

Penentuan Makna Kata dari Frase dalam Kalimat Bahasa Inggris

Jeany Harmoejanto
Sekolah Tinggi Teknik Surabaya
E-mail: ajeany@yahoo.com

Abstrak

Tokenisasi merupakan proses memecah kalimat menjadi kata, frase atau bentuk lain yang memiliki arti, hasil tokenisasi disebut sebagai token. Tokenisasi adalah langkah prapemrosesan *Word Sense Disambiguation* (WSD), proses penentuan makna suatu kata. Istilah "word" dalam WSD, dapat diartikan sebagai kata tunggal maupun rangkaian lebih dari satu kata yang disebut frase. Kategori frase sendiri dapat dibedakan berdasarkan maknanya, yakni frase yang memiliki makna baru, terlepas dari kata-kata penyusunnya (contoh: *take off*) dan yang tidak memiliki makna baru (contoh: *this morning*). Saat sebuah frase diartikan secara terpisah berdasarkan unsur kata yang membentuknya, makna yang dihasilkan menjadi kurang tepat dibandingkan jika frase tersebut diartikan secara satu kesatuan. Dalam penelitian ini, akan dilakukan penentuan frase pada kalimat berbahasa Inggris, di mana frase tersebut memiliki makna baru. Masukan sistem adalah sebuah kalimat, dan keluaran yang dihasilkan adalah daftar frase yang terdapat dalam kalimat tersebut. Langkah yang dilakukan adalah melakukan *tagging* pada kalimat dengan *Stanford PoStagger*, kemudian tokenisasi untuk menghasilkan kumpulan kata. Langkah utama penentuan frase adalah melakukan kombinasi kata, hasil dari tokenisasi, untuk menemukan frase dengan bantuan kamus wordnet. Kombinasi kata dilakukan dengan kombinasi *bubble* dan *insertion*. Uji coba dilakukan dengan membandingkan hasil dari penentuan frase yang dihasilkan oleh aplikasi yang dibuat, dibandingkan dengan validasi yang dilakukan oleh beberapa ahli tata Bahasa Inggris, terhadap 65 kalimat berbahasa Inggris. Hasil uji coba menunjukkan aplikasi mampu menentukan frase dalam kalimat dengan *precision* dan *recall* mencapai 85.14% dan 96,92%.

Kata kunci: *word sense disambiguation, frase, insertion, bubble, wordnet.*

Abstract

Tokenization is a process to get list of words, phrases or other forms needed from a sentence. The result of tokenization process usually called token. Tokenization is one of preprocessing method in Word Sense Disambiguation (WSD), process to determine meaning or sense of a word. Term "word" in WSD, can be described as single word or multiple word called phrase. Phrase can be determined by its meaning, which are a phrase that can create a new meaning from different from each word's meaning (eg: take off) and a phrase that still have the same meaning with each words (eg: this morning). When sense of a phrase is determine by sense of each words that construct that phrase, it will less accurate than if it is determined as a phrase. This research will try to find the phrase, that has new meaning, in an english sentence. Input of system is a sentence and it will generate list of phrases. The first step to find those phrases is sentence tagging using Stanford PoStagger, followed by tokenization process to generate words, the last step is to combine each words using Bubble combination or Insertion combination to find phrase based on wordnet. The experiment will compare between result generated by application validation that being done by few english lecturers, to 65 sentences. The results show that application able to generate phrase up to 85.14% precision dan 96,92% recall.

Key words: *word sense disambiguation, frase, insertion, bubble, wordnet*

Pendahuluan

Word Sense Disambiguation (WSD) merupakan salah satu topik yang mendapat perhatian dalam NLP, yang bertujuan untuk menentukan makna suatu kata. WSD banyak digunakan pada aplikasi seperti mesin penerjemah, pengekstraksi informasi/berita, mesin tanya jawab, perangkuman opini [1]. Apabila mencari arti kata dalam sebuah kamus maka akan ditemukan bahwa sebuah kata dapat mempunyai banyak makna seperti contoh kata *call* berarti *command to come*,

dapat juga berarti *a telephone connection*, bahkan dalam *wordnet* ditemukan ±30 makna *call*, di sinilah peranan WSD untuk memilih dari 30 makna *call*, manakah makna yang paling sesuai, dan hal tersebut dapat dilakukan pada saat kata dimasukkan dalam konteks kalimat.

