



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Aplikasi Penentuan Kategori dan Fungsi Sintaksis Kalimat Bahasa Indonesia

Abd. Wahab Syahrone¹, Harsono²

¹ Teknik Informatika, Universitas Madura Jl. Raya Panglegur Km 3,5 Pamekasan, Madura

² Pendidikan Bahasa dan Sastra, Universitas Madura Jl. Raya Panglegur Km 3,5 Pamekasan, Madura

KEYWORDS

"NLP, Bahasa Indonesia, Sintaksis, Teknik Chunking"

CORRESPONDENCE

Phone: +6282330511716

E-mail: roney@unira.ac.id

ABSTRACT

Bahasa Indonesia merupakan bagian penting dalam kehidupan Bangsa Indonesia Namun penelitian yang dilakukan terhadap Bahasa Indonesia masih sedikit di temukan terutama mengenai satuan sintaksis (Kata, Frasa, Klausa dan Kalimat), padahal sintaksis merupakan tata kalimat. Secara sintaksis, sebuah kata atau frasa dalam kalimat harus memperlihatkan pertautan, pertautan itu akan tampak apabila unsur-unsur dalam tata bahasa yang berfungsi sebagai penunjuk secara konsisten dipergunakan. Dengan demikian maka kalimat yang disampaikan menjadi lebih mudah untuk dipahami isinya. Sintaksis dapat juga diartikan sebagai pemahaman terhadap urutan kata dalam pembentukan kalimat dan hubungan antar kata tersebut dalam proses perubahan bentuk dari kalimat menjadi bentuk yang sistemis meliputi proses pengaturan tata letak suatu kata dalam kalimat.

Dalam penelitian ini, teknik chunking digunakan untuk memecah kalimat menjadi potongan kata atau frasa (kategori) dengan membuat chunk grammar yang berupa aturan (rule) yang mengindikasikan cara kalimat tersebut dipecah menjadi potongan frasa. Frasa yang dibentuk antara lain Frasa Nominal (NP), Frasa Verbal (VP), Frasa Adjektiva (AP), Frasa Adverbial (ADVP), Frasa Preposisional (PP) dan Frase Numeralia (NUMP). Dari hasil chunking ini akan dibuatkan algoritma penentuan klausa (fungsi) antara lain Subjek (S), Predikat (P), Objek (O), Pelengkap (PEL) dan Keterangan (K). Output dari penelitian ini dapat dilihat dalam bentuk teks atau grafik (tree).

Pemrosesan hasil POSTagger kalimat Bahasa Indonesia yang berupa level kata menjadi frasa (kategori) mendapatkan akurasi sebesar 93,32 % dan penentuan kedudukan klausa (fungsi) sintaksis mendapatkan akurasi sebesar 91,12%.

PENDAHULUAN

Natural Language Processing merupakan salah satu bidang ilmu kecerdasan buatan yang mempelajari komunikasi antara manusia dan komputer dengan menggunakan bahasa alami (bahasa manusia). Natural Language Processing (Pengolahan Bahasa Alami) bertujuan untuk memahami arti dari ucapan atau kalimat yang diberikan dalam bahasa alami dan memberikan respon yang sesuai seperti menampilkan data atau melakukan aksi tertentu. Untuk mencapai tujuan tersebut membutuhkan tiga tahap proses. Proses yang pertama adalah Parsing yaitu analisa sintaksis yang memeriksa kebenaran struktur kalimat berdasarkan suatu tata bahasa (Grammar) dan kosa kata (Lexicon) tertentu. Proses yang kedua adalah Interpretasi Semantik (Semantic Interpretation) yang bertujuan untuk merepresentasikan arti dari kalimat secara Context Independent untuk keperluan lebih lanjut. Adapun proses ketiga adalah Interpretasi Kontekstual (Contextual Interpretation)

yang bertujuan untuk merepresentasikan arti secara Context Dependent dan menentukan maksud dari penggunaan kalimat.

Penelitian yang dilakukan terhadap Bahasa Indonesia masih sedikit di temukan terutama mengenai satuan sintaksis (Kata, Frasa, Klausa dan Kalimat). Penelitian yang pernah ada seperti Part of Speech Tagging (POS Tagging) yang pernah dilakukan oleh Wicaksono [1] dan Pisceldo [2]. Namun POS Tagging itu hanya sebatas level kata. Penelitian yang dilakukan oleh Arman [3] mengenai frasa, tidak sampai kepada pembentukan klausa, selain itu frasa yang dibentuk tidak sesuai dengan buku-buku sintaksis bahasa indonesia pada umumnya.

Penelitian ini merupakan tahapan pertama dari tiga tahapan proses yang ada dalam NLP yaitu proses parsing. Manfaat penelitian ini kedepan dapat dijadikan landasan dasar untuk melakukan beberapa penelitian lanjutan seperti Question and Answering System, Text To Speech, serta dapat dilanjutkan ke

tahapan kedua dan ketiga proses yang ada dalam NLP yaitu Interpretasi Semantik dan Interpretasi Kontekstual. Selain itu, penelitian ini juga dapat digunakan oleh masyarakat yang ingin belajar sintaksis Bahasa Indonesia secara mandiri, atau digunakan oleh seorang guru Bahasa Indonesia untuk mengajarkan materi mengenai sintaksis.

Berdasarkan uraian di atas, maka diadakan penelitian yang menghasilkan aplikasi yang nantinya dapat memberikan gambaran tentang kategori dan fungsi sintaksis kalimat bahasa Indonesia dengan judul “Aplikasi Penentuan Kategori Dan Fungsi Sintaksis Kalimat Bahasa Indonesia”.

TINJAUAN PUSTAKA

Tinjauan pustaka yang digunakan dalam penelitian ini adalah sebagai berikut

Part of Speech Tagger (POS Tagger)

POSTagger merupakan sebuah aplikasi pengolahan bahasa alami yang dapat memberi tag pada setiap kata dalam kalimat seperti kata benda, kata kerja, kata ganti, kata depan, kata keterangan, dan kata sifat atau penanda kelas leksikal lain untuk setiap kata dalam kalimat[1].

