Analisis Ekstraksi Pengetahuan Eksternal untuk Question Answering System

Abdiansah
Laboratorium Sistem Cerdas, UGM
Departemen Ilmu Komputer, Universitas Sriwijaya
Email: abdiansah@unsri.ac.id

Sri Hartati
Laboratorium Sistem Cerdas, UGM
Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada
Email: shartati@ugm.ac.id

***Abstract—Question Answering System* (QAS) merupakan sistem tanya jawab yang dapat memberikan jawaban secara langsung kepada pengguna dalam bentuk bahasa alami. Untuk mencari sebuah jawaban, QAS menggunakan pengetahuan baik internal maupun eksternal. Salah satu pengetahuan eksternal adalah Internet yang memiliki sumber informasi yang berlimpah. Penelitian ini mencoba untuk melakukan analisis penggunaan pengetahuan eksternal untuk digunakan oleh QAS. Ada tiga sumber *corpus* yang digunakan yaitu: Wikipedia, Google dan Bing. Hasil dari penelitian ini adalah banyaknya data yang berhasil diperoleh dan jumlah jawaban yang dapat diekstraksi. Bing memperoleh hasil *retrieval* dan ekstraksi jawaban lebih banyak dari Google yaitu sebesar 372 dokumen dan 72 kemungkinan jawaban, sedangkan Google sebesar 345 dokumen dengan 68 kemungkinan jawaban. Sedangkan Wikipedia memberikan sedikit dokumen karena *corpus* yang digunakan berjumlah 13 file html berbeda dengan Google dan Bing yang berjumlah 130 file html. Walaupun dokumen dan ekstraksi jawaban Bing lebih besar dari Google tetapi Bing gagal mengekstraksi jawaban untuk dua *corpus*, sedangkan Google hanya gagal untuk satu *corpus*.**

Keywords—question answering system; corpus; wikipedia; Google; Bing; retrieval; ekstraksi jawaban

I. PENDAHULUAN

Question Answering System (QAS) atau Sistem Tanya Jawab merupakan sistem yang dapat memberikan jawaban langsung kepada pengguna, berbeda dengan sistem yang berbasis *Information Retrieval* (IR) seperi *search engine*, yang memberikan hasil berupa daftar tautan yang relevan. QAS memberikan jawaban dengan benar dan tepat, sehingga dibutuhkan usaha yang tidak mudah untuk membangun sistem tersebut. Secara umum, QAS terdiri dari tiga komponen utama yaitu: 1) *Question Analysis*; 2) *Passage Retrieval*; dan 3) *Answer Extraction* [7]. Setiap komponen tersebut mempunyai modul-modul spesifik terhadap permasalahan tertentu, misalnya pada komponen *Question Analysis* terdapat beberapa modul seperti *Parsing*, *Question Classification*, *Query Reformulation* [5]. QAS dapat berbentuk *open-domain* dan *closed-domain*. Pada *closed-domain*, biasanya sistem ini mempunyai basis-pengetahuan yang terstruktur dengan cakupan permasalahan yang tidak terlalu luas. Sebaliknya, pada *open-domain* umumnya menggunakan basis-pengetahuan dari data yang tidak terstruktur seperti informasi dari web, serta *open-problem* yang berarti QAS harus bisa menjawab semua pertanyaan umum. Penelitian QAS dengan *open-domain* masih terus dieksplorasi karena kompleksitasnya cukup tinggi. Telah banyak penelitian yang melakukan ekstraksi pengetahuan dari Internet seperti yang dilakukan oleh [4][8][1][2][3][9][10][6][11]. Dari penelitian yang sudah ada dapat disimpulkan bahwa modul untuk ekstraksi pengetahuan eksternal sangat diperlukan oleh QAS. Artikel ini berisi hasil penelitian tentang ekstraksi pengetahuan eksternal untuk *Question Answering System* dengan sumber data diambil dari situs Wikipedia[1], serta mesin pencari Google[2] dan Bing[3]. Pembahasan dimulai dari pemilihan domain studi kasus sampai dengan percobaan untuk melihat hasi *retrieval* dan jawaban dari sistem yang dibuat.

Dalam dunia pendidikan, pelajaran sejarah merupakan mata pelajaran penting untuk anak sekolah dasar sampai menengah ke atas, karena dengan memahami sejarah mereka akan memperoleh pengetahuan dari masa lalu. Kegiatan utama dalam belajar sejarah adalah dengan membaca secara utuh topik yang akan dipelajari, jika tidak utuh maka informasi yang diterima akan menjadi bias. Selain itu, mata pelajaran ini sarat bacaan tekstual dan terkadang butuh beberapa halaman untuk topik tertentu. Oleh karena itu, dalam penelitian ini digunakan domain sejarah karena memiliki banyak informasi tekstual yang dapat digunakan untuk percobaan ekstraksi pengetahuan. Topik sejarah yang digunakan dalam penelitian ini adalah Kerajaan Nusantara yang ada di Indonesia pada masa Hindu-Budha. Data kerajaan diambil dari Wikipedia, Google dan Bing, dimana terdapat 13 kerajaan yaitu: Medang, Majapahit, Sailendra, Sunda, Kahuripan, Sriwijaya, Dharmasraya, Singhasari, Kalingga, Kediri, Malayapura, Tarumanegara, Kutai. Data-data tersebut akan dibuat menjadi *corpus* teks. Pada Wikipedia, data diambil sesuai dengan informasi kerajaan yang ada di web tersebut, sedangkan pada Google dan Bing akuisi data menggunakan kata kunci.

Penelitian ini menggunakan aplikasi bernama QASNUS (*Question Answering System NUSantara*) untuk membantu melakukan pengujian. QASNUS dibuat menggunakan bahasa pemrograman Java versi 1.8.0_25, IDE Netbeans 8.0.2 dan basis data MySQL 5.5.4. Komputer yang digunakan mempunyai spesifikasi sebagai berikut: Intel Celeron B815 - 1.60 Ghz (2 CPU), RAM – 3.8 GiB, sedangkan sistem operasi menggunakan LINUX distro Kubuntu 14.04 LTS – 32 bit.

Selanjutnya artikel ini disusun sebagai berikut, Bagian 1 berisi latar belakang penelitian, Bagian 2 menjelaskan tentang metodologi yang digunakan, Bagian 3 menampilkan hasil

1 id.wikipedia.org
2 www.Google.com
3 www.Bing.com

percobaan yang telah dilakukan dan Bagian 4 kesimpulan.

