

Implementasi *Semantic Search* untuk Mesin Pencarian pada Ensiklopedia Masjid Bersejarah di Indonesia

A'la Syauqi

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Maulana Malik Ibrahim Malang
Jl. Gajayana No. 50 Malang
asakuuki@yahoo.co.id

Ristanti Dian Farisah
Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Maulana Malik Ibrahim Malang
Jl. Gajayana No. 50 Malang
candle_rdf@yahoo.com

Abstract- Kebutuhan manusia akan informasi menyebabkan banyak terciptanya mesin pencari sebagai sarana untuk mempermudah manusia dalam hal pencarian, namun kadangkala pencarian membutuhkan waktu yang lama karena mesin pencari menampilkan banyak hasil pencarian yang mengandung kata yang dicari bukan sesuai dengan konteks kalimat pencarian yang dimasukkan pengguna. Hal tersebut menyebabkan pengguna harus membaca satu persatu dokumen bahkan melakukan pencarian ulang dengan kata kunci yang berbeda untuk menemukan dokumen atau informasi yang sesuai dengan konteks yang diharapkan. Penelitian ini menerapkan konsep *semantic search* dalam mesin pencari yang dimana dalam prosesnya dilakukan *indexing*, pembuatan ontologi, memetakan ontologi dengan menggunakan perangkat lunak *Smore*, kemudian dari hasil metada yang berupa *triple table* akan dilakukan penabelan *SPO* (Subjek, Predikat, Objek) dengan bantuan library API *Jena*, algoritma yang digunakan dalam pencocokan string dalam aplikasi ini adalah algoritma *Boyer-Moore*. Aplikasi ini diterapkan dalam ensiklopedia masjid bersejarah yang khusus memberikan informasi mengenai masjid bersejarah yang ada di Indonesia dengan memanfaatkan teknologi *smartphone* yang ber-OS *Android* sehingga aplikasi akan mudah diakses dimanapun dan kapanpun membutuhkannya. Berdasarkan pengujian dapat disimpulkan bahwa aplikasi yang diberi nama 'EMBI' ini mampu melakukan pencarian sesuai dengan konteks kalimat pencarian yang dimasukkan oleh pengguna, sehingga pencarianpun lebih efisien dengan ditampilkannya dokumen yang tepat dan akurat sesuai dengan keinginan.

Keywords- *Indexing, Ontologi, SPO, Semantic Search, Algoritma Boyer-Moore*

I. PENDAHULUAN

Indonesia merupakan negara yang mempunyai mayoritas penduduk beragama Islam terbesar di dunia^[4], sehingga negara ini memuat banyak informasi perkembangan agama dan budaya sebagai sumber pengetahuan sejarah Islam yang salah satunya ditandai dengan persebaran masjid yang tersebar di seluruh penjuru nusantara. Masjid yang merupakan tempat ibadah khusus umat muslim ini erat kaitannya dalam sejarah Islam untuk memperluas penyebaran agama. Keberadaannya mempunyai karakteristik yang berbeda sesuai dengan kebudayaan yang telah ditanamkan, sehingga dari sanalah kita dapat mengambil informasi mengenai nilai-nilai filosofis yang terkandung agar dapat memperdalam

pengetahuan kita mengenai perkembangan keIslaman pada jaman dahulu.

Perkembangan jaman yang menciptakan era baru di masyarakat kita lambat laun menyebabkan masyarakat kurang mengenali identitas diri daerahnya yang terkandung didalam nilai-nilai historis dan filosofis masjid-masjid bersejarah tersebut. Mengingat masjid merupakan salah satu tempat ibadah yang selalu ada disetiap daerah beragama Islam, dengan demikian masjid-masjid tersebut mengandung nilai-nilai sejarah perkembangan Islam pada masanya yang seyogyanya dipelajari dan tidak dilupakan begitu saja. Oleh karena itu, dilakukan pendokumentasian khusus akan hal ini. Dimana dalam pembuatan aplikasi ini peneliti membuat sebuah ensiklopedia khusus mengenai masjid-masjid bersejarah yang ada di Indonesia dengan mengaplikasikan sebuah mesin pencarian yang menggabungkan konsep *semantic search* didalamnya. Dengan konsep ini aplikasi akan berusaha meningkatkan akurasi hasil pencarian dengan memahami maksudserta makna kontekstual kunci pencarian^[7], dengan demikian mempermudah pencarian dan pengguna akan mendapatkan isi dokumen yang tepat sehingga tidak harus membaca satu persatu dokumen bahkan melakukan pencarian ulang dengan kata kunci yang berbeda untuk menemukan dokumen atau informasi yang diinginkan.

Aplikasi ini akan dibuat dengan memanfaatkan teknologi *smartphone* dimana menurut situs berita teknologi (2012)^[5], prosentase jumlah penggunaan *smartphone* semakin meningkat dibandingkan perangkat dekstop. *Smartphone* akan lebih mudah dibawa kemana-mana sehingga penggunaan aplikasi inipun lebih efektif. Pemanfaatan teknologi semacam inilah yang mendukung berkembangnya aplikasi-aplikasi *smartphone* yang juga dapat dimanfaatkan sebagai media penyebaran informasi, khususnya dalam hal ini mengenai penyediaan informasi dari aplikasi mesin pencari masjid-masjid bersejarah di Indonesia, dengan demikian pengguna dapat menggunakannya kapanpun dan dimanapun membutuhkannya.

