QUESTION ANSWERING SYSTEM DAN PENERAPANNYA PADA ALKITAB

Gunawan dan Gita Lovina
Jurusan Teknik Informatika, Sekolah Tinggi Teknik Surabaya
e-mail: gunawan@stts.edu; g_lovina@yahoo.com

ABSTRAK: *Question answering system (QA system)* adalah sistem yang mengijinkan user menyatakan kebutuhan informasinya dalam bentuk *natural language question* (pertanyaan dalam bahasa alami), dan mengembalikan kutipan teks singkat atau bahkan frase sebagai jawaban. Ketersediaan sumber informasi yang luas dan bervariasi, serta adanya perkembangan yang pesat dari teknik *Natural Language Processing* (NLP), *Information Extraction* (IE), dan *Information Retrieval* (IR) sangat mempengaruhi perkembangan dari QA *system*, yang mana dulunya hanya dapat menjawab pertanyaan-pertanyaan yang terbatas pada suatu bidang (*domain*) tertentu dengan berdasarkan pada sumber informasi yang terstruktur seperti database, hingga kini dapat menjawab berbagai jenis pertanyaan dengan bersumber pada informasi dari sebuah koleksi teks yang tidak terstruktur. Umumnya arsitektur *question answering system* yang berbasis teks dibangun atas enam tahapan proses, yaitu analisis pertanyaan, *preprocessing* koleksi dokumen, pemilihan kandidat dokumen, analisis kandidat dokumen, ekstraksi jawaban, dan pemberian respon. Aplikasi-aplikasi QA *system* (baik yang dapat diakses melalui internet maupun tidak) yang dikembangkan dengan ciri khasnya masing-masing memiliki urutan proses yang tidak jauh berbeda satu dengan lainnya. Jawaban yang dikembalikan oleh sebuah QA *system* sebagai respon terhadap pertanyaan perlu dievaluasi untuk menilai performansi sistem. Tulisan ini dilengkapi dengan sebuah aplikasi QA system dengan menggunakan Alkitab berbahasa Inggris versi *World English Bible* (WEB) sebagai sumber informasi untuk menjawab pertanyaan. Pemilihan domain Alkitab menyebabkan sejumlah pertanyaan yang dapat diajukan terbatas pada informasi yang tersimpan dalam Alkitab itu sendiri. Selain itu pertanyaan juga dibatasi pada tiga buah tipe jawaban, yaitu *Person, Location*, dan *Date*.

Kata kunci: *question answering system, natural language processing, information retrieval, information extraction, named entity recognition, part of speech tagging.*

ABSTRACT: *Question answering system is a system that allows user to state his or her information need in the form of natural language question, and return short text excerpts or even phrases as an answer. The availability of a wide and various information source and improvements in the techniques of natural language processing, information extraction (wrapper), and information retrieval give a big effect on the development of question answering system, from just answering questions in a specific domain by consulting to structured information source such as database, and like in this research, answering any questions based on information stored in an unstructured text collection. A general architecture of question answering system based on text consists of six processing stages, i.e. question analysis, document collection preprocessing, candidate document selection, candidate document analysis, answer extraction, and response generation. Application of question answering system like AnswerBus, Mulder, and Webclopedia that are developed with its own characteristics has similar processing steps as in the general architecture. Answers returned by a question answering system need to be evaluated for performance measure. This research completed with a simple question answering system application using english Bible in World English Bible (WEB) version as the source of information to answer some questions. Because specific domain is selected: Bible, questions that can be posed by user could ask about information in the Bible itself only. Question is also limited to three types of answers that can be supported by the application: person (who), location (where), and date (when).*

Keywords*: question answering system, natural language processing, information retrieval, information extraction, named entity recognition, part of speech tagging.*

PENDAHULUAN

Saat ini, internet telah berkembang menjadi salah satu media elektronik yang menyediakan informasi yang luas mengenai segala bidang kehidupan. Oleh karena ketersediaan informasi yang memadai, dan cara akses yang cukup mudah, sebagian besar masyarakat di dunia kerap kali memanfaatkan internet untuk memenuhi kebutuhan informasinya.

Dalam upaya untuk mengumpulkan informasi mengenai sebuah topik tertentu dalam waktu yang relatif singkat, pengguna internet sering memanfaatkan fasilitas search engine atau directory seperti halnya Google, Yahoo, AltaVista, dan lain sebagainya. User hanya perlu menginputkan sebuah *query*

yang berkaitan dengan topik dari informasi yang diinginkan pada *text box* yang disediakan, dan kemudian menekan tombol "Search" atau tombol "Cari" yang akan memerintahkan search engine untuk mengumpulkan dan menampilkan daftar dokumen yang sesuai dengan query yang diberikan oleh user.

Seiring dengan perkembangan teknologi dan ilmu pengetahuan, maka informasi yang tersimpan pada internet juga akan berkembang menjadi semakin luas. Hal ini tentu saja dapat mempengaruhi performansi dari sebuah *search engine*. Semakin banyak informasi yang terkandung pada internet, maka search engine akan mengembalikan dokumen yang lebih banyak untuk sebuah query, dan pada akhirnya tetap membiarkan user untuk menggali sendiri informasi yang diinginkan dari kumpulan koleksi teks yang besar. Hal ini tentunya sangatlah tidak efisien, baik dari segi waktu maupun dari segi keakuratan jawaban yang diinginkan. Oleh karena itu, dibutuhkan sebuah sistem baru yang tidak hanya mudah untuk digunakan, tetapi juga mampu mengembalikan jawaban dari pertanyaan user secara langsung. Sistem yang dikenal sebagai *Question Answering* (QA) *system* ini mengijinkan user untuk menginputkan pertanyaan dalam bahasa natural, yaitu bahasa yang digunakan dalam percakapan sehari-hari, dan memperoleh jawaban dengan cepat serta ringkas, atau bahkan disertai dengan kalimat yang cukup untuk mendukung kebenaran dari jawaban tersebut.