Istilah "word" dalam WSD, dapat diartikan sebagai kata tunggal maupun rangkaian lebih dari satu kata yang disebut frase. Kategori frase sendiri dapat dibedakan berdasarkan maknanya, yakni frase yang memiliki

makna baru, terlepas dari kata-kata penyusunnya dan yang tidak memiliki makna baru. Saat sebuah frase diartikan secara terpisah berdasarkan unsur kata yang membentuknya, makna yang dihasilkan menjadi kurang tepat dibandingkan jika frase tersebut diartikan secara satu kesatuan. Contoh dalam kalimat *the plane took off just now*, untuk menentukan makna *take* dalam kalimat tersebut akan lebih tepat apabila diambil satu kesatuan frase *took off*.

Dalam penelitian ini, akan dilakukan penentuan frase dalam kalimat berbahasa Inggris, di mana frase tersebut memiliki makna baru. Proses penentuan frase tersebut akan dilakukan pada proses tokenisasi kalimat. Tokenisasi merupakan proses memecah kalimat menjadi kata, frase atau bentuk lain yang memiliki arti, hasil tokenisasi disebut token. Penelitian ini berusaha menghasilkan token berupa frase yang memiliki makna baru

Metode Penelitian

Secara sederhana hubungan antara kata, frase dan kalimat dalam bahasa Inggris dapat dijabarkan sebagai berikut, sebuah kata dapat berupa kata dasar (contoh: *the, plane, took, off*) kata turunan (contoh: *took* merupakan bentukan dari kata *take*) dan frase (contoh: *took off, the plane*), sedangkan kalimat adalah kumpulan dari kata yang mengisi jabatan minimal subjek (S) dan predikat (P) (contoh: *the plane* (S) *took off* (P)). Frase sendiri memiliki beberapa karakteristik antara lain [2]:

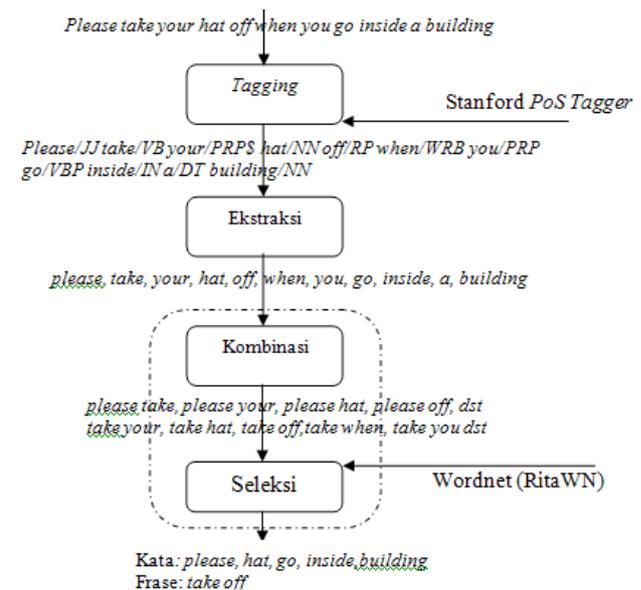
1. Pembentukan arti baru dari kata tersebut (*Two Words Verb/Phrasal Verb*)
Contoh pada kata *take*, arti umum *carry out*, namun saat bergabung dalam frase menjadi kata *take off* mempunyai arti *depart from the ground*.
Jika diperluas, pembentukan arti baru dari kumpulan kata dapat menghasilkan peribahasa (*idiom*), contoh: *false alarm* yang berarti *untrue rumor*.
2. Pengubahan jenis kata
Contoh pada kata *smile*, arti umum *express with a smile*, namun saat bergabung dalam frase menjadi kata *the smile* mempunyai arti *a facial expression characterized by turning up the corners of the mouth*. Kata *smile* tersebut mengalami perubahan jenis kata dari kata kerja menjadi kata benda
3. Tidak terjadi perubahan makna maupun jenis kata
Contoh pada frase *the moon* tetap memiliki makna yang sama dengan *moon*, yakni *the natural satellite*

