

**RANCANG BANGUN APLIKASI TANYA JAWAB MENGGUNAKAN
SEMANTIC WEB DAN ALGORITMA NAZIEF & ADRIANI SEBAGAI
METODE *STEMMER* KALIMAT TANYA**

SKRIPSI

Oleh :

AENY NURWAHDAH

NIM. 10650073



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**RANCANG BANGUN APLIKASI TANYA JAWAB MENGGUNAKAN
SEMANTIC WEB DAN ALGORITMA NAZIEF & ADRIANI
SEBAGAI METODE *STEMMER* KALIMAT TANYA**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
AENY NURWAHDAH
NIM : 10650073 / S-1**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**RANCANG BANGUN APLIKASI TANYA JAWAB MENGGUNAKAN
SEMANTIC WEB DAN ALGORITMA NAZIEF & ADRIANI
SEBAGAI METODE *STEMMER* KALIMAT TANYA**

SKRIPSI

oleh :

AENY NURWAHDAH

NIM : 10650073

Telah Diperiksa dan Disetujui untuk Diuji:

Tanggal: 13 Nopember 2014

Pembimbing I,

Pembimbing II,

A'la Syauqi, M.Kom

NIP : 19771201 200801 1 007

Zainal Abidin, M.Kom

NIP : 19760613 200501 1 004

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP : 197404242009011008

**RANCANG BANGUN APLIKASI TANYA JAWAB MENGGUNAKAN
SEMANTIC WEB DAN ALGORITMA NAZIEF & ADRIANI
SEBAGAI METODE STEMMER KALIMAT TANYA**

SKRIPSI

oleh :

AENY NURWAHDAH

NIM. 10650073

Telah Dipertahankan di Depan Dewan Penguji Skripsi

Dan dinyatakan sebagai Salah Satu Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal 20 Nopember 2014

Susunan Dewan Penguji :	Tanda Tangan
1. Penguji Utama : <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670118 200501 1 001	()
2. Ketua : <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	()
3. Sekretaris : <u>A'la Syaqui, M. Kom</u> NIP. 19771201 200801 1 007	()
4. Anggota : <u>Zainal Abidin, M. Kom</u> NIP. 19760613 200501 1 004	()

Mengesahkan,
Ketua Jurusan Teknik Informatika

Dr.Cahyo Crysdian
NIP. 197404242009011008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Aeny Nurwahdah
NIM : 10650073
Jurusan : Teknik Informatika
Fakultas : Sains Dan Teknologi

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, 13 Nopember 2014

Yang Membuat Pernyataan,

Aeny Nurwahdah
NIM. 10650073

MOTTO

إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّى يُغَيِّرُوا مَا بِأَنْفُسِهِمْ

Sesungguhnya Allah tidak merubah keadaan sesuatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri. (QS. Ar-Ra'd ayat 11)

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ۖ إِنَّ مَعَ الْعُسْرِ يُسْرًا ۖ

Karena Sesungguhnya sesudah kesulitan itu ada kemudahan, Sesungguhnya sesudah kesulitan itu ada kemudahan. (QS. Alam Nasyrah ayat 5-6)





LEMBAR PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Tak terasa lebih dari 4 tahun telah dilalui...

*Rasanya baru kemarin menginjakkan kaki masuk di Kampus Ulul Albab,
Dan kini saatnya mengakhiri perjuangan di kampus tercinta ini..*

*Bukan berhenti menuntut ilmu,
Tapi untuk berjuang di kehidupan yang lebih nyata..
Untuk mengamalkan ilmu agar bermanfaat bagi sesama...*

Kupersembahkan karya ini kepada :

“Orang tua terhebatku, umik Rosidah dan ayah Muchib (alm), terimakasih untuk doa, nasehat, semangat serta limpahan kasih sayang selama ini. Ridhoi-lah putrimu ini, semoga perjuangan kecil ini membuat bangga umik dan ayah, aamiin.. “

“Saudara-saudaraku, mas Shafy, mbak Zil, Inun, Dudy dan Ilut, terimakasih untuk doa dan motivasi nya”

“Terimakasih juga untuk Lek Mus dan Lek Tin, yang senantiasa mendoakan dan selalu mendukung perjuangan saya, jazakumullah khairan katsiir..”

“Terimakasih untuk abah Prof. Dr. Kyai H. Achmad Mudlor, SH (alm) beserta segenap keluarga besar Lembaga Tinggi Pesantren Luhur Malang (LTPLM) yang telah mengizinkan saya untuk tinggal dan memberikan bimbingan spiritual selama saya berada di Malang..”

“Kepada teman-teman seperjuangan Teknik Informatika angkatan 2010 (Infinity), semoga Allah membarokahkan ilmu yang kita dapatkan dan menjadikan kita sukses dunia akhirat, aamiin.. “

“Terimakasih kepada yang diam-diam mendoakan kebaikan kepada saya, siapapun itu, saya yakin karena doa kalian juga saya bisa menyelesaikan skripsi ini. insyaaAllah doa yang baik berbalas kebaikan juga kepada yang mendoakan..”

“Serta kepada semua pihak yang tidak dapat disebutkan satu persatu, yang telah membantu dan mensupport dari awal kuliah hingga selesainya skripsi ini, terimakasih..”



KATA PENGANTAR



Assalamualaikum Wr. Wb.

Syukur alhamdulillah penulis haturkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi ini dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring doa dan harapan *jazakumullah ahsanal jaza'* kepada semua pihak yang telah membantu terselesaikannya skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada :

1. Bapak A'la Syauqi, M.Kom, selaku Dosen Wali sekaligus Dosen Pembimbing I yang telah memberikan bimbingan dan arahan dalam menyelesaikan skripsi ini.
2. Bapak Zainal Abidin, M.Kom, selaku Dosen Pembimbing II yang bersedia meluangkan waktu untuk memberikan masukan dalam penyelesaian skripsi ini.
3. Bapak Dr. Cahyo Crysdian, selaku ketua jurusan Teknik Informatika.
4. Bapak-Ibu Dosen dan seluruh civitas akademik Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah memberikan banyak pelajaran, pengalaman, doa dan dukungan selama ini.
5. Teman-teman seperjuangan TI 2010, khususnya Khullatur Rosyidah, Anindita Caesarini, Dzakiyatur Rosyidah, Morwati, Amalia Eka Astutik,

Rizky Mauludiyah serta Rizal Furqan Ramadhan yang selalu ada dan memotivasi dalam menyelesaikan skripsi ini.

6. Serta seluruh pihak yang telah membantu dalam penyelesaian skripsi ini. Semoga Allah SWT memberikan balasan yang sesuai atas jasa dan bantuan yang telah diberikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Aamiin ya Rabbal Alamiin.*

Wassalamualaikum Wr. Wb.

Malang, 10 Nopember 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
التلخيص	xvi
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Batasan Masalah	6
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian	6
1.6. Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	
2.1. Sistem Tanya Jawab	8
2.2. Semantic Web	11
2.3. Algoritma Nazief & Adriani	21
2.4. Basis Pengetahuan DBpedia	25
BAB III DESAIN DAN PERANCANGAN SISTEM	
3.1. Deskripsi Sistem	30
3.2. Perancangan Sistem	
3.2.1. <i>Use Case Diagram</i> Aplikasi Tanya Jawab	31
3.2.2. <i>Activity Diagram</i> Aplikasi Tanya Jawab	32
3.2.3. <i>Activity Diagram</i> Teks <i>Preprocessing</i>	33
3.2.4. <i>Activity Diagram</i> <i>Tokenizing</i>	34
3.2.5. <i>Activity Diagram</i> <i>Filtering</i>	35
3.2.6. <i>Activity Diagram</i> <i>Stemming</i> Nazief & Adriani	38
3.2.7. <i>Activity Diagram</i> <i>Pencarian Resource</i>	45
3.2.8. Eksekusi <i>Query</i> SPARQL	48
3.3. Sumber Data	49

3.4. Desain Tampilan	50
3.5. Implementasi Tampilan	52

BAB IV HASIL DAN PEMBAHASAN

4.1. <i>Platform</i> yang digunakan	55
4.2. Uji Coba Aplikasi	56
4.3. Analisa Hasil dan Pembahasan	63

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	70
5.2. Saran	71

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR GAMBAR

Gambar 1.1. Contoh hasil pencarian menggunakan Google	3
Gambar 1.2. Contoh hasil pencarian menggunakan <i>Start Natural Language Question Answering System</i>	4
Gambar 2.1. Arsitektur umum <i>Question Answering System</i>	9
Gambar 2.2. <i>Layer</i> pada <i>Semantic Web</i>	14
Gambar 2.3. Contoh <i>query</i> SPARQL	19
Gambar 2.4. Hasil eksekusi <i>query</i> SPARQL	19
Gambar 2.5. Contoh tampilan data pada DBpedia	27
Gambar 2.6. Contoh tampilan data pada Wikipedia	27
Gambar 3.1. <i>Use Case Diagram</i> Aplikasi Tanya Jawab	31
Gambar 3.2. <i>Activity Diagram</i> Aplikasi Tanya Jawab	32
Gambar 3.3. <i>Activity Diagram</i> Teks <i>Preprocessing</i>	33
Gambar 3.4. <i>Activity Diagram</i> <i>Tokenizing</i>	34
Gambar 3.5. <i>Source code</i> proses <i>Tokenizing</i>	35
Gambar 3.6. <i>Activity Diagram</i> <i>Filtering</i>	37
Gambar 3.7. <i>Source code</i> proses <i>Filtering</i>	37
Gambar 3.8. <i>Activity Diagram</i> <i>Stemming</i> Nazief & Adriani	39
Gambar 3.9. <i>Source code</i> proses cek kata dasar	40
Gambar 3.10. <i>Source code</i> proses menghapus <i>Infection Suffixes</i>	40
Gambar 3.11. <i>Source code</i> proses menghapus <i>Derivation Suffixes</i>	42
Gambar 3.12. <i>Source code</i> proses menghapus <i>Derivation Prefix</i>	44
Gambar 3.13. Contoh <i>link</i> url hasil penelusuran	45
Gambar 3.14. <i>Activity Diagram</i> Pencarian <i>Resource</i>	46
Gambar 3.15. <i>Source code</i> proses pencarian <i>Resource</i>	47
Gambar 3.16. Contoh <i>query</i> SPARQL	48
Gambar 3.17. Implementasi <i>resource</i> pada <i>query</i> SPARQL	48
Gambar 3.18. Hasil eksekusi <i>query</i> SPARQL menggunakan <i>resource</i> “Internet”	49
Gambar 3.19. Desain tampilan Aplikasi Tanya Jawab	51
Gambar 3.20. Tampilan Aplikasi Tanya Jawab	53
Gambar 3.21. Contoh Penggunaan Aplikasi Tanya Jawab	54
Gambar 4.1. <i>Overstem</i> yang menyebabkan kerancuan dalam menemukan <i>keyword</i>	65
Gambar 4.2. Contoh pencarian tinggi Monas	65
Gambar 4.3. Contoh pencarian tinggi menara Eiffel	66

DAFTAR TABEL

Tabel 2.1. Kombinasi awalan & akhiran yang tidak diizinkan	24
Tabel 2.2. Aturan pemenggalan kata Algoritma Nazief & Adriani	24
Tabel 2.3. Statistik Data Wikipedia Bahasa Indonesia	26
Tabel 2.4. Tabel Contoh Klasifikasi DBpedia	26
Tabel 3.1. Contoh hasil <i>tokenizing</i>	32
Tabel 3.2. Struktur Tabel Stopword	33
Tabel 3.3. Contoh Isi Tabel Stopword	34
Tabel 3.4. Contoh hasil <i>filtering</i>	36
Tabel 3.5. Struktur Tabel Kata Dasar	36
Tabel 3.6. Contoh Isi Tabel Kata Dasar	39
Tabel 3.7. Contoh hasil <i>Stemming</i> Nazief & Adriani	48
Tabel 4.1. Klasifikasi Jenis Pertanyaan	58
Tabel 4.2. Tabel Uji Coba Pertanyaan Tentang Benda	59
Tabel 4.3. Tabel Uji Coba Pertanyaan Tentang Lokasi/Tempat.....	59
Tabel 4.4. Tabel Uji Coba Pertanyaan Tentang Orang	60
Tabel 4.5. Tabel Uji Coba Pertanyaan Tentang Waktu	60
Tabel 4.6. Tabel Uji Coba Pertanyaan Tentang Sebab	61
Tabel 4.7. Tabel Uji Coba Pertanyaan Tentang Deskripsi	61
Tabel 4.8. Tabel Uji Coba Pertanyaan Tentang Integer	62
Tabel 4.9. Tabel Perbandingan Hasil Uji Coba Pertanyaan	63

ABSTRAK

Nurwahdah, Aeny. 2014. **Rancang Bangun Aplikasi Tanya Jawab menggunakan *Semantic Web* dan Algoritma Nazief & Adriani sebagai metode *stemmer* kalimat tanya**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) A'la Syauqi, M.Kom. (II) Zainal Abidin, M.Kom.

Kata Kunci : aplikasi tanya jawab, *semantic web*, *stemming*, algoritma nazief & adriani, *natural language processing*

Salah satu teknik pencarian informasi adalah dengan menggunakan sistem tanya jawab. Sistem tanya jawab merupakan suatu sistem yang berguna untuk menjawab pertanyaan yang diajukan oleh pengguna. Sistem tanya jawab sudah ada diaplikasikan dalam bentuk *closed-domain* QAS yang menggunakan data terbatas dan *open-domain* QAS yang menggunakan data lebih banyak dan lebih luas.

Skripsi ini mengangkat topik mengenai rancang bangun suatu *open-domain* QAS, yakni implementasi *semantic web* pada aplikasi tanya jawab dan menggunakan Algoritma *stemming* Nazief & Adriani untuk mendapatkan *keyword* sebagai acuan *resource* dalam menemukan *endpoint* DBpedia. Algoritma Nazief & Adriani dipilih karena pada penelitian sebelumnya diperoleh kesimpulan bahwa algoritma ini memiliki akurasi yang lebih tinggi. Hasil analisa dan uji coba menunjukkan bahwa aplikasi mampu mengembalikan jawaban dengan baik. Akurasi ketepatan jawaban yang didapat mencapai 83,81%.

ABSTRACT

Nurwahdah, Aeny. 2014. **Engineering Question Answering Application using Semantic Web and Nazief & Adriani's Algorithm as stemmer method of question sentence**. Thesis. Informatic Engineering Department of Science and Technology Faculty. Maulana Malik Ibrahim State Islamic University of Malang.

Advisers : (I) A'la Syauqi, M.Kom

(II) Zainal Abidin, M.Kom

One of the techniques to pursue information is by using question and answer system. Question and answer system is use to answer questions posed by users. Question and answering system are already apply in the form closed-domain QAS that using limited data and open-domain QAS which use more data and more widely.

This thesis analyzed design open-domain QAS, the implementation of the semantic web in a question and answer application and using algorithm stemming Nazief & Adriani to get keywords as sign resource to find endpoints DBpedia. Algorithm Nazief & Adriani is chosen because the previous studies acquired that using of algorithm has higher accuracy. The results of the experiment show the application is able answer the questions very well. The accuracy of answers obtained accuracy reaches 83.81%.

Key words : question answering system, semantic web, stemming, nazief & adriani algorithm, natural language processing

التلخيص

عيني نور وحدة، ٢٠١٤ م، رانجانج باغون أفليكاسي سؤال وجواب يستخدم سيمانتيك ويب دان ألعارتما نازيف و أدرياني كالطريقة ستييمير الجملة السؤال، المشرف الاول : أعلى شوقي الماجستير، المشرف الثاني : زينال عابدين الماجستير.

الكلمات الأساسية : أفليكاسي السؤال وجواب، سيمانتيك ويب، ستيمينج، ألعارتما نازيف وأدرياني، العملية اللغة الطبيعية

الأسلوب واحد هو يستخدم النظام سؤال وجواب. نظام السؤال وجواب هو النظام الذي يفيد لتجيب الأسئلة المطروحة المستخدمين. نظام السؤال وجواب قد تطبق في شكل مغلق المجال QAS تستخدم بيانات محدودة ومجال مفتوح QAS الذين يستخدمون بيانات أكثر و أوسع. هذا البحث عن رنجنج تصميم والمجال مفتوح QAS ، وهي تنفيذ الشبكة الدلالية في سؤال والإجابة على تطبيق واستخدام ألعارتما ستيمينج نازيف وأدرياني للحصول كيوورد المرجعية في إيجاد نقاط النهاية DBpedia. ألعارتما نازيف وأدرياني اختيار لدراسات سابقة تلخص أن هذه ألعارتما لديها دقة أعلى. نتائج التحليل والاختبار تشير أن التطبيق قادر على استعادة إجابة جيدة. دقة الإجابات التي تم الحصول عليها دقة تصل إلى 83.81 %.

ABSTRAK

Nurwahdah, Aeny. 2014. **Rancang Bangun Aplikasi Tanya Jawab menggunakan Semantic Web dan Algoritma Nazief & Adriani sebagai metode stemmer kalimat tanya.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) A'la Syauqi, M.Kom. (II) Zainal Abidin, M.Kom.

Kata Kunci : aplikasi tanya jawab, *semantic web*, *stemming*, algoritma nazief & adriani, *natural language processing*

Salah satu teknik pencarian informasi adalah dengan menggunakan sistem tanya jawab. Sistem tanya jawab merupakan suatu sistem yang berguna untuk menjawab pertanyaan yang diajukan oleh pengguna. Sistem tanya jawab sudah ada diaplikasikan dalam bentuk *closed-domain* QAS yang menggunakan data terbatas dan *open-domain* QAS yang menggunakan data lebih banyak dan lebih luas.

Skripsi ini mengangkat topik mengenai rancang bangun suatu *open-domain* QAS, yakni implementasi *semantic web* pada aplikasi tanya jawab dan menggunakan Algoritma *stemming* Nazief & Adriani untuk mendapatkan *keyword* sebagai acuan *resource* dalam menemukan *endpoint* DBpedia. Algoritma Nazief & Adriani dipilih karena pada penelitian sebelumnya diperoleh kesimpulan bahwa algoritma ini memiliki akurasi yang lebih tinggi. Hasil analisa dan uji coba menunjukkan bahwa aplikasi mampu mengembalikan jawaban dengan baik. Akurasi ketepatan jawaban yang didapat mencapai 83,81%.

ABSTRACT

Nurwahdah, Aeny. 2014. **Engineering Question Answering Application using Semantic Web and Nazief & Adriani's Algorithm as stemmer method of question sentence**. Thesis. Informatic Engineering Department of Science and Technology Faculty. Maulana Malik Ibrahim State Islamic University of Malang.

Advisers : (I) A'la Syauqi, M.Kom

(II) Zainal Abidin, M.Kom

One of the techniques to pursue information is by using question and answer system. Question and answer system is use to answer questions posed by users. Question and answering system are already apply in the form closed-domain QAS that using limited data and open-domain QAS which use more data and more widely.

This thesis analyzed design open-domain QAS, the implementation of the semantic web in a question and answer application and using algorithm stemming Nazief & Adriani to get keywords as sign resource to find endpoints DBpedia. Algorithm Nazief & Adriani is chosen because the previous studies acquired that using of algorithm has higher accuracy. The results of the experiment show the application is able answer the questions very well. The accuracy of answers obtained accuracy reaches 83.81%.

