

# PENDETEKSIAN KETERKAITAN ANTAR KALIMAT DENGAN METODE TEMPLATE-BASED DALAM PEMBANGKIT PERTANYAAN OTOMATIS

Purba Saripudin<sup>1</sup>, Ken Kinanti Purnamasari<sup>2</sup>

<sup>1,2</sup> Universitas Komputer Indonesia

Jl. Dipati Ukur No. 102-116 Telp. (022) 2504119, 2506634, 2533603 Fax. (022)  
2533754

E-mail: purbasaripudin@yahoo.com<sup>1</sup>, ken.kinanti@email.unikom.ac.id<sup>2</sup>

## ABSTRAK

Penelitian pembangkit pertanyaan yang menggunakan analisis sintaksis sudah dilakukan namun terbatas hanya terdapat pada satu kalimat saja[4], padahal sebuah kalimat pertanyaan dapat berasal dari beberapa kalimat. Proses keseluruhan yang akan dilakukan oleh sistem terdapat dua tahap yaitu *preprocessing* yang menggunakan penelitian sebelumnya dan proses utama. *Preprocessing* berfungsi untuk mendapatkan jenis kata, entitas nama, jenis frasa, fungsi gramatikal pada tiap kalimat dan deteksi kalimat non-definisi sedangkan proses utama adalah proses deteksi keterkaitan antar kalimat yang menggunakan metode *rule-based* dan proses pembangkitan pertanyaan dengan metode *template*. Hasil yang diperoleh oleh sistem dengan menggunakan 50 teks masukan yang diambil dari Buku Sekolah Elektronik Kemendikbud RI, diperoleh nilai akurasi pertanyaan sebesar 90,80% dengan total pertanyaan yang berhasil dibangkitkan sebanyak 413 pertanyaan.

**Kata kunci** : pembangkit pertanyaan, deteksi keterkaitan antar kalimat, *rule-based*, *template*.

## 1. PENDAHULUAN

Penelitian tentang sistem pembangkit pertanyaan otomatis telah berhasil dilakukan menggunakan teks berbahasa Inggris maupun bahasa Indonesia dengan berbagai metode. Penelitian menggunakan teks berbahasa Inggris, metode yang dilakukan diantaranya menggunakan transformasi sintaksis[1] dan metode statistik[2]. Sedangkan penelitian yang menggunakan teks berbahasa Indonesia diantaranya menggunakan metode *template*[3][4] dan metode semantik[5]. Pembangkit pertanyaan yang dilakukan untuk teks berbahasa Indonesia menghasilkan pertanyaan *factoid* dan *non-factoid*.

Berdasarkan beberapa penelitian yang telah disebutkan, terdapat penelitian bahasa Indonesia yang menggunakan metode berbasis *template*[3][4]. Pada penelitian ini sudah dilakukan penelitian pembangkit pertanyaan untuk kalimat tunggal[3][4] dan kalimat majemuk[4]. Penelitian ini menggunakan metode berbasis *template* dengan fungsi sintaksis untuk mencari kata kerja[3] dan aturan gramatikal[4] yang telah ditentukan untuk membangkitkan pertanyaan.

Dari penelitian di atas[3][4], diketahui bahwa kalimat pertanyaan yang dibangkitkan, terbatas hanya dapat memproses pertanyaan dari setiap kalimat saja. Padahal, terdapat pertanyaan yang dapat diproses dari beberapa kalimat, dikarenakan adanya kalimat yang saling berkaitan. Kalimat-kalimat tersebut seharusnya dapat diproses dan dijadikan kalimat pertanyaan. Pemrosesan banyak kalimat dapat dilakukan dengan mengambil sintaksis dari setiap kalimat, kemudian disesuaikan dengan kondisi pada setiap aturan *template* pertanyaan. Informasi sintaksis ini diambil dengan menggunakan aturan khusus untuk mendeteksi kalimat mana saja yang saling berkaitan.