Search engine yang ada saat ini dapat mengembalikan daftar dokumen yang telah diurutkan berdasarkan tingkat relevansi dari dokumen tersebut terhadap query user, tetapi tidak memberikan jawaban kepada user. Jadi, semakin sulit menemukan jawaban dengan memanfaatkan search engine, maka QA system akan semakin dibutuhkan karena memberikan banyak keuntungan dengan adanya sumber pengetahuan yang luas, dan dapat mengatasi sejumlah data yang tidak berguna.

Keanekaragaman kebutuhan informasi user dan hasil yang menjanjikan telah mendorong minat dan aktivitas internasional terhadap QA system. Minat serta aktivitas pada bidang ini semakin meningkat dengan adanya undangan pada komunitas penelitian untuk menghadiri dan berpartisipasi dalam berbagai konferensi maupun kompetisi yang bertujuan untuk membahas performasi, kebutuhan, kegunaan, dan tantangan dari QA system.

PEMAHAMAN QA SYSTEM

QA system merupakan sebuah sistem yang mengijinkan user menyatakan kebutuhan informasinya dalam bentuk yang lebih spesifik dan alami, yaitu dalam bentuk natural language question, dan tidak mengembalikan daftar dokumen yang harus disaring oleh user untuk menentukan apakah dokumen-dokumen tersebut mengandung jawaban atas pertanyaan, tetapi mengembalikan kutipan teks singkat atau bahkan frase sebagai jawaban [10].

Sebagai contoh, apabila seorang user ingin mengetahui tempat kelahiran dari mantan presiden Amerika Serikat, George Washington, user dapat memberikan input pertanyaan seperti "*Where was George Washington born?*" pada QA system, dan memperoleh output berupa potongan teks atau frase "*Westmoreland County*" sebagai jawaban. Lain halnya dengan search angine, apabila user menginputkan pertanyaan di atas sebagai query, maka akan dikembalikan daftar dokumen yang memuat keyword query, dan untuk menemukan jawaban yang diinginkan, user harus menyaring setiap dokumen dengan seksama.

Pengembangan sebuah QA system yang baik bukan merupakan tugas yang mudah. Dibutuhkan pengetahuan dari berbagai disiplin ilmu seperti Natural Language Processing, Information Extraction, dan Information Retrieval. Selain itu, ada beberapa faktor yang perlu diperhatikan, yaitu: sumber informasi yang akan digunakan untuk menjawab pertanyaan (database, corpus, Web), user, tipe pertanyaan, format dan cara menghasilkan jawaban, kebutuhan akan sumber linguistik, evaluasi, dan penyajian output. Ketujuh faktor ini baik secara langsung maupun tidak langsung akan menentukan kompleksitas dan performansi dari QA system.

Natural language question answering system bukanlah merupakan area penelitian yang baru. Berikut ini diberikan beberapa contoh aplikasi QA system yang dikembangkan sejak tahun 1960-an:

- *Natural Language Database System.*
  QA system ini menggunakan database tradisional untuk menyimpan data atau fakta-fakta yang dapat dipertanyakan. Database di-query dengan menggunakan pertanyaan bahasa alami yang terlebih dahulu diterjemahkan ke dalam sebuah query database seperti halnya SQL. Contoh: BASEBALL dan LUNAR.
- *Dialogue System.*
  Memodelkan dialog antara manusia dengan komputer. Contoh dari sistem ini ialah SHRDLU yang dibangun sebagai simulasi lengan robot yang dapat mematuhi perintah dengan menggerakkan balok-balok berwarna, dan menjawab pertanyaan seputar keadaan dari balok-balok berwarna tersebut.

- *Reading Comprehension System.*
  QA system ini terlebih dahulu akan melakukan pemahaman pada sebuah teks atau cerita yang digunakan sebagai sumber informasi. Selanjutnya, user dapat mengajukan pertanyaan untuk menguji apakah sistem telah memahami teks dengan baik atau tidak. Contoh: QUALM.

