

Pengembangan Sistem Pembangkit Metadata Citra untuk Citra pada Situs Olah Raga

Umi Laili Yuhana¹, Dian Sahputra², Diana Purwitasari¹, Daniel Oranova Siahaan¹,

¹Jurusan Teknik Informatika, ²Alumni Jurusan Teknik Informatika

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia, 60111

yuhana@if.its.ac.id, dian_sah@cs.its.ac.id, diana@if.its.ac.id, daniel@if.its.ac.id

Abstrak

Perkembangan jumlah informasi dalam bentuk citra sangatlah pesat. Hal tersebut akan menimbulkan masalah dalam proses pencarian citra berbasis metadata. Masalah yang muncul adalah proses pembubuhan metadata dari citra. Jumlah citra yang sangat banyak tentu akan membutuhkan usaha yang besar dalam proses pemberian metadata citra. Karenanya dibutuhkan sebuah pembangkit metadata otomatis yang dapat memberikan metadata pada setiap citra.

Makalah ini membahas pengembangan Sistem Pembangkit Metadata Citra (PMC) yang dapat digunakan untuk membangkitkan metadata dari sebuah citra pada situs olah raga dengan memanfaatkan informasi yang dibawanya. Informasi yang dimaksud adalah informasi berupa konten berita dari citra. Informasi tersebut diharapkan dapat menjadi kata kunci dari metadata citra. Pemanfaatan teknologi Natural Language Processing (pemrosesan bahasa alami) digunakan untuk melakukan proses pengambilan kata kunci dari konten berita. Setiap citra yang telah diproses diharapkan memiliki metadata yang digunakan sebagai acuan dalam proses pencarian citra berbasis metadata.

Berdasarkan proses uji coba yang dilakukan, hasil yang diperoleh adalah bahwa sistem dapat membangkitkan metadata citra sesuai dengan yang diinginkan dengan relevansi rata-rata sebesar 59,77% dibandingkan dengan pendapat domain ekspert. Metadata yang dibangkitkan dapat ditemukan dengan proses pencarian citra berbasis semantik yang juga dibangun dalam penelitian ini.

Keywords— Semantic Web, Metadata, RDF, pencarian citra

I. LATAR BELAKANG

Perkembangan jumlah informasi dalam bentuk citra sangatlah pesat. Hal tersebut akan menimbulkan masalah dalam proses pencarian citra berbasis metadata. Masalah yang muncul adalah proses pembubuhan metadata dari citra. Jumlah citra yang sangat banyak tentu akan

membutuhkan usaha yang besar dalam proses pemberian metadata citra.

Pemberian metadata pada citra dapat didasarkan pada judul gambar, pembuat gambar, dan isi dari gambar (pihak yang terlibat pada gambar atau kegiatan yang ada didalamnya). Namun hal ini tentu bertolak belakang dengan jumlah citra yang banyak di dalam internet. Pemberian metadata pada setiap citra tentu akan membutuhkan usaha yang besar. Sehingga dibutuhkan sebuah pembangkit metadata otomatis yang dapat memberikan metadata pada setiap citra.

Peneliti merancang sistem pembangkit metadata citra yang akan digunakan untuk menandai citra secara otomatis. Metadata citra dapat digunakan untuk bahan uji coba dari aplikasi pencarian citra berbasis semantik yang dibangun juga pada penelitian ini. Adapun susunan pada makalah ini yaitu sebagai berikut. Bagian 2 akan membahas mengenai teori-teori penunjang dalam perancangan sistem dan pembuatan sistem, bagian 3 membahas mengenai perancangan dan implementasi sistem pembangkitan metadata citra. Sedangkan pada bagian 4 dan 5 akan membahas mengenai uji coba dan kesimpulan secara berturutan.

II. TEORI PENUNJANG

Pada bagian ini, penulis akan membahas teori-teori dan hal-hal yang berkaitan dengan pembangunan sistem pembangkit metadata citra.