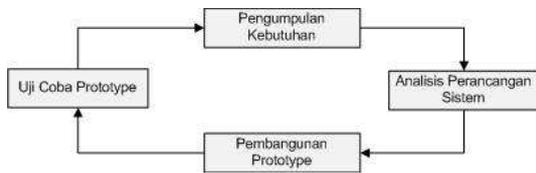
### 1.1 Maksud dan Tujuan

Berdasarkan permasalahan yang ada, maka maksud dari penelitian ini adalah untuk membangun sistem pembangkit pertanyaan yang dapat mendeteksi keterkaitan antar kalimat dan membangkitkan pertanyaan berdasarkan beberapa kalimat. Adapun tujuan dari penelitian ini adalah untuk mengetahui akurasi pertanyaan-pertanyaan yang dibangkitkan oleh sistem yang dapat membangkitkan pertanyaan informasi dari beberapa kalimat.

### 1.2 Metode Pembangunan Perangkat Lunak

Dalam pembangunan sistem pembangkit pertanyaan menggunakan metode *prototype*, karena pembangunan penelitian akan terus disempurnakan dengan adanya evaluasi dari hasil

uji keberhasilannya [6]. Metode *Prototype* sangat cocok untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus [10].



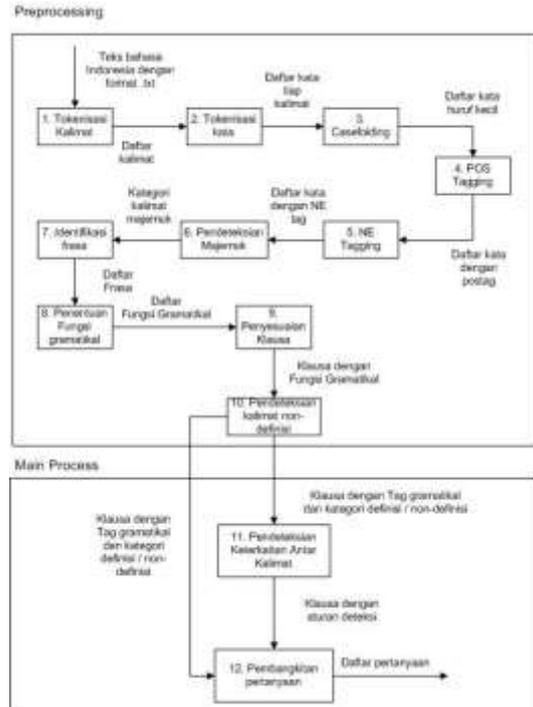
**Gambar 1.** Metode *Prototype*

Metode *Prototype* terdapat beberapa tahapan diantaranya :

1. Pengumpulan kebutuhan  
Tahapan ini merupakan proses mengidentifikasi kebutuhan sistem, pengumpulan data yang dibutuhkan, mengumpulkan kamus kata dari berbagai sumber dan analisis aturan-aturan tata bahasa Indonesia.
2. Analisis perancangan sistem  
Tahapan ini merupakan proses analisis metode, perancangan sistem, untuk mengetahui kebutuhan fungsional dan fungsional, serta rancangan dari basis data antarmuka dari pembangunan aplikasi ini.
3. Pembangunan *prototype*  
Melakukan implementasi dari perancangan sistem yang telah dibuat. Dalam penelitian ini, perangkat lunak akan dibangun menggunakan bahasa pemrograman PHP.
4. Uji coba *prototype*  
Tahap ini melakukan uji coba terhadap perangkat lunak yang dibangun. Jika dari hasil uji coba masih ditemukan kesalahan, proses pembangunan perangkat lunak akan diperbaiki.

## 2. ISI PENELITIAN

Solusi dari permasalahan mengenai pembangkit pertanyaan yang terbatas hanya pada satu kalimat saja, penelitian ini akan melakukan proses pendeteksian antar kalimat setelah proses sintaksis. Proses pendeteksian antar kalimat dilakukan dengan cara mencari subjek, predikat dan objek pada seluruh kalimat.



