
Mesin Pencari Cerdas dengan Web Semantik

Aswin Dwiono¹

¹Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
E-mail: ¹aswindwiyono@gmail.com

Abstrak

Perkembangan data (informasi) pada suatu web dapat berubah-ubah kontennya dengan cepat dan sangat sulit diikuti. Kondisi ini menjadi tantangan dalam pengembangan mesin pencari. Salah satu komponen penting untuk pengembangan mesin pencari adalah web crawler. Web crawler digunakan untuk melakukan penjelajahan dan pengambilan halaman-halaman web pada situs internet. Hasil pengumpulan situs web selanjutnya diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di internet. Penelitian ini menggunakan crawling Breadth First Search untuk menguji dan menelusuri setiap link pada halaman pertama kemudian menelusuri setiap link pada halaman berikutnya begitu seterusnya sampai setiap level pada link telah dikunjungi. Jenis data yang digunakan dalam proses crawling berupa file teks. Berdasarkan beberapa hasil pengujian terlihat bahwa web semantik dan crawling Breadth First Search dapat mengoptimalkan hasil dari mesin pencari.

Kata kunci: mesin pencari, web crawler, file teks, web semantik

Abstract

The increasing of data (information) on the internet is very difficult to follow, and information content of each website can change any time. This is the challenging in developing the search engine at this time. An important component in the development of search engine is web crawler. Web crawler is used to exploration and retrieval of web pages on the internet. The results of retrieval web pages then will be indexed by search engines; it is easier to find information on the internet. This research will present the application of web crawler search algorithm by using crawling Breadth First Search. Crawling Breadth First Search will explore and evaluate every link on the first page and then will explore every link on the next page until every level of the link has been visited. The kind of data used a text file. The results of several experiments showed that semantic web and Crawling Breadth First Search made the search engine more optimal and effective.

Keywords: search engine, web crawler, text file, semantic web.

1. Pendahuluan

Dewasa ini, teknologi internet telah berkembang dengan pesat sehingga internet menjadi salah satu kebutuhan sehari-hari bagi beberapa kalangan. Pengguna internet di seluruh dunia dapat saling mempublikasikan sumber daya yang mereka miliki di internet sehingga informasi tersebar dimana-mana. Untuk mendapatkan dan menyaring informasi yang dibutuhkan, pengguna internet dapat menggunakan aplikasi pencari yang telah tersedia. Aplikasi pencari tersebut melakukan pencarian berdasarkan kata

kunci yang dimasukkan oleh pengguna, selanjutnya mencocokkan kata kunci dengan isi konten yang tersebar di internet. Untuk mengambil informasi yang tersebar dan selalu berubah-ubah di internet diperlukan suatu *web crawler*. *Web crawler* berfungsi untuk melakukan penjelajahan dan pengambilan halaman-halaman web yang ada di internet. Hasil pengumpulan informasi situs web tersebut selanjutnya diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di internet [1].

Salah satu pendekatan dalam mengembangkan *web crawler* adalah menggunakan algoritma *crawling breadth first search*. Dengan algoritma ini, mesin pencari dapat melakukan penjadwalan dalam memperbaharui konten dan pengetahuannya di basis data. Berbagai riset mengenai tema pencarian pada web semantik telah banyak dilakukan, misalnya riset perkembangan data di internet ataupun sumber data elektronik tertentu yang sangat sulit diikuti, karena setiap saat informasi pada suatu website dapat berubah isinya [2]. Salah satu pendekatan yang dilakukan oleh [2] dalam mengatasi masalah tersebut adalah melakukan pencarian data dengan mesin pencarian berbasis semantik dengan sumber data yang terdefinisi dalam ontologi. Teknologi yang digunakan dalam ontologi antara lain, yaitu XML dan RDF.

Suatu metode pencarian berdasarkan asosiasi semantik berisi bagaimana cara menemukan informasi yang relevan dari permintaan pengguna di ontologi dan bagaimana menyediakan visualisasi yang sesuai dan metode navigasi pada hasil pencarian tersebut [3]. Pengguna dapat melakukan pencarian berdasarkan asosiasi semantik menggunakan permintaan pengguna dan menavigasi asosiasi antara sumber daya. Prosedurnya terdiri dari 2 langkah utama, yakni *querying* dan *navigation*.

Pada tahap *querying*, mereka menggunakan interrelasi antara konsep yang dipresentasikan dengan properti di ontologi untuk menemukan sumber daya yang terkait dengan kata kunci. Pada langkah *navigation*, mereka mengelompokkan hasil pencarian ke beberapa kategori berdasarkan sumber informasi dimana hasil pencarian berasal. Setelah itu, mereka merangkingkan hasil setiap kategori dengan 2 ukuran, yaitu jarak dan persamaan.