2.1. Metadata

Metadata merupakan data mengenai data, atau informasi yang dimiliki oleh suatu data. Metadata digunakan untuk menggambarkan aspek-aspek dokumen digital dan data yang terdiri dari definisi data, struktur data dan administrasi data. Metadata merupakan informasi di web yang dapat dipahami oleh mesin. Dengan mendeskripsikan konten dan konteks suatu data, kualitas dari keaslian data akan semakin meningkat. Metadata didefinisikan sebagai data yang menyediakan informasi tentang satu atau lebih potongan data seperti arti suatu data, tujuan

adanya data, waktu dan tanggal data dibuat, pembuat data, tempat atau lokasi dimana data dibuat serta lokasi data disimpan[1]. Metadata citra dapat dipilah menjadi dua bagian yaitu metadata yang menyimpan karakteristik citra itu sendiri seperti berapa besar suatu citra, kedalaman warna, resolusinya, kapan citra dibuat, dan data lain. Bagian yang lain berisi tentang informasi dari konteks data seperti siapa saja yang ada dalam citra tersebut, dimana lokasi citra diambil, apa yang dilakukan oleh objek yang ada di dalam citra, dan informasi kontekstual lainnya.

Metadata dapat dibuat secara otomatis maupun dibuat secara manual. Metadata dasar yang diambil oleh komputer meliputi informasi tentang kapan data dibuat, siapa yang membuat, kapan terakhir diperbarui, ukuran file dan ekstensinya. Untuk membangun metadata secara otomatis, terlebih dahulu harus diidentifikasi kebutuhan atau data apa yang akan disimpan dalam sebuah metadata. Selanjutnya perlu diidentifikasi proses apa yang harus dilakukan untuk menghasilkan metadata yang dimaksud.

2.2. RDF

RDF – (*Resource Description Framework*) adalah suatu bahasa untuk merepresentasikan informasi tentang *resources* (sumber daya) dalam *World Wide Web*. RDF pada dasarnya diperuntukkan secara khusus untuk merepresentasikan metadata dari sumber daya web, seperti judul, pengarang, dan tanggal modifikasi dari suatu halaman web, hak cipta dan informasi lisensi dari dokumen web.

RDF muncul berdasarkan pada ide tentang mengidentifikasi sesuatu menggunakan *web identifiers* (dikenal sebagai URI – *Uniform Resource Identifiers*) dan ide tentang mendeskripsikan sumber daya berdasarkan karakteristik dan nilainya [2]. RDF merepresentasikan sebuah pernyataan sederhana tentang sumber daya dalam bentuk sebuah graf beserta simpul dan panahnya. Metadata yang akan dihasilkan oleh pembangkit metadata citra ini diberupakan dalam bentuk RDF.

2.3. Really Simple Syndication (RSS)

RSS merupakan sebuah file berformat XML yang memberikan sebuah informasi perubahan web secara regular. RSS merupakan file berformat XML untuk sindikasi yang telah digunakan (diantaranya dan kebanyakan) situs web berita dan weblog. Singkatan ini biasanya mengarah ke beberapa protokol:

- Rich Site Summary (RSS 0.91)
- RDF Site Summary (RSS 0.9 and 1.0)
- Really Simple Syndication (RSS 2.0)

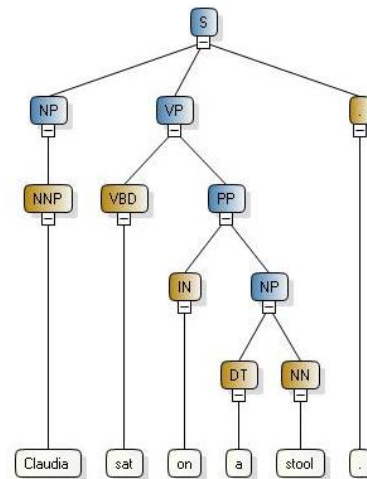
RSS memungkinkan pengguna dari suatu web mengetahui secara langsung perubahan di suatu

web. Sehingga pengguna akan selalu memperoleh informasi artikel atau konten terbaru dari sebuah web. Di dalam penelitian ini, peneliti mengambil citra dari suatu situs tertentu dan untuk mengetahui perubahan informasi yang ditampilkan dalam situs tersebut, peneliti memanfaatkan RSS. Informasi yang diperoleh dari RSS ini yang nantinya digunakan untuk membangkitkan sebuah metadata dari citra yang terdapat pada konten web yang dituju.

2.4. Treebank Parser

Untuk memproses bahasa alami yang ada dalam dokumen yang diproses, peneliti memanfaatkan fungsi *English Treebank*. *English Treebank* merupakan fungsi yang disediakan oleh OpenNLP untuk melakukan *parsing* secara penuh. Fungsi ini dapat memisahkan frasa-frasa yang ada dalam stuktur di sebuah kalimat Bahasa Inggris.