Terdapat beberapa pendekatan pada proses penentuan tag pada setiap kata dalam kalimat dalam Natural Language Processing yaitu pendekatan secara rule based, probabilistic dan transformational.

Pendekatan secara rule based atau disebut juga pendekatan yang berbasis aturan memberikan tag atau penanda pada kata dalam suatu kalimat berdasarkan pada beberapa aturan bahasa yang dibuat secara manual.

Pendekatan probabilistik menentukan tag yang paling mungkin dari kata mengingat konteks sekitarnya, berdasarkan nilai probabilitas yang diperoleh dari korpus yang diberi tag secara manual.

Pendekatan berbasis transformasional menggabungkan pendekatan berbasis aturan dan probabilistik untuk secara otomatis mendapatkan peraturan simbolis dari sebuah korpus.

Dalam penelitian ini, telah digunakan POS Tagger yang pernah dibuat oleh Wicaksono[1] sebagai landasan dasar penentuan tag kata dalam suatu kalimat.

Natural Language Toolkit

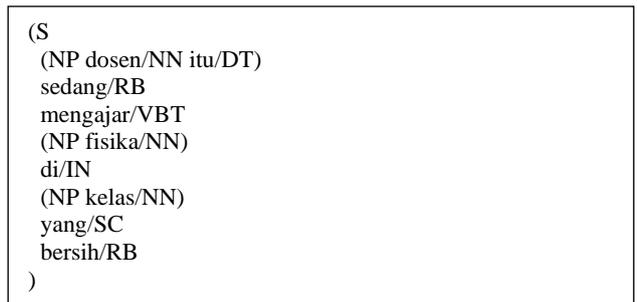
Natural Language Toolkit (NLTK) merupakan salah satu platform yang digunakan untuk membangun program bahasa manusia. NLTK digunakan untuk analisis teks. NLTK menggunakan bahasa pemrograman python, dan digunakan sebagai tools yang baik untuk pengembangan aplikasi NLP (Natural Language Processing)[4]. NLTK memberikan berbagai macam fitur yang dapat menangani pemrosesan bahasa alami. salah satu fiturnya adalah Regular Expression Parser.

Regular Expression Parser atau disebut juga RegexpParser atau RegexChunkParser bekerja dengan mendefinisikan aturan untuk mengelompokkan kata-kata yang berbeda secara bersamaan. Berikut akan diberikan sebuah contoh sederhana dalam menggunakan Regular Expression Parser.

