PEMBANGKIT PERTANYAAN OTOMATIS UNTUK TEKS BERBAHASA INDONESIA YANG MENGANDUNG KALIMAT MAJEMUK

Dendy Dwi Ginanjar[1], Ken Kinanti Purnamasari[2]

[1,2] Teknik Informatika, Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Lebakgede, Bandung, Kota Bandung, Jawa Barat 40132, Indonesia
E-mail : 2003dendy@gmail.com[1], Guinev_gs@yahoo.co.nz[2]

ABSTRAK

Fokus pada penelitian ini adalah untuk membangun pembangkit pertanyaan otomatis untuk teks Berbahasa Indonesia yang mengandung kalimat majemuk. Penelitian pembangkit pertanyaan yang menggunakan analisis sintaksis sudah dilakukan namun hanya untuk kalimat sederhana dengan klausa verba saja. Padahal dalam tata bahasa Indonesia, klausa tidak hanya dibentuk oleh predikat verba saja. Selain masalah klausa di atas, penelitian sebelumnya tidak melakukan identifikasi frasa padahal fungsi gramatikal di dalam klausa tidak hanya ditempati oleh kata tunggal saja tetapi dapat ditempati juga oleh bermacam-macam jenis frasa.

Adapun proses keseluruhan yang akan dilakukan oleh sistem terdiri dari dua tahap yaitu *preprocess* dan proses utama. *Preprocess* adalah tahap awal sistem untuk mendapatkan jenis kata dan entitas bernama untuk setiap kata pada kalimat, sedangkan proses utama adalah proses analisis sintaksis yang terdiri dari proses identifikasi frasa dengan metode pencocokkan pola, proses penentuan fungsi gramatikal dengan metode *rule-based* dan proses pembangkitan pertanyaan dengan metode template.

Dengan menggunakan 35 teks masukan yang diambil dari Buku Sekolah Elektronik Kemendikbud RI, diperoleh nilai akurasi pertanyaan sebesar 71,06% dengan rincian kesalahan yang disebabkan oleh *preprocess* sebesar 44,55%, analisis sintaksis 21,78% dan kesalahan pembangkitan pertanyaan sebesar 33,66%.

Kata kunci : pembangkitan pertanyaan, analisis sintaksis, identifikasi frasa, kalimat majemuk

1. PENDAHULUAN

Dalam kajian *text-to-question* generation, sistem pembangkit pertanyaan akan diberikan sebuah masukan berupa kalimat tunggal, kalimat majemuk, teks, sekumpulan teks bahkan transkrip percakapan atau dialog, dengan tujuan untuk menghasilkan sekumpulan pertanyaan secara otomatis yang mana jawabannya terdapat pada teks yang menjadi masukannya tersebut [1]. Pembangkit pertanyaan berguna untuk membantu proses evaluasi dalam kegiatan belajar-mengajar baik untuk peserta didik maupun untuk pengajar [2].

Manfaat pembangkit pertanyaan yang dapat diambil oleh peserta didik adalah sebagai alat bantu evaluasi pemahaman terhadap suatu teks atau materi yang dipelajari, sedangkan untuk pengajar, pembangkit pertanyaan dapat digunakan sebagai alat bantu mempersiapkan diri menghadapi pertanyaan potensial yang mungkin diajukan oleh peserta didik. Selain di dalam proses belajar-mengajar, pembangkit pertanyaan juga dapat digunakan untuk membangun *help system* seperti *Frequently Ask and Question (FAQ)*.

