

Search Engine pada Dokumen RDF Tanaman Obat Menggunakan Sesame dan Lucene

Search Engine on RDF Document of Medicinal Plants Using Sesame and Lucene

Luthfi Noviandi(G64124020)*, Julio Adisantoso

Abstrak/Abstract

Bertambahnya keanekaragaman tanaman obat menyebabkan dokumentasi hasil penelitian tanaman obat semakin bertambah, sehingga mengakibatkan kesulitan dalam hal pencarian dokumen. Diperlukan suatu sistem pencarian yang dapat menemukan kembali dokumen yang dicari dengan menggunakan kueri. Pada penelitian ini akan dilakukan pengembangan sistem pencarian pada dokumen RDF menggunakan Sesame dan Lucene. Pembobotan menggunakan Tf-idf sebagai nilai relevansi terhadap dokumen yang ditemukan kembali. Penggunaan sistem Lucene sebagai mesin pencari menghasilkan rata-rata precision pada kueri tanpa term boosting sebesar 0.862. Sedangkan untuk kueri yang diberikan term boosting menghasilkan rata-rata precision sebesar 0.877.

Increasing the diversity of medicinal plants causing documentation of the results of research medicinal plants was increasing that led to the difficulty in terms of search of a document. Required a searching system that can be retrieves documents using query. In this study, we will develop searching system for RDF documents using Sesame and Lucene. TF-idf used as relevant value about documents retrived. Query without boosting term generate average precision 0.862 while query with boosting term generate average precision 0.877.

Kata Kunci/Keywords

search engine; lucene; sesame; RDF; Tf-idf; tanaman obat
search engine, lucene, sesame, RDF, Tf-idf, medicinal plants

*Email: luthfinoviandi@gmail.com

PENDAHULUAN

Latar Belakang

Tanaman obat adalah tanaman yang mengandung bahan yang dapat digunakan sebagai pengobatan dan kandungan kimianya dapat digunakan sebagai bahan obat sintetis (DEPTAN 2009). Dengan bertambahnya keanekaragaman tanaman obat, maka dokumentasi hasil penelitian tanaman obat semakin bertambah. Oleh karena itu, dibutuhkan mesin pencari yang dapat mencari definisi dan manfaat dari tanaman obat. Herawan (2011) melakukan penelitian untuk temu kembali informasi dengan ekstraksi ciri dokumen menggunakan *chi-square* dengan klasifikasi *naive bayes* pada dokumen *eXtensible Markup Language* (XML) tanaman obat.

Dalam pengembangan temu kembali informasi format dokumen yang digunakan bermacam-macam diantaranya *freetext* atau XML. XML merupakan sintaks dan model data yang direpresentasikan dengan bentuk tree dan bergantung pada konsep *tag* seperti *Hypertext Markup Language* (HTML). XML saat ini digunakan

untuk membuat infrastruktur web semantik. Salah satu tujuan penting dari web semantik adalah untuk membuat makna informasi yang jelas, sehingga memungkinkan akses yang lebih efektif untuk pengetahuan yang terkandung dalam lingkungan informasi yang beraneka ragam (Lei *et al.* 2006). Agar kinerja mesin pencari meningkat maka dokumen yang diolah harus memiliki skema ontologi. Ontology merupakan skema metadata yang dapat menambahkan makna dari data dan memungkinkan untuk menyimpulkan informasi baru dari data yang ada. Salah satu dokumen yang dapat mendukung ontologi adalah *Resource Description Framework* (RDF).

Minack *et al.* (2008) melakukan penelitian untuk membuat full-text search dengan dokumen RDF. RDF diolah menggunakan bahasa kueri SPARQL, akan tetapi tidak cukup mampu untuk menangani jumlah data yang besar. SPARQL hanya mampu melakukan penyediaan berdasarkan *regular expression* sehingga dibutuhkan aplikasi yang dapat melakukan *indexing*, *stemming*, dan *ranking* pada *search engine*.