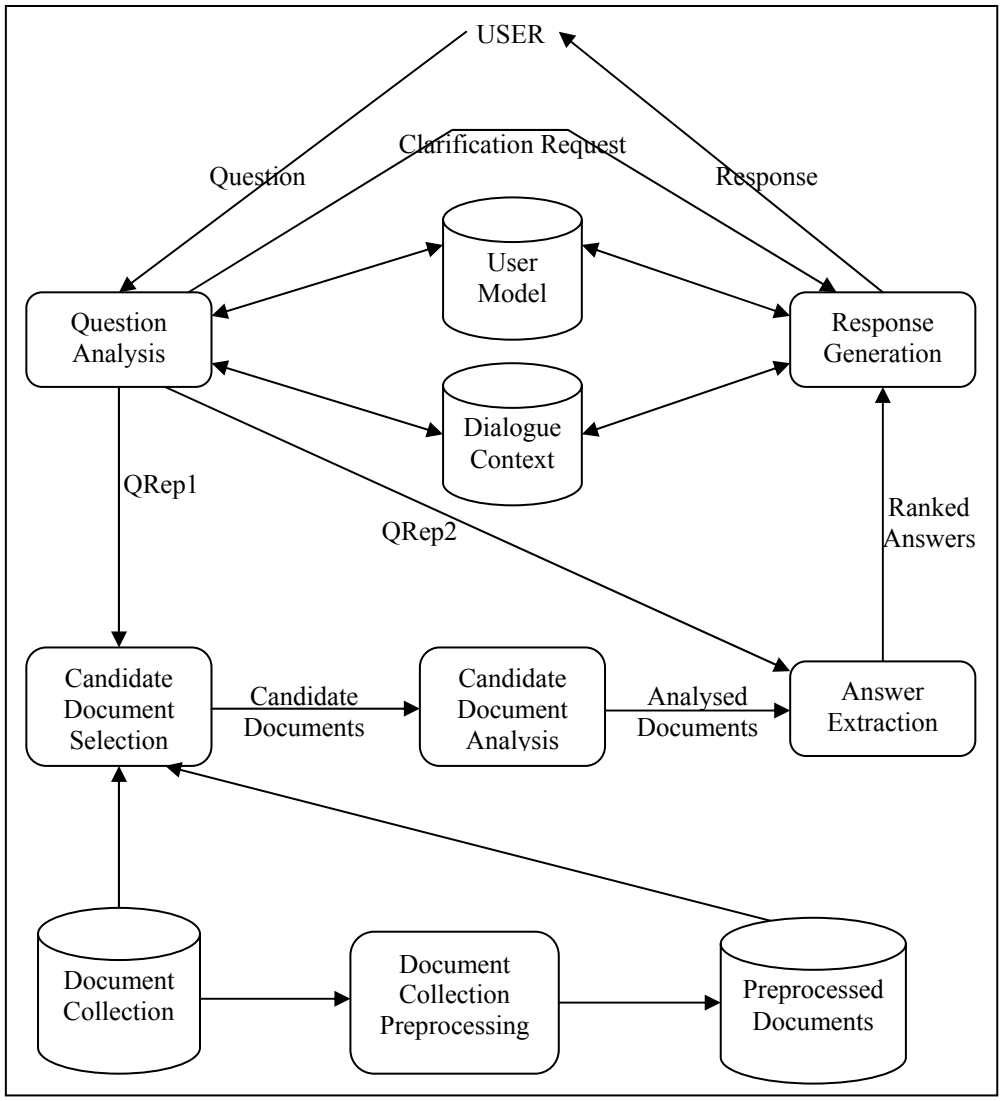
Ketertarikan akan pengembangan QA system belakangan ini banyak dipengaruhi oleh Text REtrieval Conference (TREC) dan World Wide Web (WWW), serta terfokus pada tugas-tugas yang akan menjawab pertanyaan-pertanyaan mengenai segala bidang.

ARSITEKTUR QA SYSTEM

QA system yang dikembangkan dengan tujuan, sumber informasi, dan teknik yang berbeda dapat memiliki arsitektur yang berbeda pula. Gambar 1 menunjukkan arsitektur umum dari QA system yang tersusun atas enam tahapan proses, yaitu: *question analysis*, *document collection preprocessing*, *candidate document selection*, *candidate document analysis*, *answer extraction*, dan *response generation* [5].

Question Analysis

Pertanyaan yang diinputkan oleh user terkadang harus memenuhi beberapa persyaratan tertentu, seperti terbatas pada penggunaan kosakata dan sintak. Di lain pihak, pada kalimat pertanyaan yang diinputkan secara eksplisit, mungkin saja terkandung input yang bersifat implisit. Sebagai contoh, apabila sistem mendukung dialog tertulis yang berkesinambungan, maka pada kalimat pertanyaan mungkin akan terdapat *ellipsis* (penghilangan kata dalam kalimat, yang mana artinya telah dimengerti) atau *anaphora* (penggunaan kata ganti untuk menunjukkan hal-hal yang telah disebutkan sebelumnya) yang membutuhkan akses pada komponen dialogue context untuk diterjemahkan. Selain ellipsis dan anaphora, input implisit juga dapat berupa pengetahuan sistem mengenai user yang disimpan dalam komponen user model.



Gambar 1. Arsitektur Umum QA System

Tahap question analysis akan menghasilkan dua buah representasi pertanyaan sebagai output. Representasi yang pertama berupa query yang akan diteruskan pada tahap candidate document selection, dan representasi yang kedua merupakan representasi semantik dari pertanyaan.

Adapun langkah-langkah yang dapat dilakukan untuk menghasilkan query ialah: ekstraksi keyword dari pertanyaan, mengkombinasikan keyword dengan menggunakan operator tertentu, dan mengembangkan query, misalnya dengan menemukan sinonim dan/atau variasi morfologi dari setiap keyword. Sebagai contoh, dari pertanyaan: "*Who won the 1998 Nobel Peace Prize?"*dapat dihasilkan daftar keyword query: {won|win|gain|gained|get|got|acquire|acquired| obtain|obtained, 1998, Nobel, Peace, Prize}.

Representasi semantik dari pertanyaan mengarah pada jenis informasi atau tipe jawaban yang diinginkan oleh pertanyaan. Misalnya, pertanyaan "*When*…" menginginkan jawaban berupa *Date* (tanggal) atau *Time* (waktu), "*Where*…" menanyakan *Location* (Lokasi), dan "*Who*…" menginginkan jawaban yang berupa *Person* (orang) atau *Organization* (organisasi).

