PEMANFAATAN WEBSITE PARSER TEMPLATE PADA WEB CRAWLER UNTUK MEMBANGUN METADATA PADA SISTEM PENCARIAN BERBASIS SEMANTIK

Nurhayati Masthurah [1)], Taufiq Wirahman [2)], Devi Munandar [3)]
Pusat Penelitian Informatika Lembaga Ilmu Pengetahuan Indonesia[1,2,3)]
Kompleks LIPI Gedung 20 Lantai 3 Jl Sangkuriang No 21/154D Cisitu Bandung 40135
Telepon (022) 2504711 Faks (022) 2504712
E-mail : masthurah@informatika.lipi.go.id[1] , taufiq@informatika.lipi.go.id[2)], devi@informatika.lipi.go.id[3)]

Abstrak

Seiring dengan meningkatnya jumlah halaman Web, pencarian untuk menemukan informasi yang dibutuhkan menjadi semakin sulit. Untuk mengatasi hal tersebut, banyak program telah dibangun untuk mendapatkan halaman Web secara otomatis. Web crawler adalah suatu program perangkat lunak yang menjelajahi ruang informasi WWW dengan mengikuti tautan hypertext dan mengambil dokumen Web dengan standar protokol yang ada. Sistem pencarian berbasis semantik menggunakan metadata berupa Resource Description Framework (RDF) sebagai sumber informasinya. Web crawler digunakan untuk membuat penggandaan halaman Web yang dikunjungi sebelum akhirnya mesin pencari akan mengindeks halaman yang didownload untuk memberikan hasil pencarian yang lebih cepat. Dengan memanfaatkan format Website Parser Template (WPT) memungkinkan web crawler menghasilkan RDF Semantic Web untuk halaman Web. WPT terdiri dari beberapa bagian, yaitu ontologi, template dan URL. Ontologi berisi semua konsep dan hubungannya yang digunakan dalam website. Template dan URL nantinya akan dihubungkan ke ontologi website yang dibangun. Kumpulan RDF inilah yang akan digunakan sebagai repositori metadata dalam membangun Semantic web. Sebagai contoh aplikasi adalah sistem pencarian publikasi ilmiah berbasis semantik dimana sistem pencarian ini menggunakan data publikasi ilmiah berupa file bibtex yang didapat dari hasil crawling yang kemudian dikonversi ke format RDF untuk selanjutnya disimpan direpositori.

Kata Kunci : semantic web, web crawler, website parser template, rdf, pencarian semantik.

PENDAHULUAN

Web Crawler disebut web spider, web robot, atau dalam komunitas FOAF disebut web scutter adalah program atau script yang secara otomatis mencari World Wide Web dalam suatu metode atau cara yang otomatis [5]. Proses ini disebut web crawling atau spidering. Beberapa site, khususnya search engine menggunakan spidering untuk memberikan data yang up-to-date. Web crawler umumnya digunakan untuk membuat penggandaan dari halaman yang dikunjungi dalam pemrosesan terakhir oleh search engine yang akan mengindeks halaman yang di download untuk memberikan pencarian yang lebih cepat. Crawler juga dapat digunakan secara otomatis untuk memelihara task pada website seperti pengecekan link atau validasi kode html.

Web crawler adalah layanan web yang membantu user dalam pelayarannya di web dengan otomatis tugasnya telah terhubung. Membuat pencarian indeks web dan memenuhi pencarian queri dari indeks. Secara konseptual, Web crawler adalah node dalam web graph yang memiliki hubungan ke beberapa site di internet, singkatnya penghubung antara user dan tujuannya. Simplikasi dari penggunaan web adalah penting dalam beberapa alasan.

Pertama, Web Crawler menyimpan waktu user ketika melakukan pencarian sebagai ganti dari percobaan menerka penghubung dari halaman ke halaman. Bahkan, user akan melihat hubungan yang tidak jelas antara halaman yang ditampilkan dan halaman yang dicari. Sebagai contoh, akan ditampilkan halaman dalam satu topik dan meminta halaman pada topik komplit berbeda, salah satunya tidak terhubung dari lokasi yang ada. Dalam beberapa kasus, dengan melangkah ke Web Crawler selain menggunakan alamat ini atau tombol pada browser pencarian akan lebih mudah menentukan lokasi halaman tujuan. Penghematan adalah hal yang sangat penting dengan menambahkan ukuran dan cakupan Web [2].