of the Earth, pada frase *beautiful girl*, tetap memiliki arti yang sama dengan kata penyusunnya, karena kata *beautiful* hanya menjelaskan kata *girl*
Dalam penelitian ini akan menitikberatkan pada frase yang menimbulkan makna baru, namun bukan berupa idiom.

Proses yang dilakukan dalam penentuan frase dalam kalimat, terdiri dari beberapa tahap dapat dilihat pada Gambar 1, antara lain:

1. *Tagging* kalimat
2. Ekstraksi kalimat ber-*tag* menjadi kumpulan kata
3. Kombinasi kata menjadi frase
4. Seleksi frase dan kata

Untuk selanjutnya, contoh kalimat yang akan digunakan sebagai masukan selama penjelasan setiap prosesnya adalah *please take your hat off when you go inside a building*.



Gambar 1. Proses penentuan frase

Tagging kalimat

Proses *tagging* kalimat bertujuan untuk memberikan *tag* atau penanda pada setiap kata dalam kalimat yang diinputkan [3]. *Tagging* yang dilakukan menggunakan *Stanford PoS Tagger Library*. Masukan yang diberikan adalah kalimat, dapat berupa kalimat tunggal maupun kalimat majemuk, sedangkan keluaran yang dihasilkan adalah kalimat di mana masing-masing katanya telah diberi tanda (*tag*). Pada contoh kasus, akan menghasilkan keluaran “*Please/JJ take/VB your/PRP\$ hat/NN off/RP when/WRB you/PRP go/VBP inside/IN a/DT building/NN*”. *Tagging* tersebut dilakukan dengan tujuan untuk membantu mem-*filter* pencarian frase

Tabel 1. Keluaran proses ekstraksi frase

Isi	Tag	Flag	Gloss
<i>please</i>	<i>JJ</i>	0	(v) <i>give pleasure to or be pleasing to</i> (n) -- (a) -- (r) <i>used in polite request</i>
<i>take</i>	<i>VB</i>	0	(v) <i>carry out</i> (n) <i>the income or profit arising from such transactions</i> (a) -- (r) --
<i>your</i>	<i>PRPS</i>	0	(v) -- (n) -- (a) -- (r) --
<i>hat</i>	<i>NN</i>	0	(v) <i>put on or wear a ahat</i> (n) <i>Head dress that protects the head from bad weather</i> (a) -- (r) --
dst			

pada tahap proses kombinasi kata menjadi frase.

Ekstraksi Kalimat Ber-tag Menjadi Kumpulan Kata

Proses ekstraksi kalimat merupakan proses pemotongan kalimat ber-tag menjadi kumpulan kata ber-tag. Masukan yang diberikan adalah sebuah kalimat ber-tag, sedangkan keluaran yang dihasilkan adalah kumpulan kata. Pada contoh kasus, masukan yang diberikan adalah *please/JJ take/VB your/PRP\$ hat/NN off/RP when/WRB you/PRP go/VBP inside/IN a/DT building/NN*, sedangkan keluaran yang dihasilkan adalah 11 kata yakni *please, take, your, hat, off, when, you, go, inside, a, building*.