Keluaran dari *English Treebank* direpresentasikan sebagai sebuah tree. Sebagai contoh apabila dilakukan proses *English Treebank* dalam kalimat “*Claudia sat on a stool.*” Maka akan menghasilkan tree seperti pada Gambar 1. [3]



Gambar 1. Tree Hasil English Treebank

Implementasi fungsi *English Treebank* yang digunakan adalah mengambil struktur kalimat yang merupakan *Noun Phrase* dan *Verb Phrase*. *Noun Phrase* (NP) akan diproyeksikan sebagai instance dari kelas subyek, sementara Verb yang ada di dalam *Verb Phrase* (VP) diproyeksikan menjadi als (predikat). (S=sentence, NP=noun phrase, VP=verb phrase, PP=prepositional phrase, N=noun, V=verb, AT=article, P=preposition)[3]. Dalam Gambar 1 terlihat bahwa “Claudia” terdeteksi sebagai subyek dan “sat on a stool” sebagai predikat. Tanda titik (.) menandai berakhirnya satu kalimat.

2.5. NameFinder

NameFinder merupakan fungsi dari pustaka OpenNLP yang dapat melakukan identifikasi dari sebuah entitas dalam kalimat. Entitas akan diidentifikasi termasuk dalam kelas yang sesuai dengan model yang ditawarkan OpenNLP. Model yang disediakan oleh pustaka OpenNLP antara lain : *date*, *location*, *money*, *organization*, *percentage*, *person*, dan *time*.

Implementasi dari *NameFinder* yang digunakan adalah untuk mencari kata yang termasuk dalam *event*, *location* dan *time*. Namun karena OpenNLP tidak memiliki *Maximum Models* untuk *event* maka akan dilakukan pelatihan pada model *event*.

Pelatihan model baru diperlukan apabila ingin mendapatkan entitas pada kelas yang tidak termasuk pada model yang dimiliki OpenNLP. Untuk melakukan pelatihan diperlukan *corpus* kalimat bahasa inggris yang diberi *tag* khusus. Gambar 2 menunjukkan contoh file yang digunakan sebagai data pelatihan:

```
Spain midfielder Xabi Alonso has called Nigel de Jong's 'Kung Fu' kick on him in Sunday's <START>World Cup<END> final "one of the worst tackles" he has received.
```

Gambar 2. Contoh kalimat yang digunakan sebagai data pelatihan

Pada gambar 3 terdapat script yang digunakan untuk pelatihan model baru:

```
opennlp.tools.namefind.NameFinderME
your_org_training_file
organization.bin.gz
```

Gambar 3. Script yang digunakan untuk melakukan pelatihan model baru

2.6. Jena

Jena[4] merupakan pustaka yang sangat berperan dalam proses pembangkitan metadata. Jena digunakan dalam proses pembentukan metadata RDF dan proses pembangunan query pada proses pencarian.

Jena memiliki *RDF API* (Application Programming Interface) yang berguna untuk memanipulasi model RDF baik dalam bentuk *RDF Triples* maupun *RDF Resource* dan *Properties*, memiliki *parser* dan *writer* yang terintegrasi untuk *RDF/XML*, *N3* dan *N-Triples* dsb.

2.7. SPARQL

SPARQL merupakan singkatan dari SPARQL Protocol and RDF Query Language. Secara umum sintak dari SPARQL mirip dengan bahasa SQL namun digunakan untuk file RDF.

SPARQL dapat dipanggil langsung oleh aplikasi java dengan menggunakan kelas *com.hp.hpl.jena.query package*. Dengan menggunakan *QueryFactory* dapat melakukan pembangunan query dengan masukan tekstual query dari file atau string.

Langkah selanjutnya adalah menciptakan instance *QueryExecution*. Dengan menggunakan metode *create(query,model)* maka query akan dilakukan dengan sasaran model, sehingga tidak perlu 'FROM' seperti di dalam SQL. Untuk menjalankan *QueryExecution* maka dijalankan method *execSelect()* dan menghasilkan *ResultSet*.

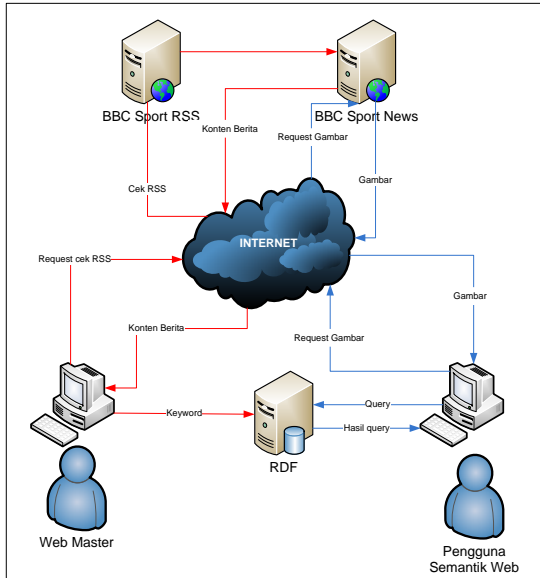
III. PERANCANGAN DAN IMPLEMENTASI

Pada bagian ini akan dijelaskan mengenai proses perancangan dan implementasi dari sistem pembangkit metadata citra (PMC).