Pemanfaatan teknologi *smartphone* dengan menggabungkan *semantic search* serta *Boyer-Moore* sebagai algoritma *string matching* dalam pembuatan aplikasi mesin pencari pada ensiklopedia masjid bersejarah di Indonesia ini diharapkan memberikan kemudahan bagi pengguna serta memberikan hasil yang akurat dalam hal pencarian karena dalam konsep ini mesin pencarian didasarkan pada makna kontekstual kunci pencarian

bukan kata, sehingga hasil yang didapatkan pun lebih tepat dan akurat.

II. SEMANTIC SEARCH

Pencarian semantik merupakan pencarian suatu konten berdasarkan konteks yang tepat. Yang dimaksud konten disini adalah teks tertulis sedangkan konteks merupakan kondisi keberadaan teks yang diinginkan pengguna.

Ada dua pencarian semantik yakni pencarian semantik dengan hasil berupa navigasi (dapat berupa *link*) yang mengarah ke dokumen yang diinginkan. Sedangkan yang kedua adalah dengan mendapatkan keseluruhan dokumen yang memberikan informasi secara lengkap. Jadi pada intinya pencarian semantik ini memberikan saran bagi pengguna berdasarkan penarikan kesimpulan yang dilakukan oleh sistem berdasarkan batasan-batasan tertentu^[7].

III. ONTOLOGY

Pada mulanya ontology berasal dari ilmu filsafat yang kemudian menjadi penelitian dibidang *computer science* dan *information system*. Menurut Aristoteles, ontologi adalah “*study of existence*”, studi tentang keberadaan, yakni suatu sistem kategori atau sistem klarifikasi untuk menjelaskan keberadaan dunia nyata.

Representasi *ontology* berupa *tree* yang tidak beraturan dan terbentuk dari sekumpulan konsep yang saling terhubung. Konsep disini merupakan entitas abstrak yang tidak bergantung pada bahasa (bersifat *language-independent*) dan bukan kata-kata. Dapat dinyatakan dengan frase bahasa Inggris atau bahasa lain sebagai konversi untuk penyederhanaan, bahkan bisa direferensikan dengan angka, simbol, atau kode sekalipun.

Tujuan dibuatnya *Ontology semantic* ini adalah untuk meningkatkan otomatisasi pemrosesan teks dengan menyediakan representasi konsep yang ada di dunia secara *language independent* dan *meaning-based*^[7].

Pembuatan ontologi dapat dilakukan dengan menggunakan perangkat lunak Protégé 3.4.5 yang dimulai dari penentuan domain dan ruang lingkungannya sampai dengan penambahan *instance* atau individual disetiap kelas yang telah terbentuk. Kemudian dari *file stored* ontologi yang disimpan dengan format *.owl* diload menggunakan perangkat lunak Smore untuk pemberian metadata di setiap halaman URL agar terbentuk *triple table* yang terdiri dari subjek, predikat dan objek.

IV. PLATFORM ANDROID

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android berlisensi di bawah *GNU, General Public Lisensi Versi2 (GPLv2)*, yang memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan *term* yang sama.

Beberapa keunggulan *Platform Android* adalah sebagai berikut:

- 1) Lengkap (*Complete Platform*). *Android* menyediakan banyak *tools* dalam membangun software dan merupakan sistem operasi yang aman.
- 2) Terbuka (*Open Source Platform*). *Platform Android* disediakan melalui lisensi *open source*.
- 3) Bebas (*Free Platform*). *Android* merupakan aplikasi yang bebas untuk dikembangkan dan tidak ada lisensi atau biaya royalti dalam mengembangkannya^[6].

V. ALGORITMA BOYER-MOORE

Menurut Christian (1997)^[2], algoritma Boyer-Moore dianggap sebagai algoritma pencocokan *string* yang paling berguna dalam berbagai aplikasi. Algoritma Boyer-Moore ini melakukan pencocokan karakter dengan teks dimulai dari karakter pertama (dalam *pattern*) dari kanan ke kiri.

Pergeseran karakter dalam Algoritma Boyer-Moore ini dapat ditentukan oleh tabel pergeseran yang dibuat dengan pendekatan *Match Heuristic* (MH) dan *Occurrence Heuristic* (OH). Nilai yang dihasilkan oleh MH selanjutnya disebut dengan *good suffixshift* dalam bmGs, sedangkan untuk OH disebut dengan *bad charactershift* yang tersimpan dalam bmBc. Aturan dari *good suffix shift* ini digunakan untuk menangani kasus pencocokan yang didalamnya terdapat pengulangan karakter pada *pattern*, sedangkan *bad charactershift* digunakan untuk menghindari pengulangan perbandingan yang gagal dari suatu karakter dalam teks dengan *pattern*^[1].

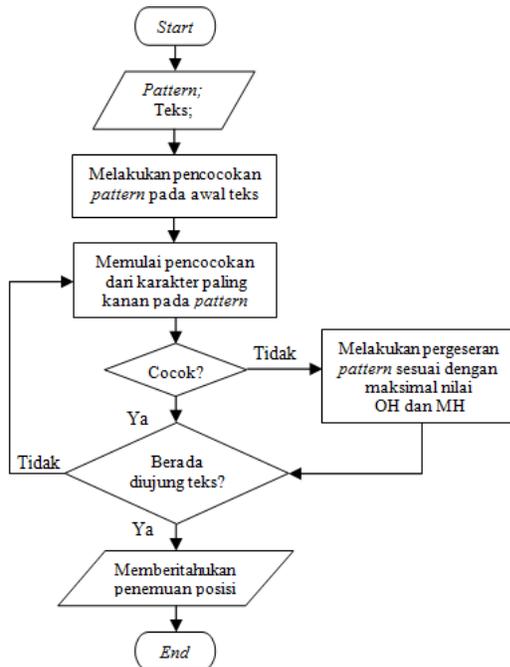
Prosedur dalam pembentukan nilai OH dalam bmBc memiliki tiga nilai penting, yakni^[5]:

1. *Pattern*, sebagai kata yang akan dicocokkan dengan teks.
2. Karakter *pattern*, sebagai karakter-karakter yang terdapat pada *pattern*.
3. *Occurrence Heuristic* (OH), sebagai nilai pergeseran yang diperoleh ketika menemukan ketidakcocokan karakter.