Banyak aplikasi mesin pencari yang sudah dikem-

bangkan antara lain Sphinx dan Lucene. Salah satu aplikasi yang dapat melakukan pencarian terhadap dokumen RDF adalah Lucene. Lucene merupakan aplikasi mesin pencari yang menerapkan konsep *full-text search*. Lucene memiliki performa yang sangat baik walaupun digunakan pada sumber daya yang rendah (Minack *et al.* 2008). Lucene dapat melakukan *stemming* dan *lemmatization*, pencarian menggunakan frase, *wildcard*, *fuzzy*, *proximity* dan *range queries*. Untuk melakukan pengindeksan dokumen RDF, Lucene membutuhkan aplikasi yang dapat mengolah data RDF salah satunya adalah Sesame. Sesame merupakan *open-source framework* untuk media penyimpanan RDF dan menyediakan bahasa kueri SeRQL dan SPARQL untuk *parsing* data. Oleh karena itu penelitian ini dilakukan untuk mengembangkan mesin pencari menggunakan Sesame dan Lucene pada dokumen RDF.

Perumusan Masalah

Penelitian ini dilakukan untuk menjawab permasalahan :

1. Bagaimana metode untuk mengkonversi format XML menjadi RDF?
2. Apakah Lucene mampu mengindeks dokumen dengan format RDF?
3. Bagaimana kinerja search engine yang dikembangkan dengan menggunakan Lucene pada dokumen RDF?

Tujuan

Tujuan penelitian ini antara lain:

1. Mengimplementasikan sistem Lucene untuk membangun search engine pada dokumen RDF.
2. Mengimplementasikan metode untuk konversi dokumen XML menjadi dokumen RDF.

Manfaat

Penelitian ini diharapkan dapat membantu seseorang dalam mencari informasi yang relevan mengenai tanaman obat di Indonesia.

Ruang Lingkup

Ruang lingkup penelitian ini yaitu proses *indexing* dilakukan terhadap semua atribut *field* yang terdapat pada dokumen RDF dengan bobot yang tidak dibedakan dan struktur dokumen RDF sama untuk setiap dokumen.

METODE PENELITIAN

Tahapan penelitian terdiri atas membangun dokumen RDF tanaman obat, penyimpanan dokumen ke dalam

aplikasi Sesame, proses *indexing* dan pencarian menggunakan Lucene, serta evaluasi.

Membangun Dokumen RDF

RDF adalah model metadata dari bahasa yang direkomendasikan oleh W3C untuk membangun infrastruktur web semantik (Gutierrez *et al.* 2007). Pada RDF, sebuah deskripsi dari sumber direpresentasikan sebagai sejumlah triple, tiga bagian dari setiap *triple* disebut subyek, predikat, dan objek. Subyek dari *triple* adalah *Uniform Resource Identifier* (URI) yang mendefinisikan sumber. Objek dapat berupa nilai literal sederhana, seperti string, numerik, tanggal, atau URI dari sumberdaya lainnya yang berkaitan dengan subyek. Predikat mengindikasikan hubungan antara subyek dan objek. RDF juga menyediakan sebuah sintaks berbasis XML yang disebut juga RDF/XML.

XML dan RDF secara umum digunakan untuk membangun infrastruktur semantik tetapi keduanya memiliki fungsi yang berbeda. XML berkaitan dengan transmisi data, sedangkan RDF berkaitan dengan konten informasi. Dokumen XML yang digunakan dalam penelitian ini adalah dokumen tanaman obat yang telah digunakan sebelumnya pada penelitian Herawan (2011). Korpus tersebut terdiri atas 93 dokumen. Data tanaman obat kemudian dikonversi menjadi dokumen RDF/XML dan disimpan ke dalam aplikasi Sesame.

Sesame

Pada penelitian ini digunakan Sesame untuk pengolahan data dokumen RDF. Sesame merupakan aplikasi yang dikembangkan oleh Aduna yang menyediakan fungsi untuk *parsing*, menyimpan, dan kueri pada data RDF. Sesame menyediakan dua bahasa kueri yaitu SeRQL dan SPARQL. SeRQL dan SPARQL merupakan bahasa kueri yang dikembangkan oleh Aduna yang digunakan untuk memanipulasi data dan *parsing* data RDF. Dokumen RDF tanaman obat disimpan pada aplikasi Sesame untuk di *parsing* menggunakan kueri SPARQL.

Lucene

Proses *indexing* dilakukan menggunakan perangkat lunak Lucene yang mencakup tokenisasi, *stemming* dan *lemmatization*, pembuangan *stopwords*, pembobotan, dan penyimpanan hasil *indexing* ke dokumen Lucene.