3.1. Arsitektur Sistem

Arsitektur sistem PMC dapat dilihat pada Gambar 4. Pada Gambar tersebut terdapat pengguna I (*web master*) yang akan melakukan pembubuhan metadata pada gambar dengan menjalankan aplikasi PMC. Citra yang diberi metadata merupakan citra yang ada di situs BBC Sport. Sistem akan melakukan cek pada RSS (*Really Simple Syndication*) dari portal berita olahraga BBC Sport tersebut. Sistem akan mendapatkan *link* dari RSS. Setiap *link* akan di cek oleh sistem apakah memiliki konten citra. Jika memiliki konten citra sistem akan melakukan pengambilan informasi dari konten beritanya. Konten yang didapat dikembalikan kepada *web master* untuk dilakukan pemrosesan bahasa alami. Dengan pemrosesan bahasa alami maka dihasilkan *keyword* yang disimpan ke dalam file RDF.

Proses ujicoba dilakukan dengan membuat mesin pencari yang digunakan untuk mencari citra yang telah dibubuhi metadata. Pencarian dilakukan oleh pengguna II (*client*) yang akan melakukan proses pencarian, dengan mencari melalui browser yang terhubung dengan web server. Web server akan mengirimkan permintaan dari *client* yang berupa query yang akan mengakses file RDF. Apabila query memiliki *result* maka akan mengembalikan URI dari citra dan URI dari web berita.



Gambar 4. Arsitektur Sistem Pembangkit Metadata Citra (PMC)

Sehingga pada makalah ini akan dijelaskan dua buah system, yaitu sistem Pembangkit Metadata Citra (PMC) sebagai pembangkit metadata citra dan Mesin Pencari Citra (MPC). Sistem PMC diperuntukkan untuk *web master* sedangkan sistem MPC diperuntukkan bagi pengguna yang akan melakukan pencarian citra berdasar metadata yang dibangkitkan. Untuk membangkitkan metadata secara otomatis, terlebih dahulu harus dirancang struktur metadata yang dimaksud. Bagian berikutnya menjelaskan tentang perancangan struktur metadata atau model RDF yang akan digunakan untuk menggambarkan metadata yang diinginkan.

3.2. Perancangan Model Data RDF sebagai model metadata Citra

Citra yang akan diproses adalah citra pada sebuah web berita. Citra yang terdapat dalam sebuah web berita merupakan sebuah foto dari suatu aktifitas. Sebuah aktifitas tentunya memiliki satu atau lebih obyek yang berbentuk Subyek atau pelaku dari aktifitas dalam foto. Sebuah aktifitas tentu akan memiliki aksi atau kegiatan yang dilakukan oleh Subyek yang terdapat pada foto. Informasi yang lebih mendetail yang terdapat di dalam foto antara lain *event* (kejadian) yaitu foto tersebut terjadi saat suatu kejadian terjadi, sebagai contoh kejadian Piala Dunia. *Place* yaitu tempat kejadian yang ada di foto tersebut terjadi. *Times* yaitu waktu yang menyatakan kapan foto tersebut diambil.

Pada bagian ini akan diberikan gambaran mengenai rancangan model RDF/RDFS berdasarkan topik seputar olahraga, perancangan

dibuat berdasarkan asumsi akan apa saja yang mungkin digambarkan dalam sebuah foto pada website yang ber-tema berita olahraga, di mana asumsi tersebut juga akan dijelaskan berikut ini. Perancangan ini menggunakan notasi-notasi dan istilah-istilah resmi web semantik. Penjelasan dari masing-masing node termasuk seluruh atribut dan hubungan antar node yang digambarkan pada model RDF/RDFS di Tabel 1. Informasi tersebut akan di definisikan sebagai kelas dalam file RDF yang akan dibangun oleh sistem PMC.