Segmen Program 1 Penggunaan RegexpParser

```
01: kalimat_postagging ← "dosen/NN itu/DT
    sedang/RB mengajar/VBT fisika/NN di/IN
    kelas/NN yang/SC bersih/RB ./."
02: sentence ← [nltk.tag.str2tuple(t) for t in
    kalimat_postagging.split()]
03: grammar ← r""NP: {<NN> (<DT>)?}""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

Dalam segmen program 1 diberikan contoh kalimat sederhana yang sudah dilakukan POS Tagging dan satu rule grammar yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa Noun Phrase (NP) dapat dibentuk jika ditemukan sebuah kata benda (NN) saja, atau NN yang disertai dengan determiner (DT) yang sifatnya optional yaitu boleh ada atau tidak. Dengan menggunakan rule grammar tersebut maka regexpParser akan memecah kalimat sesuai dengan rule yang telah dibuat. Hasil dari segmen program 1 tersebut dapat ditampilkan dalam bentuk teks atau grafik(tree) yang dapat dilihat pada gambar di bawah ini.



Gambar 1. Hasil Berbentuk Teks



Gambar 2. Hasil Berbentuk Tree

Frasa

Frasa merupakan satuan gramatik yang terdiri atas satu kata atau lebih yang tidak melebihi batas fungsi unsur klausa[5]. Batas fungsi unsur klausa ada S, P, O, Pel, dan K. Berikut beberapa contoh frasa: para guru, dosen itu, saya, kedua orang itu, sedang mengajar, di kelas.

Berdasarkan persamaan distribusi dengan golongan atau kategori kata yang menjadi intinya, frasa dapat digolongkan menjadi enam golongan, yaitu frasa nomina, frasa verba, frasa numeralia, frasa keterangan(adverbia), frasa preposisional dan frasa ajektiva.

Frasa Nomina

Frasa nomina ialah frasa yang memiliki inti berupa nomina atau kata benda. Contoh frasa nomina: Anis membeli roti bakar. Frasa roti bakar dalam kalimat di atas mempunyai inti yang berupa

nomina, yaitu roti. Kata roti termasuk nomina, karena itu frasa roti bakar termasuk golongan frasa nomina.

Frasa Verba

Frasa verba adalah frasa yang mempunyai inti berupa verba. Contoh frasa verba : Pak guru sedang mengajar fisika. Frasa sedang mengajar mempunyai inti berupa verba yaitu mengajar. Frasa mengajar termasuk golongan frasa verba, karena itu frasa sedang mengajar termasuk golongan frasa verba.

Frasa Adjektiva

Frasa adjektiva adalah frasa yang mempunyai inti berupa adjektiva sebagai Unsur Pusat, misalnya frasa sangat senang yang mempunyai unsur inti senang sebagai adjektiva dan sangat sebagai atribut.

Frasa Adverbia

Frasa Adverbia adalah frasa yang mempunyai distribusi yang sama dengan kata keterangan, ialah kata yang mempunyai kecenderungan menduduki fungsi K dalam klausa. Contoh frasa adverbial : kemarin pagi, tadi malam.

Frasa Preposisional

Frasa preposisional adalah frasa yang diawali oleh preposisi sebagai penanda dan diikuti oleh kata atau frasa kategori nomina, numeralia, atau keterangan sebagai petanda atau aksisnya. Contoh frasa preposisional : di rumah, dari Jakarta.

Frasa Numeralia

Frasa numeralia adalah frasa yang mempunyai inti berupa numeralia sebagai Unsur Pusat, misalnya frasa tiga buah dalam tiga buah kursi yang mempunyai unsur inti tiga sebagai numeralia dan buah kursi sebagai atribut.

Klausa

Klausa adalah satuan gramatik yang terdiri atas unsur S dan P. Unsur S dan P tersebut dapat disertai objek (O), pelengkap (Pel), dan keterangan (K) ataupun tidak[5].

Klausa terdiri atas unsur-unsur fungsional yang disebut S, P, O, Pel, dan K. Kelima unsur itu memang tidak selalu ada dalam satu klausa. Kadang-kadang satu klausa hanya terdiri atas S dan P, kadang-kadang terdiri atas S, P, dan O, kadang-kadang terdiri atas S, P, dan Pel, kadang-kadang terdiri atas S, P, dan K, kadang-kadang terdiri atas S, P, O, dan K, kadang-kadang terdiri atas S, P, Pel, dan K.

Subjek (S) dan Predikat (P)

Pada umumnya S terdiri dari kata atau frasa golongan kata benda (N), sedangkan P dalam kalimat paling banyak terdiri dari kata atau frasa golongan kata kerja dan yang berpredikat bukan kata kerja sangat sedikit sekali, biasanya berupa frasa adjektiva atau frasa numeralia atau frasa preposisional .

Contoh :

- (a). Ibu memasak
- (b). Wanita itu cantik

Pada contoh (a) S terdiri dari kata benda yaitu ibu, sedangkan P terdiri dari kata kerja yaitu memasak. Pada contoh (b) S terdiri dari kata benda yaitu wanita dan kata penunjuk yaitu itu ,

sedangkan P terdiri dari kata sifat yaitu cantik. S biasanya berada sebelum P dalam sebuah kalimat.

Objek (O) dan Pelengkap (P)

P mungkin terdiri atas golongan verba transitif, mungkin terdiri atas golongan verba intransitif, dan mungkin pula terdiri atas golongan-golongan yang lain. Apabila terdiri atas golongan verba transitif, diperlukan adanya O yang mengikuti P itu, misalnya:

- (a) Pemerintah akan menyelenggarakan pesta seni. Klausa (a) terdiri atas tiga unsur fungsional, yaitu pemerintah sebagai S, unsur akan menyelenggarakan sebagai P, dan unsur pesta seni sebagai O. Pada umumnya O terdiri dari kata benda dan letaknya setelah P dalam sebuah kalimat.
- (b). Ibu memberikan adik sebuah buku. Adapun Pelengkap dalam contoh klausa (b) adalah sebuah buku.

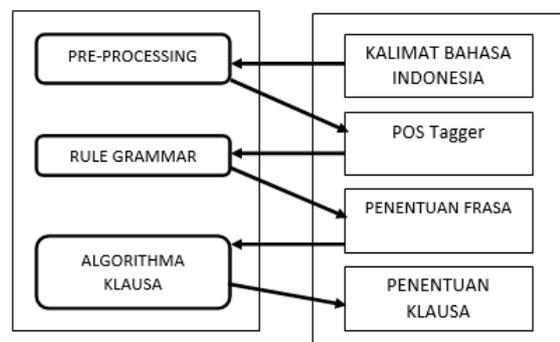
Keterangan (K)

Unsur klausa yang tidak menduduki fungsi S, P, O, dan Pel dapat diperkirakan menduduki K. Berbeda dengan O dan Pel yang selalu terletak di belakang P, dalam suatu klausa K pada umumnya mempunyai letak yang bebas, artinya dapat terletak di depan S-P, dapat terletak di antara S dan P, dan dapat juga terletak di paling belakang. Hanya sudah tentu tidak mungkin terletak di antara P dan O dan di antara P dan Pel, karena O dan PEL boleh dikatakan selalu menduduki tempat langsung di belakang P, setidaknya-tidaknya mempunyai kecenderungan demikian. Contoh:

- (a). Roney membersihkan kacamatanya dengan kain. Unsur yang menduduki fungsi K adalah unsur dengan kain yang terletak di belakang sekali. Unsur tersebut dapat dipindahkan ke depan S-P dan ke tempat di antara S dan P, menjadi:
 - (a1). Dengan kain, roney membersihkan kacamatanya
 - (a2). Roney, dengan kain membersihkan kacamatanya

METODE PENELITIAN

Penelitian ini merupakan penelitian dengan jangka waktu pelaksanaan kurang lebih 6 bulan. Adapun diagram alir dan uraian tahapan penelitian ini adalah sebagai berikut:



Gambar 3. Diagram alir penelitian

Untuk dapat menentukan kategori dan fungsi sintaksis kalimat Bahasa Indonesia diperlukan beberapa langkah yang harus dilakukan.

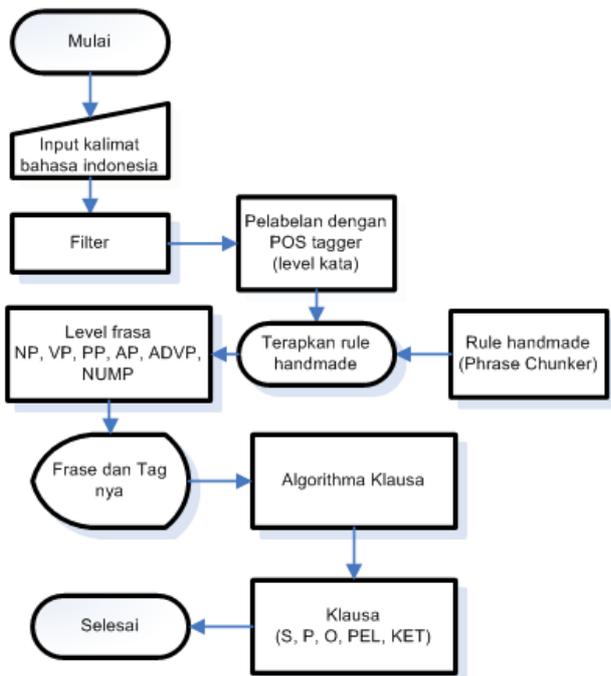
Langkah pertama adalah melakukan preprocessing dengan melakukan filter terhadap kalimat input. hal ini perlu dilakukan agar kalimat input dapat di proses oleh POSTagger.

Langkah kedua adalah memanfaatkan POSTagger Bahasa Indonesia hasil penelitian wicaksono[1] yang mampu memberikan tag pada setiap kata dari kalimat input. Output yang dihasilkan oleh POSTagger tersebut masih berupa level kata.

Langkah ketiga adalah merubah level kata hasil dari POSTagger menjadi 6 level frasa yaitu NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), AP (Adjective Phrase), ADVP (Adverbial Phrase) dan NumP (Numerial Phrase) berdasarkan pada rule handmade.

Langkah keempat adalah menentukan kedudukan fungsi sintaksis dari 6 level frasa terhadap kalimat menjadi subjek, predikat, objek, keterangan dan pelengkap.

Adapun kerangka penelitian dapat dilihat pada gambar di bawah ini. yang masing-masing bagian akan dijelaskan lebih lanjut dalam blok diagram.



Gambar 4. Blok Diagram Kerangka Penelitian

Filter dan POSTagger

Langkah pertama yang dilakukan adalah filter terhadap kalimat input. hal ini dilakukan agar kalimat input dapat diproses oleh POSTagger seperti pemisahan setiap kata ataupun tanda baca yang terdapat dalam kalimat. Kalimat input yang digunakan bisa berasal dari buku atau portal berita yang menggunakan kalimat efektif atau bahasa baku.

Langkah kedua adalah memanfaatkan POSTagger Bahasa Indonesia yang mampu memberikan tag pada setiap kata dari kalimat input. Output yang dihasilkan oleh POSTagger tersebut masih berupa level kata. Adapun blok diagram proses filter dan POSTagger dapat dilihat pada gambar dibawah ini.

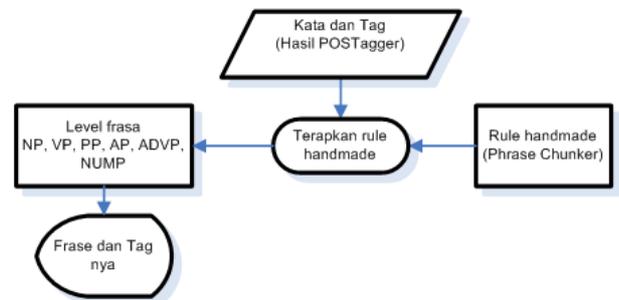


Gambar 5. Blok Diagram Filter dan POSTagger

Pembentukan Frasa

Pada langkah ketiga, phrase chunker akan merubah level kata hasil dari POSTagger menjadi 6 level frasa yaitu NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), AP (Adjective Phrase), ADVP (Adverbial Phrase) dan Numerial Phrase. Phrasa chunker yang digunakan berasal dari salah satu fungsi yang disediakan oleh NLTK yaitu regexParser dengan membuat rule handmade.

Adapun pembentukan Phrase chunker selengkapnya dapat dilihat pada gambar 6. dibawah ini.



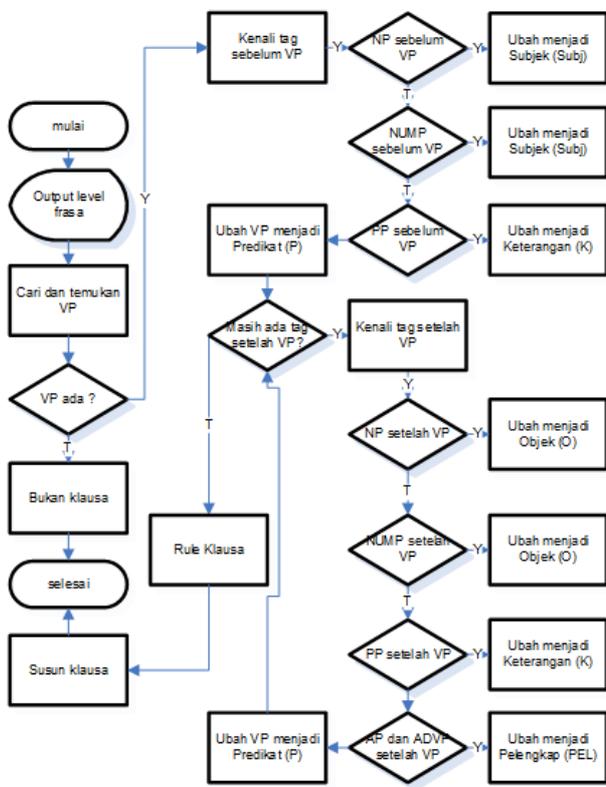
Gambar 6 Blok Diagram Pembentukan Frasa

Pembentukan Klausa

Klausa dibentuk berdasarkan dari hasil pembentukan frasa. Sehingga langkah selanjutnya yang dilakukan adalah mengolah output level frasa untuk menentukan kedudukan fungsi sintaksis dari 6 level frasa terhadap kalimat menjadi subjek (Subj), predikat (P), objek (O), pelengkap (PEL) dan keterangan (K).

Langkah pertama yang dilakukan adalah mencari verb phrase dari input level frasa, jika kalimat mengandung verb phrase, maka kenali tag sebelum verb phrase, ubah noun phrase atau numeralia phrase sebelum verb phrase menjadi subjek, jika ada preposition phrase sebelum verb phrase, ubah preposition phrase menjadi keterangan, ubah verb phrase menjadi predikat. Jika setelah verb phrase masih ada tag, maka kenali tag setelah verb phrase, ubah noun phrase atau numeralia phrase setelah verb phrase menjadi objek, jika ada preposition phrase, ubah menjadi keterangan, jika

ada adjective phrase dan adverbial phrase ubah menjadi pelengkap. Jika verb phrase sudah tidak ditemukan lagi, susun klausa berdasarkan rule klausa yang telah dibuat. Adapun pembentukan klausa selengkapnya dapat dilihat pada gambar blok diagram pembentukan klausa dibawah ini.



Gambar 7. Blok Diagram Algoritma Pembentukan Klausa

HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas tentang proses pembentukan rule handmade dari kalimat hasil POSTagging menjadi enam level frasa yaitu Noun Phrase, Verb Phrase, Preposition Phrase, Adverb Phrase, Adjective Phrase dan Numerial Phrase, serta Proses pembentukan algoritma dan rule handmade klausa kalimat Bahasa Indonesia. Di dalam setiap pembahasan akan diberi contoh dan output hasil setiap proses.

Rule Pembentukan Frasa

Rule mengadopsi dari regexParser seperti yang terdapat dalam Natural Language Toolkit (NLTK) untuk pencocokan kalimat Bahasa Indonesia menggunakan pola regular ekspresi (regex) yang disebut dengan phrase chunker. Phrase chunker akan mengekstrak level kata hasil POSTagger ke dalam 6 level frasa yaitu NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), AP (Adjective Phrase), ADVP (Adverb Phrase) dan NUMP (Numerial Phrase).

Noun Phrase Chunking

Untuk membuat Noun Phrase Chunking atau NP-Chunking maka langkah pertama yang harus dilakukan adalah mendefinisikan chunk grammar yang terdiri dari rule yang mengindikasikan

bagaimana kalimat seharusnya di pisah sesuai dengan fungsinya. Untuk lebih jelasnya akan diberikan contoh grammar sederhana sebagai berikut.

Segmen Program 2 NP-Chunker