Key words : question answering system, semantic web, stemming, nazief & adriani algorithm, natural language processing

التلخيص

عيني نور وحدة، 2014 م، رانجانج باغون أفليكاسي سؤال وجواب يستخدم سيمانتيك ويب دان ألغارتما نازيف و أدرياني كالطريقة ستييمير الجملة السؤال، المشرف الاول : أعلى شوقي الماجستير، المشرف الثاني : زينال عابدين الماجستير.

الكلمات الأساسية : أفليكاسي السؤال وجواب، سيمانتيك ويب، ستيمينج، ألغارتما نازيف وأدرياني، العملية اللغة الطبيعية

الأسلوب واحد هو يستخدم النظام سؤال وجواب. نظام السؤال وجواب هو النظام الذي يفيد لتجيب الأسئلة المطروحة المستخدمين. نظام السؤال وجواب قد تطبق في شكل مغلق المجال QAS تستخدم بيانات محدودة ومجال مفتوح QAS الذين يستخدمون بيانات أكثر و أوسع.

هذا البحث عن رنجنج تصميم والمجال مفتوح QAS ، وهي تنفيذ الشبكة الدلالية في سؤال والإجابة على تطبيق واستخدام ألغارتما ستيمينج نازيف وأدرياني للحصول كيوورد المرجعية في إيجاد نقاط النهاية DBpedia. ألغارتما نازيف وأدرياني اختيار لدراسات سابقة تلخص أن هذه ألغارتما لديها دقة أعلى .نتائج التحليل والاختبار تشير أن التطبيق قادر على استعادة إجابة جيدة. دقة الإجابات التي تم الحصول عليها دقة تصل إلى 83.81 %.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Mencari informasi telah menjadi kebutuhan manusia setiap harinya. Dengan informasi yang didapatkan, manusia menjadi tahu segala sesuatu yang sebelumnya tidak diketahui. Karena pada dasarnya, manusia terlahir dalam keadaan tidak mengetahui apa – apa. Sebagaimana salah satu ayat-Nya dalam QS. An-Nahl, Allah SWT berfirman :

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَرَ وَالْأَفْئِدَةَ
لَعَلَّكُمْ تَشْكُرُونَ ﴿٧٨﴾

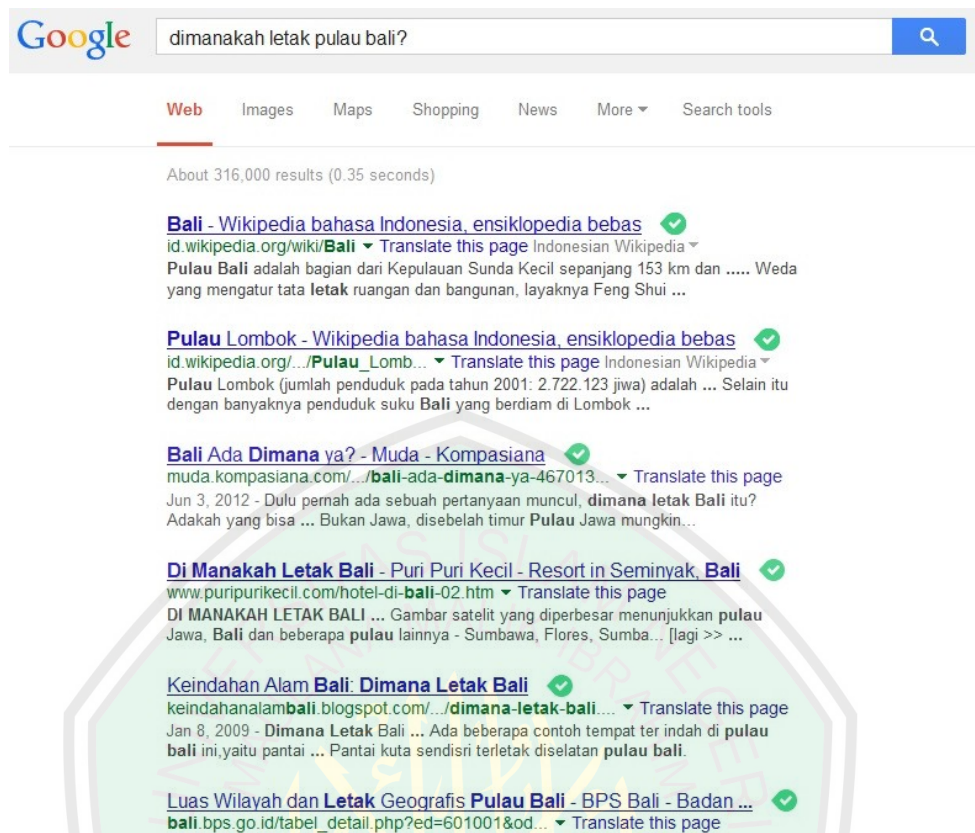
“Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatupun, dan Dia memberi kamu pendengaran, penglihatan dan hati, agar kamu bersyukur” (QS. An-Nahl : 78)

Allah SWT menciptakan manusia dalam keadaan tidak mengetahui apapun namun Allah SWT juga membekali manusia dengan segala fasilitas yang tentu memiliki tujuan. Manusia terlahir tidak serta merta paham mengenai apapun yang ada disekitarnya. Menurut Tafsir Jalalain, ayat ini mengandung penjelasan bahwa Allah melahirkan manusia dari perut ibunya, kemudian Allah menjadikan manusia dapat mengetahui segala sesuatu yang sebelumnya belum diketahui.

Untuk dapat mengetahui segala sesuatu, manusia harus melakukan pencarian informasi dan pengetahuan. Informasi sendiri dapat ditemukan di berbagai sumber, seperti mengamati sekitar, informasi dari buku, perpustakaan bahkan dari internet. Internet sendiri telah menjadi tempat penyimpanan yang universal bagi pengetahuan manusia dan kebudayaan. Sehingga pembagian informasi dan ide mudah untuk diakses. Salah satu negara terbesar pengguna internet adalah Indonesia. Berdasarkan data yang dirilis oleh Internet World Stats, Indonesia menempati urutan 8 dari 20 negara terbesar pengguna internet di seluruh dunia dengan catatan pengguna mencapai 55 juta dari jumlah populasi di Indonesia (Internet World Stats, 2014).

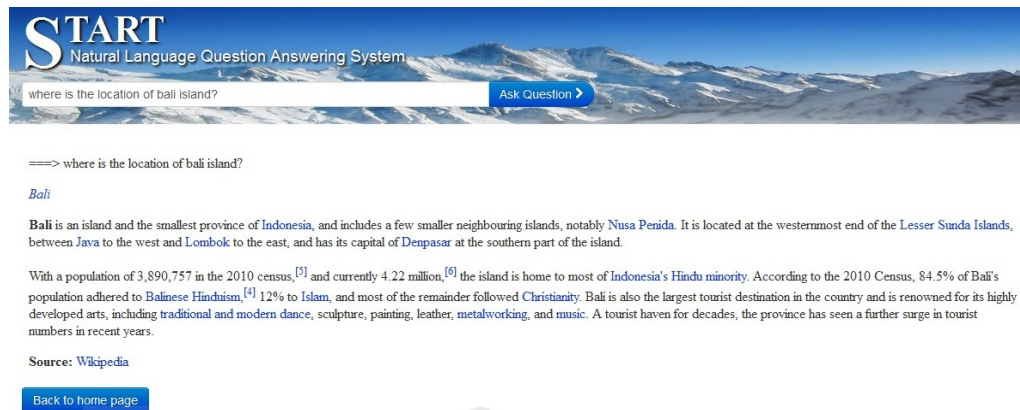
Sebagian besar masyarakat Indonesia menggunakan internet sebagai solusi untuk mendapatkan informasi sesuai dengan yang dibutuhkan. Setiap harinya, masyarakat menggunakan internet untuk mengatasi permasalahan mereka. Namun menemukan informasi yang berguna pada web bisa saja menjadi sulit. Contohnya, untuk mendapatkan informasi yang diinginkan pengguna harus memilih *links* (yang bersambung ke halaman lain yang terkadang tidak sesuai dengan kebutuhan). Hal tersebut dicontohkan pada Gambar 1.1.

Dari masalah tersebut maka perlu dilakukan pembaharuan dalam teknik pencarian. Salah satunya yaitu dengan membangun aplikasi tanya jawab. Aplikasi tanya jawab merupakan suatu sistem yang memungkinkan *user* menyatakan kebutuhan informasinya dalam bentuk kalimat pertanyaan, kemudian mengembalikan kutipan teks sebagai *output* jawaban.



Gambar 1.1. Contoh hasil pencarian menggunakan Google

Contoh sistem tanya jawab yang sudah ada ditunjukkan pada Gambar 1.2. START Natural Language Question Answering System tersebut merupakan sistem tanya jawab pertama berbasis web yang dibangun oleh MIT (*Massachusetts Institute of Technology*). Sistem ini telah di-online-kan dan beroperasi sejak Desember 1993 (START Natural Language Question Answering System, 2014). Sistem ini menggunakan bahasa Inggris sebagai bahasa teknis penggunaan aplikasi. Cara kerja sistem yaitu dengan menginputkan pertanyaan berbahasa Inggris, kemudian sistem mengembalikan jawaban dalam bentuk bahasa Inggris pula.



Gambar 1.2. Contoh hasil pencarian menggunakan *Start Natural Language Question Answering System*

Suatu *Question Answering System* dapat diklasifikasikan menjadi 2 lingkup, yaitu *open-domain QAS* dan *closed-domain QAS*. *closed-domain QAS* berhubungan dengan data terbatas pada basis pengetahuannya. Sedangkan pada *open-domain QAS*, basis pengetahuan yang digunakan lebih banyak dan lebih luas (Irwan, 2012). Untuk membangun suatu *open-domain Question Answering System* seperti *START Natural Language QAS*, dapat digunakan teknologi *semantic web* sebagai teknik untuk penggalian jawaban. Menurut Gunawan (2014), kelebihan *semantic web* adalah untuk menggali sumber pengetahuan dilakukan secara *online*, tidak perlu lagi membuka halaman web satu per satu, namun dapat memanfaatkan agen untuk menjelajahi ribuan situs-situs web.

Namun untuk melakukan penelusuran jawaban pada *semantic web*, diperlukan suatu *resource* sebagai *keyword* dalam pencarian. Untuk itu perlu dilakukan pengolahan pertanyaan yang diajukan, sehingga diperoleh *keyword* yang dibutuhkan. Salah satu tahap dalam pengolahan pertanyaan adalah proses

stemming. *Stemming* merupakan bagian penting dalam proses normalisasi dalam membangun sistem temu kembali informasi. Berdasarkan penelitian mengenai perbandingan metode *stemming* yang dilakukan oleh Agusta (2009), disimpulkan bahwa proses *stemming* dokumen berbahasa Indonesia menggunakan Algoritma Nazief & Adriani memiliki prosentase keakuratan (*precision*) lebih besar dibandingkan dengan Algoritma Porter.

Berdasarkan latar belakang tersebut, maka penelitian mengenai pembangunan Aplikasi Tanya Jawab menggunakan *Semantic Web* dan algoritma Nazief & Adriani sebagai metode *stemmer* kalimat tanya dirasa perlu dilakukan. Aplikasi dibangun dengan menggunakan *Semantic Web* sebagai proses inti penggalian jawaban, Algoritma Nazief & Adriani sebagai metode untuk *stemming* kalimat tanya dan DBpedia Indonesia sebagai sumber pengetahuan. Dari sumber pengetahuan tersebut, akan diperoleh *output* jawaban yang sesuai dengan artikel yang ada pada Wikipedia Indonesia. Dengan dibangun aplikasi tanya jawab berbahasa Indonesia ini diharapkan dapat membantu pengguna internet dari Indonesia untuk mendapat informasi sesuai dengan yang dibutuhkan.

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka rumusan masalah pada penelitian ini adalah :

- a. Bagaimana membangun aplikasi tanya jawab menggunakan *semantic web*?
- b. Seberapa akurat jawaban yang dihasilkan oleh aplikasi tanya jawab yang dibangun dengan adanya Algoritma Nazief & Adriani sebagai metode *stemmer* kalimat tanya pada sistem?

1.3. Batasan Masalah

Agar penelitian ini tidak menyimpang dari permasalahan, maka beberapa batasan masalah pada penelitian ini yaitu :

- a. Basis pengetahuan yang digunakan adalah DBpedia Indonesia
- b. Pertanyaan yang diijinkan menggunakan bahasa Indonesia baku
- c. Algoritma Nazief & Adriani digunakan sebagai metode untuk *stemming* kalimat tanya
- d. Sistem ini berfokus pada penerapan *semantic web*

1.4. Tujuan Penelitian

Adapun tujuan penelitian ini antara lain :

- a. Membangun aplikasi tanya jawab menggunakan *semantic web*
- b. Mengukur keakurasian jawaban yang dihasilkan oleh aplikasi tanya jawab yang dibangun

1.5. Manfaat Penelitian

Dengan adanya penelitian ini, diharapkan dapat memberikan manfaat diantaranya :

- a. Menawarkan kemudahan pada pengguna web untuk mendapatkan informasi yang sesuai dengan kebutuhan
- b. Pencarian informasi menjadi lebih efisien karena dilakukan dalam satu sistem

1.6. Sistematika Penulisan

Penulisan skripsi ini disusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian serta sistematika penyusunan tugas akhir.

BAB II Tinjauan Pustaka

Tinjauan Pustaka berisi beberapa teori yang digunakan pada penyusunan tugas akhir. Pada bab ini, teori yang dibahas berkaitan tentang Sistem Tanya Jawab (*Question Answering System*), *Semantic Web*, Algoritma Nazief & Adriani dan Basis Pengetahuan DBpedia.

BAB III Desain dan Perancangan Sistem

Bab ini berisikan mengenai analisa kebutuhan sistem untuk membangun aplikasi meliputi desain dan langkah – langkah pembuatan Aplikasi Tanya Jawab dengan penerapan *Semantic Web* serta Algoritma Nazief & Adriani pada sistem.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang uji coba Aplikasi Tanya Jawab menggunakan *Semantic Web* dan Algoritma Nazief & Adriani yang telah diterapkan.

BAB V Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran.

BAB II

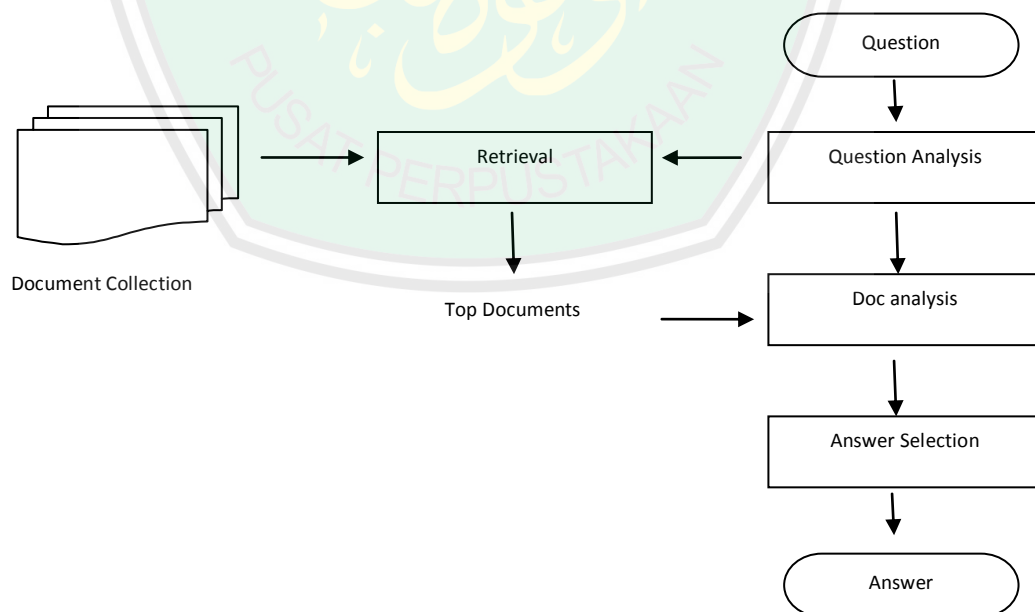
TINJAUAN PUSTAKA

2.1. Sistem Tanya Jawab

Sistem Tanya Jawab atau biasa dikenal dengan *Question Answering System* (QAS) merupakan sebuah sistem komputer yang memiliki kemampuan untuk menjawab pertanyaan yang diajukan dengan bahasa alami (*natural language*) dengan memanfaatkan berbagai teknik pemrosesan bahasa alami, seperti *Information Retrieval* dan *Information Extraction* (Irwan, 2012). Sistem Tanya Jawab (*Question Answering System*) merupakan suatu bagian dari *Information Retrieval* yang lebih menantang dari mesin pencarian (*search engine*) karena bertujuan untuk menemukan keakuratan dan konsistensi jawaban dari sebuah pertanyaan yang sesuai dengan dokumen (Li, 2002).

Question Answering System diklasifikasikan menjadi 2 lingkup kelas, yaitu *open-domain* QAS dan *closed-domain* QAS. *Open-domain* QAS berhadapan dengan pertanyaan mengenai apapun dan mempunyai lebih banyak data untuk mengekstrak jawaban. Sedangkan *closed-domain* QAS berhadapan dengan pertanyaan dalam domain yang spesifik dan dapat dilihat sebagai domain yang lebih mudah karena *closed-domain* QAS dapat memanfaatkan pengetahuan yang spesifik pada domain tertentu. Pada *closed-domain* QAS, ekstraksi jawaban diperoleh dari koleksi dokumen yang terbatas (Irwan, 2012).

Keseluruhan proses pada Sistem Tanya Jawab dijelaskan pada Gambar 2.1 Pengguna meng-*input* pertanyaan yang berupa bahasa alami yang kemudian dianalisa oleh modul *question analysis*. Selanjutnya, modul ini mengeluarkan kumpulan *keyword* yang berhasil didapat. *Keyword* tersebut kemudian dikirim ke modul *retrieval*. Modul *retrieval* merupakan suatu bagian yang berisi sistem temu balik dokumen standar yang mengidentifikasi dokumen yang mengandung *query* yang diberikan. Hasil identifikasi berupa *list* dokumen yang telah diurutkan berdasarkan kedekatan dengan *query*. Selanjutnya modul *analysis document* melakukan ekstraksi kandidat jawaban yang kemudian dikirim ke modul *answer selection*. Modul *answer selection* kemudian memilih frase yang memiliki kemungkinan besar sesuai dengan spesifikasi dari modul *question analysis*. Kemudian sistem mengembalikan jawaban yang telah diurutkan berdasarkan nilai kedekatannya kepada pengguna.