II. METODOLOGI

Pada bagian ini menjelaskan tentang metodologi penelitian yang digunakan. Terdapat 3 tahapan kegiatan yang dilakukan yaitu:

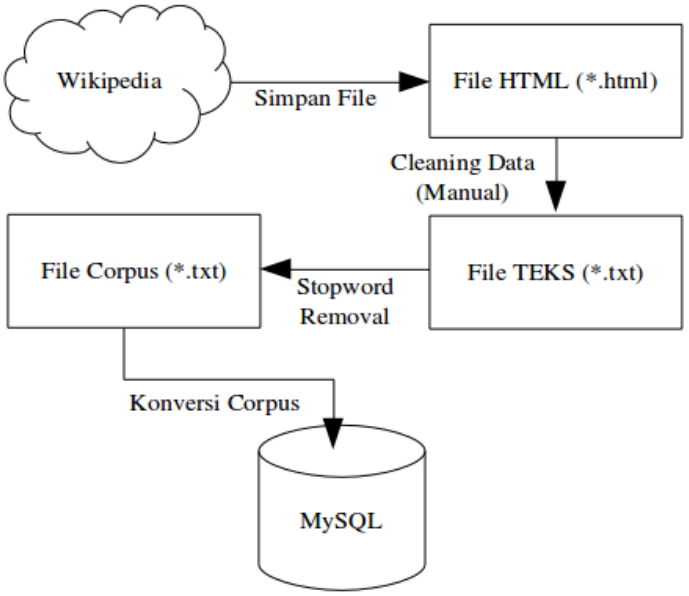
1) Pembuatan *corpus* yang meliputi: data internal (Wikipedia) dan data eksternal (Google dan Bing)
2) Pra-Pengolahan yang terdiri dari: Konversi Html ke Teks dan *Stopword Removal*
3) Proses QAS yaitu: *Question Analysis, Passage Retrieval, Answer Extraction* dan arsitektur QASNUS

A. Pembuatan Corpus

Secara garis besar ada dua jenis *corpus* berdasarkan sumber data yaitu dari Wikipedia dan dari mesin pencari Google dan Bing.

Data Internal (Wikipedia)

Data diambil dari website Wikipedia Indonesia, saat ini (tanggal akses: 20 Oktober 2014) terdapat 249.492 artikel berbahasa Indonesia. Terdapat 8 kategori utama yang didalamnya mempunyai sub-sub kategori. Salah satu kategori tersebut adalah kategori "Sejarah" yang memiliki 11 sub-kategori. "Kerajaan Nusantara" merupakan salah satu sub-kategori yang ada ditingkat 3. Sub-kategori "Kerajaan Nusantara" memiliki 64 sub-kategori yang setiap sub mewakili tautan berisi informasi suatu kerajaan. Setiap tautan kerajaan umumnya memiliki sub-kategori lagi dan maksimal 7 kategori, halaman artikel paling sedikit 1 halaman dan paling banyak 66 halaman (1 halaman 1 file html).



Gambar 1. Tahapan Akuisi Data Wikipedia

Setiap satu artikel kerajaan (termasuk sub-kategori) akan disimpan ke dalam flat file (*.txt). Dilakukan pra-pengolahan secara manual terhadap format teks tanpa mengurangi isi dari artikel. Teks tersebut dibuat perbagian (paragraf/*passage*). Untuk kegiatan analisis pertanyaan digunakan dua kerajaan yaitu kerajaan Sriwijaya (1K – satu kategori, 24H – dua empat halaman) dan Majapahit (1K, 66H). Kedua kerajaan tersebut memiliki informasi yang lebih banyak dibanding kerajaan lain. Selanjutnya dari semua Kerajaan Nusantara, hanya diambil 13 kerajaan saja (13 file html) berdasarkan label Hindu-Budha. Tabel 1 berisi nama *corpus* yang diwakili oleh nama kerajaan beserta jumlah kalimat untuk tiap-tiap *corpus*. Total keseluruhan kalimat berjumlah 1.268 kalimat.

TABEL 1. CORPUS DAN JUMLAH KALIMAT

Corpus	**Jumlah Kalimat**
Sriwijaya	247
Majapahit	241
Sunda	144
Medang	129
Sailendra	117
Kediri	78
Tarumanegara	70
Singhasari	70
Dharmasraya	54
Kahuripan	39
Kutai	37
Kalingga	35
Malayapura	7
Total	**1.268**

Pada Gambar 1 dapat dilihat tahapan akuisisi data dari Wikipedia, dimulai dari akses halaman Wikipedia yang kemudian disimpan ke dalam file html. Selanjutnya file html diubah ke dalam file teks dengan dilakukan beberapa proses *cleaning* data secara manual seperti penghilangan format teks. File teks diproses menggunakan *stopword removal* untuk dihilangkan kata-kata umum seperti kata: *yang*, *bagi*, *tapi* dan lainnya. Setelah itu file *corpus* diproses untuk diambil per-kalimat (dipisahkan oleh tanda titik) dan disimpan ke dalam basis data MySQL.

Data Eksternal (Google & Bing)

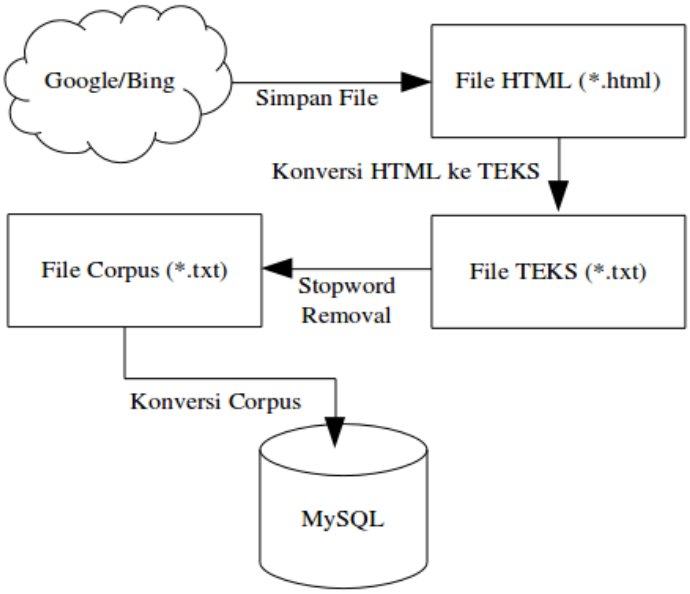
Akuisi data eksternal menggunakan mesin pencari Google dan Bing, dengan memberikan kata kunci sebagai berikut:

"*kapan kerajan <NK> berdiri?*"

dimana <NK> adalah salah satu *template* pertanyaan. Halaman pertama hasil pencarian baik Google dan Bing memberikan 10

tautan ke web yang relevan dengan kata kunci yang diberikan. Setiap kerajaan akan diambil 10 tautan sehingga diperoleh 130 file html. Semua file html tersebut dihitung total kalimatnya dan dijumlahkan sehingga didapat total untuk kalimat untuk 10 tautan.