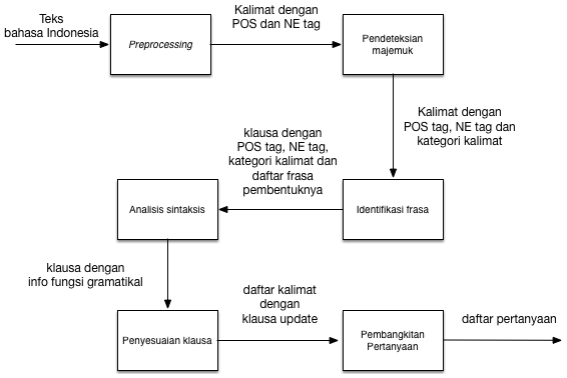
Beberapa penelitian mengenai pembangkit pertanyaan telah dilakukan. Pembangkit pertanyaan dalam bahasa Inggris dilakukan dengan metode sintaksis dan metode statistik sedangkan dalam bahasa Indonesia penelitian tentang pembangkit pertanyaan dilakukan dengan metode template dan metode semantik. Adapun pembangkit pertanyaan dalam bahasa Indonesia telah dilakukan untuk menghasilkan pertanyaan *non-factoid* [3] dan *factoid* [4]. Pertanyaan *non-factoid* adalah pertanyaan mengenai 'definisi', 'alasan', dan 'cara' sedangkan pertanyaan *factoid* adalah pertanyaan tentang 'apa', 'siapa', 'di mana', 'ke mana', 'berapa', dan 'kapan'. Tetapi, penelitian pertanyaan *factoid* baru dilakukan untuk kalimat sederhana yang terbatas untuk klausa verba saja. Klausa verba adalah klausa yang mengandung predikat kata kerja. Dalam tata Bahasa Indonesia fungsi predikat tidak hanya dapat ditempati oleh kata kerja tetapi dapat ditempati oleh frasa atau jenis kata lainnya seperti kata benda, kata sifat, kata bilangan dan lain-lain [6][8].

Pertanyaan yang dibangkitkan akan membutuhkan jawaban tentang subjek, objek keterangan bahkan tentang predikat. Oleh karena itu penentuan fungsi gramatikal di dalam sebuah kalimat menjadi sangat penting. Fungsi gramatikal di dalam kalimat dapat ditempati oleh kata tunggal maupun frasa sehingga proses identifikasi frasa menjadi

penentu utama keberhasilan penentuan fungsi gramatikal pada tiap kata maupun frasa yang ada.

2. ISI PENELITIAN

Untuk mengatasi masalah pembangkitan pertanyaan dengan kalimat majemuk diperlukan analisis sintaksis mulai dari identifikasi frasa sampai identifikasi fungsi gramatikal. Indentifikasi frasa diperlukan untuk proses penentuan fungsi gramatikal, sedangkan fungsi gramatikal sangat diperlukan dalam proses pembangkitan pertanyaan karena pertanyaan *factoid* membutuhkan jawaban tentang subjek, objek, bahkan keterangan.



Gambar 1 Alur Sistem

Secara umum, sistem ini akan memiliki alur sebagai berikut ; 1) user mengunggah file teks berisi kalimat atau kumpulan kalimat baku dalam bahasa Indonesia, 2) sistem akan melakukan *preproccess* yang akan menghasilkan informasi POS tag dan NE tag dari setiap kata di dalam teks masukan, 3) sistem akan melakukan pendeteksian kalimat majemuk berdasarkan keberadaan kata kunci yang telah didefinisikan, 4) sistem melakukan proses identifikasi frasa, 5) sistem melakukan identifikasi fungsi gramatikal untuk menentukan subjek, predikat, dan objek di dalam kalimat, 6) sistem melakukan proses penyesuaian klausa untuk kalimat majemuk setara, dan terakhir, 7) sistem melakukan proses pembangkitan pertanyaan.

2.1 *Preprocess*

Preprocessing merupakan tahapan awal yang dilakukan oleh sistem, dimana pada tahap ini, teks masukan akan ditokenisasi dan dilakukan pelabelan jenis kata (POS tag) dan pengenalan entitas bernama (NER). Proses ini dilakukan sepenuhnya oleh aplikasi indonesia NER[9][13].

Pengenalan entitas bernama atau yang disebut dengan *Named Entity Recognition (NER)* adalah sebuah proses yang digunakan untuk mengenali dan mengklasifikasi istilah dan frasa sebagai entitas nama dari sebuah teks bebas [12]. NER adalah salah satu komponen utama dari *information extration* yang bertujuan untuk mendeteksi dan mengklasifikasikan *named-entity* pada suatu teks. NER umumnya digunakan untuk mendeteksi nama orang, nama tempat dan organisasi dari sebuah dokumen, tetapi dapat juga diperluas untuk identifikasi gen, protein dan lainnya sesuai kebutuhan.