Sedangkan dalam prosedur pembentukan nilai MH dalam bmGs memiliki enam nilai penting, meliputi^[5]:

1. *Pattern*, merupakan kata yang akan dicocokkan dengan teks.
2. *Match Heuristic* (MH), merupakan nilai pergeseran yang diperoleh ketika menemukan kecocokan *suffix*.
3. *Compare*, merupakan sejumlah karakter sebelah kanan dari sebuah karakter *pattern* yang diperoleh dari pergeseran kanan ke kiri.
4. *Prefix*, merupakan awalan dari *pattern* yang diperoleh dari pergeseran dari kiri ke kanan.
5. *Suffix*, merupakan akhiran sebelah kanan *prefix*.
6. Pergeseran, merupakan nilai yang didapat ketika melakukan pergeseran dari *compare*.

Gambar 1 adalah *flowchart* proses pencocokan Algoritma Boyer-Moore secara umum.

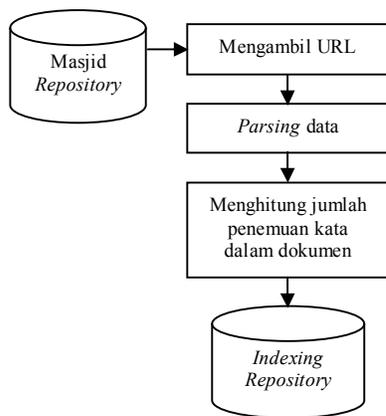


Gambar 1. Flowchart Algoritma Boyer-Moore

VI. METODE PENELITIAN

A. Perancangan Sistem

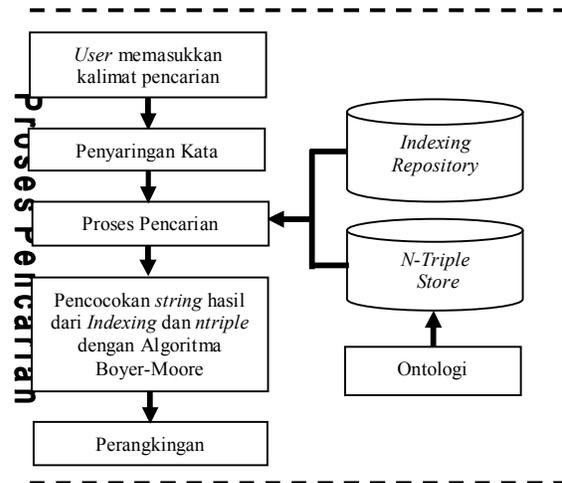
Tahapan dalam pembuatan aplikasi ini yakni tahap *preprocessing* dan tahap pencarian. Pada tahap *preprocessing* terdapat proses *indexing*, dimana dalam proses *indexing* (gambar 2) dilakukan *parsing data* URL dari *repository* masjid bersejarah berdasarkan judul dan *content*, kemudian menghitung jumlah penemuan kata pada judul dan *content* dokumen dengan daftar kata penting dalam domain masjid bersejarah. Setelah dilakukan pencocokan kata pada masing-masing URL, hasilnya disimpan ke dalam *indexing repository*.



Gambar 2. Alur Proses *Indexing*

Tahap kedua yakni proses pencarian, ketika *user* memasukkan kalimat pencarian maka aplikasi akan menyaring kalimat masukan yang selanjutnya dilakukan proses pencarian terhadap *indexing repository* dan *n-triple store*. *N-triple store* ini didapatkan

dari pembentukan *n-triple* melalui ontologi ke RDF. Selanjutnya dari kedua hasil pencarian tersebut dilakukan pencocokan *string* dengan menggunakan Algoritma Boyer-Moore, kemudian sistem melakukan perangkingan dan memunculkannya sebagai hasil. Gambaran umum proses ini dapat dilihat pada gambar 3.



Gambar 3. Diagram Blok Tahap Proses Pencarian

B. Perancangan Ontologi

Tahapan dalam perancangan ontologi aplikasi masjid bersejarah ini adalah:

- Tahap 1 : Menentukan domain dan ruang lingkup ontologi
- Tahap 2 : Mendefinisikan kelas dan tingkatan kelas
- Tahap 3 : Mendefinisikan properti kelas dan faset dari *slot*
- Tahap 4 : Membuat *instance*

Menentukan domain dan ruang lingkup ontologi

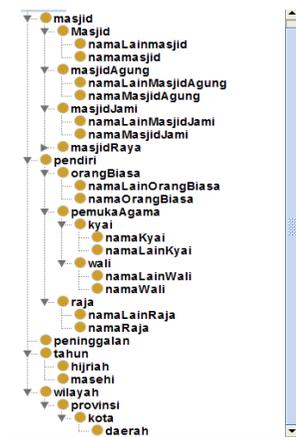
Domain dalam pembuatan aplikasi ini adalah masjid bersejarah dengan ruang lingkup ontologi sebagai berikut:

- *Class* masjid (memiliki *subclass* masjid, *subclass* masjid agung, *subclass* masjid jami, *subclass* masjid raya (dengan masing-masing *subclass* memiliki *sibling class* nama masjid dan nama lain masjid)),
- *Class* pendiri (memiliki *subclass* orang biasa (memiliki *sibling class* nama dan nama lain), *subclass* pemuka agama (memiliki *sibling class* kyai (memiliki *sibling class* nama dan nama lain kyai), *sibling class* wali (memiliki *sibling class* nama dan nama lain wali)), *subclass* raja (memiliki *sibling class* nama raja dan nama lain raja)),
- *Class* peninggalan,
- *Class* tahun (memiliki *subclass* hijriah, dan *subclass* masehi),
- *Class* wilayah (memiliki *subclass* provinsi (memiliki *sibling class* kota (memiliki *sibling class* daerah))).

Mendefinisikan kelas dan tingkatan kelas

Tahap ini dilakukan dengan mengelompokkan kata ke dalam tingkatan kelas berdasarkan kesamaan karakteristik. Terdapat tiga cara dalam pembentukan tingkatan kelas, yakni: melalui proses pengembangan *top-down*, *bottom-up*, atau melalui kombinasi antar kedua proses. Pada penelitian ini, proses pengembangan dilakukan dengan cara pengembangan *top-down* dimana pendefinisian tingkatan kelas dimulai dari yang paling umum sampai ke yang lebih khusus. Dan tiap-tiap *class* yang akan dibuat akan menjadi *subclass* dari *class* "THING".

Perancangan *class* dan *subclass* tersebut dapat dilakukan dengan menggunakan perangkat lunak Protégé 3.4.5 seperti yang terlihat pada gambar 4.

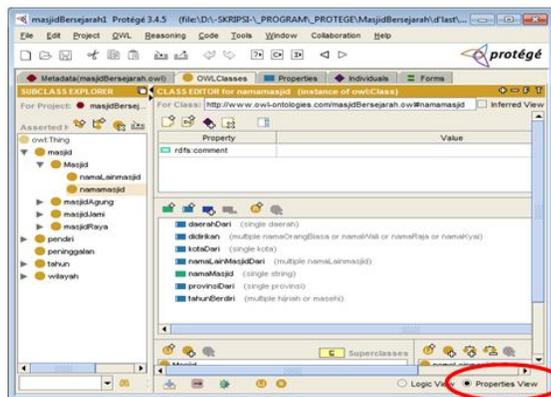


Gambar 4. Rancangan Pendefinisian *Class* dan Tingkatan *Class*

Mendefinisikan properti kelas dan faset dari slot

Properti atau bisa disebut dengan *slot* merupakan hubungan antar objek. Properti bisa berupa kata kerja atau kata sifat. Sedangkan faset merupakan informasi tentang properti berupa nilai yang dimiliki oleh properti atau tipe nilai sebuah properti, seperti *string*, *number*, *boolean*, dan lain-lain.

Rancangan properti dan *faset* dari slot dapat dibuat dengan menggunakan perangkat lunak Protégé 3.4.5. Pada tiap-tiap *subclass* yang telah dibuat ditambahkan *object* dan *datatype property* pada “Properties View” dalam tab menu “OWL Classes”. Contoh hasil pembuatan properti pada *subclass* namaMasjid *class* Masjid dapat dilihat pada gambar 5.

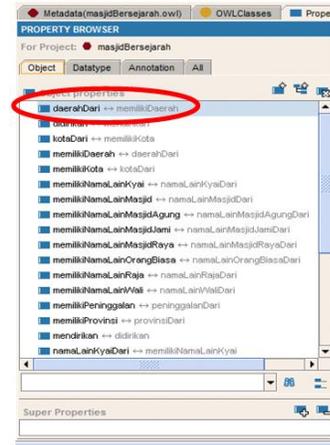


Gambar 5. Properti-Properti yang Terbentuk pada *Subclass* nama Masjid

Selanjutnya masing-masing *object* dan *datatype property* yang telah dibuat secara keseluruhan akan otomatis terisikan pada “Object” dan “Datatype” pada tab menu “Properties” yang selanjutnya dilakukan pendefinisian fungsi *Inverse property* pada masing-masing *object property* yang telah terbentuk. *Inverse property* ini bertujuan untuk memberikan nilai dari suatu properti secara otomatis jika *inverse property* diberi suatu nilai.

Sebagai contoh pada rancangan properti diatas, terdapat fungsi *inverse* pada *class* namaMasjid yakni pada properti “daerahDari”

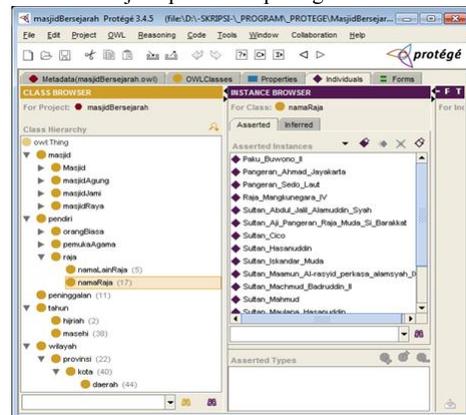
yang memiliki *inverse* “memilikiDaerah” pada *class* Daerah (gambar 6).