Gambar 2.1. Arsitektur umum *question answering system*
(Sumber : Irwan, 2012)

Sebelumnya telah dilakukan penelitian mengenai Sistem Tanya Jawab (*Question Answering System*) dengan menggunakan metode rule-based. Aplikasi yang dibangun berbasis desktop dengan menggunakan dokumen Terjemahan Al Qur'an Surah Al Baqarah sebagai databasenya. Proses penemukembalian jawaban dimulai dengan memecah (*parsing*) suatu dokumen menjadi kalimat-kalimat. Kalimat-kalimat tersebut dipecah dan di-*stem* menjadi *token-token*. Begitu pula dengan kalimat pertanyaan pada *query* dipecah dan di-*stem* menjadi *token-token*. *Token-token* dari setiap kalimat dokumen maupun kalimat *query* diproses dalam *rule* sesuai dengan tipe pertanyaannya. Proses di dalam *rule* memberikan nilai untuk masing-masing kalimat dokumen. Kalimat yang memiliki nilai tertinggi akan dikembalikan sebagai jawaban. Akurasi rata-rata *rule* terhadap *query* dalam penelitian adalah 85.69%, sedangkan akurasi rata-rata *rule* terhadap *query* pengguna umum adalah 53.14% (Anggraeny, 2008).

Question Answering System juga pernah diimplementasikan pada perangkat bergerak domain NBA (*National Basketball Association*). Aplikasi NBA QA ini merupakan aplikasi pada perangkat bergerak ber-*platform* android yang dibangun untuk menjawab pertanyaan seputar NBA. NBA QA dibuat dalam dua bagian terpisah yaitu *Database Generator* dan *Question Answering*. Hasil pengujian menunjukkan bahwa akurasi kebenaran aplikasi berdasarkan skenario uji mencapai 93,18%, sementara hasil pengujian responden menunjukkan akurasi sebesar 91,67% (Irwan, 2012).

Penelitian mengenai *Question Answering System* lainnya pernah dilakukan dengan menggunakan dokumen terjemah juz 'amma sebagai databasenya. Penelitian ini menggunakan *wordmatch scoring* dan *rule based scoring*. *Wordmatch scoring* adalah pemberian skor berdasarkan kesesuaian kata pada pertanyaan dan tipe jawaban. Skor digunakan untuk menentukan kandidat jawaban berdasarkan pertanyaan yang diajukan oleh user. Skor juga digunakan untuk perangkian jawaban. Berdasarkan hasil pengukuran relevansi pada sistem, *precision* tertinggi adalah jawaban dari kata tanya siapa dan mengapa. *Recall* tertinggi adalah jawaban dari kata tanya mengapa. *Accuracy* tertinggi adalah jawaban dari kata tanya mengapa (Lutfi, 2012).

2.2. Semantic Web

Kata semantik berasal dari bahasa Inggris *semantics*, dari bahasa Yunani yaitu *sema* (nomina) yang artinya tanda, atau dari *samaino* (verba) yang berarti menandai, berarti. Istilah tersebut digunakan pakar bahasa untuk menyebutkan bagian ilmu bahasa yang mempelajari tentang makna (Djajasudarman, 1993).

Semantic web adalah perkembangan generasi web yang disebut sebagai evolusi dari *World Wide Web* (WWW) yang dicetuskan pada tahun 2002. *Semantic web* pertama kali diperkenalkan oleh Tim Berners-Lee, penemu WWW pada tahun 1999. *Semantic web* merupakan perluasan dari *World Wide Web* dimana *content* web yang ditampilkan tidak hanya dalam bahasa format manusia (*natural language*) namun juga dalam format yang dapat dibaca dan digunakan oleh mesin.

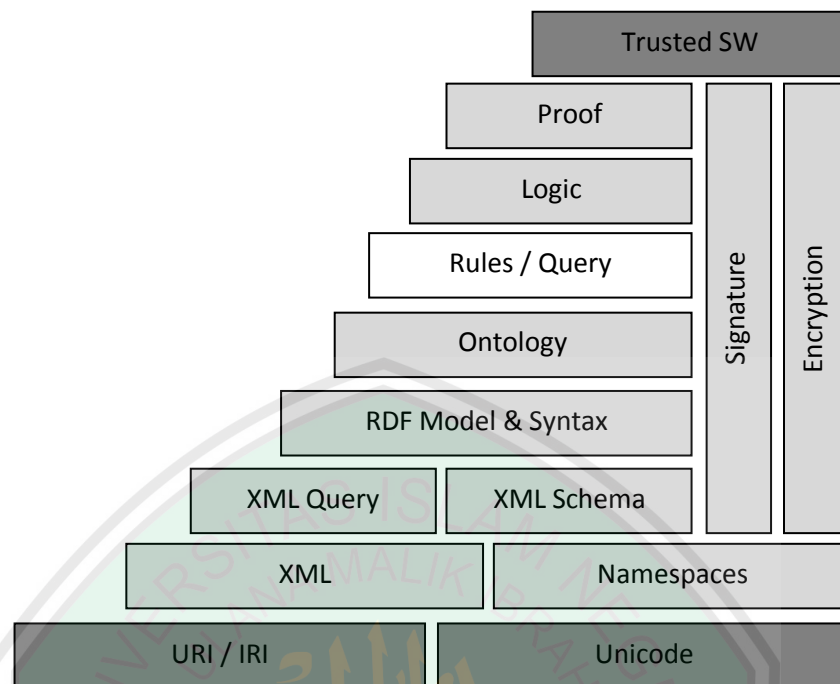
Semantic web didefinisikan sebagai sekumpulan teknologi, dimana memungkinkan komputer memahami arti sebuah informasi berdasarkan metadata, yaitu informasi mengenai informasi. Dengan adanya metadata, komputer diharapkan mampu mengartikan suatu informasi sehingga hasil pencarian menjadi lebih detail dan lebih tepat (Gunawan, 2014). *Semantic Web* mempunyai informasi yang dimiliki oleh mesin yang memiliki kecerdasan buatan sehingga mampu menemukan dan mengintegrasikan informasi dengan mudah. Tujuan dari *Semantic Web* adalah mengatur informasi dan prosedur. Fundamental dalam pembangunan *Semantic Web* adalah kreasi dan *semantic* metadata. Metadata terdiri dari 2 bagian, yaitu penggambaran sebuah dokumen dan penggambaran entitas dalam suatu dokumen (Ayuningtyas, 2009).

Menurut Gunawan (2014), kelebihan *semantic web* adalah untuk menggali sumber pengetahuan dilakukan secara *online*, tidak perlu lagi membuka halaman web satu per satu, namun dapat memanfaatkan agen untuk menjelajahi ribuan situs-situs web. Agen – agen cerdas ini dengan logis akan memilih hasil – hasil yang relevan sesuai dengan kebutuhan pengguna, kemudian menyajikannya dalam format yang diinginkan. Saat ini dokumen mulai disajikan dalam format *eXtensible Markup Language* (XML). Format lain yang mendukung selain XML adalah *Resource Description Framework* (RDF) / *Ontology Web Language* (OWL). RDF/OWL memiliki kemampuan untuk merepresentasikan makna istilah dalam web sehingga dapat di proses dalam sistem.

Selain kelebihan yang dapat menggali sumber pengetahuan secara *online*, terdapat keuntungan lain yang diperoleh dengan penggunaan *semantic web*. Menurut Ayuningtyas (2009), keuntungan yang dimiliki oleh *semantic web* adalah waktu yang diperlukan untuk mendapatkan informasi yang dicari lebih singkat dan pekerjaan pencarian yang dilakukan manusia dapat digantikan oleh mesin.

Semantic web merupakan web data yang memungkinkan mesin untuk memahami suatu informasi pada *World Wide Web* sesuai dengan maknanya. Saat pembuatan aplikasi semantik, terdapat 2 variabel yang dibangun sekaligus. Variabel pertama adalah web yang terdiri dari protocol komunikasi dan format web. Sedangkan variabel lainnya adalah semantik merepresentasikan makna dari web data (Ayuningtyas, 2009).

Semantic web adalah teknologi yang dikembangkan berbasis RDF format. Basis data yang digunakan menjadi acuan untuk pembuatan metadata. Pembentukan metadata (RDF) menggunakan format XML. Selanjutnya metadata tersebut dikonversi ke format RDF. Format RDF ini kemudian dieksekusi menggunakan *query* SPARQL. Gambar 2.2 merupakan *layer* dari *semantic web* yang direkomendasikan oleh W3C.



Gambar 2.2. *Layer pada Semantic Web*
(Sumber : <http://www.w3c.org>)

Menurut Zebua (2012), penjelasan *layer semantic web* pada Gambar 2.2 adalah sebagai berikut :

- Lapisan URI dan unicode memastikan digunakannya set karakter internasional dan menyediakan sarana untuk mengidentifikasi objek dalam web semantik.
- Lapisan XML dengan definisi skema *namespace* memastikan untuk mengintegrasikan definisi web semantik dengan standar berbasis XML lainnya.
- Dengan RDF dan RDFSchema akan memungkinkan untuk membuat pernyataan tentang objek dengan URI dan menentukan apakah kosakata

tersebut dapat disebut sebagai URI. Lapisan ini merupakan lapisan diberikannya jenis sumber daya dan link.

- Lapisan ontologi mendukung evolusi kosakata karena dapat menentukan hubungan antara konsep-konsep yang berbeda.
- Lapisan *Digital Signature* digunakan untuk mendeteksi perubahan pada dokumen.

Semantic web sedikit berbeda dengan web pada umumnya. Ada beberapa standar yang digunakan dalam pembuatan *semantic web* yang dikoordinasi oleh *World Wide Web Consortium* (W3C). Standar pokok tersebut adalah *Extensible Markup Language* (XML), *XML Schema*, *Resource Description Framework* (RDF), *RDF Schema*, *Web Ontology Language* (OWL) dan SPARQL.

a. XML dan XML Schema

Extensible Markup Language (XML) adalah bahasa *markup* yang didesain sebagai sarana untuk mengirimkan dokumen melalui web. XML berbeda dengan *Hypertext Markup Language* (HTML). XML memungkinkan pengguna untuk mendefinisikan *custom tag*. *XML Schema* merupakan bahasa yang digunakan untuk mendefinisikan sekumpulan aturan (*schema*) pada dokumen XML. Struktur dari dokumen XML yang dibuat harus sesuai dengan *schema* yang telah didefinisikan.

b. RDF dan RDF Schema

Resource Description Framework (RDF) adalah metode umum untuk memodelkan informasi dengan menggunakan sekumpulan format sintaks. Tujuan dari RDF adalah untuk membuat pernyataan mengenai *resource* web ke dalam

bentuk Subjek-Predikat-Objek. Subjek mengacu pada *resource* yang akan dideskripsikan. Predikat mengekspresikan karakteristik dari *resource* dan hubungan antara subjek dan objek. Dalam terminology RDF, Subjek-Predikat-Objek ini disebut dengan istilah *N-triple*. RDF memiliki bagian – bagian standar yang digunakan untuk membentuknya antara lain *resource* yang digunakan untuk menggambarkan apa saja yang dapat dimiliki oleh sebuah URI, *property* berisikan nama *property* – *property* yang ada dalam *resource*, *property value* yang berisikan nilai dari sebuah *property*. RDF telah digunakan pada aplikasi-aplikasi berikut :

- *RDF Site Summary (RSS)*

RSS memberikan informasi update sebuah *website* tanpa pengunjung perlu mengunjungi *website* tersebut.

- *Friend of a Friend (FOAF)*

Friend of a Friend didesain untuk mendeskripsikan orang-orang, ketertarikan dan hubungan mereka.

- *Semantically-Interlinked Online Communities (SIOC)*

SIOC menerangkan komunitas *online* dan menciptakan koneksi antara diskusi berbasis internet seperti *message board*, *blog* maupun *mailing list*. RDF *schema* dapat dipandang sebagai kamus data atau *vocabulary* untuk mendeskripsikan *properties* dan *classes* dari *resources* RDF.

c. **OWL**

Web Ontology Language (OWL) adalah bahasa yang digunakan oleh aplikasi-aplikasi yang tidak sekedar menampilkan informasi pada manusia,

melainkan juga yang perlu memproses informasi ini. *Ontology* banyak digunakan terutama untuk mendukung *semantic web* yang diarahkan untuk dapat memahami makna suatu kata atau kalimat yang diajukan oleh *user*. *Ontology* dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berisi *classes*, *properties* dan *instances*. Ketika informasi yang ada dalam dokumen diproses oleh mesin, OWL dapat digunakan untuk merepresentasikan makna suatu istilah secara eksplisit sekaligus hubungan antara istilah-istilah tersebut. OWL memiliki tiga *sub-language* yaitu :

- *OWL Lite*
OWL Lite mendukung yang memerlukan klasifikasi hirarki dan dalam batasan yang sederhana.
- *OWL DL*
OWL DL mendukung konstruksi seluruh OWL, tetapi hanya dapat digunakan pada batasan tertentu.
- *OWL Full*
OWL Full diperuntukkan bagi pengguna yang menginginkan maksimum penggunaan dan kebebasan sintaksis.

d. SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) adalah sebuah *protocol* dan bahasa *query* untuk *semantic web's resources*. SPARQL merupakan salah satu bahasa *query* yang digunakan untuk melakukan *query* pada RDF (*Resource Description Framework*). Sama halnya dengan *query SQL*, SPARQL pun digunakan untuk *me-retrieve* data-data yang diinginkan. Namun, yang

membedakan adalah SQL melakukan *query* pada data berupa database yang terdiri dari satu atau beberapa tabel sedangkan SPARQL melakukan *query* terhadap data pada RDF yang berupa *triple*. Untuk melakukan *query* SPARQL pada sebuah RDF tidaklah semudah melakukan *query* SQL.

Pada *query* SQL, hanya dengan mengetahui nama tabel dan atribut atau kolom tabel saja maka sudah bisa melakukan *query*. Sedangkan untuk melakukan *query* SPARQL pada suatu file RDF setidaknya harus mengetahui *resource*, *property* dan *value* atau *triple* pada RDF tersebut. Untuk mengetahui *triple* pada RDF bisa dengan cara melakukan bentuk *query* DESCRIBE. Sebuah *query* yang menggunakan SPARQL dapat terdiri atas *triple patterns*, konjungsi (*or*) dan disjungsi (*and*). Beberapa sintaks yang digunakan pada *query* SPARQL, yaitu :

- PREFIX

Statement PREFIX adalah sebuah metode yang digunakan sebagai penunjuk yang membawa informasi dalam suatu web. PREFIX digunakan untuk menyingkat sebuah *resource*, dalam hal ini diwakili oleh URI (*Uniform Resource Identifier*).

- SELECT

Statement SELECT adalah sebuah daftar variabel-variabel yang akan dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diwakili dengan notasi (?).

- WHERE

Statement WHERE adalah sederetan *triple pattern* yang harus dimiliki oleh setiap hasil *query* yang valid. Suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu terdiri dari Subjek Predikat dan Objek.

Gambar 2.3 merupakan contoh *query* yang digunakan untuk mengekstrak dokumen mengenai nama gubernur dan wakil gubernur DKI Jakarta. Hasil eksekusi *query* ditunjukkan pada Gambar 2.4.

```
Select ?gubernur ?wakilGubernur
Where{
  Dbpedia-id:Daerah_Khusus_Ibukota_Jakarta dbpedia-
owl:leaderName ?gubernur
  Dbpedia-id:Daerah_Khusus_Ibukota_Jakarta dbpedia-
owl:viceLeader ?wakilGubernur
}
```

Gambar 2.3. Contoh *query* SPARQL

gubernur	wakilGubernur
http://id.dbpedia.org/resource/Joko_Widodo	http://id.dbpedia.org/resource/Basuki_Tjahaja_Purnama

Gambar 2.4. Hasil eksekusi *query* SPARQL
(Sumber : Virtuoso SPARQL Query Editor)

Sebelumnya telah dilakukan penelitian mengenai *semantic web* pada aplikasi pencarian buku. Dalam penelitian ini dibuat sistem pencarian buku perpustakaan menggunakan teknologi *semantic search* berbasis web. Pembuatan aplikasi menggunakan *ontology* untuk merepresentasikan pengetahuan data terkait dalam membangun mesin pencarian. Struktur *ontology* yang digunakan adalah berupa bentuk RDF dan *query* SPARQL. Dari hasil penelitian yang dilakukan,

peneliti berhasil mencapai target yang diharapkan sehingga aplikasi dapat dikatakan berjalan dengan baik (Zebua, 2012).

Pengembangan *ontology* dan aplikasi semantik pernah dilakukan dengan membangun *ontology* web. Sistem tersebut dibangun berbasis web pada aplikasi sitasi jurnal elektronik Indonesia. Tujuannya adalah untuk mengklasifikasikan informasi pada jurnal tersebut dan menghubungkan dengan atikel yang diacu. Dari hasil pengujian, sistem yang dibangun telah melakukan fungsinya sebagai media pencarian, informasi yang ditampilkan bermanfaat dan sistem terbukti berjalan dengan baik (Ayuningtyas, 2009).

Penelitian mengenai penggunaan *semantic web* juga pernah diimplementasikan pada sistem *e-Learning*. *Semantic web* digunakan sebagai teknik untuk memenuhi kebutuhan personal masing-masing pengguna *e-Learning*. Salah satu bentuk personalisasi dalam sistem *e-Learning* cerdas adalah kemampuan untuk merekomendasikan artikel yang sesuai dengan kebutuhan tiap pengguna (*intelligent recommender*). *Semantic web* terdiri dari tiga lapis. XML (*eXtensible Markup Language*) berada pada lapisan paling bawah, kemudian RDF (*Resource Description Framework*) yang menyediakan kerangka penyajian informasi dan lapisan paling atas adalah *Schema & Ontology*. XML memungkinkan pembuatan tag untuk memberi keterangan pada web. RDF mengekspresikannya dengan menggunakan subjek, predikat dan objek dari suatu kalimat. Kemudian lapisan ketiga, merupakan kumpulan informasi yang disebut *ontology*. *Ontology* merupakan sebuah dokumen yang mendefinisikan arti dan hubungan antar *terms* (bisa berupa URI atau teks). Untuk melakukan pencarian

atau *query* yang berbasis *semantic web* tersedia bahasa *query* SPARQL. SPARQL merupakan bahasa *query* untuk mengakses RDF yang dibuat oleh W3C RDF Data Access Working Group (Paryudi, 2010).

2.3. Algoritma Stemming Nazief & Adriani

Stemming merupakan proses pemotongan imbuhan dari suatu kata untuk mendapatkan kata dasarnya. *Stemming* merupakan bagian penting dalam proses normalisasi dalam membangun sistem temu kembali informasi. Menurut Agusta (2009), *stemming* merupakan salah satu cara untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasi kata – kata ke dalam sebuah dokumen teks ke dalam bentuk kata dasarnya dengan menggunakan aturan – aturan tertentu.

Algoritma *stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* bahasa lainnya. Hal ini dikarenakan bentuk tiap – tiap bahasa memiliki morfologi yang berbeda. Proses *stemming* pada bahasa Indonesia lebih kompleks karena terdapat variasi imbuhan yang harus dibuang dari sebuah kata. Untuk kata berBahasa Indonesia terdapat beberapa macam algoritma *stemming*. Algoritma tersebut diantaranya : Algoritma Nazief dan Adriani (1996), Algoritma Vega (2001) dan Algoritma Arifin Setiono (2002).