Kedua, Simplikasi Web Crawler dari penggunaan web membuat web lebih mudah digunakan dan memiliki tool yang berguna. Navigasi web menggunakan pencarian kata kunci bahkan lebih intuitif daripada mencoba menggunakan Uniforn Resource Locator (URL) untuk mengidentifikasi halaman web seacara langsung. Jika user memiliki pengalaman yang baik maka akan lebih menggunakan web, dan mengulangi

penggunaan yang diikuti penambahan suatu medium. Search engine seperti web crawler mempunyai peranan dalam melanjutkan penyederhanaan dan pertumbuhan web.

Sehingga, web crawler berguna menyediakan beberapa konteks untuk pencarian queri tertentu. Dengan isu well-form queri, pencarian dapat menemukan informasi yang lebih luas mengenai topik tertentu dan dapat menggunakan informasi lebih jauh dalam memperbaiki tujuan ini. Pencari sering mengisukan queri besar yang disaring seperti yang lebih banyak dipelajari tentang subjek yang diharapkan.

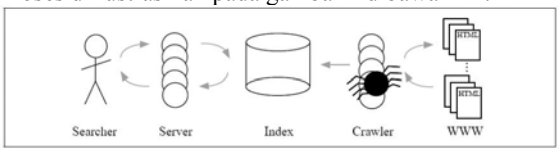
Sistem pencarian berbasis semantik menggunakan metadata berupa Resource Description Framework (RDF) sebagai sumber informasinya. Website Parser Template (WPT) XML berbasis open format yang menyediakan struktur HTML dari suatu halaman website. Format WPT memungkinkan web crawler untuk men-generate RDF semantic web pada suatu halaman web. Pada makalah ini dipaparkan pemanfaatan webcrawler yang disajikan berdasar website parser template untuk membangun repositori metadata pada sistem pencarian publikasi ilmiah berbasis semantik.

ARSITEKTUR WEBCRAWLER

Dalam implementasi web crawler, pada perspektif pengguna web khususnya web keseluruhan tidak terdapat elemen sisi client di luar browser yang dianggap perlu. Layanan ini terdiri atas 2 bagian penting, yaitu :

- crawling, proses menemukan dokumen dan konstruksi indeks.
- Serving, proses menerima queri dari pencari dan menggunakan indeks untuk menentukan hasil yang diinginkan.

Proses diilustrasikan pada gambar 1 dibawah ini:

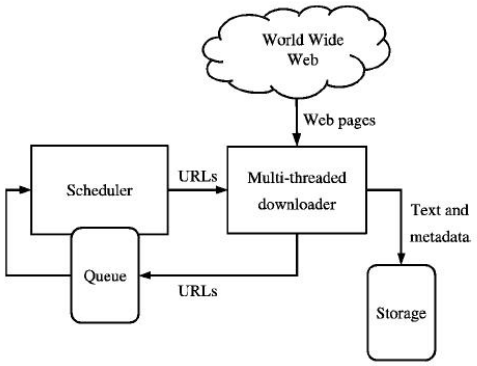


Gambar 1. Arsitektur WebCrawler [1]

Walaupun implementasi WebCrawler berkembang setiap waktu, komponen dasarnya tetap sama. Hanya cakupan, detil, dan waktu relatifnya mengalami perubahan. Crawler tidak hanya memiliki strategi crawling yang baik, tetapi juga harus memiliki arsitektur optimisasi yang tinggi.

Shkapenyuk dan Suel [3], memberi uraian bahwa : "Selama hal ini cukup mudah untuk membangun crawler lambat yang mendownload beberapa halaman per detik untuk periode waktu yang lebih singkat, membangun sistem performance tinggi dapat mendownload beribu-ribu miliar halaman selama beberapa minggu yang saat ini menjadi sejumlah tantangan dalam sistem desain, efisiensi I/O dan jaringan, ketahanan dan kemampuan mengelolanya.

Web crawler adalah bagian inti dari suatu search engine, dan detail dari algoritma dan arsitektur tetap menjadi rahasia bisnis. Ketika desain crawler di publish sering terdapat hal-hal penting karena tidak adanya detail yang mencegah hal lainnya dari reproduksi pekerjaan. Selalu timbul hal tentang "*search engine spamming*" yang mencegah search engine utama dari dipublikasikannya algoritma peringkatnya.



Gambar 2. Arsitektur Level Tinggi pada Web Crawler Standar [6]

WEBSITE PARSER TEMPLATE

Website Parser Template (WPT) XML berbasis open format yang menyediakan struktur HTML dari suatu halaman website. Format WPT memungkinkan web crawler untuk men-generate RDF semantic web pada suatu halaman web. WPT kompatibel dengan konsep semantic web yang di definisikan oleh W3C (RDF dan OWL) dan spesifikasi UNL.