**Gambar 2.** Blok Diagram alur sistem

Pada sistem yang akan dibangun ini, pengguna memilih teks berisi kalimat atau kumpulan kalimat baku dalam bahasa Indonesia, kemudian sistem ini akan mengolah teks masukan yang memiliki alur sebagai berikut : 1) sistem akan melakukan proses tokenisasi kalimat, 2) sistem akan melakukan proses tokenisasi kata, 3) Sistem melakukan *Case folding* (menyeragamkan semua menjadi huruf kecil), 4) Sistem melakukan pendeteksian jenis kata (*POS Tagging*), 5) sistem melakukan proses pengenalan entitas bernama (NER), 6) sistem melakukan proses pendeteksian kalimat majemuk, 7) sistem melakukan proses identifikasi frasa, 8) sistem melakukan proses penentuan fungsi gramatikal untuk menentukan subjek, predikat, objek, keterangan dan pelengkap, 9) sistem melakukan proses penyesuaian klausa khusus untuk kalimat majemuk setara, 10) sistem melakukan deteksi kalimat non-definis, 11) sistem melakukan deteksi banyak kalimat, 12) sistem melakukan proses pembangkit pertanyaan.

### 2.1 Preprocessing

*Preprocessing* adalah proses awal yang dilakukan oleh sistem, dimana pada tahap ini teks masukan akan dipisahkan menjadi kata-kata tunggal dan dilakukan pelabelan jenis kata (POS tag), pengenalan entitas bernama (NER), pendeteksian kalimat majemuk, identifikasi frasa, penentuan fungsi gramatikal, penyesuaian klausa dan deteksi kalimat non-definis. Proses

ini dilakukan berdasarkan penelitian sebelumnya [3][4].

### 2.1.1 Tokenisasi Kalimat

Tokenisasi kalimat digunakan untuk memecah string masukan menjadi kumpulan kalimat-kalimat berdasarkan tanda titik (.), tanda tanya (?) dan tanda seru (!). Tanda pemisah kalimat tidak akan digunakan dalam proses selanjutnya.

### 2.1.2 Tokenisasi Kata

Setelah kalimat dipisahkan dalam tokenisasi kalimat, tokenisasi kata digunakan untuk memisahkan kata dalam kalimat menggunakan tanda pemisah spasi, sehingga akan terbentuk kata-kata tunggal dalam setiap kalimat.

### 2.1.3 Casefolding

Casefolding digunakan untuk mengubah semua huruf kapital menjadi karakter dengan huruf kecil. Hasil dari proses ini hanya digunakan pada proses *pos tagging* saja.

### 2.1.4 POS Tagging

Setelah memiliki kata dengan huruf kecil, proses selanjutnya yang akan dilakukan sistem adalah pendeteksian jenis kata, proses ini menggunakan aplikasi Indonesia NER [7][8]. Proses ini akan menghasilkan label kelas kata secara otomatis pada sebuah kata, hasil dari tag ini sangat berpengaruh terhadap proses-proses selanjutnya terutama identifikasi frasa.

### 2.1.5 NE Tagging

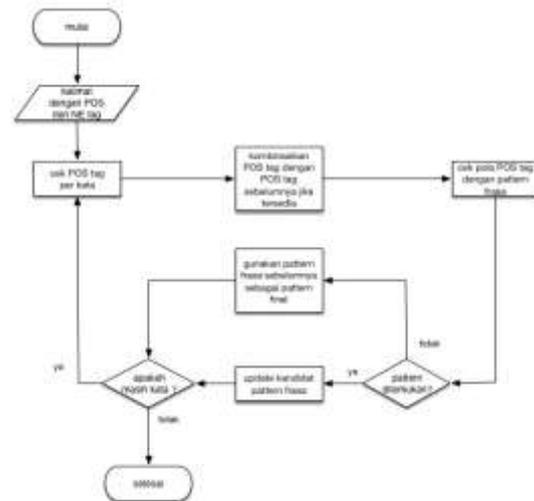
Proses selanjutnya yang akan dilakukan sistem adalah pengenalan entitas bernama, proses ini menggunakan aplikasi Indonesia NER [7][8]. Proses ini digunakan untuk mendeteksi orang, tempat dan organisasi dari sebuah teks, hasil dari tag ini sangat berpengaruh terhadap penentuan *template* pertanyaan yang akan digunakan.