Sebelumnya telah dilakukan mengenai perbandingan 2 Algoritma *stemming* Bahasa Indonesia. Agusta (2009) telah melakukan penelitian mengenai perbandingan antara dua algoritma *stemming* yaitu Algoritma Nazief & Adriani dengan Algoritma Porter. Hasil dari penelitian tersebut diperoleh kesimpulan bahwa proses *stemming* dokumen berbahasa Indonesia menggunakan Algoritma

Nazief & Adriani memiliki prosentase keakuratan (*precision*) lebih besar dibandingkan dengan Algoritma Porter. Pada proses *stemming* menggunakan Algoritma Nazief & Adriani, kamus yang digunakan sangat mempengaruhi hasil *stemming*. Semakin lengkap kamus daftar kata dasar yang digunakan, semakin akurat pula hasil *stemming*.

Algoritma *stemming* Nazief & Adriani dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan dan mengenkapsulasi imbuhan-imbuhan, termasuk awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan-akhiran (*confixes*). Algoritma Nazief & Adriani ini memiliki tahap – tahap sebagai berikut :

1. Pertama cari kata yang akan di-*stem* dalam kamus kata dasar. Jika ditemukan dalam kamus maka diasumsikan kata adalah *root word*. Maka algoritma berhenti, jika tidak maka dilanjutkan langkah selanjutnya.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu” atau “-nya”) dibuang. Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu” atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
 - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

- b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hilangkan *Derivation Prefixes* (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te-”)
 - a. Langkah 4 berhenti jika :
 - Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 2.1.
 - Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
 - Tiga awalan telah dihilangkan.
 - b. Identifikasikan tipe awalan dan hilangkan. Awalan ada tipe :
 - Standar : “di-”, “ke-”, “se-” yang dapat langsung dihilangkan.
 - Kompleks : “me-”, “be-”, “pe-”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Untuk itu, digunakan aturan pada Tabel 2.2 untuk mendapatkan pemenggalan yang tepat.
 - c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.
5. Apabila setelah langkah 4 kata dasar masih belum ditemukan maka proses *recoding* dilakukan dengan mengacu pada aturan pada Tabel 2.2. *Recoding* dilakukan dengan menambahkan karakter *recoding* di awal kata yang dipenggal. Pada Tabel 2.2, karakter *recoding* adalah huruf kecil setelah tanda hubung (‘-’) dan terkadang berada sebelum tanda kurung.

6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

Tabel 2.1. Kombinasi awalan & akhiran yang tidak diizinkan

Awalan	Akhiran
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

Tabel 2.2. Aturan pemenggalan kata Algoritma Nazief & Adriani

Aturan	Format Kata	Pemenggalan
1	berV ...	ber-V... be-rV...
2	berCAP...	ber-CAP... dimana C!= 'r' & P!= 'er'
3	berCAerV...	ber-CAerV... dimana C!= 'r'
4	Belajar	bel-ajar
5	beC ₁ erC ₂ ...	be-C ₁ erC ₂ ... dimana C ₁ != {'r', 'l'}
6	terV...	ter-V... te-rV...
7	terCerV...	ter-CerV... dimana C!= 'r'
8	terCP...	ter-CP... dimana C!= 'r' dan P!= 'er'
9	teC ₁ erC ₂ ...	te-C ₁ erC ₂ ... dimana C ₁ != 'r'
10	me{llr w y}V...	me-{llr w y}V...
11	mem{b f v}...	mem-{b f v}...
12	mempe{r l}...	mem-pe...
13	mem{rV V}...	me-m{rV V}... me-p{rV V}...
14	men{c j d z}...	men-{c j d z}...
15	menV...	me-nV... me-tV...
16	meng{g h q}...	meng-{g h q}...
17	mengV...	meng-V... meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... dimana V!= 'e'
20	pe{w y}V...	pe-{w y}V...
21	perV...	per-V... pe-rV...
23	perCAP...	per-CAP... dimana C!= 'r' dan P!= 'er'
24	perCAerV...	per-CAerV... dimana C!= 'r'
25	pem{b f V}...	pem-{b f V}...
26	pem{rV V}...	pe-m{rV V}... pe-p{rV V}...
27	pen{c d j z}...	pen-{c d j z}...
28	penV...	pe-nV... pe-tV...
29	peng{g h q}...	peng-{g h q}...
30	pengV...	peng-V... peng-kV...
31	penyV...	peny-sV...
32	pelV...	pe-lV... kecuali "pelajar" yang menghasilkan "ajar"
33	peCerV...	per-erV... dimana C!= {r w y l m n}
34	peCP...	pe-CP... dimana C!= {r w y l m n} dan P!= 'er'

(Sumber : Tahitoe, 2010)

Penelitian lain mengenai penggunaan Algoritma Nazief & Adriani adalah pada pencarian kosakata Bahasa Indonesia. Pada penelitian ini diterapkan proses *stemming* pada kamus. Algoritma Nazief & Adriani dipilih karena memiliki pola *stemming* yang bersifat iteratif. Algoritma Nazief & Adriani diterapkan pada kamus Bahasa Indonesia berbasis web. Hasil uji coba dan implementasi menunjukkan bahwa 98% masukkan menghasilkan keluaran yang berupa kata dasar sesuai dengan yang diharapkan (Rahman, 2013).

Penelitian mengenai penerapan Algoritma Nazief & Adriani lainnya pernah dilakukan pada pembuatan aplikasi sinonim untuk Bahasa Indonesia. Pada penelitian ini digunakan kumpulan kata kerja dan kata benda yang berhubungan dengan teknologi. Terdapat 100 kata dasar yang dipasangkan dengan jenis imbuhan. Jenis imbuhan terdiri dari awalan, sisipan, akhiran serta awalan-akhiran. Hasil uji coba dan implementasi menunjukkan bahwa tingkat keberhasilan aplikasi untuk awalan adalah 72%, untuk imbuhan yang berupa akhiran adalah 91% dan untuk imbuhan yang berupa awalan-akhiran adalah 79% (Mait, 2012).

2.4. Basis Pengetahuan DBpedia

DBpedia merupakan bentuk terstruktur dari Wikipedia. Maksud terstruktur adalah bentuk yang memiliki susunan baku yaitu *triple* (subjek, predikat dan objek) yang digunakan dalam RDF (*Resource Definition Format*) sebagai bagian dasar pada web semantik. Wikipedia merupakan salah satu situs pencarian yang cukup sering digunakan oleh masyarakat di dunia hingga saat ini.

Untuk Wikipedia dalam Bahasa Indonesia, tersedia sejak tahun 2003. Pada bulan Mei 2014, Wikipedia Bahasa Indonesia merupakan Wikipedia terbesar keempat di Asia Tenggara, ketujuh di seluruh Asia dan ke-21 dari seluruh dunia (Wikipedia Bahasa Indonesia, 2014). Tabel 2.3 menunjukkan statistik data Wikipedia tahun 2014.

Tabel 2.3. Statistik Data Wikipedia Bahasa Indonesia

Artikel	Pengguna	Halaman	Suntingan	Berkas	Pengguna aktif	Pengurus
349.059	614.191	1.447.319	9.223.237	50.148	1.873	24

(Sumber : Wikipedia Statistik, 2014)

Berdasarkan statistik data Wikipedia Bahasa Indonesia, diketahui bahwa jumlah artikel saat ini mencapai 349.059 artikel. Wikipedia pada umumnya hanya berupa teks biasa dan di dalamnya terkandung informasi yang sangat luas. Informasi tersebut bisa berasal dari *link website* yang berasal dari informasi-informasi yang terdapat dalam Wikipedia itu sendiri. DBpedia memiliki klasifikasi berdasarkan kelasnya masing-masing. Tabel 2.4 merupakan contoh tabel klasifikasi DBpedia.

Tabel 2.4. Tabel Contoh Klasifikasi DBpedia

Klasifikasi	Contoh
Kota	Surabaya, Jakarta, Bali
Country	Indonesia, Singapura, Inggris
Musisi	Diana Ross, Michael Jackson
Film	Harry Potter, Transformer 3
Permainan Komputer	Warcraft III, Fifa 2011, PES 2012

Masing-masing sumber data yang ada pada DBpedia diidentifikasi ke dalam URI dalam bentuk http://id.dbpedia.org/resource/nama_resource, nama_resource tersebut mewakili suatu artikel dari Wikipedia dalam bentuk http://id.wikipedia.org/wiki/nama_resource.

Content-Length: 115723
About: Internet
 An Entity of Type : [Thing](#), from Named Graph : <http://id.dbpedia.org>, within Data Space : [id.dbpedia.org](#)

Internet (kependekan dari interconnection-networking) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global Transmission Control Protocol/Internet Protocol Suite (TCP/IP) sebagai protokol pertukaran paket (packet switching communication protocol) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaidah ini dinamakan internetworking ("antarjaringan").

Property	Value
dbpedia-owl:abstract	Internet (kependekan dari interconnection-networking) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global Transmission Control Protocol/Internet Protocol Suite (TCP/IP) sebagai protokol pertukaran paket (packet switching communication protocol) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaidah ini dinamakan internetworking ("antarjaringan").
dbpedia-owl:thumbnail	http://upload.wikimedia.org/wikipedia/commons/thumb/d/d2/Internet_map_1024.jpg/200px-Internet_map_1024.jpg
dbpedia-owl:wikiPageExternalLink	http://en.wikipedia.org/wiki/Internet http://www.isoc.org/internet/history/brief.shtml
dbpedia-owl:wikiPageID	2828 (xsd:integer)
dbpedia-owl:wikiPageRevisionID	6783463 (xsd:integer)
dbpedia-owl:wikiPageWikiLink	- dbpedia-id:Tim_Berners-Lee - dbpedia-id:Armin_Meiwes - dbpedia-id:Camfrog - dbpedia-id:Komputer - dbpedia-id:Milis - dbpedia-id:Surat_elektronik - dbpedia-id:Facebook - dbpedia-id:Google - dbpedia-id:ICQ - dbpedia-id:Pengirim_pesanan_instan - dbpedia-id:Internet - dbpedia-id:Internet_Engineering_Task_Force - dbpedia-id:ISOC - dbpedia-id:Kopete - dbpedia-id:Laptop - dbpedia-id:Perpustakaan - dbpedia-id:Pomografi - dbpedia-id:SSH - dbpedia-id:Sistem

Gambar 2.5. Contoh tampilan data pada DBpedia

Buat akun baru Masuk log

Halaman Pembicaraan Baca Sunting Sunting sumber Versi terdahulu Cari

Mari bergabung dengan komunitas Wikipedia bahasa Indonesia! [tutup]

Internet

Dari Wikipedia bahasa Indonesia, ensiklopedia bebas

? Artikel ini sudah memiliki daftar referensi, bacaan terkait atau pranala luar, tapi **sumbernya masih belum jelas karena tak memiliki kutipan pada kalimat (catatan kaki)**. Mohon tingkatkan kualitas artikel ini dengan memasukkan rujukan yang lebih mendetail bila perlu.

Internet (kependekan dari *interconnection-networking*) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protokol pertukaran paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan **Internet**. Cara menghubungkan rangkaian dengan kaidah ini dinamakan *internetworking* ("antarjaringan").

Daftar isi

- Sejarah Internet
- Internet pada saat ini
- Budaya Internet
- Tata tertib Internet
- Isu moral dan undang-undang
- Akses Internet
- Penggunaan Internet di tempat umum
- Tokoh-tokoh Internet

Visualisasi dari beberapa route pada jaringan Internet.

Gambar 2.6. Contoh tampilan data pada Wikipedia

Gambar 2.5 merupakan contoh tampilan data pada DBpedia dengan menggunakan nama *resource* “Internet”. Sedangkan dengan menggunakan nama *resource* yang sama, jika dilakukan penelusuran pada website pada Wikipedia, akan diperoleh tampilan seperti Gambar 2.6.

Data yang terdapat di Wikipedia tersimpan dalam bentuk pengetahuan terstruktur berbentuk RDF (DBpedia). Data tersebut tersimpan dalam model *semantic web*. Dengan model *semantic web* tersebut DBpedia dapat memproses *query* untuk me-*retrieve* data – data yang saling terkait. Untuk memproses *query* pada data berbentuk RDF tersebut, dibutuhkan SPARQL yang merupakan bahasa *query* untuk *semantic web*.

Menurut Zebua (2012), dengan menggunakan SPARQL memungkinkan suatu sistem untuk dapat melakukan proses berikut :

- Mengambil nilai dari data yang terstruktur maupun data yang semi terstruktur
- Mengembangkan data dengan melakukan *query* terhadap suatu relasi yang tidak diketahui
- Dapat melakukan *query* operasi *join* yang kompleks pada database yang berlainan dengan lebih sederhana
- Dapat mengubah suatu data RDF menjadi *vocabulary* yang lain. Hasil dari *query* SPARQL dapat mengembalikan nilai dalam beberapa format data yaitu XML, RDF dan HTML.

DBpedia memiliki cakupan pengetahuan yang luas. Dengan keunggulan ini, DBpedia dapat menjadi penghubung *dataset*, tempat *dataset* luar dapat bertaut

dengan konsep-konsepnya. *Dataset* DBpedia saling bertaut pada tingkat RDF dengan sejumlah *dataset* Open Data lainnya pada web. Pada Januari 2011, ada lebih dari 6,5 juta antartautan (*interlink*) antara DBpedia dengan dataset luar, seperti Freebase, OpenCyc, UMBEL, GeoNames, Musicbrainz, CIA World Fact Book, DBLP, Project Gutenberg, Dbtune Jamendo, Eurostat, Uniprot, Bio2RDF dan US Cencus (DBpedia, 2014).



BAB III

DESAIN DAN PERANCANGAN SISTEM

3.1. Deskripsi Sistem

Subbab ini membahas mengenai deskripsi sistem yang dikerjakan pada skripsi. Tujuan pembangunan sistem ini adalah membuat suatu media yang dapat membantu *user* dalam mendapatkan informasi pada internet. Pada penelitian ini, sistem dibangun dengan konsep Aplikasi Tanya Jawab (*Question Answering System*) dengan menggunakan teknologi *semantic web* sebagai metode penggalian jawaban dan Algoritma Nazief & Adriani sebagai metode pemenggalan kata (*stemming*) guna mendapatkan potongan kata yang kemudian menjadi *keywords* pada proses penggalian jawaban. Sistem dibangun berbasis web bertujuan agar ketika telah di-online-kan, aplikasi dapat dijalankan tanpa batas ruang dan waktu.

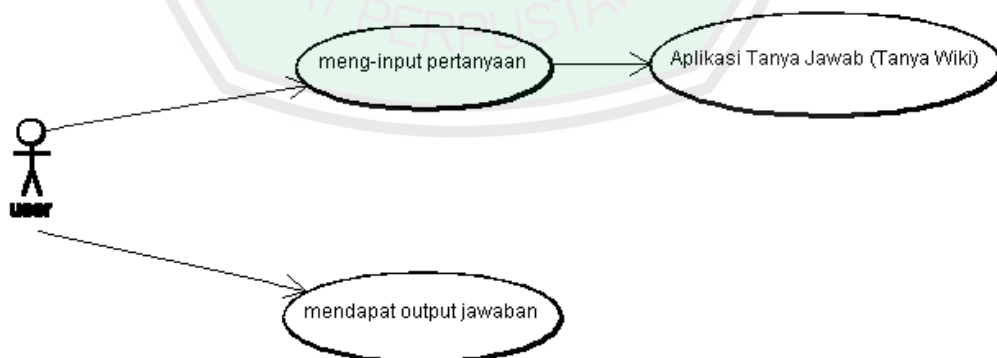
Sistem yang dibangun mengizinkan *user* untuk meng-input-kan pertanyaan dengan menggunakan bahasa baku. Pertanyaan tersebut kemudian diolah pada tahap *pre-processing* untuk mendapatkan potongan kata sebagai *keywords*. *Pre-processing* terdiri dari beberapa tahap, yaitu *tokenizing*, *filtering* dan *stemming*. Potongan kata yang diperoleh dari tahapan sebelumnya kemudian menjadi *keywords* yang diperlukan pada proses penelusuran sehingga diperoleh *resource* DBpedia yang dibutuhkan. *Resource* merupakan bagian penting pada *semantic web*. *Resource* digunakan sebagai penunjuk yang membawa informasi dalam suatu web. *Resource* diproses pada tahap penggalian jawaban menggunakan

teknologi *semantic web*. Penggalian jawaban menggunakan DBpedia sebagai sumber data. DBpedia merupakan suatu sumber data yang didapatkan dengan cara mengekstrak informasi terstruktur yang terdapat pada Wikipedia. Pada penelitian ini DBpedia Indonesia dipilih menjadi sumber pengetahuan sehingga data ekstrak yang didapat akan sesuai dengan data yang ada pada Wikipedia Indonesia. Maka ketika *user* menginputkan pertanyaan, *output* yang diterima merupakan kutipan jawaban berbahasa Indonesia sesuai dengan informasi yang telah diekstrak dari basis pengetahuan DBpedia Indonesia.

3.2. Perancangan Sistem

Berikut ini adalah penjelasan proses pada masing - masing tahap perancangan sistem :

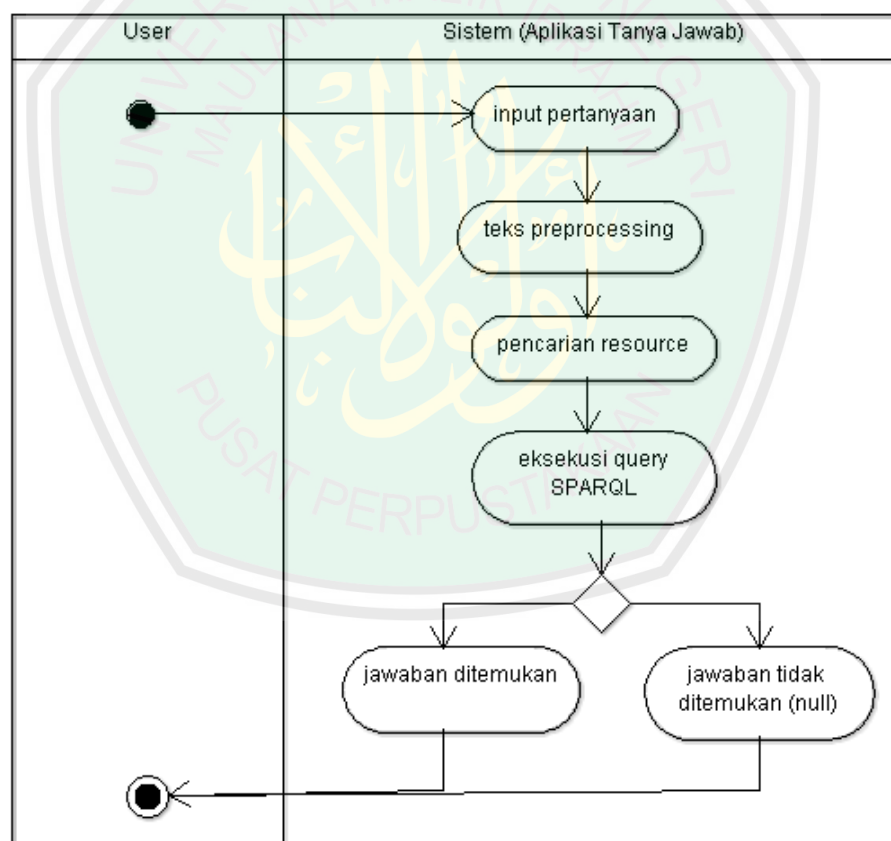
3.2.1. Use Case Diagram Aplikasi Tanya Jawab



Gambar 3.1. Use Case Diagram Aplikasi Tanya Jawab

Gambaran umum penggunaan aplikasi tanya jawab dijelaskan pada Gambar 3.1. Diawali dengan *user* menginputkan pertanyaan kepada sistem. Kalimat pertanyaan berbahasa Indonesia yang diinput harus menggunakan susunan bahasa baku. Kemudian sistem memproses pertanyaan tersebut dan mengembalikan *output* berupa kutipan jawaban sesuai dengan pertanyaan yang telah diinputkan oleh *user*.