Gambar 6. Pendefinisian Fungsi *Inverse* pada Tiap *Object Property*

Membuat instance

Instance merupakan objek maupun individual dari tiap-tiap kelas yang telah terbentuk sebelumnya. Pembentukan *instance* dapat dibuat melalui tab menu “Individuals” dalam Protégé. Contoh hasil pembuatan *instance* dari kelas pendiri dengan *subclass* raja dan *sibling class* nama raja dapat dilihat pada gambar 7.



Gambar 7. Individual-Individual yang Terdapat pada namaRaja

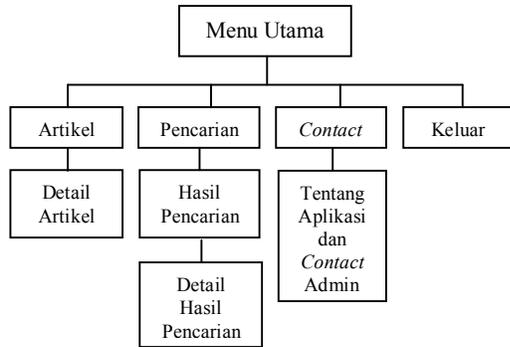
C. Pembuatan Metadata

Penambahan metadata pada ontologi yang telah dibuat sebelumnya ini dipergunakan untuk pembuatan *triple table* yang terdiri dari SPO (Subjek, Predikat, Objek) pada masing-masing halaman URL aplikasi yang telah dibuat. Penambahan metadata ini bisa menggunakan perangkat lunak Smore, dengan *meload* ontologi yang telah dibuat sebelumnya. Pada masing-masing halaman URL aplikasi dibuatlah individual baru pada menu “New Individuals” dengan memberikan name ID sesuai dengan nama .html nya, kemudian masing-masing diisikan objek propertis dan data propertisnya. Contoh pembuatan metadata untuk penambahan *object properties* dan *data properties* pada halaman URL Masjid Wapauwe terlihat pada gambar 8 dan gambar 9.

Setelah didapatkan nilai-nilai OH dan MH pada masing-masing karakter, selanjutnya memasuki tahap pencocokan *string* dalam Algoritma Boyer-Moore. Pada tahapan ini, algoritma akan melakukan pengecekan *pattern* dari awal teks di mulai dari karakter paling kanan pada *pattern*. Apabila menemukan kesamaan *pattern*, maka algoritma akan memberitahukan penemuan posisi. Namun, jika terdapat perbedaan karakter, algoritma akan melakukan pergeseran *pattern* sesuai dengan nilai pergeseran maksimal dari OH karakter teks dengan MH karakter *pattern*.

E. Desain Menu Aplikasi Mobile

Menu yang disediakan dalam aplikasi *mobile* yang dapat diakses oleh *user* yakni: menu artikel, menu pencarian, menu *contact* dan menu keluar dengan skema desain pada gambar 12.



Gambar 12. Skema Desain Aplikasi

F. Parameter Uji Coba

Uji coba dilakukan untuk mengetahui kinerja aplikasi yang telah dibuat, dengan parameter pengujian sebagai berikut:

- Yang diujikan dalam aplikasi ini adalah mengenai keakuratan aplikasi dalam menampilkan data hasil pencarian, dengan menggunakan aplikasi wordpress sebagai aplikasi pembanding.
- Jaringan internet yang digunakan adalah jaringan WiFi yang disediakan oleh Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang Jurusan Teknik Informatika dengan spesifikasi *download speed* sebesar 0,24 Mbps dan *upload speed* sebesar 0,12 Mbps (sumber: <http://www.speedtest.net>).

VII. PEMBAHASAN

A. Implementasi Indexing

Indexing dilakukan dengan menggunakan bantuan *function* php 'simple_html_dom.php'. Dengan menggunakan *function* ini aplikasi *indexing* yang dibuat dapat membedakan dokumen judul dan *content* pada masing-masing tag yang terdapat pada URL dalam masjid *repository*, sehingga mempermudah jumlah penemuan kata pada masing-masing tag dengan daftar kata penting yang telah dibuat pada tabel 'word'. Hasil dari proses *indexing* ini seperti yang ditunjukkan pada gambar 13.

word	url	count_title	count_content
raya	http://embi.web-semantik.com/masjid-raya-baiturrah...	1	3
redeb	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
redep	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
rembang	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
riau	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
rumah	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	1
salingnata	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
sayyid	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
secang	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
sedo	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
segar	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0
cabalmenek	http://embi.web-semantik.com/masjid-raya-baiturrah...	0	0