Pada Gambar 2 dapat dilihat tahapan akuisi data dari Google dan Bing mirip dengan akuisisi data dari Wikipedia, yang membedakan hanyalah proses perubahan data dari html ke teks. Proses konversi HTML ke TEKS otomatis menggunakan pustaka JSoup[4]. Tabel 2 berisi informasi jumlah kalimat untuk tiap *corpus* berdasarkan sumber yang diambil.



Gambar 2. Tahapan Akuisi Data Google dan Bing

TABEL 2. CORPUS DAN JUMLAH KALIMAT

Corpus	**Jumlah Kalimat**	
	Google	**Bing**
Sriwijaya	1.290	568
Majapahit	1.957	969
Sunda	1.366	948
Medang	2.226	1.606
Sailendra	1.399	1.771
Kediri	966	1.051
Tarumanegara	866	784
Singhasari	1.051	877
Dharmasraya	1.060	1.262
Kahuripan	1.342	1.731
Kutai	623	919
Kalingga	1.002	658
Malayapura	907	5.048
Total	**1.6055**	**18.192**

B. Pra-Pengolahan

Secara teknis terdapat dua pra-pengolahan dalam penelitian ini yaitu: 1) proses konversi file Html ke file Teks; 2) proses penghilangan kata yang menjadi kata *stopword*. Untuk pra-pengolahan pertama ada dua jenis yaitu secara manual dan otomatis. Sedangkan untuk proses kedua diproses secara otomatis.

Konversi Html ke Teks

Untuk proses konversi otomatis dibuat aplikasi menggunakan bahasa java yang memanggil pustaka Jsoup untuk melakukan tugas konversi file Html ke file Teks. Proses konversi Html ke Teks secara manual dilakukan dengan prosedur sebagai berikut:

- Simpan file Html menjadi file Teks menggunakan aplikasi web browser
- Lakukan proses *cleaning* data yaitu dengan menghilangkan format-format teks seperti paragraf, spasi yang berlebihan, teks yang bertipe list (daftar) dan lainnya.
- Simpan file

Stopword Removal

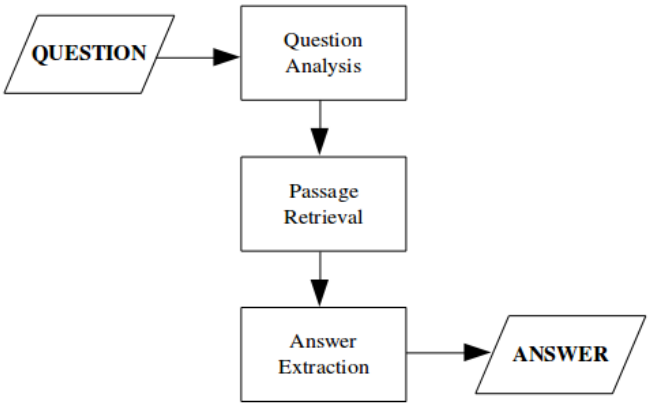
Setelah *corpus* dikonversi menjadi file teks, tahap berikutnya adalah menghilangkan kata-kata *stopword* yaitu kata-kata yang tidak berpengaruh dengan kalimat seperti: *yang*, *akan*, *di*, *ke* dan lainnya. Daftar *stopword* disimpan dalam file stopwords.txt dan sumber *stopword* diambil dari Google[5] yang berbahasa Indonesia. Proses penghilangan kata *stopword* disebut dengan *Stopword Removal*. Dalam penelitian ini aplikasi *stopword* dibuat menggunakan bahasa pemrograman java. Luaran dari aplikasi ini akan bertipe file teks tetapi ekstensi filenya diberi nama *stopword* (*.stopword). Alasan penggunaan ekstensi tersebut adalah sebagai penanda bahwa itu adalah file hasil dari proses *stopword removal*.

C. Question Answering System (QAS)

Arsitekstur *Question Answering System* (QAS) secara umum dapat dilihat pada Gambar 3.

4 www.jsoup.org

5 https://code.Google.com/p/stop-words/



Gambar 3. Arsitekstur Umum *Question Answering System*

Pada Gambar 3 dapat dilihat terdapat tiga komponen utama yaitu: 1) *Question Analysis*; 2) *Passage Retrieval*; dan 3) *Answer Extraction* [7]. *Question Analysis* (QA) bertujuan untuk memproses pertanyaan dari pengguna. Beberapa pekerjaan yang ada dalam QA [5] adalah: *parsing*, *question classification*, dan *query reformulation*. *Passage Retrieval* atau *Document Analysis* (DA) bertujuan untuk mengambil *passage*/document yang mengandung jawaban dari pengguna, pekerjaannya adalah *extract candidate documents* dan *identify answers*. Terakhir *Answer Extraction* atau *Answer Analysis* (AA) yang bertujuan untuk mengambil jawaban yang tepat dari sekumpulan dokumen yang tersedia. Pekerjaan AA adalah: *extract candidate answers* dan *rank the best one*. Pada sub-bagian berikutnya akan dibahas ketiga komponen QAS terkait dengan penelitian yang dilakukan.

Question Analysis

Dalam penelitian ini, ada beberapa pekerjaan yang dilakukan pada tahapan *Question Analysis* yaitu:

- Menentukan Pola Pertanyaan
- Konversi Dokumen
- Penyimpanan Pola dan Term Pertanyaan
- Verifikasi Pertanyaan Pengguna

Pertanyaan yang diberikan pengguna berbentuk bahasa alami, oleh karena itu untuk memudahkan validasi pertanyaan dibuatlah validasi *grammer*. Metode yang digunakan adalah *Backus Naur Form* (BNF) sebagai aturan produksi yang dimengerti oleh sistem. Berikut ini contoh dari pertanyaan pengguna:

Q: Kapan kerajaan sriwijaya berdiri?

BNF: kapan kerajaan <NK> berdiri

Simbol <NK> merupakan simbol variabel yang bisa diisi dengan nama kerajaan seperti sriwijaya, majapahit, kediri dan lainnya. Pertanyaan-pertanyaan dalam bentuk BNF akan dijadikan pola yang nantinya akan disimpan ke dalam basis data.