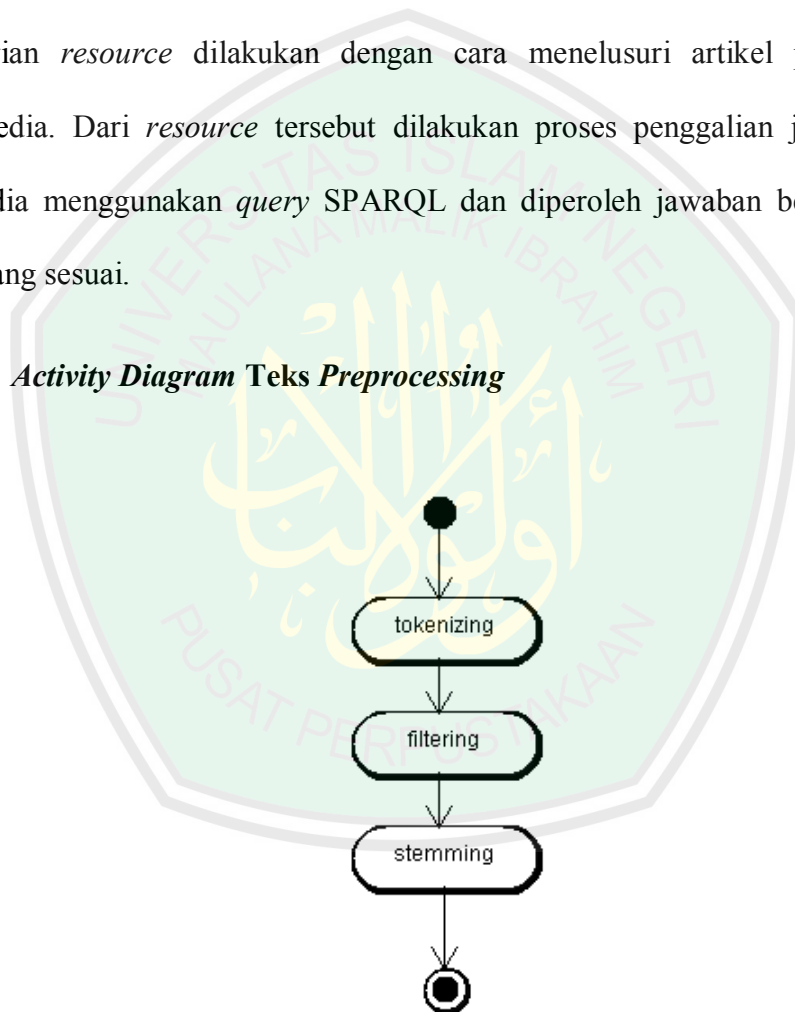
3.2.2. Activity Diagram Aplikasi Tanya Jawab



Gambar 3.2. Activity Diagram Aplikasi Tanya Jawab

Gambaran umum proses sistem diperlihatkan pada Gambar 3.2. Proses yang terjadi dalam sistem diawali dengan tahapan teks *pre-processing*. Kemudian pencarian *resource*, *resource* dibutuhkan pada proses selanjutnya yaitu pada tahap eksekusi *query* SPARQL. Potongan kata hasil teks *pre-processing* disimpan sebagai *keywords* yang dibutuhkan pada proses pencarian *resource*. Proses pencarian *resource* dilakukan dengan cara menelusuri artikel pada website Wikipedia. Dari *resource* tersebut dilakukan proses penggalian jawaban pada DBpedia menggunakan *query* SPARQL dan diperoleh jawaban berupa kutipan teks yang sesuai.

3.2.3. Activity Diagram Teks Preprocessing

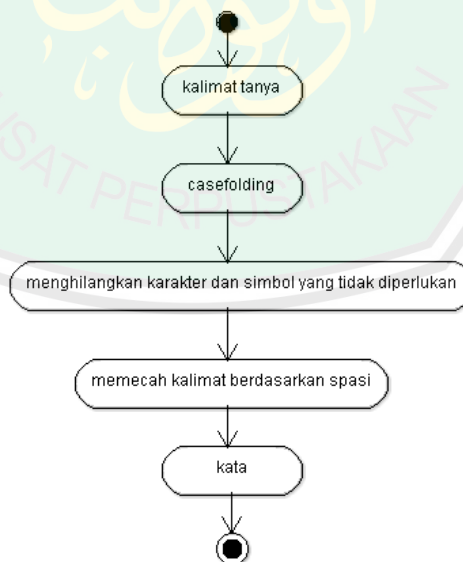


Gambar 3.3 Activity Diagram Teks Preprocessing

Teks *pre-processing* merupakan tahap pengolahan kalimat tanya yang diinputkan oleh *user*. Tahap ini terdiri dari *tokenizing*, *filtering* dan *stemming*. Tahapan teks *preprocessing* dijelaskan pada Gambar 3.3.

3.2.4. Activity Diagram Tokenizing

Tokenizing merupakan tahap teks *pre-processing* yang bertujuan untuk memecah kalimat menjadi satuan kata atau *token*. Proses ini diawali dengan menghilangkan karakter dan simbol yang tidak diperlukan. Kemudian dilakukan pemecahan kalimat berdasarkan spasi sehingga diperoleh potongan-potongan kata. *Tokenizing* pada sistem ini dilakukan untuk memecah kalimat tanya yang telah diinput oleh *user*. Proses *tokenizing* dapat dilihat pada Gambar 3.4 dan *source code* untuk proses *tokenizing* ditunjukkan pada Gambar 3.5.



Gambar 3.4. Activity Diagram Tokenizing

```

if (!empty($_POST['soal'])) {
    $lower = strtolower($_POST['soal']);
    $tanda = array(" ", "/", "?", "-", "!", ".", "*");
    $hapus = str_replace($tanda, '', $lower);
    $potongansoal = explode(" ", $hapus);
}

```

Gambar 3.5. Source code proses *Tokenizing*

Untuk memudahkan sistem dalam melakukan memecah kata, proses diawali dengan melakukan *case folding*. *Case folding* merupakan proses merubah semua huruf pada teks menjadi huruf kecil. Kemudian menghapus karakter dan simbol yang tidak diperlukan. Selanjutnya, kalimat tanya tersebut dipecah berdasarkan spasi. Sehingga diperoleh satuan kata (*token*). Contoh hasil *Tokenizing* dapat dilihat pada Tabel 3.1.

Tabel 3.1. Contoh hasil *tokenizing*

Kalimat Tanya	Hasil <i>Tokenizing</i>
Apa yang dimaksud dengan internet?	Apa yang dimaksud dengan internet

3.2.5. Activity Diagram *Filtering*

Filtering merupakan salah satu tahap pada teks *preprocessing* yaitu menghilangkan kata – kata yang dianggap kurang penting. Kata – kata hasil *tokenizing* dicocokkan dengan database daftar stopwords. Struktur tabel database

stopword dapat dilihat pada Tabel 3.2. Sedangkan contoh isi dari tabel database stopwords dapat dilihat pada Tabel 3.3. Setiap kata hasil *tokenizing* diproses pada tahap *filtering*. Jika terdapat kata yang sama dengan daftar stopwords, maka kata tersebut dihapus. Jika tidak ditemukan kata yang sama dengan daftar stopwords, maka kata tersebut ditampilkan kembali.

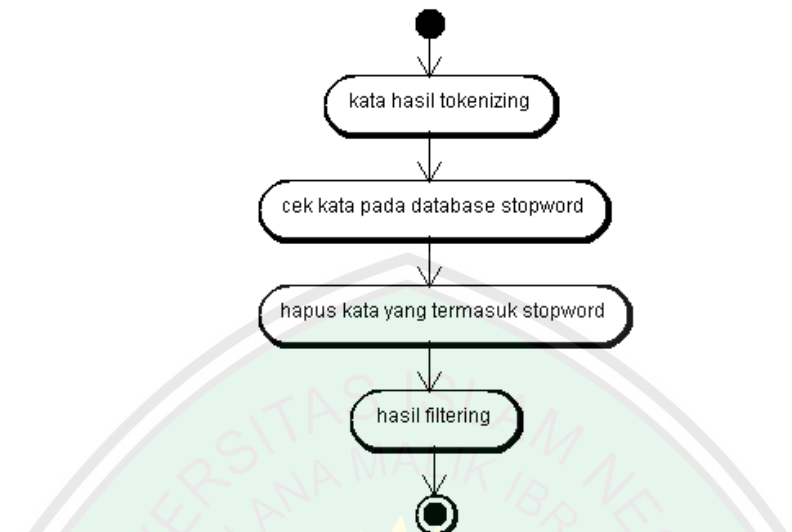
Tabel 3.2. Struktur Tabel Stopword

Atribut	Tipe	Panjang
id_stoplist	Integer	10
stoplist	Varchar	50

Tabel 3.3. Contoh Isi Tabel Stopword

id_stoplist	Stoplist
1	Yang
2	Mampu
3	Tentang
4	Juga
5	Untuk
6	Memang
7	Dari
8	Setiap
9	Seperti
10	Antara
11	Punya
12	Telah
13	Ke
14	Itu
15	Agak

Sedangkan *Activity diagram* proses *filtering* dijelaskan pada Gambar 3.6.



Gambar 3.6. *Activity Diagram Filtering*

Gambar 3.7 menunjukkan *source code* untuk proses *filtering*.

```

function cekStopword($kata){
    $koneksi3 = mysql_connect("localhost", "root", "");
    mysql_select_db("dbbaru", $koneksi3);
    $sql = "SELECT * from tb_stoplist where stoplist
    ='$kata' LIMIT 1";
    $result = mysql_query($sql) or die(mysql_error());
    if (mysql_num_rows($result) == 1) {
        $hstop = str_replace($result,"",$result);
        return $hstop;
    } else {
        return $kata; // tidakada
    }
}

```

Gambar 3.7. *Source code proses filtering*

Contoh hasil *Filtering* dapat dilihat pada Tabel 3.4.

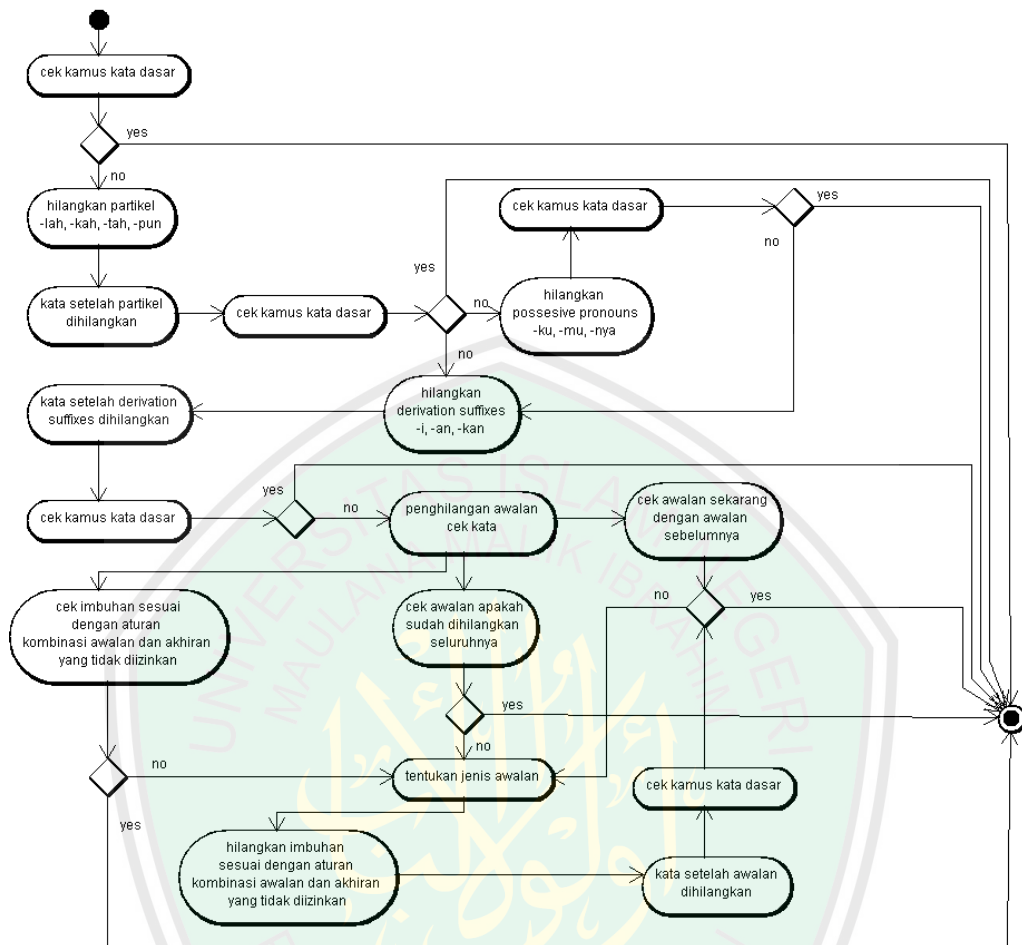
Tabel 3.4. Contoh hasil *filtering*

Kalimat Tanya	Hasil <i>Filtering</i>
Apa yang dimaksud dengan internet?	dimaksud internet

3.2.6. Activity Diagram *Stemming* Nazief & Adriani

Stemming merupakan proses pemotongan imbuhan dari suatu kata untuk mendapatkan kata dasarnya. Untuk kata berbahasa Indonesia terdapat beberapa macam algoritma *stemming*. Pada sistem ini digunakan algoritma *stemming* Nazief & Adriani sebagai aturan pemotongan imbuhan. Algoritma ini dipilih karena pada penelitian sebelumnya (Agusta, 2009) diperoleh kesimpulan bahwa proses *stemming* dokumen berbahasa Indonesia menggunakan Algoritma Nazief & Adriani memiliki prosentase keakuratan (*precision*) lebih besar. Activity Diagram proses *stemming* dapat dilihat pada Gambar 3.8.

Proses *stemming* Algoritma Nazief & Adriani diawali dengan proses cek tiap potongan kata pada kamus kata dasar. Jika potongan kata tersebut ditemukan pada database kamus kata dasar, maka kata tersebut dianggap kata dasar dan algoritma berhenti. Namun jika tidak ditemukan, maka potongan kata tersebut di proses pada tahap selanjutnya. Struktur tabel database kata dasar dapat dilihat pada Tabel 3.5. sedangkan contoh isi dari tabel database kata dasar dapat dilihat pada Tabel 3.6.



Gambar 3.8. Activity Diagram Stemming Nazief & Adriani

Tabel 3.5. Struktur Tabel Kata Dasar

Atribut	Tipe	Panjang
id_katadasar	Integer	10
Katadasar	Varchar	70
tipe_katadasar	Varchar	25

Tabel 3.6. Contoh Isi Tabel Kata Dasar

id_katadasar	Katadasar	tipe_katadasar
1	Abad	Nomina
2	Abadi	Adjektiva
3	Abang	Nomina
4	Abdi	Nomina
5	Abdomen	Nomina

Gambar 3.9 merupakan *source code* untuk proses cek kata pada database kata dasar.

```
function cekKamus($kata) {
    $koneksi = mysql_connect("localhost", "root", "");
    mysql_select_db("dbbaru", $koneksi);
    $sql = "SELECT * from tb_katadasar where katadasar
    ='$kata' LIMIT 1";
    $result = mysql_query($sql) or die(mysql_error());
    if (mysql_num_rows($result) == 1) {
        return true; // ada
    } else {
        return false; // tidakada
    }
}
```

Gambar 3.9. *Source code* proses cek kata dasar

Apabila kata dasar belum ditemukan, maka dilanjutkan proses selanjutnya yaitu menghapus *Inflection Suffixes*. Imbuhan yang merupakan *Inflection Suffixes* adalah (“-lah”, “-kah”, “-ku”, “-mu”, “-nya”).

```
function del_inf_suffixes($kata) {
    $kataasal = $kata;
    if (eregi('([km]u|nya|[kl]ah|pun)$', $kata)) {
        $kata1 = eregi_replace('([km]u|nya|[kl]ah|pun)$', '',
        $kata);
        if (eregi('([klt]ah|pun)$', $kata)) {
            if (eregi('([km]u|nya)$', $kata1)) {
                $kata2 = eregi_replace('([km]u|nya)$', '', $kata1);
                return $kata2;
            }
        }
        return $kata1;
    }
    return $kataasal;
}
```

Gambar 3.10. *Source code* proses menghapus *Inflection Suffixes*

Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu” atau “-nya”), jika ada. Gambar 3.10 menunjukkan kutipan *source code* tahap penghapusan *Inflection Suffixes*.

Setelah proses penghapusan *Inflection Suffixes*, potongan kata tersebut di cek kembali pada database kamus kata dasar sebagaimana proses sebelumnya. Jika ditemukan kata yang sesuai, maka kata dasar ditemukan dan proses algoritma berhenti. Namun jika kata dasar masih belum ditemukan, maka kata tersebut diproses pada tahap selanjutnya yaitu menghapus *Derivation Suffixes*. *Derivation Suffixes* merupakan imbuhan kata yang berupa (“-i”, “-an” atau “-kan”). Sama halnya dengan proses sebelumnya, setelah dilakukan penghapusan imbuhan, potongan kata di cek kembali pada database kamus kata dasar. Jika ditemukan pada kata dasar, maka proses algoritma berhenti. Namun jika belum ditemukan, maka dilanjut pada penerapan aturan berikut :

1. Jika imbuhan “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti.
2. Jika tidak ditemukan maka akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan.

Gambar 3.11 menunjukkan *source code* proses penghapusan *Derivation Suffixes*.

```
function del_der_suffixes($kata) {
    $kataasal = $kata;
    if (preg_match('/(kan)$/', $kata)) { // Cek Suffixes
        $kata1 = preg_replace('/(kan)$/', '', $kata);
        if (cekKamus($kata1)) { // Cek Kamus
            return $kata1;
        }
    }
    if (preg_match('/(an|i)$/', $kata)) { // cek -kan
        $kata2 = preg_replace('/(an|i)$/', '', $kata);
        if (cekKamus($kata2)) { // Cek Kamus
            return $kata2;
        }
    }
    if (cek_prefix_disallowed_sufixes($kata)) {
        return $kataasal;
    }
    return $kataasal;
}
```

Gambar 3.11. *Source code* proses menghapus *Derivation Suffixes*

Proses yang dilakukan apabila setelah dilakukan proses penghapusan *Derivation Suffixes* kata dasar belum juga ditemukan, maka dilanjutkan dengan proses menghapus *Derivation Prefixes*. *Derivation Prefix* merupakan awalan yang berupa imbuhan (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te-”).