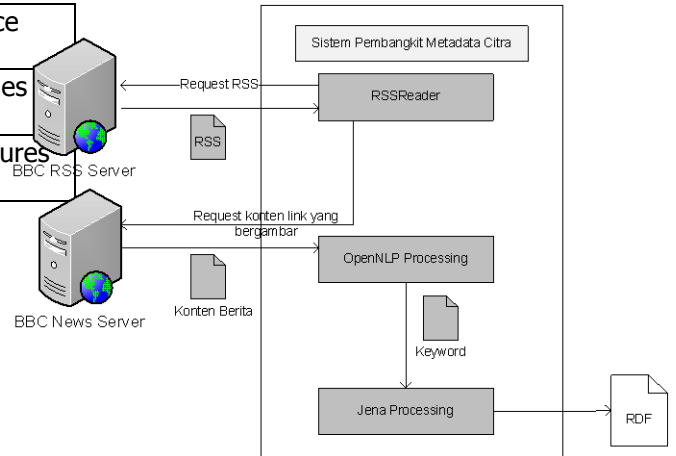
Tabel 1. Kelas pada model metadata citra

Nama	Penjelasan
KeyEvents	<i>Instance</i> pada kelas ini adalah himpunan semua aktifitas yang terjadi didalam sebuah citra foto.
Subjects	<i>Instance</i> pada kelas ini adalah daftar pelaku yang ada di sebuah citra foto.
Actions	<i>Instance</i> pada kelas ini adalah daftar kegiatan dari pelaku di sebuah citra foto.
Events	<i>Instance</i> pada kelas ini adalah daftar event/ kegiatan didalam sebuah citra foto
Place	<i>Instance</i> pada kelas ini adalah daftar tempat terjadinya foto.
Times	<i>Instance</i> pada kelas ini adalah daftar waktu terjadinya foto
Pictures	<i>Instance</i> pada kelas ini berupa URL dari citra dan URL dari berita yang memuat citra tersebut

Tabel 2 Properti pada model metadata citra

Nama	Tipe	Domain	Range
rdfhasLabel	<i>Datatype Property</i>	Subjects Actions Events Place Times	String
rdfhasPictureURI	<i>Datatype Property</i>	Pictures	String
rdfhasNewsURI	<i>Datatype Property</i>	Pictures	String
rdfhasType	<i>Datatype Property</i>	KeyEvents Subjects Actions Events Place Times Pictures	String
rdfeventHasSubjects	<i>Object Property</i>	KeyEvents	Subjects
rdfeventHasActions	<i>Object Property</i>	KeyEvents	Actions
rdfeventHasOccassion	<i>Object Property</i>	KeyEvents	Events

rdfeventHasPlace	Object Property	KeyEvents	Place
rdfeventHasTimes	Object Property	KeyEvents	Times
rdfeventPicture	Object Property	KeyEvents	Pictures



3.3. Perancangan Informasi yang Digunakan Sebagai Instance RDF class

Pada bagian 3.2 telah dijelaskan bahwa model data RDF yang dirancang memiliki beberapa class antara lain : *KeyEvents*, *Subjects*, *Actions*, *Events*, *Places*, *Times*, *Pictures*. Instance dari kelas yang telah dirancang merupakan hasil pemrosesan informasi yang di ambil dari berita. Sebuah instance yang di bangkitkan tentunya mempunyai domain tertentu. Sebagai contoh *KeyEvents* dalam sistem ini merupakan sebuah kejadian yang berada di suatu citra dan dijelaskan pada judul dari gambar yang bersangkutan.

Untuk lebih jelas berikut adalah domain informasi dari kelas-kelas RDF yang dibuat.

Tabel 3 Domain kelas RDF citra

Nama	Penjelasan
KeyEvents	Sebuah kejadian yang berada di suatu citra dan dijelaskan pada judul dari gambar yang bersangkutan.
Subjects	Didapatkan dari proses <i>English Treebank</i> diambil pada <i>Noun Phrases</i> .
Actions	Didapatkan dari proses <i>English Treebank</i> diambil dari kata-kata yang merupakan verb yang terdapat di <i>Verb Phrases</i> .
Events	Didapatkan dengan menggunakan fungsi <i>NameFinder</i> dengan tipe <i>Event</i> .
Place	Didapatkan dengan menggunakan fungsi <i>NameFinder</i> dengan tipe <i>Location</i> .
Times	Didapatkan dengan menggunakan fungsi <i>NameFinder</i> dengan tipe <i>Times</i> .
Pictures	Untuk URL didapatkan dengan cara menggunakan fungsi filter dari <i>HTML Parser</i> .