Dua buah pendekatan sederhana yang dapat dilakukan untuk menentukan tipe jawaban dari sebuah pertanyaan ialah:

- Membangun hirarki class pertanyaan berdasarkan tipe jawaban yang diinginkan, dan berusaha untuk menempatkan pertanyaan pada class yang tepat, sesuai dengan hirarki yang ada, sehingga dapat diperoleh representasi semantik yang tepat pula. Tabel 1 menunjukkan contoh hirarki class pertanyaan dan tipe jawaban pada LASSO QA system [9].

Tabel 1. Contoh Hirarki Class Pertanyaan dan Tipe Jawaban pada LASSO QA System

Q-Class	Q-Subclass	Tipe Jawaban
who		Person/Organization
how		
	how many	Number
	how long	Time/Distance
name		
	name-where	Location

- Membangun daftar pola pertanyaan yang didukung oleh sistem dan mengarahkan pola pertanyaan tersebut pada sebuah tipe jawaban yang sesuai. Kecocokkan antara pertanyaan user dengan salah satu pola yang ada akan menentukan tipe jawaban yang diinginkan. Tabel 2 menunjukkan contoh pola klasifikasi pertanyaan pada TEQUESTA QA system [10].

Tabel 2. Contoh Pola Klasifikasi Pertanyaan pada TEQUESTA QA System

Tipe Jawaban	Pola
agent	/[Ww]ho /, / by whom[\.\?]/
date	/[Ww]hen/,/[Ww](hat|hich)year/
location	/[Ww]here(\'s)?/, / is near what /

Selain pembentukan hirarki class pertanyaan dan pencocokan pola, terdapat alternatif lain yang tentunya lebih rumit dan lebih kompleks. Ada sistem yang menggunakan pendekatan *machine learning* yang dikenal dengan Support Vector Machine (SVM), dan ada juga sistem yang melakukan proses *parsing* terhadap pertanyaan, dan kemudian menerapkan sejumlah *rule* (aturan) terhadap parse tree yang dihasilkan untuk menentukan representasi semantik dari pertanyaan.

Apabila sistem mengalami kesulitan dalam menganalisis pertanyaan untuk menghasilkan dua buah representasi yang diharapkan, sistem dapat meminta klarifikasi dari user untuk lebih memperjelas, atau bahkan mengubah pertanyaannya agar dapat dianalisis kembali oleh sistem, sehingga pada akhirnya dapat menyajikan informasi yang diinginkan.

Document Collection Preprocessing

Proses kedua yang menyusun arsitektur umum QA system adalah preprocessing koleksi dokumen. Proses ini bertujuan untuk mempermudah akses pada koleksi dokumen dalam upaya menemukan jawaban yang tepat atas pertanyaan user dalam waktu yang sesingkat mungkin. Sebagian besar QA system yang berpartisipasi pada TREC question answering track melakukan preprocessing dengan mengandalkan *document indexing engine*. Akan tetapi, tahap preprocessing tidak hanya terbatas pada indexing saja.

Berikut ini adalah tiga buah alternatif lain yang dapat digunakan untuk melakukan preprocessing terhadap koleksi dokumen:

- *Part-of-speech tagging*.
  Proses ini bertujuan untuk mengidentifikasi jenis (part-of-speech) dari tiap kata yang menyusun kalimat. Contoh:
  The/DT telephone/NN was/VBD invented/
  VBN by/IN Alexander/NNP Graham/NNP
  Bell/NNP in/IN 1876/CD ./. [1]
  Sebagai proses lanjutan dari part-of-speech tagging, dapat dilakukan *chunking*, yaitu melaku-

kan pengenalan frase (noun phrase, verb phrase, dan lain sebagainya) dengan berdasarkan pada informasi part-of-speech yang telah diidentifikasi. Adapun hasil dari proses chunking terhadap kalimat diatas adalah sebagai berikut:

[NP The/DT telephone/NN] [VP was/VBD invented/VBN] [P by/IN] [NP Alexander/NNP Graham/NNP Bell/NNP] [P in/IN] [NP 1876/CD] ./. [1]

- *Named entity recognition*.
  Melakukan pengenalan sejumlah entity seperti Person, Location, Organization, Date, Money, dan entity lainnya pada teks. Gambar 2 menunjukkan contoh hasil dari tugas pengenalan entity pada Message Understanding Conference-6 (MUC-6) terhadap sebuah kalimat [4]. Kata atau frase yang dikenali sebagai entity ditandai dengan menyisipkan pasangan tag SGML sesuai dengan tipe atau jenis dari entity yang bersangkutan.

```
Mr. <ENAMEX
TYPE="PERSON">Dooner</ENAMEX> met with
<ENAMEX TYPE="PERSON">Martin
Puris</ENAMEX>, president and chief executive
officer of <ENAMEX
TYPE="ORGANIZATION">Ammirati &
Puris</ENAMEX>, about <ENAMEX
TYPE="ORGANIZATION">McCann</ENAMEX>'s
acquiring the agency with billings of <NUMEX
TYPE="MONEY">$400 million</NUMEX>, but
nothing has materialized.
```

Gambar 2. Contoh Named Entity Recognition pada MUC-6

- Mengubah teks dokumen ke dalam bentuk logika, atau ke dalam bentuk lain agar mudah diakses. Sebagai contoh, pada START QA system, teks kalimat diubah ke dalam sebuah ekspresi yang dikenal dengan istilah *T-expression*. Ekspresi ini menunjukkan hubungan antara subyek dan obyek dalam kalimat melalui bentuk <subject relation object> seperti: <<Bill surprise Hillary> with answer> yang terbentuk dari kalimat "*Bill surprised Hillary with his answer*" [6].