Proses penghapusan *Derivation Prefix* ini memiliki beberapa aturan khusus. Adapun aturan pada proses penghapusan *Derivation Prefix* ini yaitu proses berhenti jika:

1. Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 2.1.

2. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
3. Tiga awalan telah dihilangkan.

Kemudian dilanjutkan dengan proses identifikasi tipe awalan dan hilangkan. Awalan terdiri dari beberapa tipe, yaitu :

1. Standar : “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata.
2. Kompleks : “me-”, “be-”, “pe-”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, digunakan aturan pada Tabel 2.2. untuk mendapatkan pemenggalan yang tepat.

Selanjutnya cek kata yang telah dihilangkan awalan ini di dalam kamus database kata dasar. Apabila tidak ditemukan, maka langkah penghapusan *Derivation Prefix* ini diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan. Namun jika masih saja belum ditemukan maka kata awal diasumsikan sebagai kata dasar dan proses selesai. Kutipan Source code *Derivation Prefix* ditunjukkan pada gambar 3.12 dan contoh hasil *Stemming* Nazief & Adriani dapat dilihat pada Tabel 3.7.

```

function del_der_prefixes($kata) {
    $kataasal = $kata;

    /* ----- Menentukan tipe ----- */

    if (preg_match('/^(di|[ks]e)\S{1,}/', $kata)) { //Jika di-
, ke-, se-
        $katal = preg_replace('/^(di|[ks]e)/', '', $kata);
        if (cekKamus($katal)) {
            return $katal; // Jika ada balik
        }
        $kata2 = del_der_suffixes($katal);
        if (cekKamus($kata2)) {
            return $kata2;
        }
    }

    if (preg_match('/^([tmbp]e)\S{1,}/', $kata)) {
        /* Awalan be- */
        if (preg_match('/^(be)\S{1,}/', $kata)) { //Jika awalan be-
        if (preg_match('/^(ber)[aueo]\S{1,}/', $kata)) { //aturan
1
            $katal = preg_replace('/^(ber)/', '', $kata);
            if (cekKamus($katal)) {
                return $katal;
            }
            $kata2 = del_der_suffixes($katal);
            if (cekKamus($kata2)) {
                return $kata2;
            }
            $katal = preg_replace('/^(ber)/', 'r', $kata);
            if (cekKamus($katal)) {
                return $katal;
            }
            $kata2 = del_der_suffixes($katal);
            if (cekKamus($kata2)) {
                return $kata2;
            }
        }
        if (preg_match('/^(ber)[^aueor][[:alpha:]](?!er)\S{1,}/',
$kata)) { //aturan 2
            $katal = preg_replace('/^(ber)/', '', $kata);
            if (cekKamus($katal)) {
                return $katal;
            }
            $kata2 = del_der_suffixes($katal);
            if (cekKamus($kata2)) {
                return $kata2;
            }
        }
    }
}

```

Gambar 3.12. Source code proses menghapus Derivation Prefix

Tabel 3.7. Contoh hasil *Stemming* Nazief & Adriani

Kalimat Tanya	Hasil <i>Stemming</i>
Apa yang dimaksud dengan internet?	Maksud internet

3.2.7. Activity Diagram Pencarian Resource

Pencarian *resource* dibutuhkan sebagai URI *resource* pada proses eksekusi *query* SPARQL. Untuk mendapatkan *resource* yang sesuai, digunakan Google Custom Search Engine. Google Custom Search Engine akan melakukan penelusuran pada artikel yang terdapat pada website Wikipedia. Sehingga akan diperoleh indeks *link* yang dirasa sesuai dengan *keyword* yang diproses. Gambar 3.13 menunjukkan contoh *link* url hasil penelusuran menggunakan *keyword* “Internet”.

<http://www.wikipedia.org/Internet>

Gambar 3.13. Contoh *link* url hasil penelusuran

Gambar 3.13 merupakan contoh suatu url dari salah satu dokumen pada Wikipedia. Secara umum link url terdiri dari 3 bagian, yaitu :

1. Protokol

Protokol merupakan persetujuan bersama yang digunakan untuk berkomunikasi dengan *Hypertext Transfer Protocol*. Dalam hal ini, protokol ditunjukkan dalam bentuk `http://`.

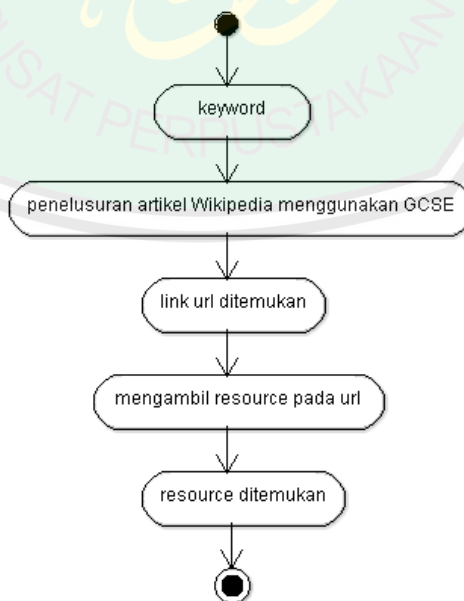
2. Alamat Server

Alamat server merupakan dimana sumber data terletak. Karena sumber data yang dicari berasal dari Wikipedia, maka alamat server yang digunakan adalah website Wikipedia yaitu www.wikipedia.org.

3. Path File

Path File menunjukkan lokasi dan nama *document* pada server tersebut. *Path File* ini yang kemudian digunakan sebagai *resource*. Karena *Path File* pada Wikipedia sama dengan nama *resource* suatu *document* pada sumber data DBpedia.

Untuk lebih jelasnya, gambaran alur proses pencarian *resource* dijelaskan pada Gambar 3.14.



Gambar 3.14. Activity Diagram Pencarian Resource

Untuk mendapatkan *resource* dilakukan beberapa tahap yaitu *input keyword* pada sistem, *keyword* tersebut kemudian diproses menggunakan Google Custom Search Engine. Dari proses tersebut diperoleh indeks *link* yang selanjutnya diekstrak menggunakan format JSON. Dari hasil data berformat JSON tersebut kemudian dilakukan proses pemotongan url dengan mengambil bagian *path file* atau *resource*-nya. Sehingga didapatkan suatu *resource* yang dibutuhkan untuk proses selanjutnya. Gambar 3.15 merupakan potongan *source code* dari proses pencarian *resource*.

```
$ganti = preg_replace('/( )/', '+', $gabungansoal);
$alamat =
file_get_contents("https://www.googleapis.com/customsearch/v1?q=".$ganti."&cx=00000000000000000000:xxxxxxxxxxx&
key=yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy");
$jsonoon = json_decode($alamat,true);
$linknya = $jsonoon['items'][0]['link'];
$arResource = explode("/", $linknya);
$resourcee = $arResource[4];
```

Gambar 3.15. *Source code* proses pencarian *resource*

Cara kerja dari Google Custom Search Engine sendiri diawali dengan Google mengirimkan *web crawler* yang bernama Googlebot, kemudian Googlebot akan menelusuri semua *website* yang ada pada World Wide Web. Googlebot bekerja secara terus menerus dalam melakukan *crawling website*. Googlebot juga mendeteksi semua *links* yang ada pada *website* dan menyimpannya karena links ini penting untuk menentukan ranking pencarian dari *website*. Selanjutnya semua halaman di *copy* dan di simpan pada Google Index's Database. Ketika mesin mendapat inputan sebuah *keyword*, maka indeks tersebut akan mencocokkan pada

data yang ada pada database dan mengambil hasil jawaban. Hasil yang ditampilkan berupa indeks *links* yang tersusun berdasarkan data yang dianggap paling relevan dengan *keyword* yang telah diinputkan.

3.2.8. Eksekusi *Query* SPARQL

Resource yang telah diperoleh pada proses sebelumnya, kemudian di proses sebagai variabel URI pada *query* SPARQL. Dalam istilah SPARQL, *resource* merupakan bagian dari URI *resource* yang digunakan sebagai penunjuk yang membawa informasi dalam suatu web. Contoh query SPARQL pada sistem ditunjukkan pada Gambar 3.16.

```
$query =
"select ?Document_Extraction
where {
dbpedia-id:". $resourceee." dbpedia-owl:abstract
?Document_Extraction
}";
```

Gambar 3.16. Contoh *query* SPARQL

Jika pada proses sebelumnya, *resource* yang ditemukan adalah kata “Internet”, maka ketika diimplementasikan pada *source code* sebagaimana Gambar 3.16, implementasi *query* SPARQL ditunjukkan pada Gambar 3.17.

```
$query =
"select ?Document_Extraction
where {
dbpedia-id:Internet dbpedia-owl:abstract
?Document_Extraction
}";
```

Gambar 3.17. Implementasi *resource* pada *query* SPARQL

Hasil eksekusi *query* SPARQL tersebut, ditampilkan pada Gambar 3.18.

Document Extraction
"Internet (kependekan dari interconnection-networking) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global Transmission Control Protocol/Internet Protocol Suite (TCP/IP) sebagai protokol pertukaran paket (packet switching communication protocol) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaidah ini dinamakan internetworking ("antarjaringan")."@id

Gambar 3.18. Hasil eksekusi *query* SPARQL menggunakan *resource* “Internet”

3.3. Sumber Data

Sistem tanya jawab membutuhkan basis pengetahuan sebagai media penggalian informasi untuk menemukembalikan jawaban sesuai dengan pertanyaan yang diajukan *user*. Basis pengetahuan yang digunakan pada pembangunan aplikasi tanya jawab ini adalah DBpedia. DBpedia merupakan sebuah komunitas yang bergerak untuk mengekstrak informasi terstruktur dari Wikipedia dan menyediakan informasi tersebut dalam sebuah web.

Untuk mengakses pengetahuan (*knowledge*) DBpedia, digunakan URI (*Uniform Resource Identifier*) *resource* dari entitas tersebut. Apabila URI *resource* diakses menggunakan *Semantic Web agents*, maka hasil yang ditampilkan adalah RDF *descriptions*. Sedangkan apabila diakses menggunakan *Web Browser* akan ditampilkan informasi dari entitas yang diakses dalam tampilan HTML sederhana (DBpedia Indonesia, 2014).

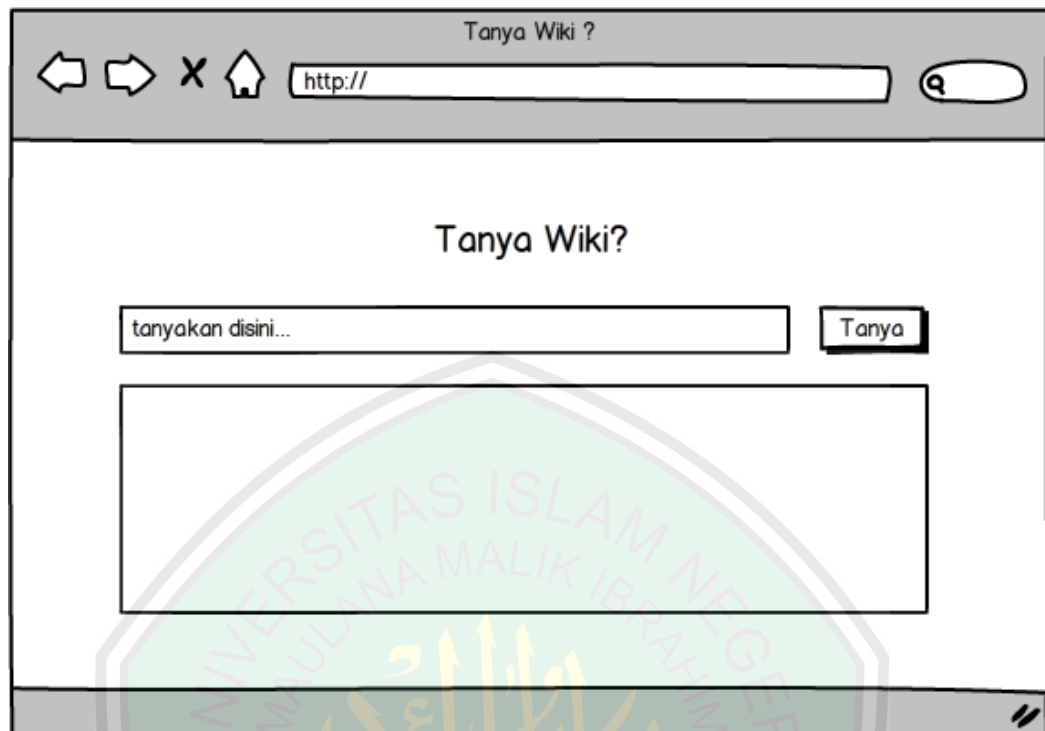
DBpedia menyimpan data dalam model *semantic web* yaitu berbentuk RDF. Dengan model *semantic web* tersebut DBpedia dapat memproses *query* pada data tersebut. Bahasa yang dibutuhkan untuk melakukan *query* tersebut adalah SPARQL.

3.4. Desain Tampilan

Setiap aplikasi yang berhubungan langsung dengan *user* pasti memiliki suatu *interface* atau tampilan antarmuka. *Interface* yang baik adalah tampilan yang sederhana, bagus dan mudah dipahami. Tampilan Aplikasi Tanya Jawab yang dibangun didesain sesederhana mungkin dengan tujuan memudahkan *user* dalam mengakses.

Aplikasi Tanya Jawab ini hanya dibangun dalam 1 halaman utama saja. Tampilan halaman utama ini memuat logo aplikasi sebagai identitas dari aplikasi, *textfield* untuk menginputkan pertanyaan, *button* ‘tanya’ untuk memulai pencarian dan *textarea* hasil untuk menampilkan *output* jawaban. Desain tampilan aplikasi digambarkan pada Gambar 3.19.

Aplikasi Tanya Jawab yang dibangun diberi nama Tanya Wiki?. Nama ini didasari oleh tujuan utama dari pembangunan aplikasi ini, yaitu menyediakan media untuk bertanya. Kemudian kata Wiki? digunakan karena sumber pengetahuan yang digunakan adalah DBpedia yang merupakan basis pengetahuan pada Wikipedia.



Gambar 3.19. Desain tampilan Aplikasi Tanya Jawab

Selain nama aplikasi, halaman utama memuat satu *textfield* sebagai tempat untuk *user* menginputkan kalimat tanya. Kemudian *button* “tanya” untuk memulai proses penggalian jawaban. Jawaban yang ditemukan selanjutnya ditampilkan pada suatu *textarea* yang terletak di bagian bawah *textfield* pertanyaan. Pada *textarea* ini akan dimuat beberapa bagian yaitu :

1. Input Interpretation

Input Interpretation merupakan bagian yang menampilkan interpretasi dari pertanyaan yang telah diajukan. Pada bagian ini akan ditampilkan :

- Pertanyaan yang telah diajukan
- *Keyword* yang ditemukan

- *Link* document pada Wikipedia
- *Resource* pada DBpedia

2. Document Extraction

Document Extraction memuat hasil eksekusi *query* SPARQL yang merupakan jawaban untuk pertanyaan yang telah diajukan.

3. Source

Source berisi *links* Wikipedia yang dirasa sesuai dengan jawaban yang dimunculkan.

3.5. Implementasi Tampilan

Halaman utama aplikasi memuat form untuk menginput pertanyaan sekaligus menampilkan jawaban beserta interpretasi dari inputan. Pada saat aplikasi dijalankan, sistem akan menampilkan suatu halaman seperti yang diperlihatkan pada Gambar 3.20.

Disini *user* dapat menginputkan pertanyaan dengan memasukkan kalimat tanya pada *textfield* yang disediakan kemudian meng-klik *button* tanya untuk memulai proses. Selanjutnya sistem akan memproses kalimat tanya tersebut dan menampilkan jawaban. Jawaban ditampilkan pada beberapa bagian, yaitu Input Interpretation, Document Extraction dan Source.



tanyakan disini... tanya!

Input Interpretation

Your Question	
Keyword Found	
Document on Wikipedia	
DBpedia Endpoint	

Source

© 10650073 - Teknik Informatika UIN Maulana Malik Ibrahim Malang - 2014

Gambar 3.20. Tampilan Aplikasi Tanya Jawab

Input Interpretation merupakan interpretasi dari inputan pertanyaan yang telah diajukan. Pada bagian ini ditampilkan kalimat tanya, *keyword* yang ditemukan, dokumen Wikipedia yang relevan dengan *keyword* dan DBpedia Endpoint (*resource* pada DBpedia). Pada bagian selanjutnya ditampilkan Document Extraction, bagian ini merupakan jawaban inti yang diperoleh dari mengekstrak data pada DBpedia.



tanya!

Input Interpretation	
Your Question	apa yang dimaksud dengan internet?
Keyword Found	maksud internet
Document on Wikipedia	http://id.wikipedia.org/wiki/Internet http://id.wikipedia.org/wiki/Internet_Explorer http://id.wikipedia.org/wiki/Internet_Download_Manager
DBpedia Endpoint	Internet Internet_Explorer Internet_Download_Manager

Document_Extraction

"Internet (kependekan dari interconnection-networking) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global Transmission Control Protocol/Internet Protocol Suite (TCP/IP) sebagai protokol pertukaran paket (packet switching communication protocol) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaidah ini dinamakan internetworking ("antarjaringan")."@id

Document_Extraction

Gambar 3.21. Contoh Penggunaan Aplikasi Tanya Jawab

Pada bagian akhir, ditampilkan Source, yakni link artikel Wikipedia yang dirasa sesuai dengan data DBpedia. Contoh penggunaan aplikasi ditunjukkan pada Gambar 3.21. Contoh inputan pertanyaan yang diajukan berupa kalimat tanya "Apa yang dimaksud dengan internet?". Aplikasi kemudian menampilkan hasil berupa jawaban dan interpretasi pertanyaan.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil uji coba sistem yang telah dirancang dan dibangun. Uji coba dilakukan untuk mengetahui apakah sistem telah berhasil dibangun dan sistem dapat berjalan sebagaimana mestinya dengan lingkungan uji coba yang sesuai dengan skenario pengujian. Uji coba pada aplikasi ini juga dilakukan untuk mengetahui akurasi aplikasi dalam menjawab pertanyaan.

4.1. Platform yang digunakan

Pada proses pembangunan dan uji coba aplikasi tanya jawab ini digunakan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut :

1. Perangkat Keras (*Hardware*)

- Processor Intel (R) Core (TM) Duo CPU T6500 @2.10 GHz
- RAM 2.00 GB
- VGA SiS Mirage 3 Graphics
- Hard disk drive 230 GB

2. Perangkat Lunak (*Software*)

- Sistem Operasi Windows 7 Ultimate 32-bit SP 1

- XAMPP 1.6.7, sebagai web server pada proses pembuatan aplikasi versi *offline*
- Notepad++, yaitu aplikasi yang digunakan untuk melakukan *coding* aplikasi tanya jawab
- Mozilla Firefox Browser 32.0.2, sebagai aplikasi *browser* untuk menjalankan aplikasi tanya jawab
- Adobe Photoshop CS3, yaitu aplikasi yang digunakan untuk mendesain logo nama aplikasi
- ArgoUML 0.34, sebagai aplikasi untuk mendesain diagram model UML

4.2. Uji Coba Aplikasi

Kriteria untuk pengujian performa dari aplikasi tanya jawab ini yaitu dengan mengukur keakurasian jawaban yang ditemukan dengan adanya algoritma *stemming* Nazief & Adriani atau tanpa menggunakan algoritma *stemming*. Uji coba dilakukan berdasarkan input pertanyaan dan perhitungan nilai akurasi hasil jawaban yang ditemukan.