### 2.1.6 Deteksi Majemuk

Proses selanjutnya yang akan dilakukan sistem adalah pendeteksian kalimat majemuk. Proses ini akan mendeteksi kategori terhadap seluruh kalimat, terdapat tiga kategori yang akan didefinisikan yaitu : majemuk setara, majemuk bertingkat dan lain-lain. Proses ini dilakukan dengan cara mendeteksi kata hubung pada kalimat majemuk setara dan majemuk bertingkat.

### 2.1.7 Identifikasi Frasa

Proses selanjutnya yang akan dilakukan sistem adalah identifikasi frasa, proses ini bergantung pada hasil POS tag dan pola frasa yang telah didefinisikan. Proses ini dilakukan berurutan dengan cara menyesuaikan kombinasi jenis kata (POS tag) dengan pola frasa yang telah didefinisikan. Tahapan identifikasi frasa dapat dilihat pada Gambar 3 berikut.



Gambar 3. Flowchart Identifikasi Frasa [4]

### 2.1.8 Penentuan Fungsi Gramatikal

Setelah mendapatkan frasa pada seluruh kalimat, proses selanjutnya adalah penentuan fungsi gramatikal. Proses ini hampir sama dengan proses identifikasi frasa, yaitu melakukan pengecekan setiap kata atau frasa dari awal kalimat hingga akhir secara berurutan. Aturan penentuan fungsi gramatikal dapat dilihat pada Gambar 4 berikut.



**Gambar 4.** Flowchart Penentuan Fungsi Gramatikal [4]

### 2.1.9 Penyesuaian Klausa

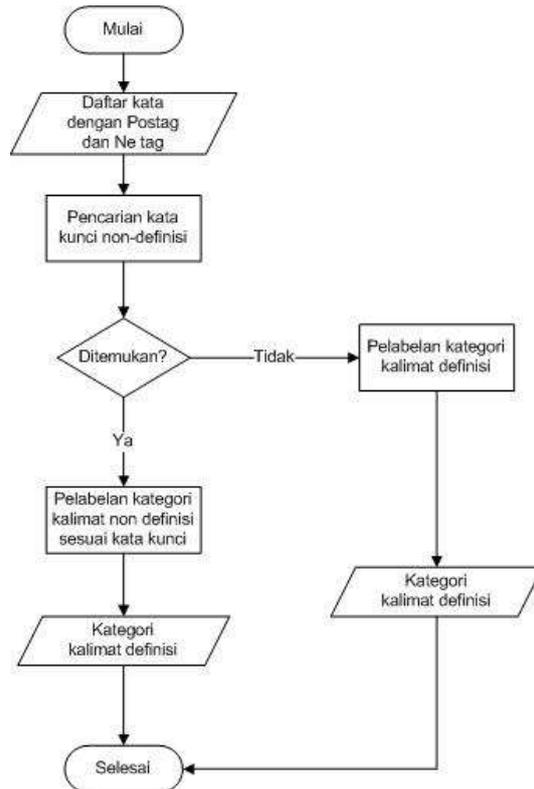
Penyesuaian klausa hanya dilakukan pada kalimat yang terdeteksi kalimat majemuk setara. Fungsi dari proses ini adalah untuk menyesuaikan subjek klausa kedua dengan klausa pertama agar pertanyaan dan jawaban yang dibangkitkan pada klausa kedua tidak ambigu. Aturan penyesuaian klausa dapat dilihat pada Tabel 1 berikut.

**Tabel 1.** Aturan Penyesuaian Klausa [4]

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.

### 2.1.10 Deteksi Kalimat Non-Definisi

Proses deteksi kalimat non-definisi berguna untuk menentukan bahwa kalimat yang memiliki kata 'adalah' dan 'ialah' tidak selalu merupakan kalimat definisi. Proses ini dilakukan secara berurutan dari kata pertama hingga kata terakhir dengan mencari kata pada sebelum dan sesudah kata kunci. Aturan deteksi kalimat non-definisi dapat dilihat pada Gambar 5 berikut.