Proses ekstraksi tidak hanya menghasilkan potongan kata, tetapi juga mengambil makna dari kata tersebut pada *wordnet*, yang selanjutnya pada karya tulis ini akan dikenal dengan sebutan *gloss*. Seperti kita ketahui, *wordnet* terdapat 4 kategori kata yakni *verb(v)*, *noun(n)*, *adjective(a)*, *adverb(r)*, dimana tiap kata juga dapat memiliki >1 makna. Oleh karena itu pada proses ekstraksi ini hanya diambil makna yang pertama saja untuk tiap kategori kata. Contoh keluaran yang dihasilkan dapat dilihat pada Tabel 1.

Kombinasi Kata Menjadi Frase

Setelah proses ekstraksi kalimat yang menghasilkan 11 kata, proses berikutnya adalah melakukan kombinasi terhadap kata-kata tersebut. Seberapa jauh kombinasi dilakukan ditentukan oleh jarak kombinasi. Jarak minimal adalah 1 sedangkan jarak maksimal adalah $(N-1)$, di mana N adalah jumlah kata. Pada penelitian ini, proses kombinasi kata dilakukan dengan 2 (dua) metode yaitu kombinasi *insertion* dan kombinasi *bubble*. Penamaan tersebut diambil dari metode pengurutan data (*sorting*) dikarenakan langkah-langkah yang dilakukan hampir serupa, namun bukan berarti dalam proses kombinasi tersebut juga melakukan proses pengurutan data.

Pada Gambar 2, terlihat bahwa kombinasi *insertion* dilakukan dengan mencari semua kombinasi untuk $K1$ terlebih dahulu sebanyak jarak (J) yaitu 5, kemudian dilanjutkan ke $K2$, $K3$, dst. Dari $K1$, untuk jarak 5, menghasilkan kombinasi $K1-K2$, $K1-K3$, $K1-K4$, $K1-K5$ dan $K1-K6$, apabila jarak yang ditentukan hanya 2, maka hasilnya adalah $K1-K2$ dan $K1-K3$. Dengan demikian dapat disimpulkan bahwa:

1. Pasangan kombinasi dari K step adalah K ($step+1$) sampai dengan $(step+J)$, di mana $(step+J) \leq N$, $1 \leq step \leq N-1$

Tabel 2. Keluaran Proses Kombinasi Kata

Jenis Tag	Kata 2						
	<i>NN</i>	<i>VB</i>	<i>RB</i>	<i>JJ</i>	<i>IN</i>	<i>RP</i>	
Kata 1	<i>NN</i>	<i>church bell</i>	-	<i>day off</i>	-	<i>switch on</i>	-
	<i>VB</i>	<i>open door</i>	-	<i>doing well</i>	-	<i>break in</i>	<i>work out</i>
	<i>RB</i>	-	-	<i>back up</i>	-	-	-
	<i>JJ</i>	-	-	-	-	<i>more than</i>	-
	<i>IN</i>	<i>at home</i>	-	-	<i>as usual</i>	-	-
	<i>RP</i>	-	-	-	-	<i>up on</i>	-

2. Banyaknya langkah adalah $N-1$, karena kata terakhir tidak perlu dicari kombinasinya.
3. Setiap langkah, dalam hal ini setiap kata, dapat menghasilkan jumlah kombinasi maksimal sebanyak J dan minimal 1.
4. Kompleksitas waktu yang dihasilkan adalah $O(N^2)$, karena waktu yang diperlukan sama dengan algoritma *insertion sort* pada *worst case*

Pada proses kombinasi, perlu dilakukan pembatasan kata mana saja yang sebaiknya dikombinasikan dan mana yang tidak, karena tidak semua kata perlu dikombinasi. Hal tersebut ditentukan oleh jenis *tag* tiap kata yang dihasilkan pada proses *tagging* kalimat. Pada penelitian ini, kombinasi kata hanya akan dilakukan pada kategori *tag* tertentu saja, beberapa ketentuan yang ada yaitu [4]:

1. *tag* yang mengandung '*NN*' untuk mewakili *noun*, '*VB*' untuk mewakili *verb*, '*RB*' untuk mewakili *adverb*, '*JJ*' untuk mewakili *adjective* pada *wordnet*
2. *tag* '*RP*' yang mewakili partikel seperti *off*, *out*
3. *tag* '*IN*' yang mewakili proposisi seperti *about*, *from*

4. kata yang bukan termasuk kategori '*be*', seperti *be*, *is*, *was*, *were*, *are*. Dalam *Stanford PoS Tagger*, '*be*' termasuk dalam kategori '*VBD*' atau '*VBZ*', sehingga bertentangan dengan ketentuan pertama. Oleh karena itu kategori '*be*' ini dibuat pengecekan khusus dalam aplikasi yang dibuat.

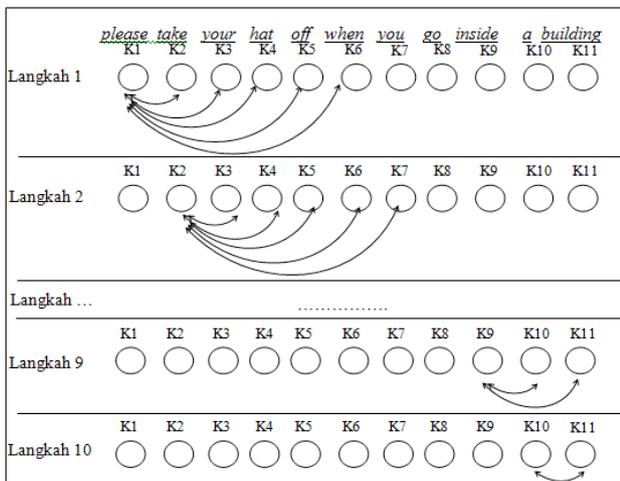
Pada Tabel 2, dapat dilihat contoh keluaran dari proses kombinasi *buble*.

Seleksi Frase dan Kata

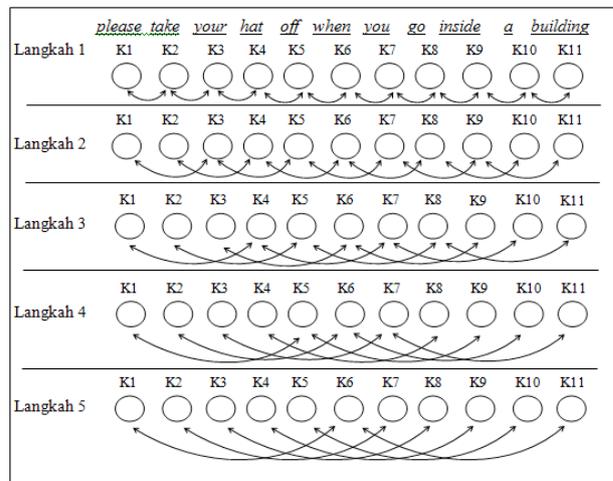
Proses seleksi adalah proses untuk menentukan apakah kombinasi kata yang dihasilkan termasuk frase yang menimbulkan makna baru pada *wordnet*. Masukan yang diperlukan adalah kombinasi kata sedangkan keluaran yang dihasilkan adalah daftar frase dan kata. Pada contoh kasus *please take your hat off when you go inside a building*, keluaran yang dihasilkan antara lain:

1. Daftar Frase: *take off*
2. Daftar Kata: *please, hat, go, building*

Sedangkan untuk kata *your, when, you, inside* dan *a* tidak terseleksi karena tidak termasuk dalam



Gambar 2. Kombinasi *insertion*



Gambar 3. Kombinasi *bubble*

kategori *noun*(n/NN), *verb*(v/VB), *adjective*(a/JJ), *adverb*(r/RB).