Candidate Document Selection

Dengan berdasarkan pada query yang dihasilkan dari proses analisis pertanyaan, tahap candidate document selection akan mengidentifkasi dokumen-dokumen yang mungkin mengandung jawaban atas sebuah pertanyaan. Proses ini biasanya diselesaikan dengan bantuan satu atau lebih search engine, baik yang sudah ada, maupun yang dikembangkan sendiri sesuai dengan kebutuhan sistem.

Candidate Document Analysis

Dokumen-dokumen yang dikembalikan oleh search engine akan dianalisis lebih lanjut dengan tujuan untuk memperkecil ukuran dokumen yang nantinya akan diakses untuk ekstraksi jawaban. Proses analisis kandidat dokumen ini sebenarnya tidak diperlukan lagi apabila sistem telah melaksanakan preprocessing secara lengkap terhadap semua dokumen, atau analisis ini memang tidak didesain sebagai bagian dari sebuah QA system.

Umumnya, QA system melakukan analisis kandidat dokumen dengan menerapkan teknik yang serupa dengan alternatif teknik yang dapat digunakan pada tahap preprocessing koleksi dokumen. Sebagai tambahan, dapat juga dilakukan pemisahan kalimat (*sentence splitting*), paragraf, atau bagian teks tertentu terhadap setiap dokumen.

Answer Extraction

Tahap answer extraction bertugas untuk mencocokkan kandidat dokumen dengan representasi semantik dari pertanyaan, dan kemudian menghasilkan daftar jawaban yang telah diurutkan berdasarkan probabilitas kebenarannya.

Proses pertama yang dilakukan pada tahap ini adalah menemukan sebuah unit teks (mungkin berupa kalimat apabila pada proses sebelumnya dilakukan sentence splitting) dari kandidat dokumen yang mengandung string dengan tipe semantik sesuai dengan tipe jawaban yang diharapkan.

Selanjutnya, sejumlah batasan akan diterapkan pada unit teks yang ditemukan, seperti kesesuaian antara term (keyword) query dengan term penyusun unit teks. Penerapan batasan ini akan menentukan kelayakan dari setiap unit teks untuk dipertimbangkan sebagai kandidat jawaban.

Respon Generation

Setelah sistem berhasil mengekstrak jawaban, maka proses selanjutnya ialah menentukan respon yang akan dihasilkan, yaitu bagaimana jawaban tersebut akan disajikan. Bentuk penyajian jawaban dari QA system yang berbeda tentunya juga berbeda. Ada sistem yang menyajikan daftar dokumen (contoh: AskJeeves [16]), daftar paragraf (contoh: IONAUT), daftar kalimat (contoh: AnswerBus [15]), atau daftar frase jawaban (contoh: Mulder [7]) sebagai respon.

Respon yang dihasilkan, baik itu berupa jawaban atau permintaan klarifikasi, dapat dipengaruhi dan/atau mempengaruhi komponen user model dan dialogue context. Apabila sistem menemui kesulitan dalam menghadapi sebuah pertanyaan, sistem dapat

meminta konfirmasi dari user untuk lebih memperjelas pertanyaannya, atau meminta konfirmasi dengan memberikan sejumlah alternatif pertanyaan lain dengan berdasarkan pada pengetahuan mengenai minat atau pekerjaaan dari user yang bersangkutan, atau dengan berdasarkan pada sejumlah pertanyaan serupa yang pernah dipertanyakan kepada sistem.

EVALUASI

Performansi dari sebuah QA system diukur dengan melakukan evaluasi terhadap jawaban yang dikembalikan sebagai output dari input pertanyaan. Evaluasi dapat dilaksanakan sendiri oleh sistem yang bersangkutan dengan melakukan perbandingan dengan output jawaban yang dihasilkan oleh QA system yang lain, atau dengan berpartisipasi pada konferensi evaluasi yang umum digelar setiap tahunnya seperti Text REtrieval Conference.

TREC-8 question answering track merupakan evaluasi berskala besar pertama terhadap sistem yang mengembalikan jawaban, bukan daftar dokumen, sebagai respon atas sebuah pertanyaan [13]. Adapun spesifikasi dari TREC-8 question answering track adalah sebagai berikut:

- Setiap peserta disediakan sebuah *question set* yang memuat 200 pertanyaan *factoid* dan memiliki jawaban singkat.
- Sebagai sumber informasi, digunakan corpus TREC yang memuat kumpulan dokumen artikel surat kabar.
- Dijamin bahwa pasti akan ada minimal satu dokumen yang menyediakan jawaban dari sebuah pertanyaan.
- Untuk setiap pertanyaan, masing-masing peserta diminta untuk mengembalikan daftar maksimum lima buah jawaban dengan format pasangan kalimat jawaban dan id dokumen dimana jawaban tersebut diekstrak: [*answer-string*, *doc-id*].
- Answer-string dibatasi pada panjang maksimum 50 byte (karakter) bagi sistem yang mengembalikan jawaban pendek, atau maksimum 250 byte bagi sistem yang mengembalikan output berupa jawaban panjang.
- Setiap pertanyaan akan menerima sebuah nilai yang merupakan kebalikan dari ranking posisi dimana jawaban yang benar pertama kali ditemukan. Penilaian ini disebut dengan istilah *reciprocal rank*. Jadi, variasi nilai reciprocal rank yang dapat diberikan adalah: {1; 0,5; 0,33; 0,25; 0,2; 0}.
- Performansi dari setiap sistem peserta diukur dengan melakukan perhitungan *Mean Reciprocal Rank* (MRR) yang merupakan nilai rata-rata dari ke 200 nilai reciprocal rank yang telah ditentukan.