Tokenisasi merupakan proses pemotongan teks untuk mendapatkan token dari suatu berkas (Manning *et al.* 2008). Tokenisasi melakukan pemisahan terhadap isi dokumen menjadi unit yang lebih kecil yang biasa dise-

but juga kata. *Stemming* dan *lemmatization* merupakan proses pengolahan linguistik tambahan yang dapat ditangani dengan tokenisasi. Tokenisasi dilakukan untuk semua korpus tanaman obat yang telah tersedia. Pada tokenisasi juga dilakukan pembuangan *stopwords*.

Stopwords merupakan kata umum yang sering muncul dalam suatu dokumen dengan jumlah besar tetapi tidak memiliki makna. *Stopwords* dibuang karena dianggap akan mengurangi akurasi dari informasi yang di temukembalikan (Manning *et al.* 2008). Contoh dari *stopwords* antara lain “yang”, “dan”, “atau”, “di”, dan lain-lain. Kata yang sudah melalui proses tokenisasi dan pemotongan *stopwords* akan diberikan pembobotan.

Pembobotan merupakan proses untuk memberikan nilai bobot pada suatu term untuk merepresentasikan ciri suatu dokumen. Hasil pembobotan akan membentuk suatu sistem peringkat yang akan mengurutkan term dengan tingkat kemiripan tertinggi ke tingkat kemiripan terendah.

Pada perangkat lunak Lucene digunakan pembobotan *term frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term frequency* melakukan pembobotan untuk menghitung jumlah kemunculan term pada suatu dokumen dan sebagai ukuran untuk tingkat relevansi dokumen (Minack *et al.* 2008). *Inverse document frequency* (IDF) akan menghitung jumlah dokumen yang memiliki suatu term tertentu untuk dibandingkan dengan jumlah semua dokumen. Untuk menghitung term t pada dokumen d digunakan

$$tf.idf_{td} = tf_{td} \times \log \frac{N}{df_t} \quad (1)$$

dengan N adalah jumlah dokumen tanaman obat, df_t adalah jumlah dokumen yang mengandung term t .

Proses pencarian dapat dilakukan jika dokumen sudah terindeks pada dokumen Lucene. Pencarian dilakukan menggunakan kueri yang berhubungan dengan tanaman obat, kemudian dihitung nilai kemiripannya. Nilai kemiripan akan berpengaruh terhadap hasil temu kembali oleh sistem. Lucene menggunakan fungsi *Vector Space Model* (VSM) untuk menentukan *similarity* hasil pencarian seperti pada persamaan

$$sim(q, d) = \frac{V_q \cdot V_d}{|V_q| |V_d|} \quad (2)$$

dengan V_q merupakan vektor dokumen q , V_d merupakan vektor dokumen d , $|V_q|$ merupakan panjang vektor dokumen q , dan $|V_d|$ merupakan panjang vektor dokumen

d . Untuk skoring pada Lucene dapat dilihat pada persamaan

$$sim(q, d) = \frac{tf_{td}}{tf_{tq}} \cdot \frac{1}{|q|} \sum_{t \in q} (tf_{td} idf_t^2) \cdot boost() \cdot \frac{1}{|d|} \quad (3)$$

dengan $boost()$ merupakan nilai *booster* yang diberikan terhadap term pada kueri dengan nilai default 1.0. Nilai *booster* akan dikalikan terhadap term t yang diberikan *boost*.

Evaluasi

Evaluasi dilakukan terhadap dokumen yang ditemukembalikan oleh mesin pencari berdasarkan kueri yang diberikan. Jumlah kueri yang digunakan yaitu 29 kueri yang didapatkan dari penelitian Herawan (2011). Pada penelitian ini dilakukan evaluasi temu kembali informasi menggunakan *recall* dan *precision*. *Precision* didefinisikan sebagai rasio dokumen yang ditemukembalikan adalah relevan dengan persamaan

$$precision = \frac{a}{b} \quad (4)$$

dengan a merupakan banyaknya dokumen relevan yang ditemukembalikan dan b adalah jumlah semua dokumen dari hasil pencarian.

Recall didefinisikan sebagai rasio dokumen relevan yang ditemukembalikan dengan persamaan

$$recall = \frac{a}{c} \quad (5)$$

dengan c adalah banyaknya dokumen relevan yang terdapat pada korpus.