Berdasarkan kajian literatur mengenai web semantik telah memberikan strategi baru untuk meningkatkan efektivitas dalam pencarian dokumen di internet ataupun sumber data elektronik tertentu. Mesin pencari cerdas tersebut diharapkan dapat membantu pengguna internet dalam melakukan pencarian dokumen yang diperlukan.

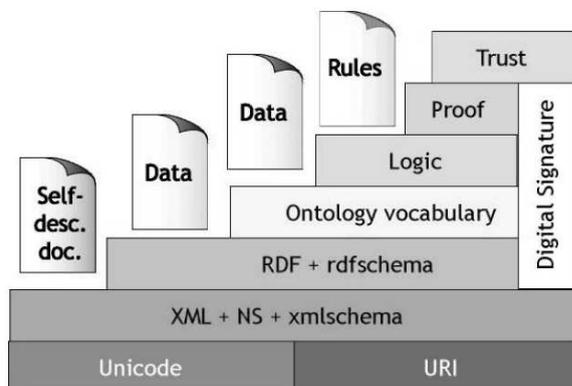
2. Web Semantik

Web semantik merupakan suatu kumpulan teknologi yang memungkinkan komputer dapat memahami arti dari suatu informasi berdasar pada metadata, yaitu informasi mengenai isi informasi tersebut. Adanya metadata, komputer diharapkan mampu secara otomatis membantu manusia mengartikan hasil proses informasi sehingga hasil pencarian informasi dapat lebih akurat [4].

Metode web semantik membantu terjadinya revolusi dalam hal penyampaian dan pemanfaatan informasi pada World Wide Web (WWW). Sebagai contoh, jika seseorang ingin menyusun informasi dari beberapa situs web sekaligus, maka dengan teknologi web yang sekarang, dia harus mengunjungi situs-situs tersebut satu persatu dan melakukan *cut* dan *paste* pada konten dari masing-masing situs untuk menciptakan suatu informasi yang menyeluruh. Hal ini sangatlah membuang waktu dan tenaga, karena dilakukan secara manual oleh manusia, dan dengan berbasis teknologi yang sudah ada tidak mungkin dibuat otomatisnya, mengingat halaman web berbasis HTML hanya dirancang untuk dipahami oleh manusia bukan mesin [4].

2.1 Komponen Web Semantik

Standar paling penting dalam membangun web semantik adalah XML, XML Schema, RDF, OWL, dan SPARQL [5]. Gambar 1 berikut ini menampilkan lapisan pada web semantik sebagaimana direkomendasikan oleh World Wide Web Consortium (W3C).



Gambar 1: Lapisan Web Semantik [6]

Gambar 1 dapat dijelaskan sebagai berikut [6]:

1. *Internationalized Resource Identified* (URI) merupakan penamaan yang unik untuk identifikasi web semantik;
2. *Unicode* mempresentasikan dan memanipulasi teks ke banyak bahasa;
3. XML sangat cocok dalam pengiriman dokumen melalui web;
4. RDF menyediakan sebuah pemaknaan sederhana untuk model data, yang dapat dipresentasikan dalam sintaks XML;
5. RDF Schema menyediakan dasar-dasar kosa kata untuk RDF yang memungkinkan untuk membuat hirarki kelas dan propertinya;
6. Ontologi *vocabulary* memperluas RDF Schema dengan menambahkan konsep yang lebih canggih untuk menambahkan sebuah batasan, seperti kardinalitas, batasan nilai, karakteristik dari properti seperti *transitive*. Ini didasarkan pada logika sehingga memberikan kekuatan *reasoning* pada web semantik;
7. *Logic* digunakan untuk meningkatkan bahasa ontologi lebih lanjut dan mengizinkan penulisan dari deklarasi pengetahuan;
8. *Proof* melibatkan proses pengurangan nyata seperti halnya penyajian bukti di bahasa web dan validasi bukti;
9. *Trust* bertujuan untuk memastikan dan memverifikasi bahwa pernyataan web semantik berasal dari sumber yang terpercaya. Ini dapat dicapai dengan tepat menggunakan ‘*digital signature*’ dari pernyataan RDF.

2.1.1 Extensible Markup Language (XML)

Extensible Markup Language (XML) adalah sebuah meta bahasa universal untuk mendefinisikan *markup*. Ini menyediakan sebuah kerangka kerja umum, dan didesain untuk menjadi sarana yang mudah dalam mengirimkan data dan metadata melalui Web [6]. Berbeda dengan HTML, XML memungkinkan penggunaanya untuk mendefinisikan *custom tag* atas keinginnya sendiri. Namun, standard XML tidak memiliki batasan semantik pada arti dari dokumen tersebut [5].