IMPLEMENTASI

Melengkapi tulisan ini diimplementasikan sebuah aplikasi QA system yang menggunakan dokumen Alkitab berbahasa Inggris versi *World English Bible* (WEB), yang merupakan penyempurnaan dari versi *American Standart* (ASV) sebagai sumber informasi untuk ekstraksi jawaban.

Aplikasi yang diberi nama WEBqa ini hanya dapat menjawab pertanyaan-pertanyaan mengenai orang, lokasi, dan hari dari injil-injil perjanjian lama dan baru yang dimuat pada Alkitab. Dengan demikian, kalimat pertanyaan yang dapat diinputkan oleh user juga terbatas pada kalimat pertanyaan seperti "Who…", "Where…", dan "When…".

WEBqa berusaha mengimplementasikan keenam tahapan proses yang membangun arsitektur umum QA system, seperti yang ditunjukkan pada gambar 1. Adapun fungsi-fungsi yang dijalankan pada setiap tahapan proses yang menyusun aplikasi WEBqa akan dijelaskan sebagai berikut.

Preprocessing

Preprocessing koleksi dokumen dijalankan dengan terlebih dahulu melakukan pengenalan kalimat dan tiga buah entity tipe jawaban, yaitu dengan menambahkan pasangan tag:
<Sentence></Sentence>,<Person></Person>,
<Location></Location>, dan <Date></Date> pada teks dokumen. Proses pengenalan kalimat dan entity pada aplikasi ini dilakukan dengan menggunakan ANNIE (A Nearly-New Information Extraction System), yaitu sebuah sistem ekstraksi informasi yang didistribusikan sebagai salah satu komponen dari GATE (General Architecture for Text Engineering) [2]. Gambar 3 menunjukkan contoh pengenalan kalimat dan entity Person, Location, dan Date pada sebuah potongan dokumen Alkitab dengan menggunakan ANNIE pada WEBqa.

Selanjutnya, koleksi dokumen akan di-index dengan menggunakan Lucene [8], yaitu sebuah library Java untuk information retrieval yang akan memberikan kemampuan indexing dan searching pada aplikasi-aplikasi yang dikembangkan dengan menggunakan library ini.

```
<HTML>
    …
    <TITLE>
        <Sentence>John Chapter 11 – World English
        Bible</Sentence>
    </TITLE>
    …
    <BODY>
        …
        <H4>1</H4><Sentence> Now a certain man
        was sick, Lazarus of
        <Person>Bethany</Person>, of the village of
        <Location>Mary</Location> and her sister,
        <Person>Martha</Person>.</Sentence>
        …
        <H4>19</H4><Sentence>Many of the Jews had
        come to <Person>Martha</Person> and
        <Person>Mary</Person>, to console them
        concerning their brother.</Sentence>
        …
        <H4>14</H4><Sentence>So Jesus said to them
        plainly then, 'Lazarus is dead.</Sentence>
        …
        <H4>55</H4><Sentence>Now the
        <Date>Passover</Date> of the Jews was at
        hand.</Sentence> <Sentence>Many went up to
        <Location>Jerusalem</Location> out of the
        country before the <Date>Passover</Date>, to
        purify themselves.</Sentence>.
        …
    </BODY>
</HTML>
```

Gambar 3. Contoh Pengenalan Kalimat dan Entity dengan ANNIE

Question Analysis

Representasi semantik atau tipe jawaban dari pertanyaan ditentukan dengan menerapkan sejumlah aturan sederhana, seperti: apabila kalimat pertanyaan diawali dengan kata tanya "Who", maka dapat ditentukan bahwa tipe jawaban dari pertanyaan tersebut ialah Person. Aturan ini juga berlaku untuk kalimat pertanyaan yang diawali dengan kata 'Where" dan "When", yang mana tipe jawaban dari kedua pertanyaan ini adalah Location dan Date.

Selain ketiga kata tanya di atas, user juga dapat menggunakan kata tanya lain untuk merepresentasikan kebutuhan informasi yang sama. Sebagai pengganti "Who", user dapat menggunakan kata tanya "What <be> the name of", dan untuk menggantikan kata tanya "Where" dan "When", user dapat menggunakan kata tanya "In/On what <noun>" atau "What <be> the <noun>".

Untuk menghasilkan query, sistem terlebih dahulu mengekstrak satu atau lebih keyword dari kalimat pertanyaan dengan membuang kata-kata yang termasuk di dalam daftar *stopword* yang telah didefinisikan sebelumnya, seperti: *a*, *and*, *he*, *in*, *is*, *the*, *what* dan lain sebagainya. Selanjutnya, WEBqa akan menemukan daftar sinonim dari setiap keyword yang berhasil diekstrak dengan menggunakan WordNet [17] yang merupakan sebuah sistem database leksikal untuk bahasa Inggris. Query dibentuk dengan menyisipkan operator "AND" dan "OR" diantara keyword dan sinonimnya.