Nilai rata-rata *interpolated precision* dapat mencerminkan urutan dari dokumen-dokumen yang relevan pada perangkikan. Standar yang digunakan adalah 11 level *recall standar* yaitu 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan 1.0. Nilai *precision* hasil interpolasi maksimum didefinisikan dengan persamaan

$$P_{interp}(r_j) = \max_{r_j \leq r \leq r_{j+1}} P(r) \quad (6)$$

dengan $P(r)$ adalah nilai *precision* pada suatu titik *recall* r .

Lingkungan Pengembangan

Spesifikasi perangkat lunak dan perangkat keras yang digunakan pada penelitian ini yaitu:

1. Perangkat lunak:

- Sistem Operasi Windows 8.1 x64
- Bahasa pemrograman PHP
- XAMPP v3.2.1
- ZendLucene, digunakan untuk search engine
- Sesame, digunakan untuk pemrosesan RDF
- Sublime Text 3, digunakan sebagai *editor* kode program
- Codeigniter 2.2.0, digunakan sebagai *framework* PHP

2. Perangkat keras berupa komputer personal dengan spesifikasi sebagai berikut:

- *Processor* Intel Core i5
- RAM 4 GB DDR3
- *Monitor* LCD 14.0" 16:9 HD
- *Harddisk* 500GB

HASIL DAN PEMBAHASAN

Arsitektur Dokumen RDF

Penelitian ini menggunakan dokumen tanaman obat yang didapat dari penelitian Herawan (2011). Jumlah dokumen yang digunakan berjumlah 93 dokumen dengan format XML. Dokumen dikelompokkan menjadi *tag-tag* seperti pada Tabel 1.

Tabel 1. Deskripsi dokumen XML tanaman obat

Nama Tag	Deskripsi
<dok>	Mewakili keseluruhan dokumen
<id>	Menjelaskan id dokumen
<nama>	Nama tanaman obat
<namal>	Nama latin tanaman obat
<deskripsi>	Deskripsi tanaman obat yang terdiri dari manfaat, habitus, bagian yang digunakan dan kandungan zat kimia
<fam>	Famili tanaman obat
<penyakit>	Penyakit yang dapat disembuhkan oleh tanaman obat.

Dokumen XML kemudian dikonversi menjadi format dokumen RDF/XML dengan deskripsi seperti pada Tabel 2. Proses konversi dilakukan secara manual sesuai dengan tag yang digunakan pada dokumen XML. Pada bagian tag <deskripsi> terdapat manfaat, kandungan,

Tabel 2. Deskripsi dokumen RDF tanaman obat

Nama Tag	Deskripsi
<rdf:RDF>	Merupakan <i>namespace</i> untuk dokumen RDF
<rdf:Description>	Mewakili keseluruhan dokumen
<rdf:about>	ID Dokumen atau subjek pada RDF
<tanaman:famili>	Famili tanaman obat
<tanaman:nama>	Nama tanaman obat
<tanaman:latin>	Nama latin tanaman obat
<tanaman:habitus>	Habitus dari tanaman obat.
<tanaman:bagian>	Bagian yang digunakan pada tanaman obat
<tanaman:manfaat>	Manfaat dari tanaman obat
<tanaman:kandungan>	Kandungan dari tanaman obat
<tanaman:lokasi>	Lokasi tanaman obat ditemukan
<tanaman:deskripsi>	Deskripsi tanaman obat
<tanaman:penyakit>	Penyakit yang dapat disembuhkan oleh tanaman obat

dan lokasi yang dapat dipisahkan menjadi field yang berbeda pada dokumen RDF.

Dokumen RDF tanaman obat diberikan namespace dengan nama "tanaman". Pada field <tanaman:manfaat>, <tanaman:kandungan>, dan <tanaman:lokasi> dibuat dalam bentuk **rdf:Bag** karena pada beberapa dokumen tanaman obat memiliki manfaat dan kandungan yang banyak. **Rdf:Bag** merupakan tipe data dari RDF yang mendefinisikan bentuk *unordered-list*.