3.4. Sistem Pembangkit Metadata Citra (PMC)

Gambar 7. Gambaran Umum Sistem Pembangkit Metadata Citra

Gambar 7 merupakan gambaran umum sistem dalam proses pembangkitan metadata dari suatu citra. Proses diawali dari pengecekan RSS. Sistem akan melakukan permintaan untuk mendapatkan script XML dari RSS yang sedang aktif. Setelah mendapatkan script XML dari RSS sistem mengecek setiap item RSS. Dari proses ini didapatkan link yang dibutuhkan untuk melakukan akses konten berita yang dimuat oleh RSS. Bentuk XML RSS terdapat pada Gambar 8.

```
<item>
  <title>Wenger puts pressure on Fabregas</title>
  <description>Arsene Wenger says it is now up to Cesc Fabregas
  <link>http://localhost/ujicoba/football/1.htm</link>
  <guid isPermaLink="false">http://news.bbc.co.uk/sport1/hi/fo
  <pubDate>Sat, 31 Jul 2010 19:13:08 GMT</pubDate>
  <category>Arsenal</category>
  <media:thumbnail width="66" height="49" url="http://news.bbc
</item>
```

Gambar 8. Tag Link dalam XML RSS

Proses pengambilan RSS dalam sistem secara default akan dilakukan setiap satu jam sekali. Dari link yang didapat, sistem akan melihat apakah dalam konten yang dimiliki oleh link memiliki citra. Konten yang memiliki citra akan diambil dan yang tidak memiliki citra tidak akan diambil. Proses pengecekan konten citra dilakukan dengan melakukan pengecekan tag html yang dimiliki oleh web berita. Suatu citra diapit oleh tag img seperti pada Gambar 9.

```
<table width="226" align="right" border="0" cellpadding="0" cellspacing="0">
  <tbody><tr><td>
  <div>
  Wenger admits the Fabregas saga has made hi:
  </div>
  </td></tr>
</tbody></table>
```

Gambar 9. Tag img dalam web berita BBC

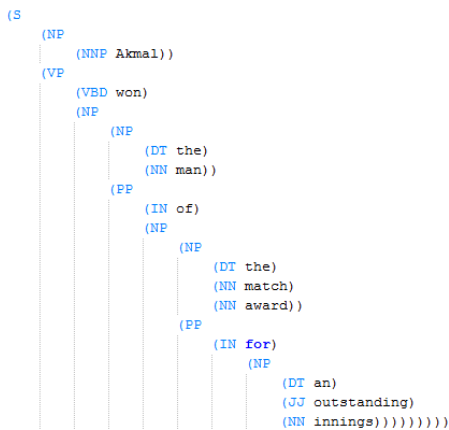
Untuk mendapatkan tag yang tepat maka akan difilter berdasar tag htm yang berhubungan dengan tag *img* tersebut. Sebagai contoh, tag *img* tersebut merupakan saudara dalam tree atau

berkedudukan sama dengan tag *div* dengan kelas “cap”.

Untuk pengambilan konten berita dilakukan dengan cara yang sama dengan pengecekan gambar, yaitu dengan melakukan pengecekan pada tag htmlnya.

Keluaran dari proses *RSSReader* adalah informasi dari citra, meliputi : alamat citra dan konten berita.

Hasil keluaran dari *RSSReader* akan dilanjutkan pada proses *OpenNLPProcessing* (pemrosesan bahasa alami dengan pustaka OpenNLP). Fungsi *Treebank Parser* digunakan untuk mendapatkan kata kunci berdasar pada *Part of Speech* dari kalimat bahasa Inggris. Untuk fungsi *Treebank* akan menghasilkan *tree* pada Gambar 10. Sedangkan untuk proses *NameFinder* digunakan untuk mendapatkan suatu jenis entitas kata dalam sebuah kalimat.



Gambar 10. Tree hasil dari TreebankParsing

Hasil dari proses *Treebank* di setiap kalimat konten berita akan menghasilkan tree. Seperti yang telah dijelaskan pada Tabel 3.2. Maka proses ini seharusnya akan menghasilkan Subyek dan Aksi. Pada Tabel 3 dijelaskan bahwa Subyek didapatkan dengan mengambil kata yang terdapat pada *Noun Phrase*. Dari Gambar 10, “Akmal” akan disebut sebagai Subyek *verb* yang berada di dalam *Verb Phrase* adalah “won”

Untuk fungsi *NameFinder* harus melalui training data terlebih dahulu. Hal ini disebabkan tidak semua model disediakan oleh OpenNLP. Sedangkan salah satu entitas yang ingin dicari adalah “event” sehingga diperlukan training model terlebih dahulu.