Candidate Document Selection

Dokumen-dokumen Alkitab yang sesuai dengan query akan ditemukan dengan memanfaatkan kemampuan searching dari Lucene. Sebagai contoh, apabila dari tahap question analysis dihasilkan query: (jesus OR christ OR savior) AND (mother), maka Lucene akan mengembalikan dokumen-dokumen yang mengandung pasangan kata jesus dan mother, atau christ dan mother, atau savior dan mother.

Candidate Document Analysis

Tahap analisis kandidat dokumen akan memecah dokumen ke dalam daftar kalimat dengan menggunakan informasi pasangan tag <Sentence></Sentence> yang telah ditambahkan sebelumnya pada tahap preprocessing. Kalimat-kalimat yang tidak mengandung tag tipe jawaban tidak akan diteruskan pada proses berikutnya karena dianggap tidak dapat menjawab pertanyaan.

Answer Extraction dan Respon Generation

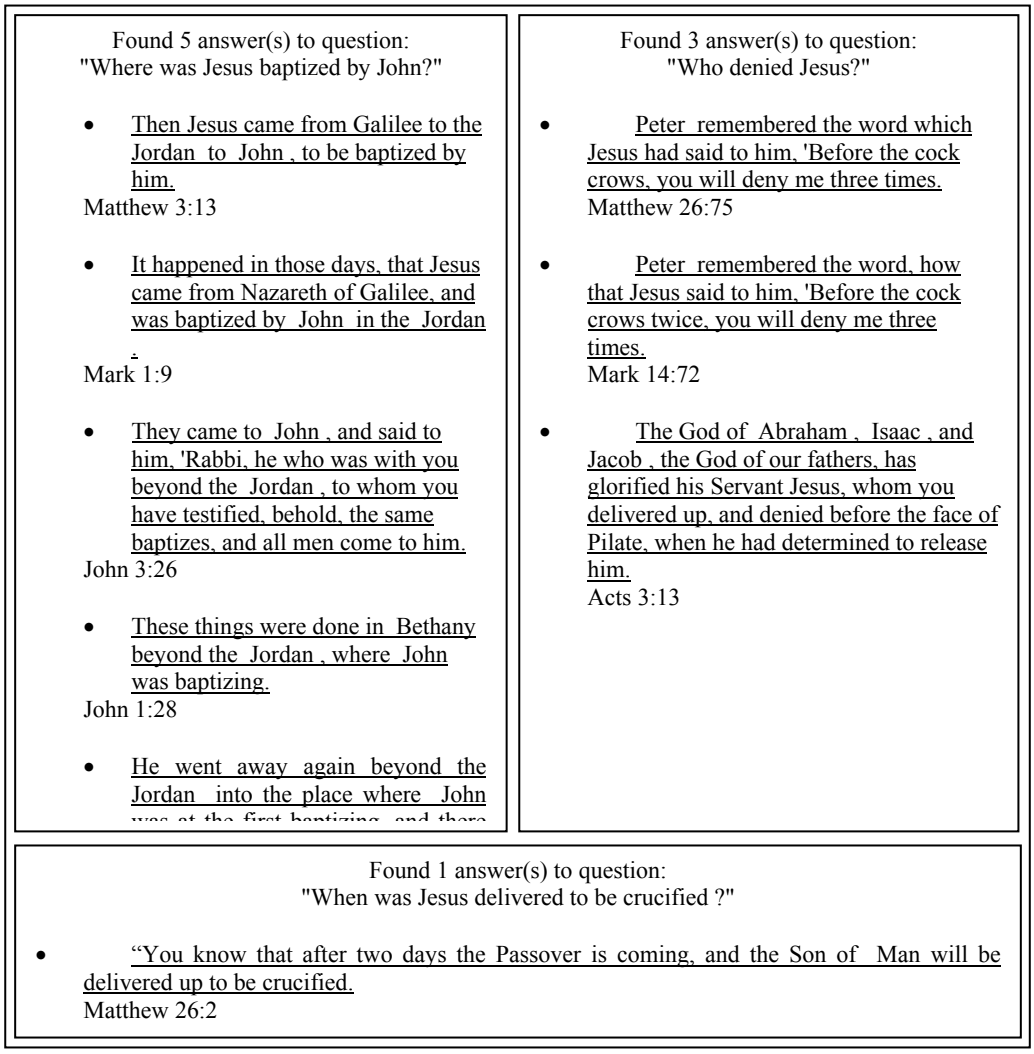
WEBqa akan menetapkan sebuah kalimat sebagai kandidat jawaban apabila kalimat tersebut memenuhi persamaan berikut:

$$q \geq \lfloor \sqrt{Q-1} \rfloor + 1$$

Q merupakan jumlah keyword penyusun query, dan q merupakan jumlah keyword query yang ditemukan pada kalimat. Persamaan diatas merupakan persamaan yang digunakan oleh Zheng dalam menentukan kandidat jawaban pada AnswerBus [18] question answering system.

Selanjutnya, kandidat jawaban akan diurutkan berdasarkan nilai q yang diperoleh, dan maksimum sepuluh kalimat terbaik, dilengkapi dengan link yang menunjuk pada dokumen dari mana kalimat diekstrak akan ditampilkan sebagai respon.

Gambar 4 menunjukkan contoh respon yang dihasilkan oleh WEBqa untuk menjawab pertanyaan: "*Where was Jesus baptized by John*?", "*Who denied Jesus*?", dan "*When was Jesus delivered to be crucified*?"