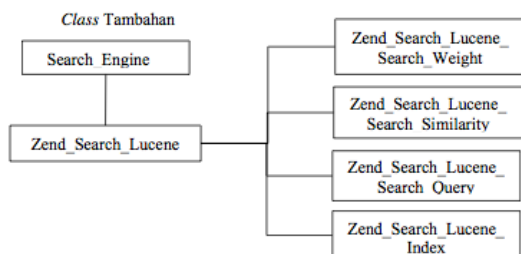
Dokumen RDF didefinisikan menggunakan subjek, predikat, dan objek. Berikut merupakan contoh definisi pada dokumen obat Pandan Wangi:

- tanaman_1 memiliki nama Pandan Wangi.
- tanaman_1 memiliki famili *Pandanaaceae*.
- tanaman_1 memiliki nama latin *Pandanus amaryllifolius Roxb.*
- Bagian yang digunakan pada tanaman_1 adalah daun.
- tanaman_1 memiliki manfaat rambut rontok, menghitamkan rambut, menghilangkan ketombe, lemah saraf, tidak nafsu makan, rematik, pegal linu, dan sakit disertai gelisah.
- tanaman_1 memiliki kandungan alkaloida, saponin, flavonoida, tannin, polifenol dan zat warna

Dokumen RDF yang telah tersedia disimpan ke dalam aplikasi Sesame dengan nama repositori tanaman-obat. Kueri dibutuhkan untuk *parsing* data pada dokumen RDF tanaman obat. Pada penelitian ini menggunakan bahasa kueri SPARQL.

Indexing

Indexing dan pencarian dilakukan dengan menggunakan Lucene. Lucene merupakan sebuah mesin pencari yang digunakan dalam membangun aplikasi ini, untuk proses *indexing*, *searching*, dan *ranking*. Pada penelitian ini digunakan beberapa *class* yang terdapat pada pustaka Lucene dan dilakukan penambahan *class* yang digunakan sebagai penghubung antara pengguna dan Lucene. Struktur penggunaan *class* dapat dilihat pada Gambar 1.



Gambar 1. Struktur penggunaan *class*

Class Search Engine memiliki fungsi sebagai berikut:

1. **indexing()**, digunakan untuk melakukan fungsi indexing
2. **getManfaat()**, digunakan untuk parsing data RDF pada field manfaat dengan tipe data <rdf:Bag>
3. **getKandungan()**, digunakan untuk parsing data RDF pada field kandungan dengan tipe data <rdf:Bag>
4. **doSearch**, digunakan untuk melakukan proses pencarian

Pengguna memasukkan kueri yang selanjutnya akan diproses oleh fungsi **doSearch** yang terdapat pada *class Search.engine*. Fungsi **doSearch** dijalankan ketika terdapat kueri yang ingin dicari di dalam koleksi dokumen RDF. Fungsi **doSearch** yang selanjutnya diproses melalui *search engine* Lucene. Setelah kueri diproses Lucene akan menemukembalikan dokumen yang relevan berdasarkan ranking tertinggi.

Untuk melakukan *indexing*, dokumen RDF yang akan diindeks disimpan dalam sebuah aplikasi penyimpanan dokumen RDF yaitu Sesame. Proses *indexing* dilakukan oleh fungsi **indexing()** yang terdapat pada *class Search.engine*. Fungsi **indexing()** akan melakukan *parsing* data dokumen RDF pada media penyimpanan Sesame dengan menggunakan kueri SPARQL. Hasil *parsing* data selanjutnya akan diindeks melalui *search engine* Lucene. Hasil *indexing* akan disimpan pada folder **tmp/rdf-indeks** yang terdapat pada direktori **/xampp/htdocs/Lucene/**.

Pencarian

Setelah dilakukan proses *indexing*, maka dapat dilakukan pencarian pada dokumen RDF tanaman obat. Pencarian dilakukan dengan memasukkan kueri pada sistem *search engine*. Jumlah kueri yang digunakan pada penelitian ini adalah 29 kueri. Kueri pada sistem ini dapat berupa kata tunggal, frase, dan gabungan dari *field* yang dipilih dengan kata tunggal atau frase. Kueri akan diproses oleh sistem Lucene yang akan me-*retrieve* dokumen yang relevan beserta skoringnya. Pada Lucene, jika hasil skoring lebih besar dari 1.0 maka nilai skoring akan dibulatkan menjadi 1.0. Tabel 3 ditunjukkan hasil temu kembali terhadap dokumen RDF tanaman obat.

