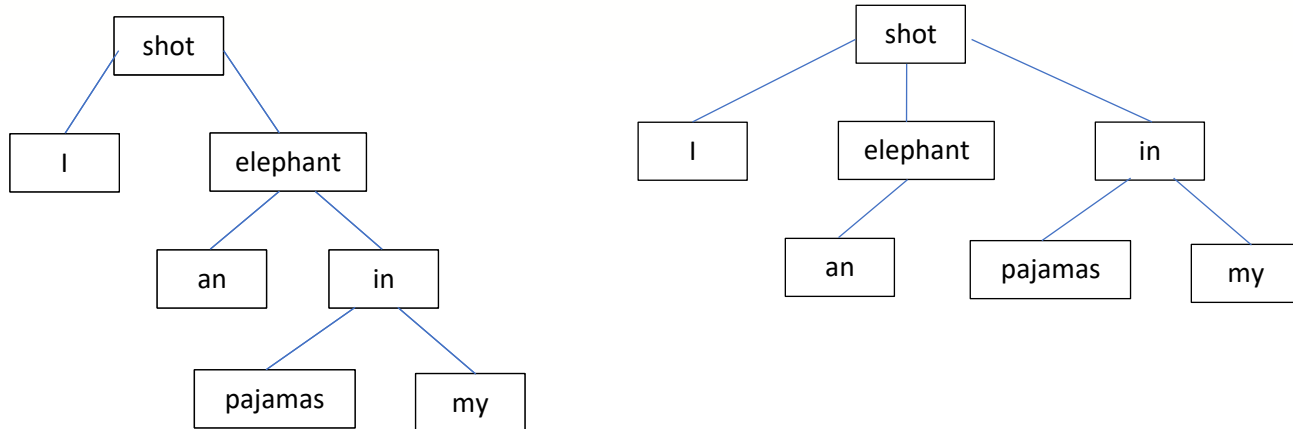


# LAPORAN TUGAS PEKAN 7 NLP : DEPENDENCY PARSER

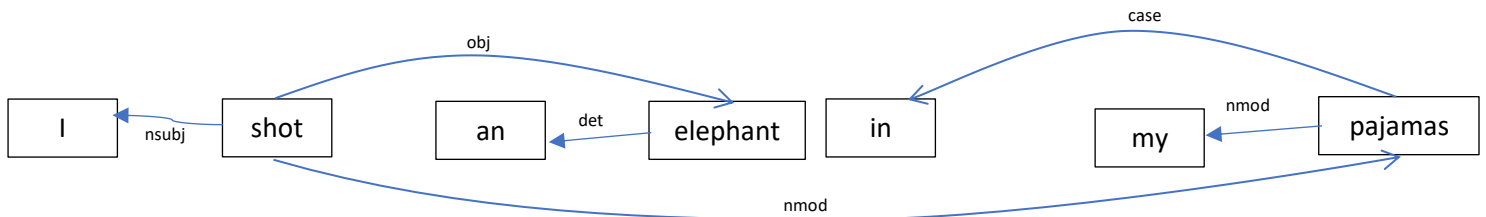
CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

## Bagian I

1. Pada hasil parsing kalimat uji 'I shot an elephant in my pajamas' pada dep\_grammar\_1 terdapat 2 dependency tree, artinya kalimat tersebut ambigu. Dependency tree pertama adalah (shot I (elephant an (in (pajamas my)))) pada gambar sebelah kiri, dan dependency tree kedua adalah (shot I (elephant an) (in (pajamas my))) pada gambar sebelah kanan

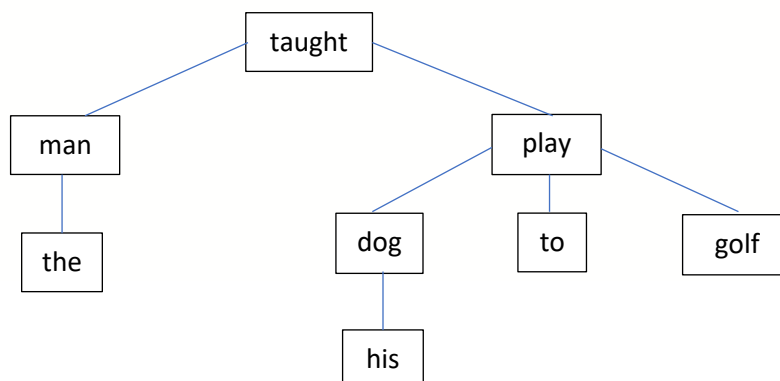


Kemudian relasi head-dependent nya adalah sebagai berikut



Kata 'shot' bergantung pada kata 'I', 'elephant', dan 'pajamas'. Kata 'elephant' bergantung pada kata 'an'. Kata 'pajamas' bergantung pada kata 'my' dan 'in'.