Elemen XML mempresentasikan “hal” yang ditulis dalam dokumen XML, menyusun konsep utama di dokumen XML. Elemen XML tersusun dari tag pembuka, isi konten, dan tag penutup. Contoh tag XML: <dosen>David Billington</dosen>

XML Schema merupakan bahasa yang digunakan untuk mendefinisikan sekumpulan aturan yang harus dipatuhi oleh dokumen XML. Struktur dokumen XML yang dibuat harus sesuai dengan aturan yang telah didefinisikan tersebut. XML Schema memperbolehkan seseorang untuk mendefinisikan tipe baru sebagai tambahan atau menggantikan tipe yang telah ada. Kombinasi dengan sintak XML, fitur dapat membangun aturan dari aturan lain tertentu dapat mengurangi beban kerja.

2.1.2 Resource Description Framework (RDF)

XML tidak menyediakan arti dan persamaan makna dalam data. Oleh karena itu, RDF-lah yang dapat menutupi kekurangan itu dalam web semantik [6]. RDF memberikan interoperabilitas antar aplikasi dalam mempertukarkan informasi yang *machine-understandable* di web. RDF memfasilitasi pemrosesan ter-otomatisasi terhadap sumber daya web. RDF dapat digunakan dalam berbagai aplikasi. Sebagai contoh, dalam *resource discovery* untuk memberikan kemampuan lebih baik pada *search engine*, dalam katalog untuk menjelaskan tentang konten dan hubungan konten [4]. Menurut [5], RDF dapat membuat pernyataan mengenai sebuah sumber daya web dalam bentuk ekspresi “Subjek-Predikat-Objek”. Dalam terminologi RDF, SPO ini seringkali disebut dengan istilah triple.

RDF didesain untuk mempresentasikan informasi dalam batasan yang minimal dan fleksibel. RDF memiliki kemiripan dengan pemodelan data semantik karena keduanya membicarakan tentang semantik. Pada pemodelan data semantik, terdapat sebuah subjek yang memiliki *value (object)* untuk predikat tertentu. Perbedaannya adalah RDF, memiliki hubungan M-to-N. Sementara pemodelan data semantik, hubungannya adalah N-to-1. Perbedaan kedua, pada RDF sebuah properti dapat menjadi sub-properti dari properti yang lain. Konsep seperti ini tidak diimplementasikan pada database, dan memungkinkan spesialisasi atas properti. Yang terakhir, RDF mengizinkan sebuah sumber daya diinstantiasikan oleh lebih dari satu kelas atau properti [4].

2.1.2.1 Tripel dan Graph

RDF merepresentasikan sebuah pernyataan sederhana tentang sumber daya dalam bentuk sebuah *graph* beserta node dan busur-nya. Sekumpulan *triple* disebut dengan RDF *graph*, dimana setiap *triple* merepresentasikan sebuah pernyataan hubungan antara hal-hal yang dinotasikan oleh node yang ditujunya [4].

Elemen dasar model RDF adalah *triple*: sebuah *resource* (sebagai *subject*) yang dihubungkan dengan *resource* yang lain (sebagai *object*) melalui sebuah *resource* ketiga (sebagai *predicate*). Dengan demikian dapat dikatakan bahwa sebuah *resource* <*subject*> memiliki *property* <*predicate*> dengan *value* <*object*> [4].

Sebuah node dapat memiliki sebuah URI dengan pilihan *fragment identifier (URI reference* atau *URIref)*, *literal* atau *blank*. Referensi URI atau *literal* digunakan sebagai sebuah node yang mengacu pada apa yang direpresentasikan oleh node tersebut. Referensi URI digunakan sebagai predikat yang mengidentifikasi hubungan antara sesuatu yang diwakili oleh node dan sesuatu dimana node tersebut terhubung. Sebuah predikat URI dapat juga berupa node dalam *graph* [4].

k

2.1.2.2 Literal

Dalam rekomendasi RDF, target sebuah *graph* dapat berupa potongan teks, potongan teks tersebut disebut literal. Hal ini secara sintak sangat berguna dalam RDF (dapat diisi teks daripada menempatkannya di sumber daya lain). Potongan teks mungkin mempunyai sebuah *fragment identifier*, maka sebuah literal node dapat digantikan dengan standard URI node dalam *graph* RDF [7].

2.1.3 Skema *Resource Description Framework* (RDF)

RDF adalah bahasa universal yang mengizinkan pengguna untuk menjelaskan sumber daya menggunakan kosakata mereka sendiri. Namun RDF tidak dapat mengasumsikan mengenai suatu domain tertentu, begitu juga dalam melakukan pendefinisian makna dari banyak domain. Kekurangan ini dapat dilakukan dengan skema RDF. Skema RDF dapat dipandang sebagai kamus data untuk mendeskripsikan properti dan kelas dari sumber daya RDF [5].