Gambar 13. Indexing Repository

B. Penabelan SPO Hasil Metadata Ontologi

Berdasarkan ontologi yang telah dibuat sebelumnya dengan menggunakan perangkat lunak Protégé yang tersimpan dalam format *.owl*, dilakukannya pemberian metadata pada masing-masing halaman url aplikasi menggunakan perangkat lunak Smore dengan *meload* hasil ontologi tersebut. Kemudian setelah pemberian metadata masing-masing halaman url dilakukan maka terbentuklah *triple table* seperti yang terlihat pada gambar 10 dan hasilnya disimpan dalam format *.owl* (gambar 14).

```

1 <?xml version="1.0"?>
2 <!DOCTYPE owl [
3 <ENTITY owl "https://www.w3.org/2002/07/owl#" />
4 <ENTITY rdfs "https://www.w3.org/2001/XMLSchema#" />
5 <ENTITY rdf "https://www.w3.org/1999/02/22-rdf-syntax-ns#" />
6 <ENTITY rdfs "https://www.w3.org/2000/01/rdf-schema#" />
7 ]>
8 <rdf:RDF
9 xmlns:og="https://www.w3.org/2001/07/owl#"
10 xmlns:rdfs="https://www.w3.org/2001/XMLSchema#"
11 xmlns:masjidBersejarah="https://www.w3.org/2002/07/owl#"
12 xmlns:rdfs="https://www.w3.org/1999/02/22-rdf-syntax-ns#"
13 xmlns:owl="https://www.w3.org/2002/07/owl#"
14 xmlns:rdfs="https://www.w3.org/2000/01/rdf-schema#"
15 >
16 >
17 <owl:Ontology rdf:about="http://www.owl-ontologies.com/masjidBersejarah.owl">
18 </owl:Ontology>
19 <owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing">
20 </owl:Class>
21 <owl:ObjectProperty rdf:about="http://www.owl-ontologies.com/masjidBersejarah.owl#dasarBani">
22 </owl:ObjectProperty>
23 <owl:ObjectProperty rdf:about="http://www.owl-ontologies.com/masjidBersejarah.owl#ditiyakan">
24 </owl:ObjectProperty>
25 </owl:Ontology>
  
```

Gambar 14. Hasil Metadata dengan Menggunakan Smore

Triple table inilah (gambar 10) merupakan SPO (Subjek, Prediket, Objek) yang akan dilakukan pembacaan dan penabelan kedalam database MySQL tabel 'ntriple' melalui program berbasis Java dengan bantuan *library* API Jena.

Gambar 15 merupakan *interface* dari proses diatas, dengan mengisikan nama domain masjid bersejarah pada *textfield* yang disediakan dan *meload* dokumen masjid bersejarah *.owl* hasil Smore, maka sistem akan otomatis menabelkan subjek, prediket, objek, beserta link (URL) masing-masing data.


```

$CompareSama = array_search($SuffixCompare, $data_suffix);
if($CompareSama != null){
for($i=$$1; $i>0; $i--){
$goodSuffix[$SuffixCompare[$i]] = $Pergeseran;
$sm_gs[$ig] = $goodSuffix[$SuffixCompare[$i]];
$ig++;
}else if($i==$e-1){
$goodSuffix[$i] = 1;
$sm_gs[$ig] = $goodSuffix[$i];
$ig++;
}else{
$goodSuffix[$i] = $a;
}
}

```

Gambar 20: Potongan Source Code untuk Pemberian Nilai MH pada Karakter

Setelah mendapatkan nilai OH dan MH, tahap selanjutnya adalah memasuki proses pencocokan string. Dalam prosesnya, Algoritma Boyer-Moore melakukan pengecekan dari awal teks dimulai dari karakter paling kanan pada *pattern*. Selanjutnya melakukan pergeseran *pattern* sesuai dengan maksimal nilai OH karakter teks dan nilai MH karakter *pattern*.

D. Implementasi Interface

1) Implementasi Interface untuk User

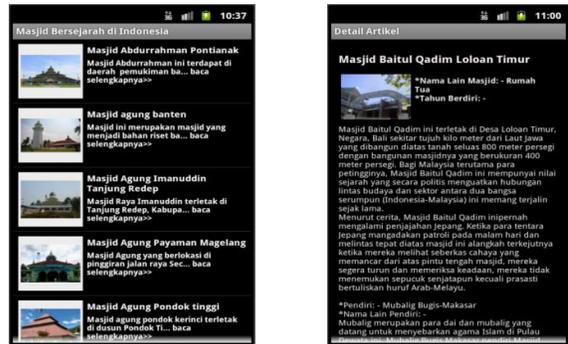
Interface menu awal dari aplikasi mobile yang telah dibuat ini dapat dilihat pada gambar 21.



Gambar 21: Menu Awal Aplikasi

Berikut penjelasan setiap aksi menu pada gambar 21:

1. Menu artikel, menu ini menyediakan halaman *list view* untuk *user* yang berisi artikel-artikel mengenai masjid-masjid bersejarah di Indonesia dan detail artikel (gambar 22).
2. Menu pencarian, merupakan menu yang menyediakan mesin pencarian bagi *user* (gambar 23 sebelah kiri). Disini *user* dapat melakukan pencarian artikel dengan memasukkan kalimat pencarian yang selanjutnya diproses oleh sistem untuk menampilkan hasil.
3. Menu *Contact*, menu ini menyediakan halaman tentang aplikasi dan *contact administrator* seperti yang terlihat pada gambar 23 sebelah kanan.
4. Menu keluar, menu ini menyediakan untuk akses keluar aplikasi. Ketika menu ini dipilih maka akan tampil kotak dialog untuk keluar aplikasi atau tidak.



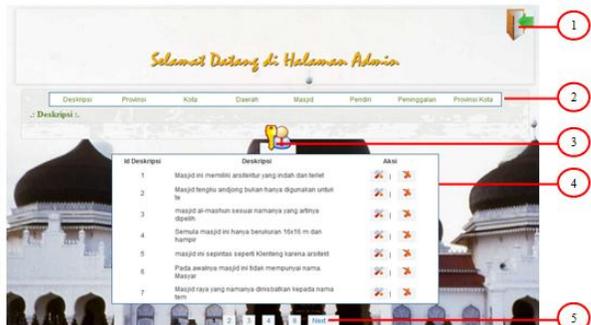
Gambar 22: Menu Artikel dan Detail Artikel



Gambar 23: Menu Pencarian dan Menu Contact

2) Implementasi Interface untuk Administrator

Adapun implementasi *interface* untuk admin ditunjukkan pada gambar 24.