Menentukan Pola Pertanyaan yaitu mencari dan menambah pola pertanyaan. Satu pertanyaan bisa berbentuk banyak pola. Untuk sementara terdapat 23 pola pertanyaan yang dibagi ke dalam 4 kelompok pertanyaan yaitu:

- Kelompok-1: tahun berdirinya kerajaan (8 pola)
- Kelompok-2: nama raja pertama/pendiri suatu kerajaan (7 pola)
- Kelompok-3: asal kerajaan dari seorang raja (4 pola)
- Kelompok-4: asal kerajaan dari suatu prasasti (4 pola)

Berikut ini contoh masing-masing pertanyaan tiap kelompok:

- Kelompok-1: pada tahun berapa kerajaan <NAMA_KERAJAAN> berdiri?
- Kelompok-2: siapa pendiri kerajaan <NAMA_KERAJAAN>?
- Kelompok-3: raja <NAMA_RAJA> berasal dari kerajaan mana?
- Kelompok-4: Berasal dari kerajaan mana prasasti <NAMA_PRASASTI>?

Setelah diperoleh pola pertanyaan, pekerjaan berikutnya adalah melakukan konversi dokumen *corpus* pertanyaan dan menyimpan pola pertanyaan ke dalam basis data MySQL. Untuk memudahkan proses komputasi maka sumber data yang berbentuk dokumen teks akan diubah dan disimpan ke dalam basis data. Proses konversi dilakukan secara otomatis menggunakan aplikasi yang telah dibuat (Corpus2MySql.java). Masukan aplikasi berupa file teks yang berisi *corpus* dan luaran berupa *record*/baris yang berisi kalimat teks. Hasil Konversi Dokumen untuk uji coba dua dokumen *corpus* yaitu: Untuk *corpus* 1 (Sriwijaya): Total 247 kalimat, yang berhasil disimpan sebanyak 236 kalimat. Sedangkan untuk *corpus* 2 (Majapahit): Total 241 kalimat, yang berhasil disimpan sebanyak 240 kalimat. Ada 12 kalimat yang gagal di konversi sehingga perlu dilakukan analisis kesalahan. Kegagalan konversi terletak pada *string* mysql, karena ada kalimat yang mengandung karaktek baku sintaks mysql seperti karakter (,), ", ' dan lainnya. Misalnya ada kalimat:

> "*Sumber utama yang digunakan oleh para sejarawan adalah Pararaton ('Kitab Raja-raja') dalam bahasa Kawi dan Nagarakretagama dalam bahasa Jawa Kuno*"

Pada kalimat di atas mengandung karakter (,) dan ' sehingga akan terjadi kesalahan pada saat dilakukan proses memasukan data. Untuk mengatasi masalah diatas maka harus dilakukan lagi pembersihan data (*cleaning*) khusus untuk karakter-karakter yang sama dengan karakter baku mysql. Proses *cleaning* dilakukan dengan cara mengganti karakter ' " ( ) menjadi karakter _ . Untuk selanjutnya harus diingat bahwa karakter _ merupakan pengganti dari karakter-karakter: ' " ( ) dan bisa dikatakan sebagai karakter pengganti. Pada pengujian pertama masih terdapat kesalahan yaitu kapasitas penampung data yang sedikit (VARCHAR - 500 karakter), setelah diubah menjadi tipe data TEXT (65.535 karakter) maka hasil yang didapat lebih baik dan semua kalimat bisa masukan ke dalam basis data. Setelah diperbaiki maka diperoleh *record* sebanyak

488 kalimat yang sebelumnya berjumlah 476 kalimat.

Berikutnya adalah proses penyimpanan pola dan *term* pertanyaan. Terdapat 23 pola kalimat pertanyaan yang dibagi ke dalam 4 kelompok. Pola-pola tersebut dimasukan ke dalam basis data secara manual menggunakan aplikasi yang telah dibuat (EntryPolaTerm.java). Kalimat yang dimasukan pengguna akan dilakukan verifikasi dengan pola-pola tersebut. Data *term* seperti nama kerajaan, nama raja dan nama prasasti disimpan ke dalam basis data secara manual. Data *term* digunakan untuk membantu proses verifikasi *template*, apakah pertanyaan yang dimasukan pengguna sudah sesuai atau belum dengan *template* pertanyaan. Contoh: Sriwijaya, Majapahit, Singasari = NK, Brawijaya, Wijaya = NR.

Tahap terakhir *Question Analysis* adalah verifikasi pertanyaan pengguna. Verifikasi bertujuan untuk mendeteksi apakah pertanyaan pengguna sudah sesuai dengan pola pertanyaan yang tersimpan. Beberapa pengujian telah dilakukan dan hasil yang didapatkan cukup baik. Algoritma verifikasi pola menggunakan teknik *query* untuk mendeteksi apakah pertanyaan sudah sesuai dengan pola yang tersimpan. Teknik *query* juga digunakan untuk mengganti *term* yang ada dalam pertanyaan menjadi simbol *term*, seperti: Kerajaan sriwijaya = kerajaan <NK>. Hasil dari pengujian sangat akurat karena verifikasi berdasarkan pola yang tersimpan.

Algoritma ini akan lambat serta tidak efisien (*entry* manual) jika data pola dan basis data *term* berjumlah besar. Diperlukan optimasi algoritma untuk mengatasi kekurangan tersebut. Salah satu solusinya adalah *generate* pola otomatis dari sekumpulan dokumen menggunakan *machine learning*. Menggunakan *Named Entity Recognition* (NER) untuk pengenalan objek *term* dalam dokumen. Pada Tabel 3 berisi algoritma yang digunakan.