Dilihat dari jenis keluaran yang dihasilkan maka proses seleksi ada 2 tahap yaitu seleksi frase dan seleksi kata. Seleksi kata dijalankan apabila seleksi frase menyatakan bahwa kombinasi kata itu bukan frase. Pada proses seleksi frase, ketentuan yang perlu diperhatikan antara lain [5]:

1. Gloss pertama dari Kata 1 ≠ gloss pertama Kata 3, di mana gloss pertama Kata 3 ≠ kosong
2. Gloss pertama dari Kata 2 ≠ gloss pertama Kata 3, di mana gloss pertama Kata 3 ≠ kosong
3. Gloss pertama dari Kata 3 ≠ gloss pertama Kata 4, di mana gloss pertama Kata 3 dan 4 ≠ kosong
4. Jika ketentuan 1/2 tidak terpenuhi, maka hitung jumlah gloss dari Kata 1/2 dan Kata 3, hasilnya tidak boleh sama. Jika jumlah gloss sama maka cek gloss kedua dari Kata 1/2 ≠ gloss kedua.
5. Ketentuan nomer 1-4 terjadi pada minimal 1 dari 4 kategori gloss yang ada (*verb*(v), *noun*(n), *adjective*(a), *adverb*(r))

Kata 1 dan 2 adalah kata hasil yang dikombinasikan, sedangkan Kata 3 adalah Kata 1 + <spasi> + Kata 2, sedangkan Kata 4 adalah Kata 1 + Kata 2 (tanpa spasi). Kata 3 dan 4 diperlukan karena karakteristik dari *RitaWN* yang mampu mendeteksi otomatis, contoh Kata 1 adalah “do”, Kata 2 adalah “or”, maka *RitaWN* akan mendeteksi gabungan kedua kata tersebut sebagai Kata “door”. Kelima ketentuan tersebut dibuat untuk menghindari karakteristik dari *RitaWN* tersebut. Selanjutnya, ketentuan yang perlu diperhatikan pada seleksi kata antara lain:

1. Tag dari kata termasuk dalam kategori NN, VB, JJ, RB, mengingat *wordnet* hanya mengenal kategori *verb*(v), *noun*(n), *adjective*(a), *adverb*(r).
2. Makna dari kata tersebut sesuai dengan tagnya, ditemukan dalam *wordnet*.

Pada Gambar 1, proses kombinasi dan seleksi, digabungkan dengan kotak bergaris putus-putus, hal tersebut untuk menunjukkan bahwa proses kombinasi dan seleksi sebenarnya bukanlah proses berurutan murni. Yang dimaksud dengan berurutan yakni kondisi dimana setelah semua kombinasi kata ditemukan, baru diseleksi satu per satu, tetapi yang dilakukan adalah setiap kombinasi kata yang dihasilkan diseleksi terlebih dahulu, kemudian dicari kombinasi berikutnya.

Tujuan dilakukan metode kombinasi dan seleksi, dengan tidak berurutan, adalah untuk menghindari kata yang sama dipakai beberapa frase, sebagai contoh

kalimat *She just got in her car and drove away* terdapat 2 kondisi penerapan:

1. Kondisi 1: Jika kombinasi kata dilakukan semua terlebih dahulu, maka akan muncul kombinasi *got in* dan *got away*, kemudian dilakukan seleksi karena kedua-duanya terdapat dalam *wordnet*, maka hasil kombinasi dan seleksi menjadi tidak tepat sebab dalam kalimat tersebut yang benar adalah *get in* saja.
2. Kondisi 2: Jika setiap kombinasi kata yang ditemukan, dilakukan seleksi terlebih dahulu, maka ketika muncul kombinasi *got in* dan ternyata dianggap sebagai frase maka *got* dan *in* akan ditandai agar tidak diikutkan pada proses kombinasi berikutnya.

Hasil dan Pembahasan

Pada proses penentuan frase, dilakukan uji coba terhadap 65 kalimat. Kalimat tersebut diambil dari referensi buku dan *website* yang khusus mempelajari mengenai frase terutama frase verbal, karena frase yang memiliki makna baru sebagian besar berupa frase verbal. Kalimat yang digunakan tersusun atas minimal 4 kata dan maksimal 17 kata, di mana pemilihan kalimat tersebut dilakukan secara acak. Untuk mendukung hasil dari uji coba, maka dilakukan pengisian kuesioner untuk memvalidasi hasil dari uji coba. Kuesioner diisi oleh 10 pengajar Bahasa Inggris dari Fakultas Bahasa Inggris, Universitas Petra Surabaya.