Mekanisme yang diadopsi dalam RDF untuk mengatur ekspresi tentang batas adalah dengan membuat *resources*, *properties*, *types* dan *statement* sebagai objek utama di dalam web. Artinya, ekspresi tersebut memiliki URI dan tidak dibatasi pada level dasar untuk dikombinasikan sedemikian rupa.

2.1.3.1 Class

Sumber daya dapat dikelompokkan menjadi grup-grup disebut kelas. Anggota kelas dikenal sebagai *instances* dari kelas. Kelas adalah sumber daya itu sendiri, kelas sering diidentifikasi dengan RDF *URI References* dan dapat dijelaskan menggunakan RDF *properties*. *Property* rdf:type dapat digunakan untuk menyatakan bahwa sebuah sumber daya adalah sebuah *instance* dari kelas [4].

RDF membedakan antara sebuah kelas dan sekumpulan *instance*-nya. Sesuatu yang berhubungan dengan setiap kelas adalah sebuah kumpulan yang disebut perluasan sebuah kelas. Dua buah kelas dapat memiliki sekumpulan *instance* yang sama namun menjadi kelas yang berbeda. Sebagai contoh, kantor pajak dapat mendefinisikan kelas orang yang hidup di alamat yang sama. Kantor pos dapat mendefinisikan kelas orang yang alamatnya memiliki kode pos yang sama. Hal ini sangat memungkinkan memiliki *instance* yang sama, namun memiliki properti yang berbeda.

2.1.3.2 Properties

RDF *property* adalah relasi antara *subject resource* dan *object resource*. Spesifikasi berikut menjelaskan tentang konsep *subproperty*. Rdfs:subPropertyOf dapat digunakan untuk menyatakan bahwa sebuah *property* adalah *subproperty* dari *property* yang lain. Jika *property* P adalah *subproperty* dari *property* P', maka semua pasangan *resource* yang dihubungkan dengan P berhubungan juga dengan P' [4].

2.1.4 *Ontology Web Language* (OWL)

Ontologi adalah sebuah spesifikasi dari sebuah konseptual [8], dengan kata lain ontologi adalah penjelasan dari sebuah konsep dan keterhubungannya dari sebuah ilmu tertentu. Ontologi merupakan suatu teori tentang makna dari suatu objek, properti dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada suatu domain pengetahuan [9]. Ontologi memudahkan dalam berbagi pengetahuan dan penggunaan

kembali yakni suatu pemahaman umum dari berbagai jenis konten yang menjembati aplikasi dan manusia [10].

Menurut [10] secara teknis, ontologi adalah suatu potongan pengetahuan secara teks, yang diletakan di web agar agent dapat berkonsultasi ke ontologi ketika diperlukan, dan mempresentasikan menggunakan sintak dari suatu penyajian bahasa ontologi. Menurut [5], ontologi sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berisi kelas, properti, dan *instance*. Deskripsi ini dapat membantu sistem komputer dalam menggunakan istilah-istilah tersebut dengan cara yang lebih mudah [4].

Awal tahun 2004 W3C resmi melepaskan OWL sebagai bahasa rekomendasi untuk mempresentasikan ontologi. OWL dikembangkan dimulai dari uraian logika dan DAM+OIL. OWL adalah satu set dari elemen dan atribut XML, mendeskripsikan arti dan relasi diantaranya [10]. OWL adalah suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi tersebut pada manusia, melainkan juga yang perlu memproses isi informasi isi [4]. Dengan menggunakan OWL, dapat menambah kosakata tambahan disamping semantik formal yang telah dibuat sebelumnya menggunakan XML, RDF, dan RDF *Schema* [5]. Hal ini sangat membantu penginterpretasian mesin yang lebih baik terhadap isi web.

2.1.5 SPARQL

SPARQL adalah sebuah bahasa *query* RDF, ini dapat digunakan untuk *query* banyak data RDF (termasuk pernyataan RDF dan OWL) [11]. Bahasa *query* diperlukan untuk merujuk untuk informasi dari aplikasi web semantik.

Menurut [5], *SPARQL Protocol and RDF Query Language* adalah sebuah protokol dan bahasa *query* untuk sumber daya web semantik. Sebuah *query* yang menggunakan SPARQL dapat terdiri atas *triple patterns*, konjungsi, dan disjungsi.

Hasil dari *query* SPARQL dapat mengembalikan nilai dalam beberapa format data, antara lain : XML, JSON, RDF, dan HTML [4]. Untuk menjalankan SPARQL, dapat menggunakan beberapa tools dan APIs seperti: ARQ, Rasqal, RDF::Query, twingql, Pellet, dan KAON2 [5].

Tools tersebut memiliki API yang memungkinkan pemrogram untuk memanipulasi hasil *query* dengan berbagai aplikasi yang ada. Namun, sebagai standar dapat menggunakan SPARQL Query Results XML Format [5] yang direkomendasikan oleh W3C.