Training diawali dengan membuat corpus kalimat yang akan di jadikan entitas. Data training pada Gambar 11 merupakan bagian dari data *training model event*.

```
Great Britain's Jessica Ennis won
<START> European heptathlon <END> gold after
dominating the 800m finale to fend off her
closest rival Nataliya Dobrynska.
```

Gambar 11. Kalimat Training

Pada kalimat Gambar 11, antara tag <START> dan <END> terdapat kata “European heptathlon”. Kata tersebut yang akan dimasukkan ke dalam model yang akan dibuat. Untuk melakukan training dengan kata lain dapat dilakukan pada tempat yang sama. Namun kata lain itu harus memiliki entitas yang sama dengan kata lain dalam 1 *corpus*.

Sehingga pada satu file *corpus* kalimat hanya digunakan untuk melakukan training pada satu jening entitas saja.

Setelah membuat data training maka dengan fungsi dari *NameFinderME* dapat dilakukan trining model. Pemanggilan fungsi dapat dilakukan melalui command prompt seperti pada Gambar 12.

```
java -classpath opennlp-tools-1.4.5.jar;
trove.jar; maxent-2.5.2.jar; jwnl-
1.3.3.jar
opennlp.tools.namefind.NameFinderME
training.txt event.bin.gz
```

Gambar 12. Fungsi yang dijalankan pada CMD

Fungsi tersebut akan melakukan perhitungan probabilitas sehingga dihasilkan model *event.bin.gz*. Untuk lebih jelasnya terdapat pada gambar 13. Setelah mendapatkan model dari hasil training maka model akan digunakan sebagai dasar pencarian entitas yang diinginkan antara lain “event”, “location” dan “time”.

Sehingga sistem telah mendapatkan kata kunci antara lain Subjects, Actions, Event, Location dan Time dari setiap gambar yang didapatkan. Setelah didapatkan kata kunci, maka proses akan berlanjut pada proses penyimpanan kata kunci. Proses ini dilakukan dengan bantuan *Jena*. *Jena* digunakan dalam proses manipulasi file RDF.

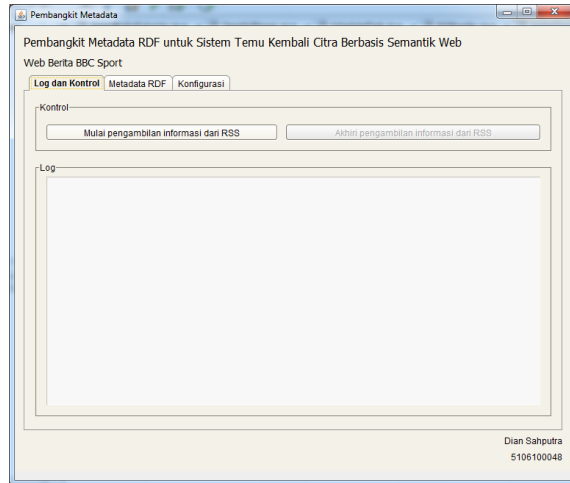
IV. UJI COBA

Uji coba dilakukan dengan terhadap citra yang ada di situs BBC Sport [5]. Sumber berita diambil dari portal berita RSS BBC Sport yang diambil pada tanggal 1 Agustus 2010 pukul 13.00. Dengan rincian berita sebagai berikut :

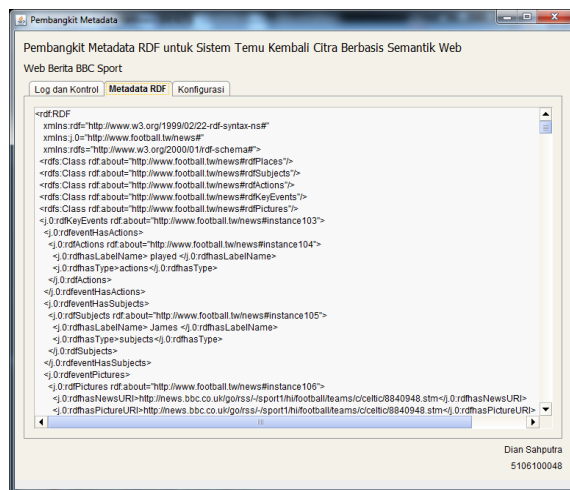
1. Berita yang diambil adalah berita olahraga dengan sub bagian : football, rugby_union, rugby_league, dan athletics.
2. Jumlah total berita ada 45 buah berita.
Dengan rincian :
 - a. Football = 15 berita
 - b. Rugby_union = 15 berita
 - c. Rugby_league = 15 berita
3. Jumlah berita yang memiliki konten gambar ada 32 buah berita.
4. Jumlah total berita yang tidak memiliki konten gambar ada 13 buah berita.