2.2 Deteksi Majemuk

Setelah proses pelabelan kata selesai, sistem akan melakukan pendeteksian majemuk terhadap semua kalimat yang menjadi masukan sistem. Terdapat tiga kategori kalimat yang telah didefinisikan sistem, yaitu; majemuk setara, majemuk bertingkat, dan lain-lain. Pendeteksian majemuk dilakukan berdasarkan pada keberadaan kata penghubung. Kata penghubung yang dimaksud adalah kata penghubung kalimat majemuk setara dan majemuk bertingkat.

Tabel 1 Contoh Deteksi Majemuk

Kalimat	Presiden tidak bisa menghadiri peresmian tol barito karena Presiden baru kembali besok.
Kategori	Majemuk bertingkat
Kata kunci	karena

2.2 Identifikasi Frasa

Pembentukan frasa mengacu pada pola frasa yang telah didefinisikan oleh sistem. Pencarian pola frasa dilakukan berurutan dari kata pertama sampai kata terakhir di dalam klausa menggunakan *rule-based*. Aturan untuk menentukan frasa di dalam klausa dapat dilihat pada Tabel 2 di bawah ini.

Tabel 2 Aturan Identifikasi Frasa

NO	Aturan
1	Pointer berada pada kata ke-*n*, kemudian sistem menyimpan kata ke-*n* tersebut ke variabel *x* dan menyimpan POS tag dari kata ke-*n* ke variabel *y*. Selanjutnya *n* ditambah satu. Jika kata ke-*n* masih ada maka pointer bergerak ke-*n*, jika tidak ada, salin variabel *x* lalu tambahkan ke variabel *z['frasa']* sedangkan variabel *y* disalin dan ditambahkan ke variabel *z['kategori']*.
2	Pointer berada pada kata ke-*n*, kemudian sistem akan mengkombinasikan POS tag kata ke-*n* dengan POS tag yang ada pada variabel *y*, kombinasi tersebut disimpan ke variabel *k*. Selanjutnya sistem akan mencari pola frasa yang sesuai dengan kombinasi pada variabel *k*. Jika tidak ditemukan pola frasa yang sama, sistem akan melanjutkan proses ke poin (3). Namun, jika ditemukan pola frasa yang sama, ganti nilai variabel *y* dengan nilai variabel *k* lalu tambahkan kata ke-*n* ke variabel *x*. Nilai *n* ditambah satu dan sistem akan mengulang proses poin (2). Variabel *k* selalu dikosongkan di setiap perulangan.
3	Susunan kata pada variabel *x* ditentukan sebagai frasa kemudian variabel *x* disalin

NO	Aturan
	dan ditambahkan ke variabel *z['frasa']* sedangkan kategori frasa yang sesuai dengan pola frasa pada variabel *y* disimpan ke variabel *z['kategori']*. Variabel *x* dan *y* dikosongkan lalu sistem kembali melakukan proses pada poin (1).

Contoh identifikasi frasa untuk klausa *Presiden Joko Widodo sedang melaksanakan tugas Negara ke Singapura*

Langkah 1 (aturan 1) :
Pointer berada pada kata pertama (*n=1*), yaitu *Presiden* dengan POS tag NN. Kata '*Presiden'* disimpan ke variabel *x* dan Tag 'NN' disimpan ke variabel *y*. Kemudian *n* ditambah satu *(n = 2*) dan pointer bergerak ke-*n.*

var x = Presiden
var y = NN
var k = ''
var z = ''

Langkah 2 (aturan 2) :
Pointer berada pada kata kedua (*n=2*), yaitu *Joko* dengan POS tag NN. Kombinasi antara POS tag kata kedua dengan POS tag pada variabel *y* disimpan ke variabel *k* lalu sistem mencari pola frasa yang sesuai dengan kombinasi yang terdapat pada variabel *k*. Pola yang sesuai sehingga kata *'Joko'* ditambahkan ke variabel *x* dan nilai variabel *y* diganti dengan nilai variabel *k*.