TABEL 3. ALGORITMA VERIFIKASI PERTANYAAN

1. Input kalimat T = String.
   - T = Nama kerajaan dari raja 'Balaputra dewa'?
2. Hapus semua karakter selain huruf, spasi dan tanda kuote.
   - T = T.replaceAll("[^\\w^\\s^\\']", "")
3. Pecah T menjadi array kata-kata.
   - kata_kata[] = T.split("\\s+")
4. Cek apakah T berisi term jamak (tanda kuote).
   - Jika TIDAK lanjutkan langkah 8.
   - Jika YA lanjutkan langkah berikutnya
5. Modifikasi term jamak (TJ).
   - TJ = 'Balaputra dewa'.
   - TJ = Balaputra_dewa
6. Modifikasi kalimat.
   - T = Nama kerajaan dari raja 'Balaputra dewa'.
   - T = Nama kerajaan dari raja Balaputra_dewa
7. Pecah T menjadi array kata-kata.
   - kata_kata[] = T.split("\\s+")
8. Lakukan perulangan sebanyak N kali.
   - N = kata_kata.length() = 5
9. Buat variabel untuk menampung kata-kata.
   - Temp = Temp.concat(kata_kata[i])
10. Lakukan query.
    - SELECT * FROM db_pola where kalimat like '%Temp%'
11. Apakah query berisi record?
    - Jika YA lanjutkan langkah 15.
    - Jika TIDAK lanjutkan langkah berikutnya
12. Lakukan query.
    - SELECT * FROM db_term where kata = kata_kata[i]
13. Apakah query berisi record?
    - Jika TIDAK maka set Flag=TRUE dan lanjutkan langkah 15.
    - Jika YA lanjutkan langkah berikutnya
14. Ganti term dengan template term.
    - temp=temp.replace(kata_kata[i], "<"+db_term.getString("term")+">").
    - Balaputra_dewa = <NR>
15. Cek variabel Flag?
    - Jika FALSE, lanjutkan perulangan.
    - Jika TRUE, stop perulangan
16. Lakukan query.
    - SELECT * FROM db_pola where kalimat = Temp
17. Apakah query berisi record?
    - Jika YA maka POLA VALID.
    - Jika TIDAK maka POLA TIDAK VALID

Passage Retrieval

Dalam penelitian ini, tahapan *Passage Retrieval* adalah membuat *query reformulation* berdasarkan pertanyaan yang sudah valid, meliputi: analisis kelompok pertanyaan, menentukan kata kunci yang mewakili kelompok pertanyaan, dan membuat *query*. Tahap berikutnya adalah melakukan *query corpus* berdasarkan *query* yang telah dirumuskan tadi. *query* dibuat statis dan dieksekusi berdasarkan kelas pertanyaan yang diseleksi menggunakan aturan IF-THEN, misalnya ada pertanyaan: “siapa pendiri kerajaan sriwijaya?” maka pertanyaan tersebut akan masuk ke kelompok 2 dan *query* yang dibuat berbentuk:

*Query: “select * from db_kalimat like '%raja pertama%' or like '%pendiri%' or ...*

Hasil dari *answer retrieval* merupakan satu atau lebih kalimat hasil *query*. Keluaran dari *answer retrieval* akan dijadikan masukan untuk modul *answer extraction* yang akan mengekstraksi jawaban benar.

Answer Extraction

Pada tahapan ini sistem akan mencari jawaban yang benar berdasarkan hasil kueri dari tahap passage *retrieval*. Karena keterbatasan waktu penelitian, maka pengujian untuk modul ini dilakukan secara manual yaitu dengan menilai secara subjektif berdasarkan hasil yang diperoleh. Untuk akurasi tentu sangat tinggi karena dinilai langsung oleh seorang pakar yang mengetahui dengan jelas apakah suatu dokumen mempunyai jawaban atau tidak dari suatu pertanyaan.

QASNUS

Untuk membantu pengujian data maka dibuatlah aplikasi QASNUS (*Question Answering System NUSantara*). Pada sub bagian ini akan dijelaskan secara singkat tampilan serta alur kerja QASNUS. Pada Gambar 4 menunjukan alur kerja penggunaan QASNUS yaitu dimulai dari pra-pengolahan (manual dan otomatis), *stopword removal*, konversi *corpus*, entri pola dan term pertanyaan serta melakukan proses pencarian jawaban. Hasil akhir dari QASNUS masih pada

tahapan *Passage Retrieval*, untuk *Answer Extraction* dilakukan secara manual berdasarkan penilaian subjektif pakar.

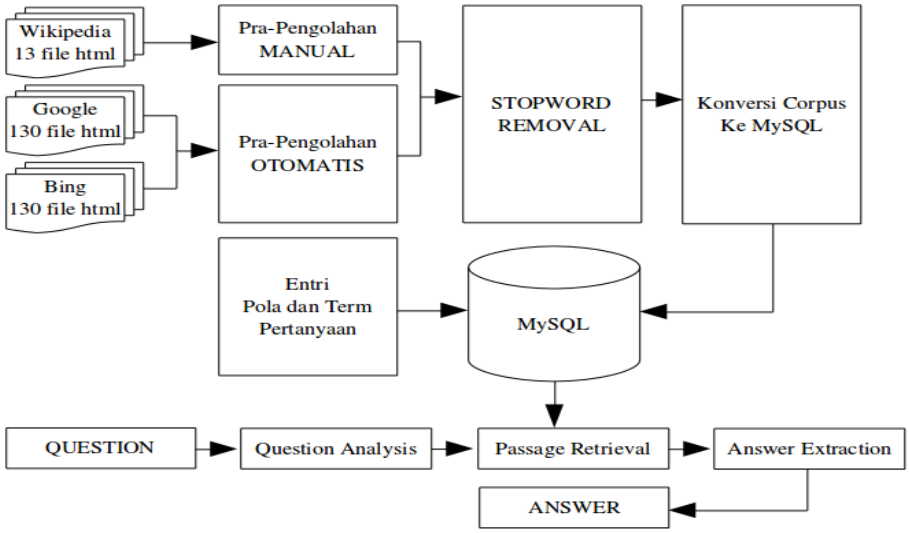
III. HASIL PERCOBAAN

Ada tiga percobaan yang telah dilakukan berdasarkan *corpus* Wikipedia, Google dan Bing. Untuk lebih singkat disebut dengan Percobaan-1 untuk *corpus* Wikipedia, Percobaan-2 untuk *corpus* Google dan Percobaan-3 untuk *corpus* Bing. Dari percobaan tersebut akan ditampilkan jumlah total *passage*/kalimat *corpus* yang berhasil disimpan ke basis data, jumlah kalimat yang berhasil di *retrieval* dan jumlah kalimat yang mengandung jawaban dari pertanyaan. Pertanyaan yang diajukan untuk ketiga percobaan hanya satu yaitu: "*kapan kerajaan <NK> berdiri?*". Dimana simbol <NK> akan diganti dengan nama ke-13 kerajaan. Jadi total pertanyaan ada 13 pertanyaan. Untuk percobaan-1, pertanyaan langsung diproses oleh QASNUS karena datanya sudah tersedia, sedangkan untuk percobaan-2 dan 3, pertanyaan diajukan ke mesin pencari (Goole dan Bing), hasil dari pencarian tersebut akan disimpan 10 link html yang akan diproses menjadi *corpus* sesuai dengan arsitektur QASNUS yang dapat dilihat pada Gambar 4.