Gambar 24: Halaman Admin

Berikut penjelasan masing-masing menu yang tersedia pada halaman admin dalam gambar 24 diatas:

1. Menu keluar, menu ini menyediakan akses *logout* halaman pengolahan data.
2. Menu Data, menu-menu ini terdiri dari menu deskripsi, provinsi, kota, daerah, masjid, pendiri, peninggalan dan provinsi kota yang masing-masing menu tersebut digunakan untuk akses pengolahan data-data masjid bersejarah berdasarkan menu data yang dipilih.
3. Menu tambah data, merupakan menu yang menyediakan akses *insert* data bagi admin.

- Tabel data, menyediakan tabel data-data masjid bersejarah sesuai dengan menu data yang dipilih. Pada kolom 'aksi' menyediakan akses *update* datadan *delete* data bagi admin.
- Paging*, merupakan komponen menu yang digunakan untuk melihat tabel data berikutnya maupun sebelumnya.

VIII. UJI COBA APLIKASI

Pengujian dalam aplikasi 'EMBI' mesin pencari berbasis *semantic search* pada masjid bersejarah di Indonesia ini dilakukan dengan melihat keakuratan sistem mendapatkan data hasil pencarian dan membandingkannya dengan aplikasi pencarian masjid bersejarah lainnya. Peneliti menggunakan wordpress sebagai aplikasi pembandingan.

A. Perbandingan Hasil Penemuan Data

Jumlah data hasil pencarian antara kedua aplikasi tersebut beserta keterangan jumlah kesalahan dokumen yang ditampilkan dapat dilihat pada tabel 1.

TABEL 1: TABEL PERBANDINGAN JUMLAH HASIL PENEMUAN DATA

No	Kalimat Pencarian	Pencarian EMBI			Pencarian Wordpress		
		Jumlah penemuan data	Hasil	Ket.	Jumlah penemuan data	Hasil	Ket.
1	Masjid bersejarah di Surabaya	1	Masjid Agung Sunan Ampel Surabaya	Tepat	-	-	Tidak ada hasil
2	Masjid di Jawa Timur	2	Masjid Sunan Giri Gresik, Masjid Agung Sunan Ampel Surabaya	Tepat	4	Masjid Besar Kauman, Masjid Baitul Qadim Loloan Timur, Masjid Agung Sunan Ampel Surabaya, Masjid Agung Pondok tinggi	3 salah
3	Masjid dengan pendiri Sultan Qasim	1	Masjid Sultan Siak	Tepat	1	Masjid Sultan Siak	Tepat
4	Peninggalan sumur di Jawa Timur	1	Masjid Agung Sunan Ampel Surabaya	Tepat	1	Masjid Agung Sunan Ampel Surabaya	Tepat
5	Masjid bersejarah di Surabaya yang memiliki peninggalan sumur	1	Masjid Agung Sunan Ampel Surabaya	Tepat	-	-	Tidak ada hasil
6	Masjid yang didirikan oleh Raden Paku	1	Masjid Sunan Giri Gresik	Tepat	-	-	Tidak ada hasil
7	Masjid Air Mata	1	Masjid Air Mata Kupang	Tepat	3	Masjid Besar Kauman, Masjid Air Mata Kupang, Masjid Agung Sunan Ampel Surabaya	2 salah
8	Masjid bersejarah yang didirikan pada tahun 1991	1	Masjid Jami Kudus	Tepat	-	-	Tidak ada hasil
9	Masjid bersejarah yang ada di provinsi Kalimantan Barat	3	Masjid Jamiatul Khair Kalimantan Barat, Masjid Sintang Kapuas Hilir, Masjid Abdurrahman Pontianak	Tepat	-	-	Tidak ada hasil
10	Masjid Baiturrahman	1	Masjid Raya Baiturrahman	Tepat	1	Masjid Raya Baiturrahman	Tepat
11	Masjid di Bukit Tinggi	1	Masjid Syuhada Pasar Usang	Tepat	2	Masjid Syuhada Pasar Usang, Masjid Sunan Giri Gresik	1 salah
12	Masjid bersejarah yang memiliki peninggalan prasasti	1	Masjid Baitul Qadim Loloan Timur	Tepat	-	-	Tidak ada hasil