Var n ditambah satu (*n=3*) lalu pointer bergerak ke-*n*.

update variabel
var x = 'Presiden Joko'
var y = 'NN+NN'
var k = ''
var z = ''

Langkah 3 (aturan 2) :
Pointer berada pada kata ketiga yaitu *Widodo* dengan POS tag NN. Kombinasi antara POS tag kata ketiga dengan POS tag pada variabel *y* disimpan ke variabel *k* lalu sistem mencari pola frasa yang sesuai dengan kombinasi yang terdapat pada variabel *k*. Pola yang sesuai ditemukan sehingga kata *'Widodo'* ditambahkan ke variabel *x* dan nilai variabel *y* diganti dengan nilai variabel *k*.

Var n ditambah satu (*n=4*) lalu pointer bergerak ke-*n*.

update variabel
var x = 'Presiden Joko Widodo'
var y = 'NN+NN+NN'
var k = ''
var z = ''

Langkah 4 (aturan 2,3) :
Pointer berada pada kata keempat yaitu *sedang* dengan POS tag RB. Kombinasi antara POS tag kata ketiga dengan POS tag pada variabel *y* disimpan ke variabel *k* lalu sistem mencari pola frasa yang sesuai dengan kombinasi yang terdapat pada variabel *k*. Pola yang sesuai tidak ditemukan.

var x = 'Presiden Joko Widodo'
var y = 'NN+NN+NN'
var k = 'NN+NN+NN+RB'
var z = ''

Langkah 5 (aturan 3) :
Pola frasa tidak ditemukan pada Tabel 2.12 sehingga daftar kata yang ada pada variabel *x* ditetapkan sebagai frasa dengan kategori frasa sesuai dengan pola pada variabel *y*. Salin variabel *x* ke variabel *z[frasa]* dan simpan kategori frasa ke variabel *z[kategori*] lalu kosongkan variabel *x* dan *y*. Kemudian, sistem mengulang proses aturan poin (1).

update variabel
var x = ''
var y = ''
var k = ''
var z[frasa] = 'Presiden Joko Widodo'
var z[kategori] = 'FN'

Langkah 6 :
Proses pencarian frasa diulang dari langkah pertama untuk kata-kata selanjutnya yang masih ditemukan di dalam klausa.

Hasil Akhir identifikasi frasa untuk klausa *Presiden Joko Widodo sedang melaksanakan tugas Negara ke Singapura* adalah sebagai berikut :
Frasa = 1. *Presiden Joko Widodo (FN)*
2. sedang melaksanakan (FVB)
3. tugas Negara (FN)
4. ke Singapura (FIN)

2.3 Fungsi Gramatikal

Kalimat adalah satuan bahasa yang disusun konstituen dasar yang umumnya berupa klausa, kata penghubung (jika ada) dan intonasi final. Dalam ragam tulisan intonasi final ini dinyatakan dengan titik (.), tanda tanya (?), atau tanda seru (!) [12]. Menurut strukturnya, sebuah kalimat sederhana dalam bahasa Indonesia memiliki pola sebagai berikut[14][12].

Tabel 3 Pola Kalimat

STRUKTUR
Subjek + Predikat (S-P)
Subjek + Predikat + Objek (S-P-O)
Subjek + Predikat + Pelengkap (S-P-Pel)
Subjek + Predikat + Objek + Pelengkap (S-P-O-Pel)
Subjek + Predikat + Objek + Keterangan (S-P-O-K)
Subjek + Predikat + Keterangan (S-P-K)

Identifikasi fungsi gramatikal dilakukan setelah proses identifikasi frasa. Proses ini dilakukan menggunakan aturan tata bahasa pembentukan kalimat seperti yang telah dipaparkan pada Tabel 3.2. Penentuan fungsi gramatikal khusus pada kalimat majemuk bertingkat hanya dilakukan pada klausa induk saja sedangkan pada kalimat majemuk setara, penentuan fungsi gramatikal akan dilakukan pada kedua klausa yang telah dipisahkan pada tahap proses deteksi majemuk.