A. Percobaan-1 (Wikipedia)

Data *corpus* untuk seluruh kerajaan yang bersumber dari Wikipedia berjumlah 1.268 kalimat. Rinciannya dapat dilihat pada Tabel 1. Sedangkan pada Tabel 4 dapat dilihat hasil *retrieval* dan jawaban untuk setiap kerajaan, dengan total *retrieval* berjumlah 9 dokumen yang dapat diambil. Pada Tabel 4 dapat dilihat bahwa terdapat 6 kerajaan yang berhasil diambil walaupun berjumlah sedikit yaitu: Sriwijaya, Medang, Kahuripan, Sunda, Singhasari dan Majapahit, sedangkan sisa kerajaan lainnya tidak bisa diambil.

Berdasarkan analisis hasil percobaan, kerajaan Sriwijaya dengan jumlah dokumen terbanyak memberikan hasil *retrieval* yang sedikit dibandingkan dengan kerajaan Kahuripan dengan ranking ke-10 dokumen terbanyak dari total 13 kerajaan. Ini menunjukan bahwa jumlah dokumen tidak berpengaruh terhadap *retrieval*. Salah satu penyebab *retrieval* yang sedikit dapat disebabkan oleh beberapa hal antara lainnya: 1) kalimat yang relevan yang ada dalam *corpus* memang tidak tersedia, contohnya untuk *corpus* Sriwijaya yang tidak memberikan secara spesifik kapan kerajaan tersebut berdiri. Dapat dibandingkan dengan *corpus* Majapahit yang ekplisit menjelaskan kapan kerajaan tersebut berdiri; dan 2) *query reformulation* yang belum spesifik dan akurat.



Gambar 4. Arsitekstur QASNUS

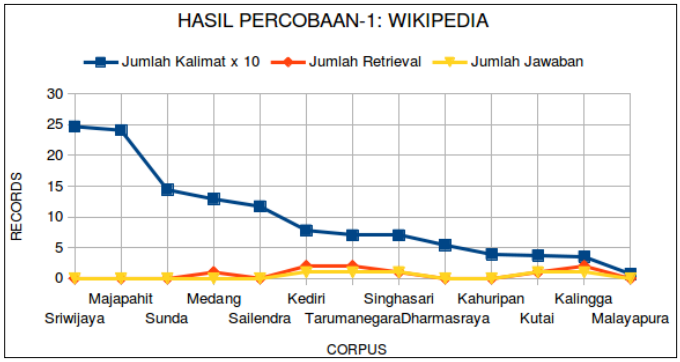
TABEL 4. JUMLAH RETRIEVAL & JAWABAN WIKIPEDIA

Corpus	Jumlah	
	Retrieval	Jawaban
Kutai	0	0
Tarumanegara	0	0
Kalingga	0	0
Sriwijaya	1	0
Sailendra	0	0
Medang	2	1
Kahuripan	2	1
Sunda	1	1
Kediri	0	0
Dharmasraya	0	0
Singhasari	1	1
Majapahit	2	1
Malayapura	0	0
Total	**9**	**5**

Proses pencarian jawaban dilakukan secara manual dengan melihat relevansi isi dokumen terhadap pertanyaan pengguna. Pada tabel tersebut dapat dilihat bahwa kerajaan sriwijaya yang memiliki satu dokumen hasil *retrieval* (lihat Tabel 4) ternyata dokumen tersebut tidak mengandung jawaban yang tepat. Sedangkan untuk kerajaan Medang, Kahuripan dan Majapahit yang masing-masing mempunyai 2 dokumen *retrieval*, masing-masing mempunyai satu dokumen yang mengandung jawaban yang tepat terhadap pertanyaan pengguna.

TABEL 5. HASIL KESELURUHAN PERCOBAAN-1

Corpus	Jumlah			
	Kalimat	Kalimat x10	Retrieval	Jawaban
Sriwijaya	247	24.7	0	0
Majapahit	241	24.1	0	0
Sunda	144	14.4	0	0
Medang	129	12.9	1	0
Sailendra	117	11.7	0	0
Kediri	78	7.8	2	1
Tarumanegara	70	7	2	1
Singhasari	70	7	1	1
Dharmasraya	54	5.4	0	0
Kahuripan	39	3.9	0	0
Kutai	37	3.7	1	1
Kalingga	35	3.5	2	1
Malayapura	7	0.7	0	0
Total	**1268**	**126.8**	**9**	**5**



Gambar 5. Grafik hasil percobaan-1

Percobaan-1 ini cukup memberikan gambaran secara umum bagaimana proses *Question Answering System* bekerja, walaupun hasil *retrieval* sedikit dan pencarian jawaban masih dilakukan secara manual. Pada Tabel 5 diberikan hasil secara keseluruhan untuk *corpus* Wikipedia. Jumlah kalimat yang ada pada grafik dibagi 10 yang berfungsi untuk menormalkan grafik, oleh karena itu dibuat label (kalimat x 10). Data diurutkan dimulai dari jumlah *record* paling banyak (Sriwijaya) sampai paling sedikit (Malayapura). Grafik 5 menunjukan perbandingan jumlah kalimat, jumlah *retrieval* dan jumlah jawaban secara grafik.

B. Percobaan-2 (Google)

Hasil dari percobaan-2 dapat secara keseluruhan dapat dilihat pada Tabel 6. Total jumlah kalimat yang berhasil disimpan ke dalam basis data MySQL sebanyak 16.055 *record* dengan kerajaan Medang yang paling banyak dan kerajaan Kutai paling sedikit. Total *record* yang berhasil diambil sebanyak 345 dengan kerajaan Majapahit tertinggi dan kerajaan Sailendra terendah. Sedangkan untuk total ekstraksi jawaban berjumlah 68 *record* dengan kerajaan Majapahit tertinggi dan kerajaan Sailendra terendah. Berdasarkan grafik yang ada pada Gambar 6 dapat dilihat jumlah kalimat yang diurutkan dimulai dari *record* paling banyak. Garis jumlah jawaban tampak mengikuti jumlah *retrieval* yang berarti semakin banyak jumlah *retrieval* maka akan semakin besar peluang untuk mendapatkan jawaban.