```
01: kalimat_postagging ← "dosen/NN itu/DT
    sedang/RB mengajar/VBT fisika/NN di/IN
    kelas/NN yang/SC bersih/RB ./."
02: sentence ← [nltk.tag.str2tuple(t) for t in
    kalimat_postagging.split()]
03: grammar ← r"""NP: {<NN>(<DT>)?}"""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

Dalam segmen program 2 diberikan contoh kalimat sederhana yang sudah dilakukan POSTagging dan satu rule handmade yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa NP dapat dibentuk jika ditemukan sebuah kata benda (NN) saja, atau NN yang disertai dengan determiner (DT) yang sifatnya optional yaitu boleh ada atau tidak. Dengan menggunakan rule handmade tersebut maka regexParser akan memecah kalimat sesuai dengan rule handmade yang telah dibuat. Hasil dari segmen program 2 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks atau grafik yang dapat dilihat pada gambar 8 dan 9. Noun Phrase (NP) dari kalimat input diatas adalah frasa Dosen itu, fisika serta kelas

Verb Phrase Chunking

Berikut contoh grammar Untuk membuat Verb Phrase Chunking atau VP-Chunking.

Segmen Program 3 VP-Chunker

```
01: kalimat_postagging ← "dosen/NN itu/DT
    sedang/RB mengajar/VBT fisika/NN di/IN
    kelas/NN yang/SC bersih/RB ./."
02: sentence ← [nltk.tag.str2tuple(t) for t in
    kalimat_postagging.split()]
03: grammar ← r"""
    NP: {<NN>(<DT>)?}
    VP: {( <RB> )? <VB.*>}
    """
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

Dalam segmen program 3 diberikan contoh kalimat sederhana lanjutan dari contoh 2 sebelumnya dan tambahan satu rule handmade yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa VP dapat dibentuk jika ditemukan sebuah kata kerja (VBT/VBI), atau kata kerja yang disertai dengan adverb(RB) yang sifatnya optional yaitu boleh ada atau tidak. Dengan menggunakan rule handmade tersebut maka regexParser akan memecah kalimat sesuai dengan rule handmade yang telah dibuat. Hasil dari segmen program 5.2 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks atau grafik.

```
(S
(NP dosen/NN itu/DT)
(VP sedang/RB mengajar/VBT)
(NP fisika/NN)
di/IN
(NP kelas/NN)
yang/SC bersih/RB)
```

Gambar 8 Hasil VP-Chunking Berbentuk Teks



Gambar 9 Hasil VP-Chunking Berbentuk Tree

Berdasarkan pada gambar 8 dan 9, Verb Phrase (VP) dari kalimat input diatas adalah frasa sedang mengajar. Adapun kata yang lain dibiarkan apa adanya karena belum adanya rule handmade yang menanganinya.

Preposition Phrase Chunking

Berikut contoh grammar Untuk membuat Preposition Phrase Chunking atau PP-Chunking.

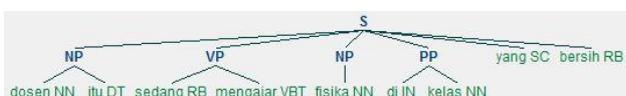
Segmen Program 4 Contoh Program PP-Chunker

```
01: kalimat_postagging="dosen/NN itu/DT
sedang/RB mengajar/VBT fisika/NN di/IN
kelas/NN yang/SC bersih/RB ./."
01: sentence ← [nlk.tag.str2tuple(t) for t in
kalimat_postagging. split()]
02: grammar ← r"""
PP: {<IN><NN>}
NP: {<NN>(<DT>)?}
VP: {(<RB>)?<VB.*>}
"""
03: cp ← nltk.RegexpParser(grammar)
04: result ← cp.parse(sentence)
05: print(result)
06: result.draw()
```

Dalam segmen program 4 ditambahkan satu rule handmade yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa PP dapat dibentuk jika ditemukan sebuah preposisi(IN) dan diikuti kata benda(NN). Hasil dari segmen program 4 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks atau grafik.