Proses penentuan fungsi gramatikal hampir sama dengan proses identifikasi frasa, yaitu dilakukan pengecekan setiap kata maupun frasa yang terdapat pada klausa dari awal sampai akhir secara berurutan. Aturan-aturan dalam menentukan fungsi gramatikal dapat dilihat pada Tabel 3.4 berikut ini.

Tabel 4 Aturan Penentuan Fungsi Gramatikal

No	Aturan
1	Jika kata pertama pada klausa adalah konjungsi, jadikan kata tersebut sebagai klausa *keterangan* sampai tanda baca koma (,).
2	Jika belum ada subjek, dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk subjek, jadikan kata atau frasa tersebut sebagai subjek.
3.	Jika kata atau frasa sebelumnya adalah subjek, dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk predikat, jadikan kata atau frasa tersebut sebagai predikat.
4	Jika kata atau frasa sebelumnya adalah predikat berjenis verba transitif (VBT) dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk objek, jadikan kata atau frasa tersebut sebagai objek.
5	Jika kata atau frasa sebelumnya adalah predikat berjenis verba intransitif (VBI), jadikan kata atau frasa sekarang sebagai pelengkap.
6	Jika fungsi subjek, predikat, objek sudah ada. Dan Jika kata atau frasa sekarang adalah konjungsi, jadikan kata atau frasa tersebut sebagai klausa keterangan sampai akhir klausa.
7	Jika fungsi subjek, predikat, objek sudah ada. Dan Jika kata atau frasa sekarang mengandung kata depan, jadikan kata atau frasa tersebut sebagai keterangan sampai akhir klausa.
8	Jika fungsi subjek, predikat, objek sudah ada. Dan Jika kata atau frasa sekarang **TIDAK** mengandung kata depan, jadikan kata atau frasa tersebut sebagai pelengkap sampai akhir klausa.

2.3 Penyesuaian Klausa (*call reference*)

Proses penyesuaian klausa hanya dilakukan untuk kalimat majemuk setara. Fungsinya adalah untuk melakukan penyesuaian klausa kedua dengan klausa pertama. Kalimat majemuk setara dapat dipecah menjadi klausa-klausa bebas. Kemudian, klausa-klausa bebas ini akan diproses pada modul pembangkit pertanyaan satu persatu secara bergantian sehingga memungkinkan dihasilkannya pertanyaan maupun jawaban yang ambigu.

Aturan untuk melakukan proses penyesuaian klausa dapat dilihat pada Tabel 5 berikut ini.

Tabel 5 Aturan Penyesuaian Klausa

No	Aturan
1	Jika subjek klausa kedua adalah kata ganti (PR), maka kata ganti tersebut mengacu pada subjek di klausa pertama.
2	Jika klausa kedua tidak memiliki subjek, maka subjek klausa kedua sama dengan subjek klausa pertama.
3	Jika objek pada klausa kedua adalah kata milik (NNG), maka objek tersebut sama dengan objek pada klausa pertama.

Contoh penyesuaian klausa

Kalimat : *Presiden akan berkunjung ke lokasi kemudian Dia akan kembali ke Jakarta .*

Klausa 1 : *Presiden akan berkunjung ke lokasi*

Klausa 2 : *Dia akan kembali ke Jakarta*

Tabel 6 Contoh Penyesuaian Klausa

No	Proses
1	Subjek klausa kedua *Dia* adalah kata ganti (PR). Kondisi ini sesuai dengan aturan ke-1 . Maka kata *Dia* mengacu pada subjek klausa pertama yaitu *Presiden*. Klausa 2 (sebelum) : Dia akan kembali ke Jakarta Klausa 2 (setelah) : Presiden akan kembali ke Jakarta

2.4 Pembangkit Pertanyaan

Setelah tahapan *preprocessing* dan analisis sintaksis selesai dilakukan, maka kalimat yang menjadi masukan sistem akan memiliki berbagai macam informasi penting yang akan digunakan dalam proses pembangkitan pertanyaan. Pertanyaan yang dibangkitkan adalah pertanyaan *factoid* dan *non-factoid.*

Pembangkitan kalimat pertanyaan dibantu dengan *template* berdasarkan keberadaan kata kunci dan fungsi gramatikal yang didapatkan dari hasil analisis sintaksis. Pembangkitan pertanyaan *non-factoid* akan dilakukan dengan cara mengidentifikasi kata kunci yang terdapat pada kalimat masukan. Setelah kata kunci ditemukan, sistem akan memilih template pertanyaan yang sesuai dengan kategori kata kunci lalu membangkitkan pertanyaan dan jawaban menggunakan template yang telah dipilih tersebut. Berikut adalah daftar kata kunci dan template pertanyaan yang dapat dipakai untuk membangkitkan pertanyaan *non-factoid* dari sebuah kalimat.