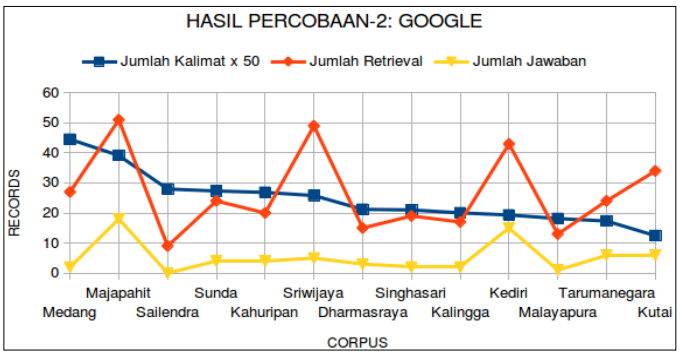
TABEL 6. HASIL KESELURUHAN PERCOBAAN-2

Corpus	Jumlah			
	Kalimat	Kalimat x10	Retrieval	Jawaban
Medang	2.226	44.52	27	2

Majapahit	1.957	39.14	51	18
Sailendra	1.399	27.98	9	0
Sunda	1.366	27.32	24	4
Kahuripan	1.342	26.84	20	4
Sriwijaya	1.290	25.8	49	5
Dharmasraya	1.060	21.2	15	3
Singhasari	1.051	21.02	19	2
Kalingga	1.002	20.04	17	2
Kediri	966	19.32	43	15
Malayapura	907	18.14	13	1
Tarumanegara	866	17.32	24	6
Kutai	623	12.46	34	6
Total	**1.6055**	**321.1**	**345**	**68**

C. Percobaan-3 (Bing)

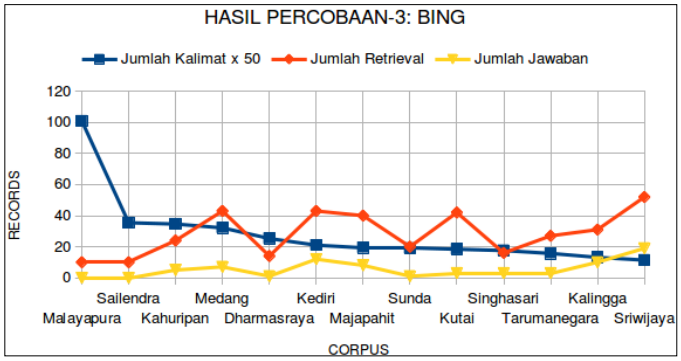
Hasil dari percobaan-3 dapat dilihat pada Tabel 7. Total jumlah kalimat yang berhasil disimpan ke dalam basis data MySQL sebanyak 18.192 *rekod* dengan kerajaan Malayapura yang paling banyak dan kerajaan Sriwijaya paling sedikit. Total *record* yang berhasil di *retrieval* sebanyak 372 dengan kerajaan Sriwijaya tertinggi dengan dua kerajaan terendah yaitu Malayapura dan Sailendra. Sedangkan untuk total ekstraksi jawaban berjumlah 72 *record* dengan kerajaan Sriwijaya tertinggi dengan kerajaan Malayapura dan Sailendra terendah.



Gambar 6. Grafik hasil percobaan-2

TABEL 7. HASIL KESELURUHAN PERCOBAAN-3

Corpus	Jumlah			
	Kalimat	Kalimat x10	Retrieval	Jawaban
Malayapura	5.048	100.96	10	0
Sailendra	1.771	35.42	10	0
Kahuripan	1.731	34.62	24	5
Medang	1.606	32.12	43	7
Dharmasraya	1.262	25.24	14	1
Kediri	1.051	21.02	43	12
Majapahit	969	19.38	40	8
Sunda	948	18.96	20	1
Kutai	919	18.38	42	3
Singhasari	877	17.54	16	3
Tarumanegara	784	15.68	27	3
Kalingga	658	13.16	31	10



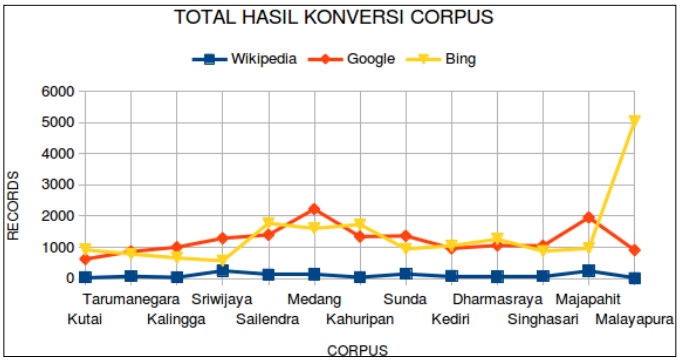
Sriwijaya	568	11.36	52	19
Total	**18.192**	**363.84**	**372**	**72**

Gambar 7. Grafik hasil percobaan-3

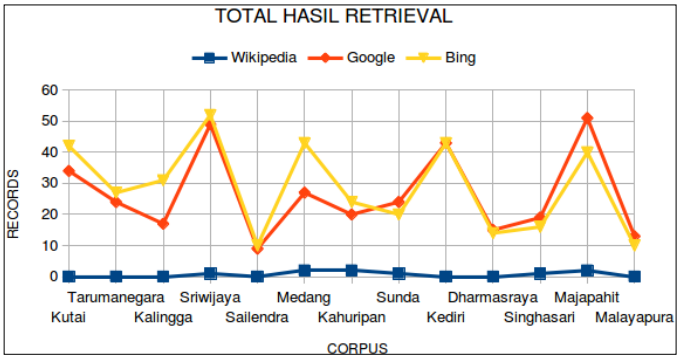
Berdasarkan grafik yang ada pada Gambar 7 dapat dilihat jumlah kalimat yang diurutkan dimulai dari *record* paling banyak. Kerajaan sriwijaya memiliki jumlah *record* yang paling sedikit tetapi tingkat *retrieval* dan ekstraksi jawaban paling tinggi diantara yang lainnya.