2.2 Pencarian Konten dan Perangkingan

Penggambaran pencarian konten dan perangkingan ditunjukkan melalui arsitektur mesin pencarian dan analisis pencarian konten dan perangkingan berikut ini.

2.2.1 Arsitektur Mesin Pencarian

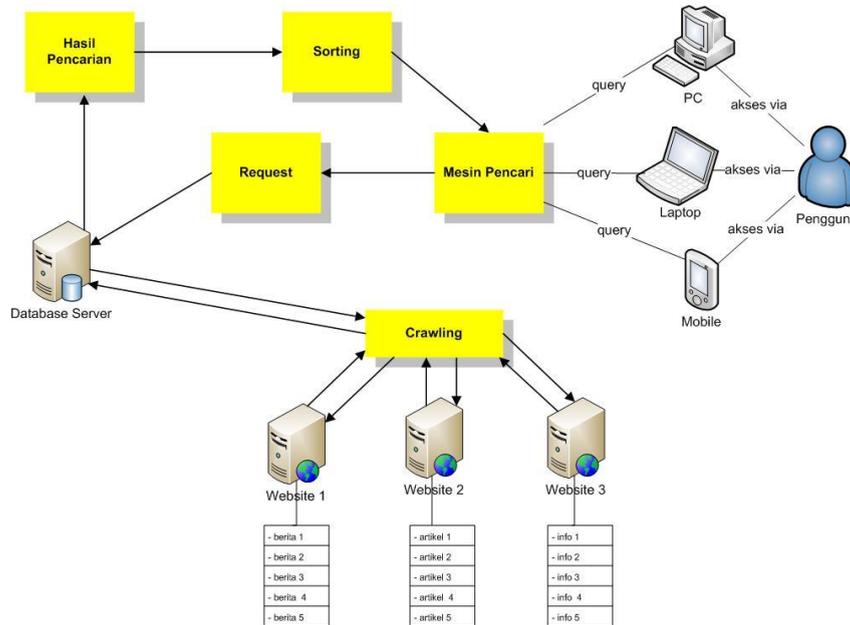
Arsitektur mesin pencarian yang digunakan ditampilkan pada Gambar 3 berikut ini.

2.2.2 Analisis Pencarian Konten dan Perangkingan

Tahapan pencarian konten dan perangkingan dilakukan sebagai berikut. Proses pencarian konten ini dilakukan dengan mencocokkan kata kunci yang dimasukkan oleh pengguna dengan isi konten hasil proses *crawler* yang telah disimpan di basis data.

Sedangkan perangkingan hasil proses pencarian dilakukan berdasarkan jumlah frekuensi kata kunci yang terkandung dalam judul konten dan isi konten. Jumlah frekuensi kata yang paling sering muncul ditampilkan pada halaman teratas.

Tahapan pencarian konten dan perangkingan dapat dijelaskan pada Gambar 4.



Gambar 3: Arsitektur Mesin Pencari

2.3 Web Crawler

Web crawler, *web spider* atau *web robot* merupakan salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama *web crawler* adalah melakukan penjelajahan dan pengambilan halaman-halaman web yang ada di internet. Hasil pengumpulan situs web selanjutnya diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di Internet [1].

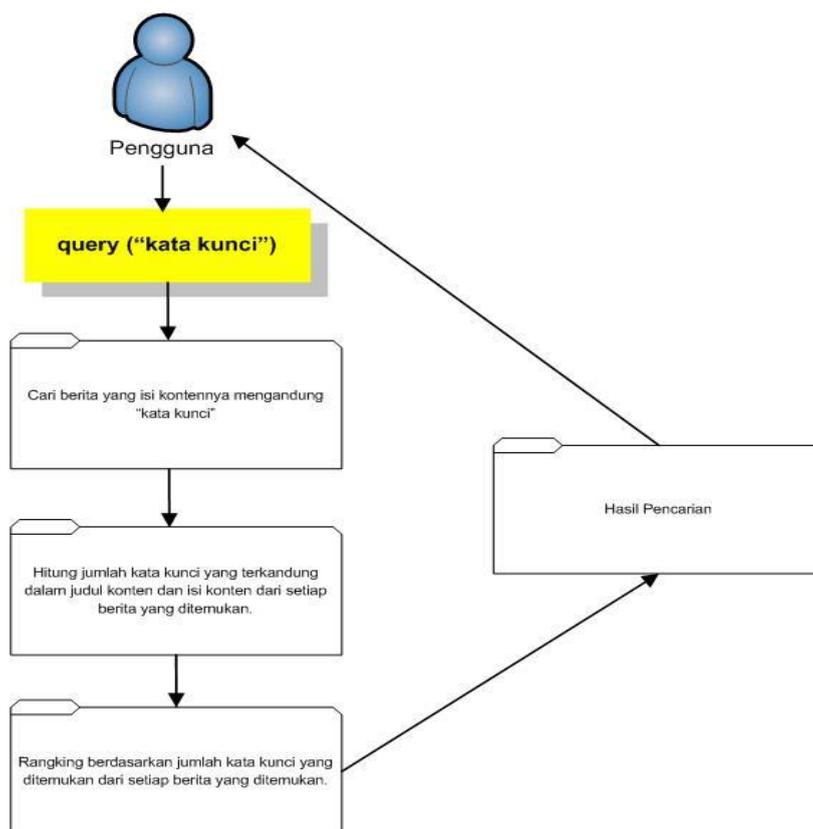
Penelitian ini menggunakan *web crawler* dengan skema *crawling breadth first search* dengan langkah sebagai berikut.