2.4.1 Deteksi *Non-factoid*

Pada proses ini akan dilakukan proses pelabelan kandidat *non-factoid.* Caranya sama dengan deteksi majemuk, yaitu dengan cara mengidentifikasi keberadaan kata kunci khusus *non-factoid*. Tabel 7 berikut adalah daftar kata kunci untuk pertanyaan *non-factoid* [5].

Tabel 7 Kata Kunci Non-factoid

Kategori non-factoid	Kata Kunci	
	Sebelum target	Sesudah target
Definisi	disebut, dikenal, dinamakan, mendefinisikan	adalah, yaitu, ialah, merupakan, diartikan
Alasan	oleh sebab itu, jadi, memungkinkan adanya, dengan demikian, maka, dikatakan, penyebab	sebab, karena, bertujuan
	terjadinya, sehingga, walau demikian, namun demikian	
Metode	berfungsi untuk, berguna untuk	dengan cara

2.4.2 Pembangkitan pertanyaan *Non-factoid*

Klausa atau kalimat yang terdeteksi sebagai *non-factoid* melalui proses deteksi *non-factoid* akan membuat sistem membangkitkan pertanyaan menggunakan template *non-factoid.* Template *non-factoid* dapat dilihat pada Tabel 8 di bawah ini.

Tabel 8 Template Non-Factoid

Jenis	Template
Definisi	Apa yang dimaksud...?
Alasan	Mengapa ...?
Cara	Bagaimana cara. ..?

2.4.3 Pembangkitan pertanyaan *factoid*

Untuk pembangkitan pertanyaan *factoid* akan dilakukan menggunakan bantuan *template* dan analisis sintaksis.

Tabel 9 Template Factoid

Target	Template
Subjek	Siapa yang ...? Apa yang ....? Berapa ...?
Objek	Siapa yang ...? Apa yang ....? Berapa ...?
Keterangan	Di mana ...? Ke mana ...? Kapan ...?

3. PENUTUP

Sistem pembangkit pertanyaan otomatis dari teks berbahasa Indonesia berhasil membangkitkan pertanyaan *factoid* berupa pertanyaan siapa, apa, mana, kapan, berapa dan pertanyaan *non-factoid* berupa pertanyaan definisi, cara dan alasan. Sistem ini dapat menghasilkan pertanyaan dengan akurasi sebesar 71,06% dengan rincian kesalahan yang disebabkan oleh *preprocess* sebesar 44,55%, analisis sintaksis 21,78% dan kesalahan pembangkitan pertanyaan sebesar 33,66%. Dari hasil tersebut dapat disimpulkan bahwa proses analisis sintaksis yang secara garis besar terdiri dari proses identifikasi frasa dan penentuan gramatikal masih memiliki tingkat kesalahan yang cukup tinggi sehingga mengurangi akurasi pertanyaan yang dihasilkan sistem.

Saran untuk penelitian berikutnya untuk

meningkatkan akurasi sistem adalah sebagai berikut. Pertama, diperlukan penambahan aturan untuk memproses kalimat majemuk rapatan dan campuran. Kalimat majemuk rapatan sebenarnya adalah kalimat majemuk setara yang klausa-klausanya memiliki kesamaan subjek atau predikat sehingga penulisannya di dalam kalimat seakan-akan terdiri dari satu klausa padahal terdiri dari dua klausa atau lebih yang dirapatkan subjek atau predikatnya. Meskipun sama, penanganan kalimat majemuk rapatan berbeda dengan majemuk setara dan ini perlu analisis yang lebih mendalam lagi supaya proses identifikasi frasa pada kalimat majemuk rapatan bisa lebih akurat.