Skenario pengujian aplikasi tanya jawab yang dibangun dilakukan terhadap hasil jawaban yang ditemukan. Objek pengujian adalah kalimat pertanyaan berbahasa Indonesia. Kalimat pertanyaan yang diproses harus

menggunakan bahasa Indonesia baku. Kata tanya yang dianjurkan adalah bentuk kata tanya 5W+1H, yaitu :

- Apa (*what*)

Kata tanya “Apa” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban bertipe benda.

- Dimana (*where*)

Kata tanya “Dimana” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa tempat atau lokasi.

- Siapa (*who*)

Kata tanya “Siapa” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa orang.

- Kapan (*when*)

Kata tanya “Kapan” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa waktu.

- Mengapa (*why*)

Kata tanya “Mengapa” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa sebab.

- Bagaimana (*how*)

Kata tanya “Bagaimana” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa deskripsi.

- Berapa

Kata tanya “Berapa” digunakan untuk mengajukan pertanyaan yang membutuhkan jawaban berupa integer ataupun kuantitas suatu objek.

Tabel 4.1. menunjukkan klasifikasi jenis pertanyaan yang diproses aplikasi tanya jawab. Aplikasi tanya jawab ini juga dapat memproses pertanyaan yang membutuhkan jawaban bertipe *integer*, seperti pertanyaan mengenai jarak, tinggi, usia dll. Akan tetapi, aplikasi tanya jawab ini tidak dapat memproses pertanyaan yang berbentuk perhitungan dan aritmatika.

Tabel 4.1. Klasifikasi Jenis Pertanyaan

Kata Tanya	Tipe Jawaban	Contoh Pertanyaan
Apa	Benda	Apa yang dimaksud dengan <i>smartphone</i> ?
Dimana	Lokasi	Dimana letak makam Bung Karno?
Siapa	Orang	Siapa nama walikota Surabaya?
Kapan	Waktu	Kapan Indonesia dinyatakan merdeka?
Mengapa	Alasan	Mengapa Jepang menyerah kepada Belanda tahun 1945?
Bagaimana	Deskripsi	Bagaimana proses metamorfosis berlangsung?
Berapa	Integer (jarak, tinggi, usia dll)	Berapa tinggi tugu Monas?

Berikut ini uji coba yang telah dilakukan terhadap sistem, tabel menunjukkan pertanyaan yang diinputkan beserta ketepatan jawaban yang dihasilkan.

Tabel 4.2. Tabel Uji Coba Pertanyaan Tentang Benda

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Apa yang dimaksud dengan <i>smartphone</i> ?	√	
2	Apa warna bendera negara Jerman?	√	
3	Apakah pengertian dari <i>hardware</i> ?		√
4	Apa nama maskot kota Surabaya?	√	
5	Apa arti semboyan Bhineka Tunggal Ika?	√	
6	Apa nama tarian khas Bali?		√
7	Apa makanan khas Yogyakarta?	√	
8	Apakah judul lagu kebangsaan Indonesia?		√
9	Apa teori yang ditemukan oleh Albert Einstein?		√
10	Apa nama planet yang paling dekat dengan matahari?	√	
11	Apa nama planet yang memiliki cincin?	√	
12	Apa nama kota yang disebut kota hujan?	√	
13	Apa makanan khas Lamongan?	√	
14	Apa nama gunung di kota Batu?	√	
15	Apa nama jembatan di kota Palembang?	√	

Tabel 4.3. Tabel Uji Coba Pertanyaan Tentang Lokasi/Tempat

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Dimana letak sungai Musi?	√	
2	Dimana letak tugu Monas?	√	
3	Dimana letak museum Radya Pustaka?	√	
4	Dimana letak makam Bung Karno?	√	
5	Dimanakah letak Jembatan Ampera?	√	
6	Dimanakah Pangeran Diponegoro dimakamkan?	√	
7	Dimana letak kerajaan Majapahit?	√	
8	Dimana letak Gunung Bromo?	√	
9	Dimana tugu pahlawan berada?	√	
10	Dimana tempat ibadah umat muslim?	√	
11	Dimana letak jembatan suramadu?	√	
12	Dimana letak pulau Raja Ampat?	√	
13	Dimana lokasi Institut Teknologi Telkom?	√	
14	Dimana lokasi Taman Safari Indonesia 2?	√	
15	Dimana kota asal bahasa Osing?	√	

Tabel 4.4. Tabel Uji Coba Pertanyaan Tentang Orang

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Siapa nama gubernur Jawa Timur?	√	
2	Siapa nama istri Bung Karno?	√	
3	Siapa pembuat sosial media Facebook?	√	
4	Siapakah nama walikota Surabaya?	√	
5	Siapakah CEO Apple Inc?	√	
6	Siapakah nama kiper Arema?	√	
7	Siapa pencipta lagu Indonesia Raya?	√	
8	Siapa pencetus teori gravitasi?		√
9	Siapa pencetus teori relativitas?	√	
10	Siapakah CEO Microsoft saat ini?	√	
11	Siapakah proklamator kemerdekaan Indonesia?	√	
12	Siapa vokalis Ungu band?	√	
13	Siapa penemu listrik?		√
14	Siapa nama istri SBY?	√	
15	Siapa nama gubernur Bank Indonesia?	√	

Tabel 4.5. Tabel Uji Coba Pertanyaan Tentang Waktu

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Kapan Indonesia dinyatakan merdeka?	√	
2	Kapan Gus Dur mulai menjabat sebagai presiden?	√	
3	Kapan Dude Harlino lahir?	√	
4	Kapan Nagita Slavina memulai karir?	√	
5	Kapan tanggal lahir Bunga Citra Lestari?	√	
6	Kapan sholat Idul Adha berlangsung?	√	
7	Kapan perayaan natal berlangsung?	√	
8	Kapan masa jabatan Jokowi sebagai Gubernur DKI Jakarta?	√	
9	Kapan Hari Kebangkitan Nasional diperingati?		√
10	Kapan Apollo 11 tiba di Bulan?	√	
11	Kapan Soeharto berhenti menjadi presiden?	√	
12	Kapan PKI dibubarkan?	√	
13	Kapan TNI dibentuk?	√	
14	Kapan terjadinya perjanjian hudaibiah?	√	
15	Kapan terjadinya perang badar?	√	

Tabel 4.6. Tabel Uji Coba Pertanyaan Tentang Sebab

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Mengapa Jepang menyerah terhadap Belanda tahun 1945?	√	
2	Mengapa terjadi perang sampit?	√	
3	Mengapa Idul Adha disebut juga hari raya haji?	√	
4	Mengapa Jogja dijuluki sebagai daerah istimewa?		√
5	Mengapa terjadi gerhana bulan?	√	
6	Mengapa terjadi gerakan Aceh merdeka?	√	
7	Mengapa Bogor disebut kota hujan?	√	
8	Mengapa Bali disebut pulau Dewata?		√
9	Mengapa Pontianak disebut kota Khatulistiwa?	√	
10	Mengapa terjadi krisis moneter tahun 1998?	√	
11	Mengapa Korea berpisah menjadi 2 wilayah?	√	
12	Mengapa Malin Kundang dikutuk menjadi batu?	√	
13	Mengapa Surabaya disebut kota Pahlawan?	√	
14	Mengapa Edelweiss disebut bunga abadi?		√
15	Mengapa Timor Leste berpisah dari Indonesia?	√	

Tabel 4.7. Tabel Uji Coba Pertanyaan Tentang Deskripsi

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Bagaimana proses metamorfosis berlangsung?	√	
2	Bagaimana proses membuat tempe?	√	
3	Bagaimanakah proses terbentuknya pelangi?		√
4	Bagaimana cara menanam jagung?		√
5	Bagaimana cara menginstal ulang Windows?		√
6	Bagaimanakah cara merawat Kucing Anggora?		√
7	Bagaimana cara mengukur presisi?	√	
8	Bagaimana aturan permainan gobak sodor?		√
9	Bagaimana cara mengukur akurasi?	√	
10	Bagaimana aturan permainan catur?	√	
11	Bagaimana proses terjadinya hujan?	√	
12	Bagaimana proses terjadinya fotosintesis?	√	
13	Bagaimana cara mencangkok tumbuhan?	√	
14	Bagaimana cara menanam padi?		√
15	Bagaimana proses terjadinya gerhana matahari?	√	

Tabel 4.8. Tabel Uji Coba Pertanyaan Tentang Integer (panjang, jarak, tinggi dll)

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Berapa panjang jalan tol Surabaya-Gempol?	√	
2	Berapa tinggi tugu Monas?	√	
3	Berapa banyak pulau di Indonesia?	√	
4	Berapa luas Kebun Raya Purwodadi?	√	
5	Berapa banyak suku yang ada di Indonesia?	√	
6	Berapa tinggi gunung Bromo?	√	
7	Berapa luas wilayah Taman Nasional Bromo Tengger Semeru?	√	
8	Berapa diameter bunga Raflesia?	√	
9	Berapa jarak Bumi dengan Matahari?	√	
10	Berapa lama proses rotasi Bumi?	√	
11	Berapa jarak dari surabaya ke jakarta?	√	
12	Berapa personel JKT48?	√	
13	Berapa tekanan darah normal manusia?	√	
14	Berapa warna yang ada pada pelangi?		√
15	Berapa jumlah episode Running Man?	√	

Setiap jawaban yang ditampilkan dari masing-masing pertanyaan dilakukan pengukuran performansi. Pengukuran yang digunakan adalah *recall*, *precision* dan *accuracy*. *Recall* merupakan kemampuan sistem untuk memanggil dokumen yang relevan, sedangkan *precision* berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan.

Dari hasil pengujian akan didapatkan beberapa kemungkinan, yaitu :

1. *True Positive* (TP) yaitu jawaban yang dihasilkan sistem benar
2. *False Positive* (FP) yaitu jawaban yang dihasilkan salah atau sistem tidak menghasilkan jawaban
3. *True Negative* (TN) yaitu pertanyaan yang diajukan tidak sesuai dengan ketentuan dan sistem tidak menghasilkan jawaban

4. *False Negative* (FN) yaitu pertanyaan yang diajukan tidak sesuai dengan ketentuan tetapi sistem menghasilkan jawaban

Berikut rumus yang digunakan untuk menghitung *precision* dan *recall* beserta *accuracy* pada sistem :

$$Precision = \frac{TP}{TP + FP} \times 100\%$$

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%$$

Tabel 4.9 menunjukkan hasil pengukuran *recall*, *precision* dan *accuracy* pada pengujian sistem.

Tabel 4.9. Tabel Perbandingan Hasil Uji Coba Pertanyaan Keseluruhan

<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>
83,81%	100%	83,81%

4.3. Analisa Hasil dan Pembahasan

Berdasarkan uji coba yang telah dilakukan, diketahui bahwa aplikasi tanya jawab yang telah dibangun mampu menemukembalikan jawaban dengan baik untuk setiap pertanyaan yang diajukan. Keakurasian aplikasi tanya jawab dengan adanya Algoritma *stemming* Nazief & Adriani pada pemrosesan pertanyaan mencapai 83,81%.

Hasil analisa terhadap data uji coba diketahui bahwa akurasi jawaban dipengaruhi oleh hasil *stemming* yang menjadi *keyword* dalam pencarian *resource*. Jenis kalimat tanya tidak mempengaruhi hasil jawaban, karena kata tanya yang digunakan tidak diproses pada sistem. Sistem memproses *keyword* hasil *text-preprocessing*. Apabila terdapat *overstem* pada hasil penemuan *keyword*, maka terjadi ketidaksesuaian *resource* yang ditemukan sehingga jawaban yang ditampilkan menjadi rancu.

Gambar 4.3 merupakan contoh hasil *overstem* yang menyebabkan *keyword* tidak sesuai dengan pertanyaan yang diajukan. Pada proses *stemming* tersebut, seharusnya *keyword* yang didapatkan adalah kata “arti hardware” atau “pengertian hardware”. Namun, karena terjadi *overstem* maka *keyword* beserta jawaban yang diperoleh tidak sesuai dengan yang diharapkan.

Selain terjadinya *overstem*, penyebab tidak ditemukannya jawaban adalah data tidak ditemukan pada dokumen yang ada pada DBpedia. Hal ini dikarenakan tidak semua artikel yang terdapat pada DBpedia mencakup segala pengetahuan yang dibutuhkan. Sebagai contohnya, ketika *user* bertanya mengenai tinggi Monas seperti Gambar 4.4 aplikasi memberikan jawaban yang sesuai. Namun ketika *user* bertanya mengenai tinggi menara Eiffel, aplikasi hanya memberikan jawaban berupa deskripsi singkat mengenai menara Eiffel dan tidak memberikan jawaban yang diharapkan seperti Gambar 4.5. Hal ini dikarenakan pada dokumen menara Eiffel yang terdapat pada DBpedia tidak terdapat data mengenai tinggi menara tersebut



tanyakan disini...

tanya!

Input Interpretation	
Your Question	Apakah pengertian dari hardware?
Keyword Found	erti hardware
Document on Wikipedia	http://id.wikipedia.org/wiki/BASIC http://id.wikipedia.org/wiki/Properti http://id.wikipedia.org/wiki/Perangkat_perusak
DBpedia Endpoint	BASIC Properti Perangkat_perusak

Document_ Extraction
"BASIC, adalah singkatan dari Beginners' All-purpose Symbolic Instruction Code adalah sebuah kelompok bahasa pemrograman tingkat tinggi. Secara harfiah, BASIC memiliki arti "kode instruksi simbolis semua tujuan yang dapat digunakan oleh para pemula". Memang, istilah "Bahasa BASIC" di sini juga bisa diartikan menjadi bahasa untuk pemula, atau dengan kata lain, disebut sebagai bahasa dasar, tapi hal tersebut dirasa kurang tepat, mengingat BASIC dapat juga digunakan oleh para pemrogram ahli. BASIC pertama kali dikembangkan pada tahun 1963 oleh John George Kemeny dan Thomas Eugene Kurtz yang berasal dari Dartmouth College, untuk mengizinkan akses terhadap komputer bagi para mahasiswa jurusan selain jurusan ilmu eksakta. Pada waktu itu, hampir semua komputer membutuhkan perangkat lunak, dan waktu itu belum ada perangkat lunak yang dijual secara bebas, sehingga hanya orang-orang tertentu yang dapat menggunakan komputer, yakni para matematikawan dan ilmuwan, karena mereka dapat membangun perangkat lunak sendiri. Bahasa

Gambar 4.1. Overstem yang menyebabkan kerancuan dalam menemukan *keyword*



tanyakan disini...

tanya!

Input Interpretation	
Your Question	Berapa tinggi tugu Monas?
Keyword Found	tinggi tugu monas
Document on Wikipedia	http://id.wikipedia.org/wiki/Monumen_Nasional http://id.wikipedia.org/wiki/Menara_Jakarta http://id.wikipedia.org/wiki/Jalur_kereta_api_Jakarta_Kota-Manggarai
DBpedia Endpoint	Monumen_Nasional Menara_Jakarta Jalur_kereta_api_Jakarta_Kota-Manggarai

Document_ Extraction
"Monumen Nasional atau yang populer disingkat dengan Monas atau Tugu Monas adalah monumen peringatan setinggi 132 meter (433 kaki) yang didirikan untuk mengenang perlawanan dan perjuangan rakyat Indonesia untuk merebut kemerdekaan dari pemerintahan kolonial Hindia Belanda. Pembangunan monumen ini dimulai pada tanggal 17 Agustus 1961 di bawah perintah presiden Sukarno, dan dibuka untuk umum pada tanggal 12 Juli 1975. Tugu ini dimahkotai lidah api yang dilapisi lembaran emas yang melambangkan semangat perjuangan yang menyala-nyala. Monumen Nasional terletak tepat di tengah

Gambar 4.2. Contoh pencarian tinggi Monas



tanya!

Input Interpretation

Your Question	Berapa tinggi menara Eiffel?
Keyword Found	tinggi menara eiffel
Document on Wikipedia	http://id.wikipedia.org/wiki/Menara_Eiffel http://id.wikipedia.org/wiki/Menara_Tokyo http://id.wikipedia.org/wiki/Daftar_bangunan_tertinggi_di_Las_Vegas
DBpedia Endpoint	Menara_Eiffel Menara_Tokyo Daftar_bangunan_tertinggi_di_Las_Vegas

Document Extraction

"Menara Eiffel merupakan sebuah menara besi yang dibangun di Champ de Mars di tepi Sungai Seine di Paris. Menara ini telah menjadi ikon global Perancis dan salah satu struktur terkenal di dunia."@id
--

Gambar 4.3. Contoh pencarian tinggi menara Eiffel

Aplikasi tanya jawab yang telah dibangun ini dapat diintegrasikan dengan islam. Berdasarkan tujuan utama pembangunan aplikasi yaitu membantu *user* dalam mencari informasi. Dengan informasi, manusia bisa menemukan peluang dan kesempatan. Selain itu, manusia jadi mengetahui apa yang sebelumnya tidak diketahui. Aplikasi tanya jawab ini melakukan pencarian informasi yang ada dalam DBpedia untuk dikembalikan sebagai jawaban berdasarkan pertanyaan yang diajukan oleh *user*. Jika ditinjau dari sisi Alqur'an, proses pencarian informasi untuk memperoleh keyakinan diceritakan dalam kitab suci Alqur'an. Sebagaimana pada QS. Al-An'am ayat 74 – 79, yaitu kutipan kisah Nabi Ibrahim ketika membuktikan pencarian Tuhan pada kaumnya.

وَإِذْ قَالَ إِبْرَاهِيمُ لِأَبِيهِ ۖ أَزَّرَ اتَّخِذْ أَصْنَامًا ءَالِهَةً ۖ إِنِّي أُرِيدُ أَنْ مَبْنِي ۖ ضَلَّلَ مُبِينٍ ﴿٧٤﴾
 وَكَذَلِكَ نُرِي إِبْرَاهِيمَ مَلَكُوتَ السَّمَوَاتِ وَالْأَرْضِ وَلِيَكُونَ مِنَ الْمُوقِنِينَ ﴿٧٥﴾ فَلَمَّا جَنَّ
 عَلَيْهِ اللَّيْلُ رَأَى كَوْكَبًا ۖ قَالَ هَذَا رَبِّي ۖ فَلَمَّا أَفَلَ قَالَ لَا أُحِبُّ الْآفِلِينَ ﴿٧٦﴾ فَلَمَّا رَأَى الْقَمَرَ
 بَازِغًا قَالَ هَذَا رَبِّي ۖ فَلَمَّا أَفَلَ قَالَ لَئِنْ لَمْ يَهْدِنِي رَبِّي لَأَكُونَنَّ مِنَ الْقَوْمِ الضَّالِّينَ ﴿٧٧﴾
 فَلَمَّا رَأَى الشَّمْسَ بَازِغَةً قَالَ هَذَا رَبِّي هَذَا أَكْبَرُ ۖ فَلَمَّا أَفَلَتْ قَالَ يُغْوِمُنِي إِلَىٰ بَرِيءٍ مِّمَّا
 تُشْرِكُونَ ﴿٧٨﴾ إِنِّي وَجَّهْتُ وَجْهِيَ لِلَّذِي فَطَرَ السَّمَوَاتِ وَالْأَرْضَ حَنِيفًا ۖ وَمَا أَنَا مِنَ
 الْمُشْرِكِينَ ﴿٧٩﴾

Artinya :

74. dan (ingatlah) di waktu Ibrahim berkata kepada bapaknya, Aazar, "Pantaskah kamu menjadikan berhala-berhala sebagai tuhan-tuhan? Sesungguhnya aku melihat kamu dan kaummu dalam kesesatan yang nyata."