```
(S
(NP dosen/NN itu/DT)
(VP sedang/RB mengajar/VBT)
(NP fisika/NN)
(PP di/IN kelas/NN)
yang/SC
bersih/RB)
```

Gambar 10 Hasil PP-Chunking Berbentuk Teks



Gambar 11 Hasil PP-Chunking Berbentuk Tree

Berdasarkan pada gambar 10 dan 11, Preposition Phrase (PP) dari kalimat input diatas adalah frasa di kelas. Adapun kata yang lain

dibiarkan apa adanya sesuai dengan penanganan rule handmade yang dibuat.

Perlu dijelaskan disini bahwa urutan dari rule handmade yang dibuat perlu diletakkan pada posisi yang tepat karena akan menghasilkan output yang berbeda. Misalnya jika segmen program 4 dirubah menjadi seperti segmen program 5 dibawah ini.

Segmen Program 5 PP-Chunker Posisi Rule yang Berbeda

```
01: kalimat_postagging ← "dosen/NN itu/DT
sedang/RB mengajar/VBT fisika/NN di/IN
kelas/NN yang/SC bersih/RB ./."
02: sentence ← [nlk.tag.str2tuple(t) for t in
kalimat_postagging. split()]
03: grammar ← r"""
NP: {<NN>(<DT>)?}
VP: {(<RB>)?<VB.*>}
PP: {<IN><NN>}
"""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

maka akan di hasilkan output yang salah sebagai berikut:

```
S
(NP dosen/NN itu/DT)
(VP sedang/RB mengajar/VBT)
(NP fisika/NN)
di/IN
(NP kelas/NN)
yang/SC
bersih/RB)
```

Gambar 12 Hasil Output Salah Berbentuk Teks



Gambar 13 Hasil Output Salah Berbentuk Tree

Dalam hasil gambar 12 dan 13, tidak ditemukannya PP dalam kalimat, itu disebabkan posisi rule handmade PP terletak setelah rule handmade NP, sehingga setiap ada NN dari kalimat input akan di proses menjadi NP terlebih dahulu.

Adverb Phrase Chunking

Dengan menggunakan langkah yang sama, berikut contoh segmen program untuk menentukan Adverb Phrase.

Segmen Program 6 ADVP-Chunker

```
01: kalimat_postagging ← "dosen/NN itu/DT
sedang/RB mengajar/VBT fisika/NN di/IN
kelas/NN yang/SC bersih/RB ./."
02: sentence ← [nlk.tag.str2tuple(t) for t in
kalimat_postagging. split()]
03: grammar ← r"""
PP: {<IN><NN>}
NP: {<NN>(<DT>)?}
VP: {(<RB>)?<VB.*>}
ADVP: {<SC><RB>}
"""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
```

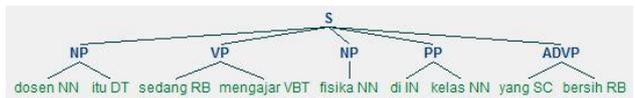
```
06: print(result)
07: result.draw()
```

Dalam segmen program 6 ditambahkan satu rule handmade yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa Adverb Phrase (ADVP) dapat dibentuk jika ditemukan sebuah subor-conjunction (SC) dan diikuti oleh adverb (RB). Hasil dari segmen program 6 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks seperti yang terlihat pada gambar 14 dan grafik yang dapat dilihat pada gambar 15 dibawah ini.

Berdasarkan pada gambar 14 dan 15, Adverb Phrase (ADVP) dari kalimat input diatas adalah frasa yang bersih.

```
(S
(NP dosen/NN itu/DT)
(VP sedang/RB mengajar/VBT)
(NP fisika/NN)
(PP di/IN kelas/NN)
(ADVP yang/SC bersih/RB)
)
```

Gambar 14 Hasil ADVP-Chunking Berbentuk Teks



Gambar 15 Hasil ADVP-Chunking Berbentuk Tree

Adjective Phrase Chunking

Dengan cara yang sama namun menggunakan contoh kalimat yang berbeda maka untuk membuat Adjective Phrase Chunking atau AP-Chunking sebagai berikut.

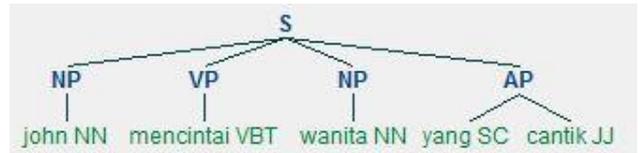
Segmen Program 7 AP-Chunker

```
01: kalimat_postagging ← "john/NN mencintai/VBT
wanita/NN yang/SC cantik/JJ"
02: sentence ← [nlk.tag.str2tuple(t) for t in
kalimat_postagging.split()]
03: grammar ← r"""
PP: {<IN><NN>}
NP: {<NN>(<DT>)?}
VP: {(<RB>)?<VB.*>}
ADVP: {<SC><RB>}
AP : {<SC><JJ>}
"""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

Dalam segmen program 7 ditambahkan satu rule handmade yang terletak pada baris ke 3. Rule tersebut menjelaskan bahwa AP dapat dibentuk jika ditemukan sebuah subor-conjunction(SC) dan diikuti oleh Adjective(JJ). Hasil dari segmen program 7 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks atau grafik.

```
(S
(NP john/NN)
(VP mencintai/VBT)
(NP wanita/NN)
(AP yang/SC cantik/JJ)
)
```

Gambar 16 Hasil AP-Chunking Berbentuk Teks



Gambar 17 Hasil AP-Chunking Berbentuk Tree

Numerial Phrase Chunking

Dengan cara yang sama namun menggunakan contoh kalimat yang berbeda maka untuk membuat Numerial Phrase Chunking atau NUMP-Chunking sebagai berikut.

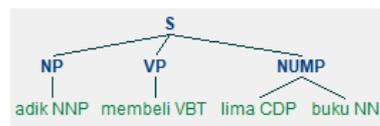
Segmen Program 8 NUMP-Chunker

```
01: kalimat_postagging ← " adik/NNP membeli/VBT
lima/CDP buku/NN "
02: sentence ← [nlk.tag.str2tuple(t) for t in
kalimat_postagging.split()]
03: grammar ← r"""
NP: {<NNP>}
VP: {<VB.*>}
NUMP: {<CDP><NN>}
"""
04: cp ← nltk.RegexpParser(grammar)
05: result ← cp.parse(sentence)
06: print(result)
07: result.draw()
```

Dalam segmen program 8 baris ke 3, terdapat rule handmade yang menjelaskan bahwa NUMP dapat dibentuk jika ditemukan sebuah Primary Numerals (CDP) dan diikuti oleh Common Noun(NN). Hasil dari segmen program 8 tersebut berupa tree yang dapat ditampilkan dalam bentuk teks atau grafik.