Kedua, diperlukan penambahan aturan pada proses identifikasi frasa untuk dapat memproses kalimat yang mengandung perluasan subjek, predikat atau perluasan objek. Hal ini diperlukan karena keterbatasan pola frasa yang ditetapkan oleh sistem belum optimal mengenali fungsi-fungsi gramatikal yang diperluas.

Ketiga, sistem yang telah dibangun pada penelitian ini, tahapan *preprocess* nya bergantung pada aplikasi penunjang, sehingga akurasi hasil *preprocess* akan berimbas pada akurasi analisis sintaksis yaitu identifikasi frasa dan penentuan fungsi gramatikal. Adapun hasil dari *preprocess* ini adalah jenis kata (POS Tag) dan entitas bernama (NE Tag) untuk setiap kata yang ada pada teks masukan. Untuk meningkatkan akurasi identifikasi frasa, dalam *preprocess* dapat ditambahkan proses pendeteksian kata ambigu.

Keempat, untuk mendapatkan pasangan pertanyan-jawaban yang relevan bisa ditambahkan proses analisis semantik antar kalimat, sehingga jika terdapat makna yang berkaitan antara satu kalimat dengan kalimat yang lainnya sistem dapat membangkitkan pertanyaan-jawaban lebih akurat. Dan jika dilakukan penambahan template pertanyaan pada proses pembangkitan pertanyaan, analisis semantik dapat dipergunakan untuk mencegah kemunculan pertanyaan-pertanyaan yang memiliki konteks yang sama dengan pertanyaan yang lainnya.

DAFTAR PUSTAKA

[1] H. D. A. D. Ali, "Thesis: Automatic question generation : a syntactical approach to the sentence-to-question generation case," 2012.

[2] W. Suwarningsih, I. Supriana, A. Purwarianti, and P. P. Informatika-lipi, "Tantangan dan Peluang pada Question Generation 1,3,4," *J. Sist. Inf.*, vol. 6, no. 2, pp. 719–730, 2014.

[3] M. Heilman, "Automatic Factual Question Generation from Text," *Dissertation*, no. June, p. 203, 2011.

[4] M. Fachrurrozi, "Sistem Pembangkit Pertanyaan Otomatis Dengan Metode Template-Based," 2013.

[5] W. Suwarningsih, I. Supriana, and A. Purwarianti, "Toward a Framework for Indonesian Medical Question Generator," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 13, no. 1, p. 357, 2015.

[6] Efendi, D. Kentjono, and B. Suhardi, *Tata Bahasa Dasar*, Pertama. Bandung: ROSDAKARYA, 2015.

[7] a a Zulen and a Purwarianti, "Study and implementation of monolingual approach on indonesian question answering for factoid and non-factoid question," *PACLIC 25 - Proc. 25th Pacific Asia Conf. Lang. Inf. Comput.*, pp. 622–631, 2011.

[8] M. Khairah and S. Ridwan, *Sintaksis : Memahami Satuan Kalimat Persfektif Fungsi*, Pertama. Jakarta: PT Bumi Aksara, 2014.

[9] M. Fachri, "Named Entity Recognition For Indonesian Text Using Hidden Markov Model," Universitas Gadjah Mada, 2014.

[10] R. A. SUKAMTO, "Penguraian Bahasa Indonesia Dengan Rosa Ariani Sukamto Nim :," vol. 23507024, 2009.

[11] A. F. Wicaksono and A. Purwarianti, "HMM Based Part-of-Speech Tagger for Bahasa Indonesia," *4th Int. MALINDO (Malay Indones. Lang. Work.*, no. October 2016, 2010.

[12] I. Budi, S. Bressan, G. Wahyudi, Z. A. Hasibuan, and B. A. A. Nazief, "Named Entity Recognition for the Indonesian language: Combining contextual, morphological and part-of-speech features into a knowledge engineering approach," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3735 LNAI, pp. 57–69, 2005.

[13] Y. Syaifudin, "Quotations Identification From Indonesian Online News Using Rule-Based Method," Universitas Gadjah Mada, 2016.