**Gambar 5.** Flowchart Deteksi Kalimat Non-Definisi [9]

## 2.2 Proses Utama

Proses utama dalam penelitian ini adalah proses pendeteksian keterkaitan antar kalimat dan proses pembangkit pertanyaan yang dilakukan oleh sistem setelah proses *preprocessing*. Proses deteksi keterkaitan antar kalimat dilakukan untuk memeriksa kalimat mana saja yang saling berkaitan, agar dapat memproses banyak kalimat.

### 2.2.1 Deteksi Keterkaitan Antar Kalimat

Proses ini dilakukan dengan cara mencari subjek, predikat dan objek yang sama di seluruh kalimat. Tahapan ini dilakukan secara berurutan, dengan mencari subjek yang sama di seluruh

kalimat kemudian mencari predikat dan objek. Proses ini akan mendapatkan kondisi kalimat mana saja yang akan dibangkitkan pada proses pembangkitan pertanyaan beberapa kalimat. Kondisi deteksi keterkaitan antar kalimat secara lengkap dapat dilihat pada Tabel 2 berikut.

**Tabel 2.** Deteksi Keterkaitan Antar Kalimat

No	Deteksi
1	1. Jika subjek sama / ditemukan 2. Jika predikat beda
2	1. Jika subjek sama / ditemukan 2. Jika predikat sama / ditemukan
3	1. Jika predikat sama / ditemukan 2. Jika objek sama / ditemukan
4	1. Jika subjek sama / ditemukan 2. Jika predikat sama / ditemukan 3. Jika objek sama / ditemukan
5	1. Jika subjek sama / ditemukan 2. Jika predikat sama / ditemukan 3. Jika objek beda

Berdasarkan deteksi keterkaitan antar kalimat pada Tabel 2, Gambar 6 berikut akan menjelaskan alur dari kondisi deteksi keterkaitan antar kalimat.



**Gambar 6.** Flowchart Deteksi Keterkaitan Antar Kalimat

## 2.2.2 Pembangkit Pertanyaan Beberapa Kalimat

Proses pembangkitan pertanyaan beberapa kalimat akan dilakukan menggunakan bantuan *template* berdasarkan aturan deteksi kalimat yang saling berkaitan. Gambar 7 ini merupakan contoh hasil deteksi keterkaitan antara kalimat.

Kondisi 2 dan Kondisi 3 :				
Ayah	membawa	Nenek	ke dokter	Kalimat
PERSON		PERSON		NE Tag
S	P	O	K	Struktur bahasa
Ayah dan Ibu	membawa	Nenek	ke rumah sakit	Kalimat
PERSON		PERSON		NE Tag
S	P	O	K	Struktur bahasa

**Gambar 7.** Contoh Hasil Deteksi Keterkaitan Antar kalimat

Klausa-klausa diatas merupakan hasil deteksi keterkaitan antar kalimat, yang termasuk dalam Kondisi 2 dan Kondisi 3. Klausa diatas dalam proses *NE tagging* dapat mengenali kata Ayah dan Nenek sebagai *PERSON*, hasil tag ini akan membantu dalam proses penentuan *template*. Tabel 3 berikut adalah contoh proses pembangkitan pertanyaan beberapa kalimat berdasarkan *template* pertanyaan.

**Tabel 3.** *Template* Pertanyaan Beberapa Kalimat

Kata Tanya	<i>Template</i> Pertanyaan
Siapa	<p><b>NE Tag :</b> PERSON, ORGANIZATION</p> <p><b>Kondisi :</b> Kondisi 2</p> <p><b>Template :</b> Siapa yang + ...?</p> <p><b>Hasil Pertanyaan :</b> Siapa yang <u>membawa</u> <u>nenek</u> <u>ke dokter</u> P O K1 dan <u>ke rumah sakit</u>? K2</p> <p><b>Jawaban :</b> Ayah</p>
Siapa	<p><b>NE Tag :</b> PERSON, ORGANIZATION</p> <p><b>Template :</b> Siapa yang + ...?</p> <p><b>Kondisi :</b> Kondisi 3</p>