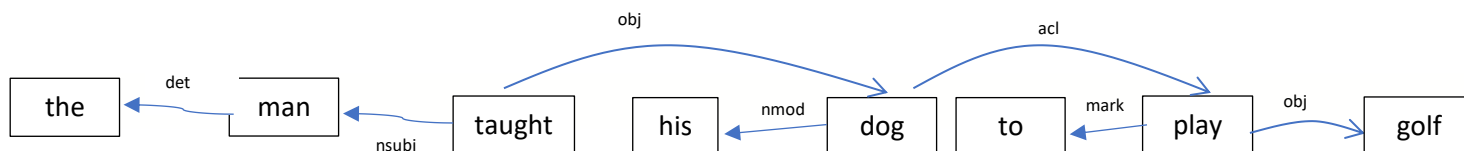
Pada hasil parsing kalimat uji 'the man taught his dog to play golf' pada dep\_grammar\_2, terdapat dependency tree yang dapat digambarkan seperti dibawah ini



# LAPORAN TUGAS PEKAN 7 NLP : DEPENDENCY PARSER

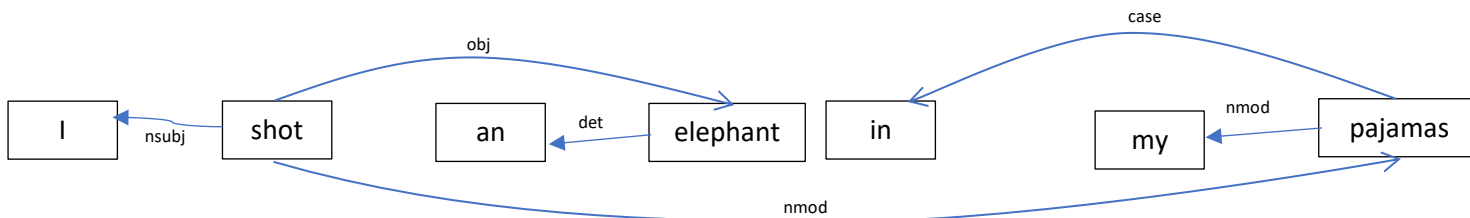
CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

Kemudian relasi head-dependent nya adalah sebagai berikut

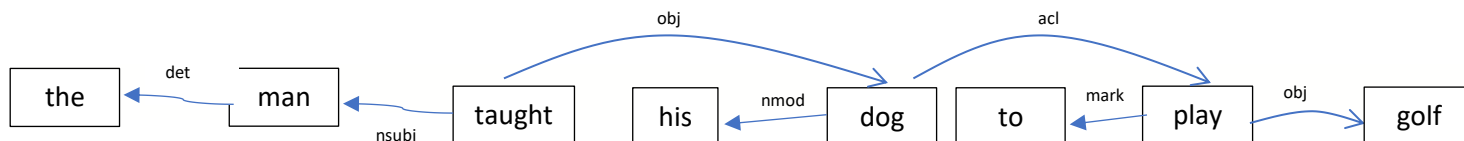


Kata 'man' bergantung pada kata 'the'. Kata 'taught' bergantung pada kata 'man' dan 'dog'. Kata 'dog' bergantung pada kata 'his' dan 'play'. Kata 'play' bergantung pada kata 'to' dan 'golf'.

2.



Pada dep\_grammar\_1 dibentuk relasi head-dependent beserta tipe relasinya. Kata 'I' dan 'shot' tipe relasinya adalah 'nsubj' karena 'I' adalah subjek. Kata 'shot' dan 'elephant' tipe relasinya adalah 'obj' karena 'elephant' adalah objek. Kata 'shot' dan 'pajamas' tipe relasinya adalah 'nmod' karena 'shot' adalah dependen nominal dari kata benda 'pajamas'. Kata 'an' dan 'elephant' tipe relasinya adalah 'det' karena 'an' adalah determiner dari 'elephant'. Kata 'in' dan 'pajamas' tipe relasinya adalah 'case' karena 'in' adalah dependen dari kata benda 'pajamas'. Kata 'my' dan 'pajamas' tipe relasinya adalah 'nmod' karena 'my' adalah dependen nominal dari kata benda 'pajamas'.



Pada dep\_grammar\_2 dibentuk relasi head-dependent beserta tipe relasinya. Kata 'the' dan 'man' tipe relasinya adalah 'det' karena 'the' adalah determiner dari 'man'. Kata 'man' dan 'taught' tipe relasinya adalah 'nsubj' karena 'man' adalah subjek. Kata 'taught' dan 'dog' tipe relasinya adalah 'obj' karena 'dog' adalah objek. Kata 'his' dan 'dog' tipe relasinya adalah 'nmod' karena 'his' adalah dependen nominal dari kata 'dog'. Kata 'dog' dan 'play' tipe relasinya adalah 'acl' karena 'dog' adalah klausa adverbial yang mengubah predikat 'play'. Kata 'to' dan 'play' tipe relasinya adalah 'mark' karena 'to' menandai klausa sebagai subordinat dari klausa lain. Kata 'play' dan 'golf' tipe relasinya adalah 'obj' karena 'golf' adalah objek.