a). Mengunduh halaman web.

Langkah pertama, sebuah *web crawler* mengambil URL dan mengunduh halaman URL dari internet berdasarkan URL yang diberikan. Halaman yang diunduh disimpan dan diindeks di basis data mesin pencari. Namun, pada tulisan ini, file yang diunduh dan disimpan di basis data mesin pencari adalah file yang berupa file teks (*.txt).

b). Memparsing halaman yang diunduh dan mengambil semua link.

Web crawler memarsing dan mengolah keseluruhan halaman yang diunduh dan mengambil link-link ke halaman lain. Setiap link dalam halaman didefinisikan dengan sebuah penanda HTML dengan sintaks: ` Link`. Setelah *crawler* mengambil link dari halaman, setiap link ditambahkan ke sebuah daftar untuk di-*crawler*. Pada tulisan ini, link yang ditambahkan ke daftar link *crawler* adalah file HTML (*.html) dan file PHP (*.php).



Gambar 4: Proses Pencarian dan Perangkingan

c). Setiap link yang diambil, ulangi langkah 1.

Mengulangi proses karena sebuah *crawler* bekerja dengan rekursif atau bentuk perulangan.

Crawling Breadth First Search menguji setiap *link* pada sebuah halaman sebelum memproses ke halaman berikutnya. Jadi, algoritma ini menelusuri setiap *link* pada halaman pertama dan kemudian menelusuri setiap *link* pada halaman pertama pada link pertama dan begitu seterusnya sampai setiap level pada *link* telah dikunjungi. Kode algoritma *crawling breadth first search* diperlihatkan pada Gambar 5 [1].

3. Metode Penelitian

Langkah-langkah penelitian dilakukan sebagai berikut.

1. mengkaji landasan teori untuk memperkuat hipotesis, meliputi *semantic association-based search*, *web crawler* dan kebutuhan perangkat lunak;
2. mengumpulkan data atau objek penelitian;
3. menganalisis, mendesain, mengkodekan dan menguji perangkat lunak menggunakan metode *Rational Unified Process*;
4. menguji keakuratan perangkat lunak (sistem) yang dibangun;
5. menganalisis hasil pengujian sistem;
6. membahas terhadap hasil yang diperoleh dan membuat kesimpulan.

```
Input : starting_url: seed URL
Procedure :
[1] enqueue (url_queue, starting_url)
[2] while (not empty(url_queue))
[3] url = dequeue (url_queue)
[4] page = crawl_page(url);
[5] enqueue (crawled_page,(url,page))
[6] url_list = extract_urls(page)
[7] foreach u in url_list
[8]   enqueue (links,(url,u))
[9]   if (u ≠ url_queue and (u,-) ≠ crawled_pages)
[10]     enqueue (url_queue,u)
[11]   reorder_queue(url_queue)
```

Deskripsi fungsi :

enqueue (queue, element): menambah element di ujung dari queue

dequeue (queue): menghilangkan elemen di awal queue dan memberikan ke program yang memanggil

reorder_queue: mengurutkan queue menggunakan informasi yang ada ditautan.