```
(S
(NP adik/NNP)
(VP membeli/VBT)
(NUMP lima/CDP buku/NN)
)
```

Gambar 18 Hasil NUMP-Chunking Berbentuk Teks



Gambar 19 Hasil NUMP-Chunking Berbentuk Tree

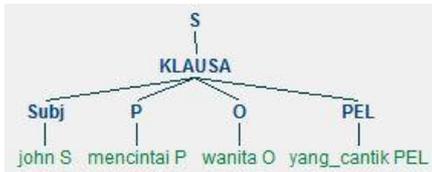
Berdasarkan pada gambar 18 dan 19, Numerial Phrase (NUMP) dari kalimat input tersebut adalah frasa lima buku.

Rule Pembentukan Klausa

Dengan menggunakan contoh pada segmen program 7 maka jika diberikan input kalimat "john/NN mencintai/VBT wanita/NN yang/SC cantik/JJ" akan dihasilkan klausa berdasarkan pada penjelasan gambar 7 Blok Diagram Algoritma Pembentukan Klausa sebagai berikut.

```
(S
(KLAUSA
(Subj john/S)
(P mencintai/P)
(O wanita/O)
(PEL yang_cantik/PEL)
)
)
```

Gambar 20 Hasil Klausa Berbentuk Teks



Gambar 21 Hasil Klausa Berbentuk Tree

Dari gambar 20 dan 21 dapat dijelaskan bahwa frasa john tidak lagi menjadi NP melainkan sudah menjadi Subjek(Subj), frasa mencintai berubah dari VP menjadi P, frasa wanita berubah dari NP menjadi O dan frasa yang cantik berubah dari AP menjadi PEL.

Adapun pembentukan rule handmade klausa dapat dilihat pada segmen contoh program dibawah ini.

Segmen Program 9 Contoh Grammar Klausa