	<p><b>Hasil Pertanyaan :</b>          Siapa yang <u>dibawa ayah dan Ibu</u>                            P                  S  <u>ke dokter dan ke rumah sakit?</u>                            K1                  K2</p> <p><b>Jawaban :</b>          Nenek</p>
--	--

*template non-factoid. Template pertanyaan non-factoid dapat dilihat pada Tabel 5 berikut.*

**Tabel 5. Template Pertanyaan Non-Factoid [4]**

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

### 2.2.3 Pembangkit Pertanyaan Setiap Kalimat

Pembangkitan pertanyaan setiap kalimat terdapat beberapa proses, yaitu deteksi *non-factoid*, pembangkitan pertanyaan *non-factoid* dan pembangkitan pertanyaan *factoid*.

#### 2.2.3.1 Deteksi Non-Factoid

Pada proses pembangkit pertanyaan, kalimat atau klausa yang menjadi masukan sistem akan dideteksi terlebih dahulu pada proses deteksi *non-factoid*. Proses deteksi *non-factoid* dilakukan berdasarkan keberadaan kata kunci khusus *non-factoid*. Daftar kata kunci *non-factoid* dapat dilihat pada Tabel 4 berikut.

**Tabel 4. Kata Kunci Non-Factoid [4]**

Kategori <i>Non-Factoid</i>	Kata Kunci	
	Sebelum Target	Sesudah Target
Definisi	Disebut, dikenal, dinamakan, mendefinisikan	Adalah, yaitu, ialah, merupakan
Alasan	Oleh sebab itu, jadi, memungkinkan adanya, dengan demikian, maka, dikatakan, penyebab terjadinya, sehingga, walau demikian, namun demikian	Sebab, karena, bertujuan
Cara	Berfungsi untuk, berguna untuk	Dengan cara

#### 2.2.3.3 Pembangkit Pertanyaan Factoid

Proses pembangkit pertanyaan *factoid* akan dilakukan apabila klausa tersebut tidak terdeteksi sebagai pertanyaan *non-factoid*. Proses pembangkit pertanyaan *factoid* dilakukan berdasarkan *template* yang disesuaikan berdasarkan hasil NE tag dan analisis sintaksis. *Template* pertanyaan *factoid* dapat dilihat pada Tabel 6 berikut.

**Tabel 6. Template Pertanyaan Factoid [4]**

Kata tanya	Template Pertanyaan
Apa	<b>Fungsi yang ditanyakan :</b> Subjek
	<b>NE Tag :</b> OTHER
	<b>Template :</b> Apa yang + ..... ?
	<b>Fungsi yang ditanyakan :</b> Objek
Siapa	<b>NE Tag :</b> OTHER
	<b>Template :</b> Apa yang + .... ?
	<b>Fungsi yang ditanyakan :</b> Subjek
	<b>NE Tag :</b> PERSON, ORGANIZATION
Mana	<b>Template :</b> Siapa yang + ....?
	<b>Fungsi yang ditanyakan :</b> Objek
	<b>NE Tag :</b> PERSON, ORGANIZATION
	<b>Template :</b> Siapa yang + ..... ?
Mana	<b>Fungsi yang ditanyakan :</b> Keterangan tempat
	<b>NE Tag :</b> LOCATION

#### 2.2.3.2 Pembangkit Pertanyaan Non-Factoid

Klausa yang terdeteksi sebagai kandidat pertanyaan *non-factoid* berdasarkan kata kunci pada proses deteksi *non-factoid* akan membuat sistem membangkitkan pertanyaan menggunakan

	<b>Template :</b> (di / ke / dari) + mana + ..... ?
Kapan	<b>Fungsi yang ditanyakan :</b> Keterangan waktu
	<b>NE Tag :</b> TIME
	<b>Template :</b> Kapan + ..... ?