Gambar 4. Tiga Contoh Output WEBqa

KESIMPULAN DAN PENGEMBANGAN LANJUT

Kesimpulan yang berhasil diperoleh dari penelitian mengenai QA system ini adalah sebagai berikut:

- QA system merupakan bagian dari information retrieval. Sistem ini dapat dipandang sebagai bentuk pengembangan terhadap kemampuan yang dimiliki oleh search engine dengan mengembalikan respon jawaban terhadap query yang berupa pertanyaan natural language.
- QA system dapat dikembangkan pada domain yang beranekaragam, tergantung dari tujuan pengembangan sistem. Domain yang dimaksudkan di sini erat kaitannya dengan sumber informasi yang akan digunakan untuk menjawab pertanyaan, yang secara langsung akan membatasi jenis informasi yang dapat dipertanyakan.
- Teknik yang digunakan dalam mengembangkan sebuah QA system dapat bervariasi, mulai dari teknik yang paling sederhana seperti halnya pencocokan pola, sampai dengan teknik-teknik lain yang lebih kompleks, tergantung dari keterbatasan dan kebutuhan sistem.
- Tiga disiplin ilmu yang sangat mempengaruhi perkembangan QA system ialah Natural Language Processing, Information Extraction, dan Information Retrieval.
- Tahapan proses yang dijalankan oleh sebuah aplikasi QA system tidak akan terlepas dari tiga tahapan proses utama yang menyusun arsitektur umum QA system, yaitu analisis pertanyaan, memilih kandidat dokumen atau segmen dokumen, dan ekstraksi jawaban.
- Dari hasil evaluasi pada TREC-8 question answering track, dan dari hasil evaluasi yang dilakukan oleh AnswerBus secara terpisah, dimana nilai performansi tertinggi hanya mencapai

60% [13,18], dapat diketahui bahwa tugas pengembangan QA system yang dimaksudkan untuk menjawab berbagai jenis pertanyaan yang mencakup segala bidang bukanlah merupakan tugas yang mudah.

Penelitian dan aplikasi QA system yang dibuat masih dapat dikembangkan lagi dengan menggali lebih dalam mengenai tugas-tugas pemahaman bacaan, karena dirasakan bahwa QA system yang dikembangkan dengan menggunakan Alkitab sebagai sumber informasi akan menjadi lebih sempurna apabila sistem terlebih dahulu melakukan pemahaman terhadap kisah-kisah yang terdapat dalam pasal dan ayat-ayat Alkitab.

Untuk memperbesar kemungkinan ditemukannya jawaban, pada tahap preprocessing dari WEBqa dapat dilakukan pengenalan anaphora. Selain itu, perumusan yang digunakan untuk ekstraksi jawaban pada WEBqa dapat diubah atau diganti untuk lebih memperhitungkan kedekatan antar keyword penyusun query pada kandidat kalimat jawaban.

DAFTAR PUSTAKA

1. Buchholz, C., dan W. Daelemans. *Complex Answers: A Case Study using a WWW Qusetion Answering system*. 1998. Cambridge University Press. United Kingdom. Tanggal akses: 31 Jun 2005.
2. General Architecture for Text Engineering. Sheffield Natural Language Processing Group. http://gate.ac.uk/
3. Gospodnetic, O., dan E. Hatcher, Lucene In Action. 2005. Manning Publications Co. Greenwich. USA. Tanggal akses: 05 Feb 2005.
4. Grisham, R., dan B. Sundheim, *Message Understanding Conference-6: A brief history*. Tanggal akses: 21 Mar 2005.
5. Hirschman, L., dan Gaizauskas, R., *Natural language question answering: the view from here*. 2001. Cambridge University Press. United Kingdom. Tanggal akses: 05 Feb 2005.
6. Katz, Boris. F*rom Sentence Processing to Information Access on the World Wide Web*. Massachusetts Institute of Technology. Tanggal akses: 05 Apr 2005.
7. Kwok, C., O. Etzioni, dan D.S.Weld, *Scaling Question Answering to the Web*. November 2000. University of Washington. Seattle. USA. Tanggal akses: 31 Jan 2005.
8. Lucene. The Apache Software Foundation. http://lucene.apache.org/
9. Moldovan, D., Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, dan Vasile Rus, *LASSO: A Tool for Surfing the Answer Net*. 2000. Dallas. Tanggal akses: 26 Feb 2005.
10. Monz, C., *From Document Retrieval to Question Answering*. ILLC Dissertation Series 2003-4. Universiteit van Amsterdam. Amsterdam. Tanggal akses: 19 Feb 2005.
11. START Natural Language Question Answering System. MIT Computer Science and Artificial Intelligence Laboratory. http://start.csail.mit.edu/
12. Text REtrieval Conference. National Institute of Standards and Technology. http://trec.nist.gov/
13. Voorhees, E.M., dan Tice, D.M., *The TREC-8 Question Answering Track Evaluation*. National Institute of Standards and Technology. Gaithersburg. Tanggal akses: 31 Jan 2005.
14. World English Bible. Speaking Bible. USA. http://www.speakingbible.com/freezip/web.zip
15. Website Answer Bus, www.answerbus.com
16. Website Ask Jeeves, www.ask.com.
17. Website WordNet, wordnet.princeton.edu.
18. Zheng, Z., *AnswerBus Question Answering System*. University of Michigan. Tanggal akses: 25 Jan 2005.