## LAPORAN TUGAS PEKAN 7 NLP : DEPENDENCY PARSER

CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

3. Projective parser pada saat dalam bentuk dependency tree tidak ada relasi yang bersilangan. Cocok untuk Bahasa yang terstruktur seperti Bahasa Inggris. Non projective parser pada saat dalam bentuk dependency tree, bias saja terdapat relasi yang bersilangan. Biasanya sering dipakai untuk Bahasa yang tidak terstruktur seperti Bahasa Jerman, Belanda, Ceko, dsb.

4. CoNLL-U Format dari kalimat uji 1 dan 2. Pada kolom pertama adalah index kata. Kolom kedua adalah kata-kata hasil tokenisasi. Kolom ketiga adalah hasil lemmatisasi kata nya. Kolom keempat adalah postag nya. Kolom keenam adalah head kata. Kolom ketujuh adalah hubungan ketergantungan terhadap head.

1	I	I	PRP	0	2	nsubj
2	shot	shoot	VBD	0	0	ROOT
3	an	a	DT	0	4	det
4	elephant	elephant	NN	0	2	dobj
5	in	in	IN	0	7	case
6	my	my	PRP\$	0	7	nmod:poss
7	pajamas	pajamas	NNS	0	2	nmod
1	the	the	DT	0	2	det
2	man	man	NN	0	3	nsubj
3	taught	teach	VBD	0	0	ROOT
4	his	he	PRP\$	0	5	nmod:poss
5	dog	dog	NN	0	3	dobj
6	to	to	TO	0	7	mark
7	play	play	VB	0	5	acl
8	golf	golf	NN	0	7	dobj

## Bagian II

1. Kalimat yang dibuat untuk projective parser adalah 'Saya mengelus seekor kucing di taman'. Karena susunan kalimatnya terstruktur. Hasil dependency tree nya tidak terdapat relasi yang bersilangan.

```
dep_grammar_3 = nltk.DependencyGrammar.fromstring("""
    'mengelus' -> 'Saya' | 'kucing' | 'di'
    'kucing' -> 'seekor'
    'di' -> 'taman'
    """)
Dependency grammar with 5 productions
'mengelus' -> 'Saya'
'mengelus' -> 'kucing'
'mengelus' -> 'di'
'kucing' -> 'seekor'
'di' -> 'taman'
```

```
1 pdp = nltk.ProjectiveDependencyParser(dep_grammar_3)
2 sent = 'Saya mengelus seekor kucing di taman'.split()
3 trees = pdp.parse(sent)
4
5 for tree in trees:
6     print(tree)
```

(mengelus Saya (kucing seekor) (di taman))

## LAPORAN TUGAS PEKAN 7 NLP : DEPENDENCY PARSER

CLARISA HASYA YUTIKA | 1301174256 | IF 41 GAB01

2. Kalimat yang dibuat untuk non projective parser adalah 'Kakak Maya belanja sayur untuk makanan kelinci'. Karena susunan kalimatnya tidak terstruktur. Dari hasil dependency parser didapatkan, 'belanja' sebagai head. 'belanja' = [Maya, sayur, kelinci], 'Maya' = ['Kakak'], 'kelinci' = ['makanan', 'untuk']

```
dep_grammar_4 = nltk.DependencyGrammar.fromstring("""
    'belanja' -> 'sayur' | 'Maya' | 'kelinci'
    'Maya' -> 'Kakak'
    'kelinci' -> 'makanan' | 'untuk'
    """)

Dependency grammar with 6 productions
    'belanja' -> 'sayur'
    'belanja' -> 'Maya'
    'belanja' -> 'kelinci'
    'Maya' -> 'Kakak'
    'kelinci' -> 'makanan'
    'kelinci' -> 'untuk'

npdp = nltk.NonprojectiveDependencyParser(dep_grammar_4)
sent = 'Kakak Maya belanja sayur untuk makanan kelinci'.split()
trees = npdp.parse(sent)

for tree in trees:
    print(tree)

{0: {'address': 0,
     'ctag': 'TOP',
     'deps': defaultdict(<class 'list'>, {'ROOT': [3]}),
     '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': 'Kakak'},
 2: {'address': 2,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {'': [1]}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'Maya'},
 3: {'address': 3,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {'': [2, 4, 7]}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'belanja'},
 4: {'address': 4,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'sayur'},
 5: {'address': 5,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'untuk'},
 6: {'address': 6,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'makanan'},
 7: {'address': 7,
     'ctag': None,
     'deps': defaultdict(<class 'list'>, {'': [5, 6]}),
     'feats': None,
     'head': None,
     'lemma': None,
     'rel': None,
     'tag': None,
     'word': 'kelinci'}})
```