Gambar 13 merupakan gambar antar muka sistem untuk mulai mengambil informasi dari situs tempat citra berada. Ketika tombol “Mulai pengambilan informasi dari RSS” ditekan, maka proses pengambilan informasi dimulai.

Proses pengambilan akan berakhir jika tombol “Akhir pengambilan informasi dari RSS” ditekan. Pada tab Metadata RDF akan ditampilkan hasil pembangkitan, seperti yang disajikan pada Gambar 14 berikut.



Gambar 13. Antar muka proses pengambilan informasi



Gambar 14. Metadata hasil pembangkitan

Metadata selanjutnya disimpan. Untuk menguji apakah metadata hasil menghasilkan informasi citra yang benar, dilakukan uji coba dengan mencari citra melalui mesin pencari berbasis web yang dikembangkan didalam penelitian ini juga. Gambar 15 dan 16 menunjukkan contoh pencarian citra dengan mesin pencari yang dibuat. Pengguna dapat memilih subyek yang dicari dengan bantuan

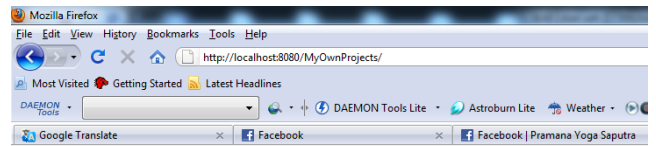
combo box yang ada dalam mesin pencari. Selanjutnya mesin pencari akan mengembalikan hasil yang didapat seperti pada Gambar 16. Jika link pada Gambar 16 diklik, maka akan muncul tampilan citra yang sedang dicari seperti pada Gambar 17.



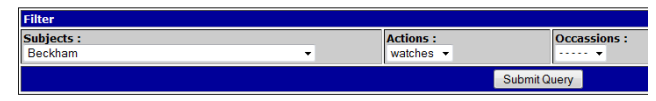
Semantic Portal



Gambar 15. Halaman awal proses pencarian



Semantic Portal



JSP Query Results

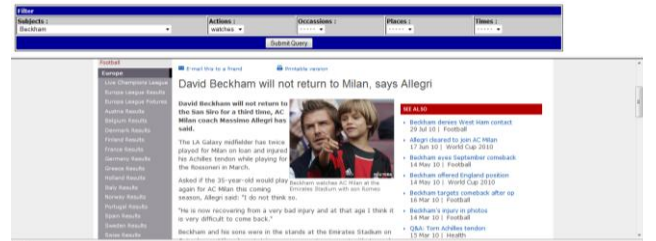
Hasil Query :



Gambar 16. Hasil proses pencarian



Semantic Portal



Gambar 17. Konten berita dari link hasil pencarian

Setelah dilakukan uji coba, selanjutnya dilakukan perbandingan terhadap pendapat domain expert. Dari hasil perbandingan didapatkan tingkat relevansi rata-rata sebesar 59,77% dengan rincian relevansi terhadap subyek sebesar 83,34 %, relevansi kata kunci

actions 28,34 %, dan relevansi kata kunci *events* 90 %.

V. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini antara lain sebagai berikut :

1. Pemrosesan konten berita dengan menggunakan sistem Pembangkit Metadata Citra dapat menghasilkan pembubuhan metadata pada suatu citra dengan tingkat relevansi sebesar 59,77 % dibandingkan dengan pendapat domain ekspert.
2. Metadata yang telah dibangun oleh sistem dapat digunakan untuk proses pencarian pada mesin pencari berbasis semantik dengan melakukan query pada metadata tersebut.

VI. DAFTAR PUSTAKA

- [1]. Understanding Metadata, <http://www.niso.org/publications/press/UnderstandingMetadata.pdf>, diakses pada bulan Mei 2010
- [2]. RDF, <http://www.w3.org/RDF/>, diakses pada bulan Mei 2010
- [3]. *Parsing*, <http://www.lancs.ac.uk/fss/courses/ling/corpus/Corpus2/2PARSE.HTM>, diakses pada Juni 2010.
- [4]. Jena, <http://jena.sourceforge.net/>, diakses pada bulan April 2010
- [5]. BBC Sport, <http://news.bbc.co.uk/sport>, diakses pada bulan Juli dan Agustus 2010