Gambar 5: Algoritma *Crawling Breadth First Search*

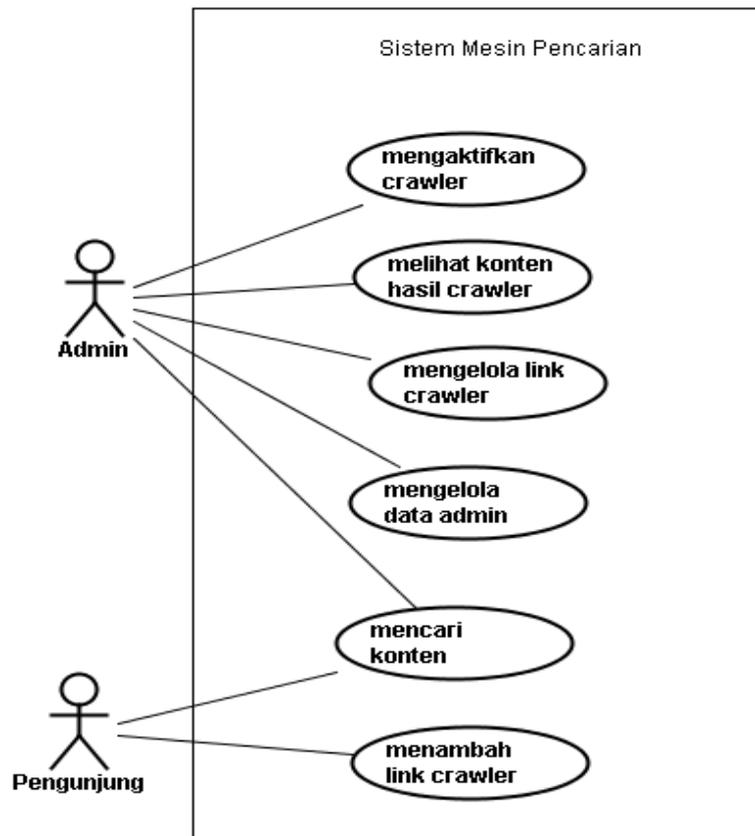
4. Analisis Perangkat Lunak

Perangkat lunak yang dikembangkan pada tulisan ini untuk memiliki fasilitas untuk: a) menambahkan URL untuk link *crawler* yang baru, b) melakukan proses pencarian dan menampilkan hasil pencarian dari kata kunci yang dimasukkan oleh pengguna, c) melihat data konten, d) menambah, mengubah, dan menghapus data *user*, dan e) melakukan proses *crawler* dari setiap URL yang tersimpan di basis data.

Perangkat lunak mesin pencarian yang dikembangkan menggunakan 2 pengguna, yaitu admin dan pengunjung. Adapun use-case digunakan sebanyak 6 buah seperti diperlihatkan pada Gambar 6.

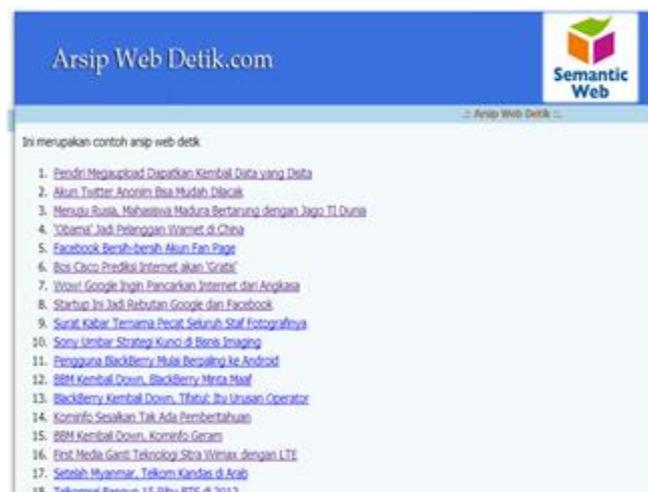
5. Hasil dan Pembahasan Pengujian Mesin Pencarian

Pengujian proses *crawler* menggunakan 2 buah komputer dengan rincian 1 buah komputer server dengan IP address 192.168.1.1 dan 1 buah komputer *client* dengan IP address 192.168.1.2. Komputer server menjalankan proses *crawler* untuk mengumpulkan informasi web yang ada di komputer *client*. Sebelumnya admin harus mencatat link-link URL yang di-*crawler* pada di komputer *client*, misalkan proses *crawler* pada halaman <http://192.168.1.2/webdetik/index.php>. Setelah disimpan di data *link crawler*, maka admin di komputer server membuka halaman mengaktifkan proses *crawler* seperti terlihat pada Gambar 7 – Gambar 9 berikut ini.



Gambar 6: Diagram *Use Case* Mesin Pencarian

Gambar 7 menunjukkan halaman web arsip berita yang dilakukan proses *crawler*. Halaman-halaman web yang di-*crawler* harus disimpan terlebih dahulu pada basis data komputer server seperti yang terlihat pada Gambar 8. Gambar 9 menunjukkan data konten hasil proses *crawler* yang sedang berjalan. Proses *crawler* akan terus berjalan dan mengecek apakah ada perubahan isi konten di setiap web yang telah *crawler*.



Gambar 7: Arsip Berita URL yang di-*crawler*

No.	Link	Status	Crawled	Aksi
1	http://192.168.1.2/webkompas/komputer.php		sudah	
2	http://192.168.1.2/webdetik/index.html		sudah	
3	http://192.168.1.2/webviva/1.html		sudah	

Gambar 8: Link URL di-crawler yang Dicatat di Komputer Server

Jika terjadi perubahan data pada isi konten web, maka *crawler* secara otomatis akan memperbaharui basis data isi konten mesin pencari sesuai dengan perubahan isi konten tersebut.