### 3. PENUTUP

Pendeteksian keterkaitan antar kalimat pada sistem pembangkit pertanyaan otomatis untuk teks berbahasa Indonesia berhasil mendeteksi kalimat mana saja yang dapat dibangkitkan menjadi pertanyaan. Template pertanyaan yang dikembangkan, untuk menangani banyak kalimat pun dapat diterapkan, sehingga sistem tidak hanya membangkitkan pertanyaan perkalimat saja. Setelah dilakukan pengujian dengan menggunakan 50 teks masukan, sistem ini menghasilkan 413 pertanyaan dengan tingkat akurasi seluruh pertanyaan sebesar 90,80%.

Saran untuk penelitian berikutnya untuk meningkatkan akurasi sistem adalah sebagai berikut. Pertama, dapat melakukan penelitian lebih mendalam pada proses pengenalan entitas bernama (NER) dan jenis kata (POS Tag) karena proses pada sistem ini bergantung pada aplikasi penunjang. Kesalahan NER akan berdampak pada penentuan *template* pertanyaan yang tidak sesuai dan kesalahan POS Tag akan berdampak pada proses identifikasi frasa sehingga gramatikal yang dibentuk menjadi salah. Kedua, pada penelitian selanjutnya dapat ditambahkan proses analisis semantik antar kalimat untuk deteksi keterkaitan antar kalimat. Karena deteksi yang dilakukan hanya mencari gramatikal yang sama di setiap kalimat. Hal ini perlu dilakukan karena terdapat kesalahan pada gramatikal yang sama tetapi mempunyai makna yang berbeda, sehingga sistem akan membangkitkan pertanyaan yang salah. Ketiga, pada penelitian selanjutnya dapat ditambahkan proses *coreference* sebelum pertanyaan dibangkitkan. *Coreference* dibutuhkan ketika terdapat dua kata atau lebih yang mengarah pada acuan yang sama, sistem dapat mempertimbangkan kata atau frasa mana yang akan dimasukkan ke pertanyaan dan jawaban. Keempat, 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 didalam kalimat seakan-akan terdiri dari satu klausa padahal terdapat lebih dari satu klausa. Diperlukan

analisis yang lebih mendalam untuk menangani kalimat majemuk rapatan dan campuran, karena diperlukan penanganan yang berbeda dengan kalimat majemuk setara.

### DAFTAR PUSTAKA

- [1] M. Heilman, "Automatic Factual Question Generation from Text," Carnegie Mellon University, 2011.
- [2] K. Beulen and H. Ney, "Automatic Question generation for Decision Tree Based State Tying," University of Technology, 1998.
- [3] M. R. Iqbal. 2016. "Pembangkit Pertanyaan Otomatis Untuk Teks Berbahasa Indonesia Berdasarkan Template Sintaksis". Skripsi. UNIKOM.
- [4] D. D. Ginanjar. 2017 "Pembangkit Pertanyaan otomatis untuk Teks Berbahasa Indonesia yang Mengandung Kalimat Majemuk". Skripsi. UNIKOM.
- [5] F. Ferdian, "Implementation of Semantic Analyzer in Indonesian Text-Understanding Evaluation System," pp. 6–10, 2012.
- [6] F. Ferdian, "Implementation of Semantic Analyzer in Indonesian Text-Understanding Evaluation System," Bandung Institute of Technology, pp. 6–10, 2012.
- [7] M. Fachri, "Named Entity Recognition For Indonesian Text Using Hidden Markov Model," Universitas Gadjah Mada, 2014.
- [8] Y. Syaifudin, "Quotations Identification From Indonesian Text Using Hidden Markov Model," Universitas Gadjah Mada, 2014.
- [9] W. Hidayat. 2017 "Pendeteksian Kalimat Non-Definisi Pada Pembangkit Pertanyaan Otomatis Untuk Teks Berbahasa Indonesia". Skripsi. UNIKOM.
- [10] R. Susanto, "Perbandingan Model Waterfall dan Prototyping untuk Pengembangan Sistem Informasi," Universitas Komputer Indonesia, 2016.