```
01: grammar_klausa ← r""
KLAUSA:
{ (<K|PEL>+)?<Subj>(<K|PEL>+)?<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?<P>(<O|PEL|K>+)?<P>(<O|PEL|K>+)?
}&}
KLAUSA:
{ (<K|PEL>+)?<Subj>(<K|PEL>+)?<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?<P>(<O|PEL|K>+)?
}
KLAUSA:
{ (<K|PEL>+)?<Subj>(<K|PEL>+)?<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?
<P>(<O|PEL|K>+)?
}
KLAUSA:
{ (<K|PEL>+)?<Subj>(<K|PEL>+)?<P>(<O|PEL|K>+)?
}
KLAUSA:
{ (<K|PEL>+)?<Subj>(<K|PEL>+)?<P>(<O|PEL|K>+)?
}
.....
```

Dari segmen program 9 dapat dijelaskan bahwa klausa minimal harus terdiri dari Subjek (Subj) dan Predikat (P). Adapun Objek (O), Pelengkap (PEL) dan Keterangan (K) sifatnya optional, boleh ada atau tidak. Subj dalam satu klausa maksimal hanya ada satu, sedangkan P, O, PEL dan K dalam satu klausa boleh lebih dari satu. Dalam penelitian ini, jumlah Predikat (P) dalam kalimat hanya dibatasi maksimal hingga lima predikat.

Hasil Evaluasi

Algoritma dalam penelitian ini diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan Natural Language Toolkit (NLTK). Algoritma ini diujikan pada kalimat Bahasa Indonesia dengan bantuan rule handmade yang mengadopsi dari NLTK ke dalam 6 level frasa yaitu NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), AP (Adjective Phrase), ADVP (Adverb Phrase) dan NUMP (Numerial Phrase). Setelah itu

menentukan kedudukan fungsi sintaksis dari 6 level frasa menjadi subjek, predikat, objek, keterangan dan pelengkap.

Metode evaluasi yang digunakan dalam penelitian ini adalah metode akurasi [9], yakni mencocokkan antara hasil dari pakar atau buku dengan hasil program mengenai susunan hasil frasa dan klausa seperti pada Persamaan (1).

$$Accuracy = \frac{(\text{sum}(\text{program benar} / \text{pakar benar}) / n \text{ Kalimat}) * 100 \% (1)}$$

hasil perhitungan akurasi frasa, didapatkan jumlah benar sebesar 466,6 dari 500 kalimat yang terdiri dari 360 kalimat tunggal, 110 kalimat kompleks, 30 kalimat majemuk dan 5.068 kata, Akurasi frasa mencapai rata-rata sukses 93,32% dan Akurasi Klausa mencapai rata-rata sukses 91,12%.

Algoritma rule handmade untuk frasa Bahasa Indonesia ini bergantung pada hasil identifikasi dari POSTagger. Jika POSTagger tidak akurat mengidentifikasi input teks atau kalimat, maka akan mempengaruhi efisiensi dan akurasi dari rule handmade yang telah dibuat.

Pada awalnya, rule handmade yang sangat terbatas digunakan untuk memproses hasil POSTagger menjadi bentuk frasa. Oleh karena itu, akurasi yang diperoleh menjadi rendah, namun ketika jumlah rule di tambah, maka level akurasi dari setiap proses yaitu frasa dan klausa menjadi meningkat.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dapat diambil kesimpulan dan saran

Kesimpulan

Berdasarkan hasil uji coba aplikasi dapat disimpulkan bahwa aplikasi penentuan kategori dan fungsi sintaksis kalimat bahasa indonesia adalah sebagai berikut.

1. Pemrosesan hasil POSTagger kalimat Bahasa Indonesia yang berupa level kata di ubah menjadi 6 level frasa yaitu NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), AP (Adjective Phrase), ADVP (Adverb Phrase) dan NUMP (Numerial Phrase) dan diperoleh akurasi sebesar 93,32 % dengan jumlah benar sebesar 466,6.
2. Penentuan kedudukan fungsi sintaksis dari 6 level frasa terhadap kalimat Bahasa Indonesia di ubah menjadi Subjek, Predikat, Objek, Keterangan dan Pelengkap untuk penentuan klausa diperoleh akurasi sebesar 91,12% dengan jumlah benar sebesar 455,61.
3. Urutan dari rule handmade frasa yang dibuat harus diletakkan pada posisi yang tepat karena akan mempengaruhi hasil akurasi pembentukan frasa.

Saran

Adapun saran yang dapat diberikan adalah sebagai berikut

1. Pada penelitian selanjutnya dapat dilakukan pengembangan klausa keterangan seperti menjadi Keterangan Waktu (KW), Keterangan Tempat (KT), Keterangan Alat (KA), Keterangan Kualitas (KQ) dan Keterangan Kuantitas (KK).
2. Penelitian ini belum membedakan kalimat aktif dan pasif.

3. Algorithma penentuan klausa masih belum akurat dalam menentukan klausa pelengkap (PEL)

Ucapan Terima Kasih

Terima kasih kami ucapkan kepada Direktorat Riset dan Pengabdian kepada Masyarakat khususnya Direktorat Jenderal Penguatan Riset dan Pengembangan, yang telah memberikan dukungan finansial kepada kami sebagai peneliti sehingga kami dapat melaksanakan penelitian ini dengan baik. Terima kasih juga kami sampaikan kepada semua pihak yang telah membantu dalam pelaksanaan penelitian ini.

DAFTAR PUSTAKA

[1] A.F. Wicaksono and A. Purwarianti. HMM Based Part-of-Speech Tagger for Bahasa Indonesia. On Proceedings of 4th International MALINDO (Malay and Indonesian Language) Workshop, 2nd August 2010.

[2] F. Pisceldo, R. Manurung, and M. Adriani, "Probabilistic part-of-speech tagging for bahasa indonesia," in Third International MALINDO Workshop, 2009.

[3] A.A.Armana, A.B.Putra, A. Purwarianti, and Kuspriyanto. "Syntactic Phrase Chunking for Indonesian Language" in The 4th International Conference on Electrical Engineering and Informatics (ICEEI), 2013.

[4] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

[5] Supriyadi, Dr. M.Pd. Sintaksis Bahasa Indonesia. UNG Press. Gorontalo. 2014.

[6] Ramlan. M., Ilmu Bahasa Indonesia : Sintaksis, CV. Karyono, Yogyakarta. 2005.

[7] A. Chaer. Sintaksis Bahasa Indonesia (Pendekatan Proses). Rineka Cipta. 2009.

[8] I. B. Putrayasa. Analisis Kalimat (Fungsi, Kategori dan Peran). Refika Aditama. 2006.

[9] A. Ali and M. A. Khan, " Knowledge Representation of Urdu Text Using Predicate Logic", 2010 6th International Conference on Emerging Technologies (ICET).

[10] Diaz, F.C., van Santen, J., Banga, E.R., Integrating phrasing and intonation modelling using syntactic and morph syntactic information. In Speech Communication , 2009, vol 51, pp. 452-465.

[11] Brill, E."A simple rule-based part-of-speech taggCher". In: Proceedings of the Third Conference on Applied Natural Language Processing (ANLP '92), Trento, Italy (1992) 152–155.

	Frasa	NP=dia, VP=akan_pergi, PP=ke_eropa
	Klausa	S=dia, P=akan_pergi, KET=ke_eropa
4	Kalimat	Adik bermain bola di lapangan
	Frasa	NP=Adik, VP=bermain, NP=bola, PP=di_lapangan
	Klausa	S=Adik, P=bermain, O=bola, KET=di_lapangan
5	Kalimat	Budi menyanyi merdu
	Frasa	NP=Budi, VP=menyanyi, AP=merdu
	Klausa	S=Budi, P=menyanyi, PEL=merdu
6	Kalimat	di area pegunungan, kita tidak memiliki jembatan
	Frasa	PP=di_area_pegunungan, NP=kita, VP=tidak_memiliki, NP=jembatan
	Klausa	KET=di_area_pegunungan, S=kita, P=tidak_memiliki, O=jembatan
7	Kalimat	semua anak-anak dan guru di sekolah berkumpul di aula .
	Frasa	NP=semua_anak-anak_dan_guru, PP=di_sekolah, VP=berkumpul, PP=di_aula
	Klausa	S=semua_anak-anak_dan_guru, KET=di_sekolah, P=berkumpul, KET=di_aula
8	Kalimat	kita mendapatkan air untuk menyiram tembakau kita
	Frasa	NP=kita, VP=mendapatkan, NP=air, VP=untuk_menyiram, NP=tembakau_kita
	Klausa	S=kita, P=mendapatkan, O=air, P=untuk_menyiram, O=tembakau_kita
9	Kalimat	si anton telah membantu sang paman menulis surat
	Frasa	NP=si_anton, VP=telah_membantu, NP=sang_paman, VP=menulis, NP=surat
	Klausa	S=si_anton, P=telah_membantu, O=sang_paman, P=menulis, O=surat
10	Kalimat	Adik sedang makan atau dia sedang minum
	Frasa	NP=Adik, VP=sedang_makan, CC=atau, NP=dia, VP=sedang_minum
	Klausa	S=Adik, P=sedang_makan, CC=atau, S=dia, P=sedang_minum

LAMPIRAN

Table 1. Lampiran Beberapa Hasil Penelitian

No.	Jenis	Hasil Uji Coba
1	Kalimat	Dosen Mengajar
	Frasa	NP=Dosen, VP=mengajar
	Klausa	S=Dosen, P=mengajar
2	Kalimat	Ayah meminum kopi
	Frasa	NP=Ayah, VP=meminum, NP=kopi
	Klausa	S=Ayah, P=meminum, O=kopi
3	Kalimat	dia akan pergi ke eropa