D. Hasil Seluruh Percobaan

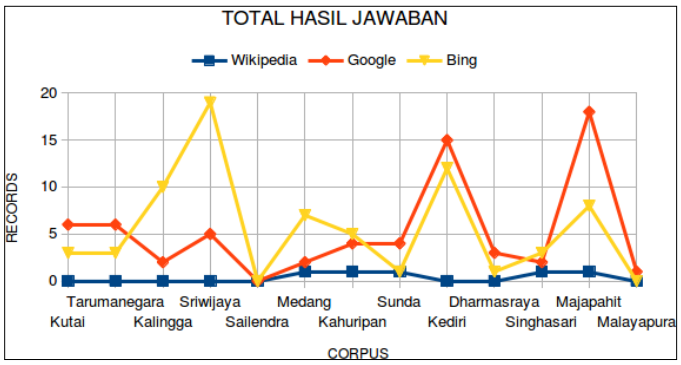
Pada bagian ini akan memberikan hasil dari keseluruhan percobaan yang dibagi menjadi tiga bagian yaitu: jumlah kalimat, jumlah *retrieval* dan jumlah jawaban untuk masing-masing *corpus* (Wikipedia, Google dan Bing). Hasilnya dapat dilihat pada Gambar 8–10. Gambar 8 menunjukan perbandingan hasil konversi *corpus* ke basis data MySQL. Hasil konversi tersebut berbentuk banyaknya *record* dalam basis data. Wikipedia memiliki *record* sedikit karena data *corpus* sedikit (13 file html) sedangkan untuk Google dan Bing tampak seragam kecuali untuk *corpus* Malayapura yang sangat menonjol.



Gambar 8. Grafik hasil percobaan-3



Gambar 9. Grafik hasil percobaan-3



Gambar 10. Grafik hasil percobaan-3

Untuk total hasil *retrieval* dapat dilihat pada Gambar 9, dimana Wikipedia menghasilkan *retrieval* yang rendah sedangkan untuk perbandingan Google dan Bing, Bing menghasilkan hasil *retrieval* yang lebih besar dibanding Google yaitu sebesar 372 dokumen sedangkan Google sebesar 345. Begitu pula untuk hasil ekstraksi jawaban yang dapat dilihat pada Gambar 10, Bing lebih besar dari Google yaitu sebesar 72 dokumen sedangkan Google sebesar 68 dokumen. Walaupun data *retrieval* dan ekstraksi jawaban Bing lebih besar dari Google tetapi *corpus* Google lebih unggul disisi ekstraksi jawaban karena berdasarkan Tabel 2, Google hanya gagal mengekstraksi jawaban untuk *corpus* Sailendra sedangkan Bing, lihat Tabel 3, gagal mengekstraksi dua *corpus* yaitu Malayapura dan Sailendra. Dari tabel-tabel tersebut dapat diketahui juga bahwa informasi yang berhubungan dengan Sailendra masih sedikit sehingga susah untuk mencari jawabannya.

IV. KESIMPULAN

Pengetahuan eksternal sangat penting untuk sistem yang berbasis pengetahuan, karena dengan pengetahuan eksternal dapat membuat sistem tersebut dinamis dan mampu beradaptasi terhadap permasalahan yang baru. Dalam penelitian *Question Answering System* (QAS), pengetahuan eksternal sangat dibutuhkan untuk QAS yang bersifat *open-domain* karena cakupan yang luas membutuhkan sumber pengetahuan yang besar pula. Penelitian ini mencoba untuk menganalisis ekstraksi pengetahuan dari Internet untuk QAS dengan tiga *corpus* utama yaitu: Wikipedia, Google dan Bing. Hasil dari penelitian ini adalah banyaknya dokumen yang diambil dan jumlah jawaban yang dapat diekstraksi. Bing memperoleh hasil *retrieval* dan ekstraksi jawaban lebih banyak dari Google, sedangkan Wikipedia paling sedikit karena *corpus*-nya berjumlah 13 file html berbeda dengan Google dan Bing yang berjumlah 130 file html. Walaupun data hasil *retrieval* dan ekstraksi jawaban Bing lebih besar dari Google tetapi Bing gagal mengekstraksi jawaban untuk *corpus* Malayapura dan Sailendra, sedangkan Google hanya gagal untuk *corpus* Sailendra.

Penelitian ini hanya fokus pada analisis ekstraksi pengetahuan dari Internet dengan batasan Wikipedia yang menjadi data internel serta Google dan Bing yang dijadikan data eksternal (dari Internet). Percobaan yang dilakukan hanya melihat jumlah dokumen yang berhasil dikonversi dan diambil serta jumlah jawaban yang berhasil diekstraksi menggunakan metode standar. Walapun demikian, dari beberapa percobaan yang dilakukan dapat memberikan hasil yang cukup untuk analisis awal tentang ekstraksi pengetahuan dari Internet. Oleh karena itu untuk penelitian selanjutnya, diutamakan untuk eksplorasi metode-metode ekstraksi teks dari web. Karena keterbatasan penelitian, untuk metode ekstraksi jawaban dilakukan secara manual yaitu dengan melihat satu persatu dokumen hasil *retrieval* dan dicari jawaban yang sesuai dengan pertanyaan pengguna. Pendekatan ini sangat akurat tetapi sangat tidak efisien dan efektif sehingga diperlukan eksplorasi lagi untuk metode ekstraksi jawaban.

REFERENSI

[1] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. 2003. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE*, *18*(1), 14-21.

[2] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Lewis, P. H., Hall, W., & Shadbolt, N. R. 2003. Automatic extraction of knowledge from web documents.

[3] Chang, C. H., Kayed, M., Girgis, M. R., & Shaalan, K. F. 2006. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, *18*(10), 1411-1428.

[4] Craven, M., McCallum, A., PiPasquo, D., Mitchell, T., & Freitag, D. 1998. *Learning to extract symbolic knowledge from the World Wide Web* (No. CMU-CS-98-122). Carnegie-mellon univ pittsburgh pa school of computer Science.

[5] Dwivedi, S. K., & Singh, V. 2013. Research and Reviews in Question Answering System. Procedia Technology, 10, 417-424.

[6] Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. 2008. Open information extraction from the web. *Communications of the ACM*, *51*(12), 68-74.

[7] Ezzeldin, A., & Shaheen, M. 2012. A Survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends. In *Proceedings of The 13th International Arab Conference on Information Technology (ACIT 2012)* (pp. 1-8).

[8] Laender, A. H., Ribeiro-Neto, B. A., da Silva, A. S., & Teixeira, J. S. 2002. A brief survey of web data extraction tools. *ACM Sigmod Record*, *31*(2), 84-93.

[9] Loh, S., Wives, L. K., & de Oliveira, J. P. M. 2000. Concept-based knowledge discovery in texts extracted from the Web. *ACM SIGKDD Explorations Newsletter*, *2*(1), 29-39.

[10] Pasca, M., Lin, D., Bigham, J., Lifchits, A., & Jain, A. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI* (Vol. 6, pp. 1400-1405).

[11] Wong, T. L., & Lam, W. 2007. Adapting web information extraction knowledge via mining site-invariant and site-dependent features. *ACM Transactions on Internet Technology (TOIT)*, *7*(1), 6.