13	Masjid yang memiliki peninggalan mimbar	5	Masjid Sultan Suriansyah, Masjid Agung Sultan Mahmud Badaruddin, Masjid jami-anawier, Masjid Jami Rambang, Masjid Besar Al-mahmudiyah	Tepat	5	Masjid Kampung Bugis Denpasar, Masjid Jamik, Masjid jami Al-atiq, Masjid Besar Al-mahmudiyah, Masjid Agung Surakarta	4 salah
14	Tiang ukiran berbentuk alif yang terbuat dari kayu kanjoli	1	Masjid Wapauwe Desa Ketetu	Tepat	1	Masjid Wapauwe Desa Ketetu	Tepat
15	Masjid bersejarah yang berada di kota Solo	3	Masjid Laweyan Solo, Masjid Agung Surakarta, Masjid Pura Mangkunegaran	Tepat	-	-	Tidak ada hasil
16	Kota Solo	3	Masjid Laweyan Solo, Masjid Agung Surakarta, Masjid Pura Mangkunegaran	Tepat	3	Masjid Pura Mangkunegaran, Masjid Laweyan Solo, Masjid Agung Surakarta	Tepat
17	Sumatera Barat	5	Masjid Nurul Falah, Masjid raya Syekh Burhanuddin, Masjid Agung Sultan Mahmud Badaruddin, Masjid Syuhada Pasar Usang, Masjid Raya Nagari Kota Nan Empat	1 salah	2	Masjid raya Syekh Burhanuddin, Masjid Nurul Falah	Tepat, hasil kurang
18	Peninggalan kitab	2	Masjid Wapauwe Desa Ketetu, Masjid Raya Sultan Ternate	Tepat	4	Masjid Wapauwe Desa Ketetu, Masjid Raya Sultan Ternate, Masjid Raya Nagari Kota Nan Empat, Masjid Baitul Qadim Loloan Timur	2 salah
19	Masjid Bukit Tinggi	1	Masjid Syuhada Pasar Usang	Tepat	2	Masjid Syuhada Pasar Usang, Masjid Sunan Giri Gresik	1 salah
20	Masjid di daerah Tanjung Redep	1	Masjid Agung Imanuddin Tanjung Redep	Tepat	-	-	Tidak ada hasil
21	Desa Bayan	1	Masjid Kuno Desa Bayan	Tepat	1	Masjid Kuno Desa Bayan	Tepat
22	Masjid bersejarah di provinsi Nusa Tenggara Timur	1	Masjid Air Mata Kupang	Tepat	-	-	Tidak ada hasil
23	Masjid yang didirikan oleh PB II	1	Masjid Agung Surakarta	Tepat	1	Masjid Agung Surakarta	Tepat
24	Masjid bersejarah yang didirikan pada abad 16	1	Masjid jami Al-atiq	Tepat	-	-	Tidak ada hasil
25	Masjid bersejarah yang ada di Jawa Tengah	7	Masjid Jami Rambang, Masjid Agung Payaman Magelang, Masjid Agung Surakarta, Masjid Jami Kudus, Masjid	Tepat	2	Masjid Jami Rambang, Masjid al-Falah	1 salah, hasil kurang

Berdasarkan hasil penemuan data antara kedua aplikasi pencarian masjid bersejarah pada tabel1 diatas dapat dilihat bahwa pencarian menggunakan aplikasi EMBI didapatkan hasil yang tepat dan lebih akurat dibandingkan dengan pencarian pada wordpress. Dari data pengujian didapatkan beberapa kesalahan penampilan data pada pencarian wordpress serta beberapa dokumen tidak dapat ditemukan, hal tersebut dikarenakan pencarian wordpress menampilkan hasil dari tiap kata yang sama dan terkandung dalam dokumen. Berbeda dengan pencarian pada aplikasi EMBI yang menampilkan hasil melalui proses pencariannya yang melalui tahap *pengindexan* sampai pencarian ke metadatanya. Kesalahan terjadi dikarenakan aplikasi ini masih menggunakan *query* pencarian yang hanya mengacu pada objek.

IX. KESIMPULAN

Aplikasi EMBI atau yang juga merupakan aplikasi mesin pencari berbasis *semantic search* dengan menggunakan Algoritma

Boyer-Moore sebagai algoritma *string matching* pada ensiklopedia masjid bersejarah ini mampu memberikan kemudahan bagi *user* dalam menemukan dokumen berdasarkan konteks kalimat pencarian yang dicari. Berdasarkan pengujian dengan memasukkan beberapa kalimat pencarian didapatkan hasil pencarian yang tepat dan akurat sesuai dengan konteks kalimat pencarian. Dari segi ontologi, apabila semakin banyak tingkatan ontologi yang digunakan maka hasil pencarian semakin akurat dan tingkat pencarianpun semakin kompleks.

Berdasarkan prosesnya, aplikasi ini melakukan penyaringan kata penting sesuai dengan ontologi yang digunakan, yakni: data nama masjid, nama lain masjid, nama pendiri, nama lain pendiri, wilayah, tahun berdiri dan peninggalan, sehingga dapat dikembangkan lagi dengan menambahkan kata yang dianggap penting dalam masing-masing kandungan deskripsi data masjid bersejarah agar didapatkan pencarian yang lebih kompleks. Sedangkan pada saat pencarian ke *repository ntriple, query* hanya mengacu pada objek sehingga dapat dilakukan pengembangan dengan mengkombinasikan ketiganya, yakni: subjek, predikat, dan objek dengan melibatkan operator *logic* sehingga didapatkan data yang tepat dengan berbagai macam kalimat pencarian.

DAFTAR PUSTAKA

- [1] Aulia, Rama. 2008. *Analisis Algoritma Knuth Morris Pratt dan Algoritma Boyer Moore dalam Proses Pencarian String*. Bandung: ITB
- [2] Charras, Christian. 1997. *Boyer-Moore Algorithm*. Available online Accessed: <http://www-igm.univ-mlv.fr/~lecroq/string/node14.html> (diakses tanggal 7 Januari 2013)
- [3] <http://www.beritateknologi.com/tahun-2012-10-persen-pengguna-internet-mengakses-dari-perangkat-mobile/> (diakses tanggal 17 Agustus 2013)
- [4] <http://www.world-insights.com/the-growing-world-of-islam/> (diakses tanggal 17 Agustus 2013)
- [5] Kurnaedi, Andri. 2011. *Penerapan String Matching Menggunakan Algoritma Boyer-Moore pada Translator Bahasa Pascal ke C*. Bandung: Unikom
- [6] Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika
- [7] Sarno, Riyanarto dkk. 2012. *Semantic Search-Pencarian Berdasarkan Konten*. Yogyakarta: ANDI