Uji coba penentuan frase dilakukan baik untuk kombinasi *insertion* maupun kombinasi *bubble* dengan jarak kombinasi 1 sampai dengan jarak maksimal 5.

Tabel 3. Pengukuran *precision* dan *recall*

		Ujicoba Program	
		(+)	(-)
Kuesioner	⊖	A	B
	⊕	C	D

Keterangan:

A = Frase ada di kuesioner dan ada di ujicoba

B = Frase ada di kuesioner tetapi tidak ada diujicoba

C = Frase tidak ada di kuesioner, tetapi ada diujicoba

D = Frase tidak ada di kuesioner & tidak ada diujicoba

$$Precision = A / (A+C)$$

$$Recall = A / (A+B)$$

$$Akurasi = (A+D) / (A+B+C+D)$$

Tabel 4. Hasil uji coba penentuan frase

Pengukuran	Kombinasi	Jarak				
		1	2	3	4	5
<i>Precision</i>	<i>Bubble</i>	78,26%	82,54%	84,29%	85,14%	84,00%
	<i>Insertion</i>	78,26%	80,95%	82,86%	82,43%	81,33%
<i>Recall</i>	<i>Bubble</i>	55,38%	80,00%	90,77%	96,92%	96,92%
	<i>Insertion</i>	55,38%	78,46%	89,23%	93,85%	93,85%
Akurasi	<i>Bubble</i>	48,00%	68,42%	77,63%	82,89%	81,82%
	<i>Insertion</i>	48,00%	66,23%	75,32%	78,21%	77,22%

Uji coba untuk jarak >5 menghasilkan frase yang sama dengan jarak 5. Hasil uji coba yang dilakukan diukur dengan menggunakan pengukuran *precision* (*P*) dan *recall* (*R*), ketentuan mengenai pengukuran *PR* dapat dilihat pada Tabel 3 sedangkan hasil uji coba dapat dilihat pada Tabel 4.

Pengukuran *recall* bertujuan untuk menunjukkan berapa persen aplikasi mampu menentukan frase yang ada di kuesioner. Contoh pada kombinasi *Bubble*, aplikasi mampu menentukan 96.92% frase yang ada di kuesioner. Sedangkan *precision* bertujuan untuk menunjukkan berapa persen aplikasi mampu menemukan frase baru yang tidak ada di kuesioner. Contoh pada kombinasi *Bubble*, 85.14% frase yang ditemukan aplikasi tersebut, ada di kuesioner, sedangkan sisanya 14.86% adalah frase baru yang tidak ada di kuesioner.

Berdasarkan Tabel 4, jika dilihat dari hasil akhir maka diperoleh hasil yang hampir sama, baik untuk kombinasi *insertion* maupun *bubble*, tetapi apabila dilihat pada detail data yang ada, maka akan terlihat perbedaan pada kata apa yang terdeteksi dengan benar dan apa yang terdeteksi salah. Sebagai contoh pada kalimat *Some schools got President's Day off but we had classes as usual* dan *Could you pick some more snacks up on your way home this evening*. Pada *bubble* kalimat pertama dapat terdeteksi dengan benar (*day off*) sedangkan kalimat kedua terdeteksi salah (*up on*). Sebaliknya pada *insertion*, kalimat pertama terdeteksi salah (*got off*), kalimat kedua terdeteksi benar (*pick up*).