Tabel 3. Hasil temu kembali dokumen RDF tanaman obat

No.	Kueri	Retrieved	Relevan
1.	Kanker	3	3
2.	Flu	2	2
3.	Diabetes	17	17
4.	Pusing	3	3
5.	Merambat	1	1
6.	Menjari	2	2
7.	Bergerigi	15	11
8.	Menyirip	19	14
9.	Vitamin	16	15
10.	Antioksidan	1	1
11.	Protein	6	3
12.	Kalsium	13	8
13.	Diseduh	12	11
14.	Ditumbuk	13	12
15.	Diperas	7	7
16.	Batuk Pilek	27	3
17.	Kencing Batu	47	4
18.	Datang Bulan	13	3
19.	Gatal-gatal	11	4
20.	Sesak Nafas	9	6

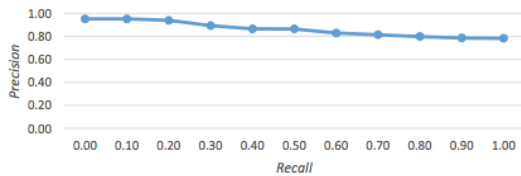
Kueri ditentukan dengan cara memilih kata tunggal atau frase yang mewakili isi setiap tanaman obat. Kata-kata tersebut berkaitan dengan penyakit yang dapat disembuhkan, kandungan kimia, karakter fisik, dan cara penggunaan tanaman obat. Pada Lucene disediakan fitur *term boosting* yang dapat meningkatkan tingkat akurasi hasil temu kembali informasi. Pada penelitian ini, tiga kueri yang memiliki tingkat akurasi yang rendah diberikan *term boosting* seperti pada Tabel 4.

Tabel 4. Kueri dengan *term boosting*

No.	Kueri	Retrieved	Relevan
1.	obat diseduh(4)	42	4
2.	obat ditumbuk(4)	39	8
3.	buah diperas(4)	60	3

Evaluasi

Evaluasi kinerja search engine dilakukan menggunakan nilai interpolasi maksimum *recall* dan *precision*. Pengujian dilakukan terhadap 29 kueri berupa kata tunggal atau frase dan dokumen yang relevan. Setiap kueri akan dihitung nilai *precision* pada setiap nilai *recall* standar yaitu 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. Setelah didapatkan nilai *precision* pada sebelas nilai *recall* untuk setiap kueri, kemudian dicari nilai interpolasi maksimum *recall* dan *precision*. Nilai interpolasi maksimum *recall* dan *precision* dapat dilihat pada Gambar 2.



Gambar 2. Rataan recall precision dengan interpolasi maksimum

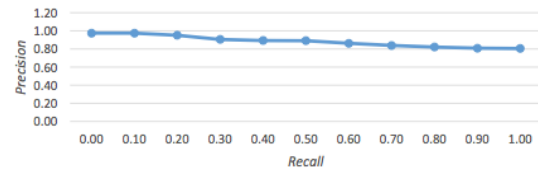
Dari percobaan yang dilakukan terhadap 29 kueri didapatkan nilai *precision* sebesar 0.862. Dapat disimpulkan bahwa kinerja sistem temu kembali informasi memiliki tingkat keakuratan yang baik untuk semua kueri yang diberikan.

Dokumen yang tidak relevan tetapi tetap ditemukembalikan terjadi pada kueri 'Batuk Pilek', 'Kencing Batu', 'Datang Bulan', 'Gatal-gatal', 'Buah Diperas', 'Tanaman Hias', 'Tumbuhan Merambat', 'Daun Elips', 'Buah Buni', 'Kalsium Oksalat', 'Zat Warna', 'Obat Diseduh', dan 'Obat Ditumbuk'. Hal ini disebabkan karena kueri tersebut memiliki banyak arti dalam setiap dokumen tanaman obat, sehingga tidak mampu mewakili informasi yang diinginkan pengguna. Misalnya pada kueri 'Tumbuhan Merambat' informasi yang diinginkan pengguna adalah mengenai tanaman obat yang tumbuh merambat, tetapi sistem akan menemukembalikan dokumen yang mengandung kata 'Tumbuhan' dan 'Merambat'. Hal tersebut yang mempengaruhi nilai *precision* yang didapat pada penelitian ini. Hal ini terbukti dengan menghilangkan kueri dua kata mengakibatkan perubahan nilai *recall* dan *precision* seperti terlihat pada Gambar 3.



Gambar 3. Recall precision untuk kueri kata tunggal

Untuk kueri dengan dua kata, perlu ditambahkan penguatan pada kata tertentu dengan menggunakan *term boosting* agar sistem dapat mengetahui kata yang dipentingkan. Dengan menambahkan *term boosting*, nilai rata-rata *precision* yang didapatkan lebih baik yaitu 0.877. Pada evaluasi ini juga dilakukan perhitungan *recall* dan *precision* dengan menggunakan nilai *term boosting* yang berbeda pada masing-masing kueri yang menggunakan dua kata yaitu 'Obat Diseduh(6)', 'Obat Ditumbuk(6)', dan 'Buah Diperas(6)'. Nilai *recall* dan *precision* yang dihasilkan setelah nilai *term boosting* ditambahkan dapat dilihat pada Gambar 4.