SYNTAKS WPT

WPT terdiri dari berapa bagian :

- Ontologi, dimana publisher mendefinisikan konsep dan hubungan yang digunakan dalam website.
- Template, dimana publisher menyediakan template untuk sekelompok halaman web yang memiliki kategori isi dan struktur yang sama. Publisher memberikan elemen HTML Xpath atau TagID dan link dengan konsep website ontologi.
- URL, dimana publisher menyediakan pola URL yang mengumpulkan kelompok halaman web yang dihubungkan dengan "Parse Template". Pada bagian URL publisher dapat memisahkan bentuk URL sebagai bagian dari konsep dan hubungan ke website ontologi.

Website Parse Template dimulai dengan tag <icdl> dan diakhiri dengan tag </icdl>. WPT tunggal menunjuk ke host yang sama, dimana host tunggal memiliki beberapa WPT yang menggambarkan struktur HTMLnya. Hal ini diperlukan untuk menetapkan host WPT pada bagian awal tag <icdl>, sebagai contoh:

```
<icdl
host="http://informatika.lipi.go.id">
. . . . . .
</icdl>
```

ONTOLOGI WPT

Pada bagian ontologi berisi penyebutan satu persatu dan pendefinisian semua konsep yang digunakan dalam website. Urutan konsep harus dilampirkan dalam tag <ontology> </ontology>. Hal ini diperlukan untuk menentukan nama ontologi (setiap string rasional) dan mengindikasikan dukungan bahasa ("icdl:ontology", "owl", atau "unl:uws") yang digunakan untuk menetapkan konsep.

Contoh 1. Konsep yang digunakan dalam bibtex untuk object "article"

```
<ontology               name="general"
language="icdl:ontology">
     <concept name="article">
          <inherit
concept="label"></inherit>
          <has object="author"></has>
          <has object="title"></has>
          <has object="journal"></has>
          <has object="year"></has>
          <has object="id"></has>
          <has
object="fullname"></has>
     </concept>
     <concept name="Logo"></concept>
     <concept name="Menu"></concept>
     <concept          name="Content">
</concept>
     <concept           name="External
Link"></concept>
</ontology>
```

Setiap definisi konsep seharusnya dimulai dengan tag <concept> diakhiri tag </concept>. Tag <inherit> menunjukkan hubungan warisan (*inheritance*) dan tag <has> menunjukkan hubungan atribut diantara dua konsep. Salah satu dari konsep yang didefinisikan memiliki atribut default *object identifier* (id) yang digunakan oleh web crawler untuk mengkoordinasikan atribut objek yang sama dalam halaman yang berbeda dari suatu website yang sama.

WPT meramalkan beberapa konsep yang sudah dikenal yang umum digunakan untuk semua website, yaitu :

- "Menu", navigation bar / menu
- "Logo", elemen desain / logo
- "Content", elemen yang berisi teks utama dalam suatu halaman
- "Advertisement", advertisement / banner
- "External Link", elemen yang berisi link eksternal

TEMPLATE WPT

Pada bagian template berisi jumlah template untuk kelompok yang memiliki struktur halaman web yang sama. Salah satu dari template tersebut menunjuk ke kelompok tunggal yang memiliki struktur halaman web yang sama. Elemen HTML, referensi Xpath atau TagID digunakan untuk menghubungkan struktur isi dengan konsep yang digambarkan. Deskripsi template dimulai dengan tag <template> dan diakhiri dengan tag </template>. Dalam tag <template>, perlu untuk menspesifikasikan nama template dan bahasa yang digunakan untuk menggambarkan templatenya. Untuk nama template dapat dipilih setiap string, tapi untuk bahasanya diperlukan untuk mengindikasikan dukungan tipe language, misal "icdl:template", "rdf", atau unl:expression.

Contoh 2. Template sederhana untuk The Journal of Stuff.

```
<template name="The Journal of Stuff"
language="icdl:template">
  <html_tag tagid=" " content="Menu"/>
  <html_tag xpath=" " content="Logo"/>
  <html_tag          xpath="          "
content="Advertisement"/>
  <html_tag          xpath="          "
content="article.author"/>
  <html_tag       tagid="articletitle"
content="article.title"/>
  <html_tag            tagid="journal"
content="article.journal"
reference="Article Journal"/>
  <html_tag                    xpath=""
content="article.year"/>
  <html_tag                    xpath=""
content="article.author"/>
</template>
```

Halaman web berisi struktur isi yang berulang (<<repeater block>>) termasuk salah satunya elemen HTML utama (<<container>>) yang dispesifikasikan sebagai berikut :