No.	Judul	Isi	File	Aksi
13	Rekor! Akun Twitter @SBYudhoyono Tembus 1 Juta Follower	Jakarta - Presiden Susilo Bambang Yudhoyono (SBY) meluncurkan akun Twitter @SBYudhoyono pada Sabtu (13/4) lalu. Sejak diluncurkan hingga saat ini, follower akun orang nomor satu di Indonesia itu sudah menembus angka 1 ...	http://192.168.1.2/webdetik/data28.txt	
14	XL Axiata Disukan Mau Akuisisi Axis	Jakarta - Axiata Group Bhd yang merupakan induk perusahaan XL Axiata dirumorkan tengah mempertimbangkan untuk mengakuisisi Axis Telekom Indonesia. Tidak disebutkan berapa nilai yang bakal disodorkan. Namun seperti ...	http://192.168.1.2/webdetik/data27.txt	
15	Isu Caplok Axis, XL: Kita Terbuka untuk Konsolidasi	Jakarta - Isu ketertarikan Axiata Group Bhd, induk perusahaan XL Axiata, terhadap Axis Telekom Indonesia bergulir panas. XL sendiri memastikan pihaknya selalu terbuka untuk peluang konsolidasi. Menurut Turina Farouk, ...	http://192.168.1.2/webdetik/data26.txt	
16	Juni, Telkomsel Tambah BTS di Perbatasan Malaysia	Samarinda - Operator selular Telkomsel menargetkan pengoperasian Base Transceiver Station (BTS) di wilayah Kabupaten Malinau, Kalimantan Utara, yang berbatasan dengan Malaysia, selamatnya bulan Juni 2013 ...	http://192.168.1.2/webdetik/data25.txt	

Gambar 9: Data Konten Hasil Proses Crawler

6. Kesimpulan

Berdasarkan hasil penggunaan perangkat lunak ini dapat disimpulkan bahwa:

1. Mesin pencari dapat menampilkan hasil pencarian yang sesuai dengan isi konten berdasarkan kata kunci yang dimasukan oleh pengguna, dan

2. Sistem *crawler* setelah diuji pada jaringan *local* berjalan dengan baik. Daftar website yang di-*crawler* berhasil dikunjungi dan diindeks di basis data mesin pencari.

Referensi

- [1] Sulatri, and E. Zuliarso, “Aplikasi *Web crawler* Berdasarkan *Breadth First Search* dan *Back-Link*”, *Jurnal Teknologi Informasi DINAMIK*. XV(1), 52-56, 2010.
- [2] H. Toba, “*Pencarian Cerdas dengan Penggunaan Semantic Web*”, *Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2005)*, Yogyakarta, 18 Juni 2005.
- [3] M. Lee, W. Kim, J. S. Hong, dan S. Park, “*Semantic Association-Based Search And Visualization Method On The Semantic Web Portal*”. *International Journal of Computer Networks & Communications (IJCNC)*, 2(1), 140-152, 2010.
- [4] M. Awaludin, “*Sistem Navigasi dan Pencarian Berbasis Konteks pada Konten E-learning Menggunakan Teknologi Web Semantik*” [skripsi]. Surabaya: Jurusan Teknik Informatika - Fakultas Teknologi Informasi-Institut Teknologi Surabaya, 2010.
- [5] N. Ibrahim, “*Pengembangan Aplikasi Semantic Web Untuk Membangun Web yang Lebih Cerdas*”, *Jurnal Informatika*, 3(1), 27-39, 2007.
- [6] G. Antoniou, and F. v. Harmelen, “*A Semantic Web Primer*”, Massachusetts: The MIT Press Cambridge, 2008.
- [7] D. O. Siahaan, “*Graphical Notations for Semantic Web Language*”, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Surabaya. Indonesia, 2006.
- [8] K. Triyantio, “*Perbandingan Tool Untuk Membangun Ontology Berbasis RDF/OWL dan Ilustrasi Implementasinya*” [skripsi]. Jakarta: Fakultas Ilmu Komputer, Universitas Gunadarma, 2006.
- [9] V. Lanin, and L. Lyadova, “*Intelligent Search and Automatic Document Classification and Cataloging Based On Ontology Approach*”, *International Journal Information Theories & Applications*, Vol.14, 2007.
- [10] F. F. M. Ghaleb, S. S. Daoud, A. M. Hasna, J. M. Jaam, dan H. F. El-Sofany, “*A Web-Based E-Learning System Using Semantic Web Framework*”, *Journal of Computer Science* 2 (8): 619-626, 2006.
- [11] R. Handayani, “*Penerapan Travel Planning dan Context-Aware Information Services Berbasis Web Semantik untuk Sistem Pariwisata di Indonesia*” [skripsi]. Surabaya: Jurusan Teknik Informatika, Fakultas Teknologi Informaski, Institut Teknologi Sepuluh Nopember, 2009.