Kombinasi *bubble* menunjukkan persentase yang lebih tinggi dibandingkan dengan kombinasi *insertion*, dengan selisih tidak lebih dari 4%. Persentase *PR* semakin meningkat dengan semakin bertambahnya nilai jarak kombinasi sedangkan untuk akurasi dari jarak kombinasi 4 ke 5 mengalami penurunan dikarenakan ditemukannya lebih banyak frase baru

pada jarak 5 dibandingkan dengan jarak 4. Frase baru yang dimaksud adalah frase yang tidak ditentukan di kuesioner namun ditemukan saat proses kombinasi.

Simpulan

Berdasarkan uji coba yang dilakukan, terdapat 3 hal yang perlu digarisbawahi pada proses penentuan frase antara lain:

1. Proses penentuan frase dengan kombinasi *bubble* dapat mencapai nilai *precision*, *recall* dan akurasi sampai 85.14%, 96.92%, 82.89, di mana jika dibandingkan kombinasi *insertion* menghasilkan selisih tidak lebih dari 4% lebih tinggi.
2. Peranan *wordnet*
Kesediaan data *wordnet* dalam *wordnet* mempengaruhi proses penentuan frase pada kalimat dan dapat menghasilkan yang tidak termasuk dalam kategori frase yang menimbulkan pengertian baru, contohnya *at home*, *this morning*.
3. Karakteristik dari *RitaWN*
Fungsi pada *RitaWN* yang digunakan pada proses penentuan frase adalah *GetGloss* dan *GetAllGloss*. Dalam mencari makna dari suatu kata, *RitaWN* mampu melakukan pencarian kata yang serupa, apabila kata yang dimaksud tidak ada pada *wordnet*.
Contoh:
 - Kombinasi *complain about*, karena *wordnet* tidak menyediakan frase tersebut, maka makna yang dihasilkan oleh fungsi *GetGloss* atau *GetAllGloss* adalah makna dari *complain* atau makna dari *about* saja, tergantung dari kategori yang diminta *verb(v)* atau *adverb(r)*
 - Kombinasi *do<spasi>or*, karena *wordnet* tidak menyediakan frase tersebut, maka makna yang dihasilkan adalah makna dari kata *door*
 - Kombinasi *is<spasi>get*, karena *wordnet* tidak menyediakan frase tersebut, maka makna yang dihasilkan adalah makna dari kata *beget*

Dengan adanya kondisi tersebut, dibuatlah 5 ketentuan pada proses seleksi frase.

4. Jarak kombinasi pencarian frase

Berdasarkan hasil penelitian, jarak pencarian frase semakin besar akan semakin banyak frase yang ditemukan, karena semakin banyak pengkombinasian dilakukan. Namun semakin banyak frase, bukan berarti frase yang ditemukan adalah frase yang benar. Contoh pada kalimat *when my computer crashed, I also lost my back up files*, jika jarak diperbesar menjadi 10, maka akan ditemukan frase *computer files* pada *wordnet*, padahal pada konteks kalimat tersebut *computer* dan *files* lebih tepat untuk konteks kalimat jika terpisah.

Melalui hasil kesimpulan tersebut, walaupun hasil yang diperoleh sudah mencapai 80-90%, namun masih dapat dilakukan variasi kombinasi yakni dengan mendahulukan kata yang berkategori *verb* untuk setiap kombinasi, karena mayoritas frase dengan makna baru adalah frase verbal.

Daftar Pustaka

- [1] Agirre, E dan Stevenson, M., 2006, “*Knowledge sources for WSD. In Word Sense Disambiguation: Algorithms and Applications*”, Eds. Springer, New York, NY, 217–251.
- [2] Allsop, J., 1990, *Test Your Phrasal Verbs, Gelatik*.
- [3] Cutting, J. Kupiec, Pedersen, J. dan Sibun, P. 1992. “A practical Part-of-Speech Tagger”. *Proceedings of the Third Conference on Applied Natural Language Processing. ANLP-92*.
- [4] Mihalcea, R., 2005, “Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling”, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, hal. 411–418.
- [5] Toutanova K. dan Manning, D.C., 2000, “Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger” In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*. hal. 63–70.