Gambar 4. Recall precision dengan term boosting

Nilai rata-rata *precision* yang dihasilkan dengan penambahan nilai *term boosting* yaitu 0.884. Dari Gambar 4 dapat terlihat bahwa kueri dengan menambahkan *term boosting* yang makin tinggi sampai pada nilai tertentu dapat meningkatkan keakuratan hasil temukembali informasi.

Contoh Penulisan Algoritme

Bagian ini adalah tambahan penulisan untuk mengetahui cara menuliskan algoritme yang diacu maupun yang tidak diacu dalam teks. Algoritme 1 dibuat untuk mendapatkan bilangan terbesar dari kumpulan bilangan yang terhitung.

Input: Himpunan $A = \{a_1, a_2, \dots, a_n\}$

Output: Bilangan terbesar

```

max ← a1
for i ← 2 to n do
    if ai > max then
        max ← ai
    end
end

```

end

return max

Algorithm 1: MAX mendapatkan bilangan terbesar

Berikut adalah algoritme tanpa referensi dan caption, yang tidak diacu dalam teks:

Input: A set $C = \{c_1, c_2, \dots, c_r\}$ of denominations of coins, where $c_i > c_2 > \dots > c_r$ and a positive number n

Output: A list of coins d_1, d_2, \dots, d_k , such that $\sum_{i=1}^k d_i = n$ and k is minimized

```

C ← ∅
for i ← 1 to r do
    while n ≥ ci do
        C ← C ∪ {ci}
        n ← n - ci
    end
end
return C

```

KESIMPULAN DAN SARAN

Kesimpulan

Pada penelitian ini dapat disimpulkan bahwa:

1. Mesin pencari menggunakan Lucene pada dokumen RDF dapat dilakukan. Lucene tidak dapat secara langsung mengolah dokumen RDF karena dokumen RDF harus disimpan dan diolah menggunakan Sesame.
2. Hasil pencarian yang dilakukan menggunakan 29 kueri yang didapat dari penelitian Herawan (2011) menghasilkan nilai rata-rata *precision* yang baik yaitu 0.862 dan menggunakan kueri dengan *term boosting* menghasilkan nilai rata-rata *precision* 0.877.
3. Penambahan nilai *term boosting* menghasilkan nilai rata-rata *precision* 0.884.
4. Dokumen XML dapat dikonversi menjadi dokumen RDF. Agar dokumen RDF yang dihasilkan memiliki struktur yang jelas, maka dilakukan konversi secara manual.

Saran

Terdapat beberapa hal yang dapat ditambahkan atau diperbaiki untuk penelitian selanjutnya, yaitu:

1. Jumlah dokumen tanaman obat yang digunakan sebagai korpus diperbanyak lagi, agar pengukuran relevansi dapat dilakukan lebih jelas.
2. Menggunakan ontologi untuk dokumen RDF agar makna dari informasi pada dokumen RDF dapat lebih spesifik.

DAFTAR PUSTAKA

- [DEPTAN] Departemen Pertanian. 2009. "Pasokan dan Permintaan Tanaman Obat Indonesia Serta Arah Penelitian dan Pengembangannya" dalam: *Perspektif* 8 (1), pp. 52–64.
- Gutierrez, C, Hurtado, C.A, dan Vaisman, A. 2007. "Introducing Time into RDF" dalam: *IEEE Transactions On Knowledge And Data Engineering* 19 (2).
- Herawan, Yoga. 2011. "Ekstraksi Ciri Dokumen Tumbuhan Obat Menggunakan Chi-Kuadrat dengan Klasifikasi Naive Bayes". Skripsi. Departemen Ilmu Komputer, Institut Pertanian Bogor.
- Lei, Y, Uren, V, dan Motta, E. 2006. *SemSearch: A Search Engine for the Semantic Web*. Milton Keynes: The Open University.
- Manning, C.D, Raghavan, P, dan Schutze. 2008. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.
- Minack, Enrico *et al*. 2008. *The Sesame LuceneSail: RDF Queries with Full-text Search*.