75. dan Demikianlah Kami perlihatkan kepada Ibrahim tanda-tanda keagungan (kami yang terdapat) di langit dan bumi dan (kami memperlihatkannya) agar Dia Termasuk orang yang yakin.

76. ketika malam telah gelap, Dia melihat sebuah bintang (lalu) Dia berkata: "Inilah Tuhanku", tetapi tatkala bintang itu tenggelam Dia berkata: "Saya tidak suka kepada yang tenggelam."

77. kemudian tatkala Dia melihat bulan terbit Dia berkata: "Inilah Tuhanku". tetapi setelah bulan itu terbenam, Dia berkata: "Sesungguhnya jika Tuhanku tidak memberi petunjuk kepadaKu, pastilah aku Termasuk orang yang sesat."

78. kemudian tatkala ia melihat matahari terbit, Dia berkata: "Inilah Tuhanku, ini yang lebih besar". Maka tatkala matahari itu terbenam, Dia berkata: "Hai kaumku, Sesungguhnya aku berlepas diri dari apa yang kamu persekutukan.

79. Sesungguhnya aku menghadapkan diriku kepada Rabb yang menciptakan langit dan bumi, dengan cenderung kepada agama yang benar, dan aku bukanlah Termasuk orang-orang yang mempersekutukan tuhan.

Menurut Kitab Tafsir Jalalain, QS. Al-An'am ayat 74-79 tersebut menceritakan bahwa pertanyaan yang diajukan oleh Nabi Ibrahim kepada bapaknya yakni "Pantaskah kamu menjadikan berhala-berhala sebagai tuhan-tuhan?" merupakan kata tanya yang bermakna celaan karena telah menjadikan berhala sebagai tuhan. Nabi Ibrahim menganggap sesat ayah dan kaum ayahnya.

Kemudian Allah SWT memperlihatkan kuasa-Nya kepada Nabi Ibrahim agar Nabi Ibrahim dapat mengambil kesimpulan tentang keesaan-Nya. Ketika malam datang, Nabi Ibrahim melihat bintang *zahrah*/venus. Nabi Ibrahim mengatakan “*Inilah tuhanku*” kepada kaumnya yang pada waktu itu menjadi penyembah bintang-bintang. Dan ketika malam telah hilang dan bintang tersebut tenggelam, Nabi Ibrahim mengatakan “*Saya tidak suka kepada yang tenggelam*” maksudnya adalah Nabi Ibrahim tidak suka menjadikan bintang sebagai tuhan sebab tuhan tidak patut mempunyai sifat yang berubah-ubah dan berpindah-pindah tempat karena kedua sifat ini hanyalah pantas disandang oleh makhluk, namun cara yang disampaikan Nabi Ibrahim ini tidak mempan pada kaum tersebut. Selanjutnya ketika bulan terbit dan menampakkan sinarnya, Nabi Ibrahim berkata “*Inilah tuhanku*”. Tetapi setelah bulan itu terbenam, dia berkata “*Sesungguhnya jika Tuhanku tidak memberi petunjuk kepadaKu, pastilah aku Termasuk orang yang sesat*”. Maksud dari perkataan ini adalah sindiran Nabi Ibrahim terhadap kaumnya bahwa mereka itu berada dalam kesesatan, akan tetapi perkataan tersebut sedikitpun tidak bermanfaat bagi kaumnya.

Kemudian ketika melihat matahari, Nabi Ibrahim berkata “*Inilah Tuhanku, ini yang lebih besar*” dan ketika matahari telah tenggelam, Nabi Ibrahim menyampaikan hujah yang kuat dan tidak dapat dibantah lagi oleh kaumnya. Nabi Ibrahim berkata “*Hai kaumku, Sesungguhnya aku berlepas diri dari apa yang kamu persekutukan*”. Maksud dari perkataan ini adalah tidak mempersekutukan Allah SWT dengan berhala-berhala dan benda-benda yang membutuhkan kepada yang menciptakannya.

Akhirnya kaum itu bertanya kepada Nabi Ibrahim *“lalu apakah yang engkau sembah?”*. Nabi Ibrahim menjawab sebagaimana ayat ke 79 yaitu *“Sesungguhnya aku menghadapkan diriku kepada Rabb yang menciptakan langit dan bumi, dengan cenderung kepada agama yang benar, dan aku bukanlah termasuk orang-orang yang mempersekutukan tuhan”*.

Proses pencarian merupakan bagian dari skripsi ini. Pada skripsi ini, dibangun suatu aplikasi tanya jawab yang berguna dalam proses pencarian suatu informasi. Dalam aplikasi tanya jawab ini, untuk mendapatkan jawaban sesuai pertanyaan yang diajukan, dilakukan tahapan-tahapan diawali dengan menganalisa pertanyaan untuk mendapatkan *keyword*, mencari *resource* hingga memperoleh jawaban yang diperlukan.

Untuk memperoleh jawaban yang dirasa sesuai, dilakukan dengan memilih 3 *resource* teratas dari hasil pencarian yang kemudian digunakan sebagai acuan *endpoint* DBpedia. Hasil ekstraksi dokumen pada *endpoint* DBpedia tersebut yang kemudian diyakini sebagai jawaban. Jika pada hasil ekstraksi dokumen pertama tidak ditemukan, kemungkinan jawaban berada pada ekstraksi dokumen pada *endpoint* kedua. Begitu juga seterusnya sampai ekstraksi dokumen pada *endpoint* DBpedia yang ketiga. Proses ini selaras dengan proses disaat Nabi Ibrahim menemukan bintang dan bulan kemudian ketika bintang dan bulan menghilang, selanjutnya melihat matahari dan Nabi Ibrahim kecewa karena matahari menghilang. Hingga Nabi Ibrahim meyakini bahwa Allah SWT merupakan Tuhan Yang Maha Pencipta yang menciptakan alam semesta.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang diperoleh dari penelitian skripsi ini adalah sebagai berikut :

- a. Aplikasi Tanya Jawab dibangun dengan menerapkan *semantic web* pada sistem. *Semantic web* menjadi proses inti penggalian jawaban. Untuk mendapatkan jawaban, dilakukan pemrosesan pertanyaan yakni diawali dengan *tokenizing*, *filtering* dan *stemming*. Metode *stemming* yang digunakan adalah Algoritma Nazief & Adriani. Dari hasil pemrosesan pertanyaan tersebut diperoleh *keyword* untuk melakukan pencarian *resource*. *Resource* digunakan sebagai URI *resource* atau petunjuk letak dokumen pada DBpedia. Setelah *resource* ditemukan, dokumen DBpedia tersebut kemudian diekstrak dengan menggunakan bahasa *query* SPARQL. Hasil ekstrak dokumen selanjutnya ditampilkan sebagai jawaban atas pertanyaan yang telah diajukan.
- b. Akurasi jawaban yang diperoleh dengan adanya pemrosesan pertanyaan menggunakan Algoritma *stemming* Nazief & Adriani mencapai 83,81%. *Stemming* merupakan bagian penting dalam suatu sistem temu balik karena hasil *stemming* mempengaruhi relevansi jawaban yang dihasilkan oleh sistem.

5.2. Saran

Untuk pengembangan aplikasi tanya jawab ini di kemudian hari diperlukan beberapa perbaikan dan tambahan untuk mendapatkan hasil yang lebih maksimal, diantaranya :

1. Menambah ekstraksi jawaban lebih dari 1 *property* yang ada pada dokumen DBpedia, sehingga data yang didapatkan bisa lebih banyak dan bisa jadi memiliki akurasi yang lebih baik.
2. Membangun aplikasi tanya jawab dalam versi *mobile* sehingga bisa diakses dimanapun dan kapanpun.
3. Membangun database mapping *resource* atau menggunakan metode lain untuk melakukan proses pencarian *keyword* sehingga lebih tepat dalam menemukan ekstrak dokumen DBpedia yang dibutuhkan.

DAFTAR PUSTAKA

- Al-Qur'an dan Terjemahannya*. Departemen Agama RI. Jakarta : PT Sygma Examedia Arkanleema.
- Agusta, Ledy. 2009. *Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk stemming dokumen teks bahasa Indonesia*. Bali : Konferensi Nasional Sistem dan Informatika.
- Anggraeny, Meinar Dwi. 2008. *Skripsi : Implementasi Question Answering dengan Metode Rule-Based pada Terjemahan Al Qur'an Surat Al Baqarah*. Bandung : IPB.
- Assuyuthi, Jalaluddin dan Jalaluddin Muhammad Ibnu Ahmad Almahally. 2008. *Tafsir Jalalain Terjemah*. Tasikmalaya : Pustaka Alhidayah.
- Ayuningtyas, Nuriana. 2009. *Skripsi : Implementasi Ontologi Web dan Aplikasi Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia*. Depok : Universitas Indonesia.
- Djajasudarman, Fatimah. 1993. *Semantik 1 : Pengantar ke Arah Ilmu Makna*. Bandung : PT REFIKA anggota IKAPI.
- Gunawan dan Fandli Halim. 2014. *Penerapan Web Semantik untuk Aplikasi Pencarian pada Repositori Koleksi Penelitian, Studi Kasus : Program Studi Sistem Informasi STMIK Mikroskil Medan*. Medan : JSM STMIK Mikrosil Vol 15, No 1.
- Irwan, Calvin dan Ayu Purwarianti. 2012. *Studi dan Implementasi Question Answering System Bahasa Indonesia pada Perangkat Bergerak Domain National Basketball Association*. Bandung : Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika, Vol. 1, No. 3 Oktober 2012.
- Li, Xin dan Dan Roth. 2002. *Learning Question Classifiers*. USA : Proceeding of the 19th International conference on Computational Linguistics – Vol 1 (COLING '02), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA.
- Lutfi, Citra Rosiana. 2012. *Skripsi : Question Answering System pada terjemah juz Amma menggunakan metode Rule Based*. Malang : UIN Maulana Malik Ibrahim.
- Mait, Hitoshi Mario Naga. 2012. *Making Stemming Synonym Indonesian using Algorithm Nazief and Adriani*. Jakarta : Universitas Gunadarma.

- Paryudi, Iman dan Naniek Andiani. 2010. *Intelligent Recommender pada Sistem E-Learning menggunakan Semantic Web*. Bandung : Seminar dan Call for Paper Munas Aptikom Politeknik Telkom.
- Rahman, Fajrul. 2013. *Skripsi : Aplikasi stemming pada kamus bahasa Indonesia dengan pendekatan Algoritma Nazief & Adriani*. Jakarta : UIN Syarif Hidayatullah.
- Tahitoe, Andita Dwiyoga dan Diana Purwitasari. *Implementasi Modifikasi Enhanced Confix Stripping Stemmer untuk Bahasa Indonesia dengan Metode Corpus Based Stemming*. Surabaya : ITS.
- Zebua, Javier dan Metty Mustikasari. 2012. *Aplikasi Pencarian Buku Perpustakaan dengan menggunakan RDF dan SPARQL*. Jakarta : UG Jurnal Vol. 6 No. 04.
- DBpedia. 2014. DBpedia. <http://id.wikipedia.org/wiki/DBpedia>. Diakses tanggal 10 Oktober 2014.
- Internet World Stats. 2014. Top 20 Internet Countries by Users. <http://www.internetworldstats.com/top20.htm>. Diakses tanggal 6 April 2014.
- START. 2014. START Natural Language Question Answering System. <http://start.csail.mit.edu/index.php>. Diakses tanggal 21 Oktober 2014.
- Statistika Wikipedia bahasa Indonesia. 2014. Wikipedia:Statistik. <http://id.wikipedia.org/wiki/Wikipedia:Statistik>. Diakses tanggal 10 Oktober 2014.
- Wikipedia bahasa Indonesia. 2014. Wikipedia bahasa Indonesia. http://id.wikipedia.org/wiki/Wikipedia_bahasa_Indonesia. Diakses tanggal 10 Oktober 2014.

LAMPIRAN

NO	PERTANYAAN	JAWABAN	
		TEPAT	TIDAK
1	Apa yang dimaksud dengan <i>smartphone</i> ?	√	
2	Apa warna bendera negara Jerman?	√	
3	Apakah pengertian dari <i>hardware</i> ?	√	
4	Apa nama maskot kota Surabaya?	√	
5	Apa arti semboyan Bhineka Tunggal Ika?	√	
6	Apa nama tarian khas Bali?		√
7	Apa makanan khas Yogyakarta?	√	
8	Apakah judul lagu kebangsaan Indonesia?		√
9	Apa teori yang ditemukan oleh Albert Einstein?		√
10	Apa nama planet yang paling dekat dengan matahari?	√	
11	Apa nama planet yang memiliki cincin?	√	
12	Apa nama kota yang disebut kota hujan?	√	
13	Apa makanan khas Lamongan?	√	
14	Apa nama gunung di kota Batu?	√	
15	Apa nama jembatan di kota Palembang?	√	
16	Dimana letak sungai Musi?	√	
17	Dimana letak tugu Monas?	√	
18	Dimana letak museum Radya Pustaka?	√	
19	Dimana letak makam Bung Karno?	√	
20	Dimanakah letak Jembatan Ampera?	√	
21	Dimanakah Pangeran Diponegoro dimakamkan?	√	
22	Dimana letak kerajaan Majapahit?	√	
23	Dimana letak Gunung Bromo?	√	
24	Dimana tugu pahlawan berada?	√	
25	Dimana tempat ibadah umat muslim?	√	
26	Dimana letak jembatan suramadu?	√	
27	Dimana letak pulau Raja Ampat?	√	
28	Dimana lokasi Institut Teknologi Telkom?	√	
29	Dimana lokasi Taman Safari Indonesia 2?	√	
30	Dimana kota asal bahasa Osing?	√	
31	Siapa nama gubernur Jawa Timur?	√	
32	Siapa nama istri Bung Karno?	√	
33	Siapa pembuat sosial media Facebook?	√	
34	Siapakah nama walikota Surabaya?	√	
35	Siapakah CEO Apple Inc?	√	
36	Siapakah nama kiper Arema?	√	
37	Siapa pencipta lagu Indonesia Raya?	√	

38	Siapa pencetus teori gravitasi?		√
39	Siapa pencetus teori relativitas?	√	
40	Siapakah CEO Microsoft saat ini?	√	
41	Siapakah proklamator kemerdekaan Indonesia?	√	
42	Siapa vokalis Ungu band?	√	
43	Siapa penemu listrik?		√
44	Siapa nama istri SBY?	√	
45	Siapa nama gubernur Bank Indonesia?	√	
46	Kapan Indonesia dinyatakan merdeka?	√	
47	Kapan Gus Dur mulai menjabat sebagai presiden?	√	
48	Kapan Dude Harlino lahir?	√	
49	Kapan Nagita Slavina memulai karir?	√	
50	Kapan tanggal lahir Bunga Citra Lestari?	√	
51	Kapan sholat Idul Adha berlangsung?	√	
52	Kapan perayaan natal berlangsung?	√	
53	Kapan masa jabatan Jokowi sebagai Gubernur DKI Jakarta?	√	
54	Kapan Hari Kebangkitan Nasional diperingati?		√
55	Kapan Apollo 11 tiba di Bulan?	√	
56	Kapan Soeharto berhenti menjadi presiden?	√	
57	Kapan PKI dibubarkan?	√	
58	Kapan TNI dibentuk?	√	
59	Kapan terjadinya perjanjian hudaibiah?	√	
60	Kapan terjadinya perang badar?	√	
61	Mengapa Jepang menyerah terhadap Belanda tahun 1945?	√	
62	Mengapa terjadi perang sampit?	√	
63	Mengapa Idul Adha disebut juga hari raya haji?	√	
64	Mengapa Jogja dijuluki sebagai daerah istimewa?		√
65	Mengapa terjadi gerhana bulan?	√	
66	Mengapa terjadi gerakan Aceh merdeka?	√	
67	Mengapa Bogor disebut kota hujan?	√	
68	Mengapa Bali disebut pulau Dewata?		√
69	Mengapa Pontianak disebut kota Khatulistiwa?	√	
70	Mengapa terjadi krisis moneter tahun 1998?	√	
71	Mengapa Korea berpisah menjadi 2 wilayah?	√	
72	Mengapa Malin Kundang dikutuk menjadi batu?	√	
73	Mengapa Surabaya disebut kota Pahlawan?	√	
74	Mengapa Edelweiss disebut bunga abadi?		√
75	Mengapa Timor Leste berpisah dari Indonesia?	√	
76	Bagaimana proses metamorfosis berlangsung?	√	
77	Bagaimana proses membuat tempe?	√	

78	Bagaimanakah proses terbentuknya pelangi?		√
79	Bagaimana cara menanam jagung?		√
80	Bagaimana cara menginstal ulang Windows?		√
81	Bagaimanakah cara merawat Kucing Anggora?		√
82	Bagaimana cara mengukur presisi?	√	
83	Bagaimana aturan permainan gobak sodor?		√
84	Bagaimana cara mengukur akurasi?	√	
85	Bagaimana aturan permainan catur?	√	
86	Bagaimana proses terjadinya hujan?	√	
87	Bagaimana proses terjadinya fotosintesis?	√	
88	Bagaimana cara mencangkok tumbuhan?	√	
89	Bagaimana cara menanam padi?		√
90	Bagaimana proses terjadinya gerhana matahari?	√	
91	Berapa panjang jalan tol Surabaya-Gempol?	√	
92	Berapa tinggi tugu Monas?	√	
93	Berapa banyak pulau di Indonesia?	√	
94	Berapa luas Kebun Raya Purwodadi?	√	
95	Berapa banyak suku yang ada di Indonesia?	√	
96	Berapa tinggi gunung Bromo?	√	
97	Berapa luas wilayah Taman Nasional Bromo Tengger Semeru?	√	
98	Berapa diameter bunga Raflesia?	√	
99	Berapa jarak Bumi dengan Matahari?	√	
100	Berapa lama proses rotasi Bumi?	√	
101	Berapa jarak dari surabaya ke jakarta?	√	
102	Berapa personel JKT48?	√	
103	Berapa tekanan darah normal manusia?	√	
104	Berapa warna yang ada pada pelangi?		√
105	Berapa jumlah episode Running Man?	√	