Contoh 3. Representasi isi berulang

```
<template name=" The Journal of Stuff"
language="icdl:template">
    . . . . . .
     <container container_xpath=" ">
         <repeatable_block
block_xpath="">
         <html_tag            xpath=""
content="artist.journal"/>
         </repeatable_block>
     </container>
    . . . . . .
</template>
```

Dalam kasus spesifikasi kompleks element HTML sudah digambarkan oleh template lainnya tag

<reference> dapat digunakan untuk menunjuk ke blok templatenya. Hal ini memungkinkan untuk membuat hubungan hirarki antara template WPT sehingga webcrawler dapat menggunakan referensi spesifik untuk mengidentifikasi objek yang sama dalam halaman yang berbeda dari website yang diberikan.

Contoh 4. Hubungan hirarki antara template WPT

```
<template name=" The Journal of Stuff"
language="icdl:template">
     . . . . . .
     <html_tag          tagid="journal"
content="article.journal"
reference="Article Journal"/>
     . . . . . .
</template>
<template    name="Article    Journal"
language="icdl:template">
     <html_tag        xpath="        "
content="article.year"/>
     <html_tag   tagid="article_author"
content="article.author"/>
     <html_tag xpath=" "/>
</template>
```

BAGIAN URL

Bagian ini mendefinisikan URL atau pola URL yang sesuai dengan kelompok yang memiliki struktur halaman web sama yang diuraikan pada bagian template. Sesuai dengan template pada bagian URL juga terdiri dari beberapa blok dan blok lain dari itu seharusnya dimulai dengan tag <urls> dan diakhiri tag </urls>.

Contoh 5. Pola URL

```
<urls name="The Journal of Stuff"
template=" The Journal of Stuff ">
     <url
url="http://visus.mit.edu/webdav/www/K
anwisherLab.rdf#LiuJia"/>
     <url
url="http://visus.mit.edu/webdav/www/K
anwisherLab.rdf#ChrisBaker"/>
<url
url="http://visus.mit.edu/webdav/www/K
anwisherLab.rdf#NancyKanwisher"/>
</urls>
```

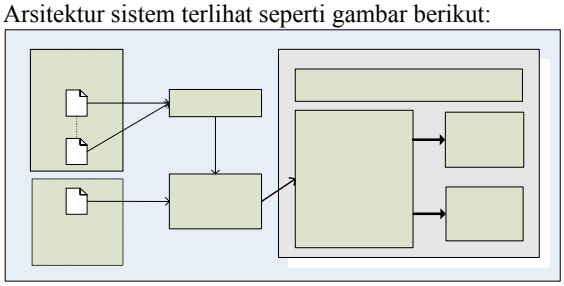
Nama blok URL dapat dipilih untuk setiap string, tetapi untuk template perlu untuk mengindikasikan nama template khusus. Pola URL diberikan pada contoh 5, yang merepresentasikan URL yang sebenarnya. Spesifikasi RegExp digunakan untuk menggambarkan pola URL. Konsep diperlukan untuk definisi pola URL.

IMPLEMENTASI

Dalam implementasi, dibangun suatu search engine dengan menggunakan WebCrawler yaitu sistem pencarian publikasi ilmiah berbasis semantik dengan menggunakan data publikasi ilmiah menggunakan bibtex yang didapat dari hasil crawling yang kemudian di konversi ke format RDF untuk disimpan di repositori.

Web crawler secara tipikal mengidentifikasi dirinya ke web server dengan menggunakan bagian user-agent dari HTTP request. Administrator website secara tipikal menguji log web server dan menggunakan bagian user agent untuk menentukan dimana crawler telah dikunjungi web server dan berapa frekuensinya. Bagian user agent dapat memasukkan URL dimana administrator web site dapat menemukan lebih banyak informasi mengenai crawler. Hal ini penting bagi web crawler untuk mengidentifikasi administrator web site yang dapat menghubungi pemiliknya jika dibutuhkan.

Data Bibtex diperoleh dari penyedia layanan data publik seperti CiteSeer , ACM , BibSonomy , Google Scholar dll serta dari data milik sendiri. Data-data tersebut tersedia dalam bentuk file teks dalam format .bib. Modul pemrosesan data akan mengubah data tersebut menjadi format RDF. Hasil pemrosesan kemudian disatukan di modul integrasi data dan data inilah yang nantinya digunakan di repositori metadata. Arsitektur sistem terlihat seperti gambar berikut:



Gambar 3. Arsitektur sistem untuk pemrosesan metadata

Gambar 4 adalah contoh aplikasi yang mengembangkan web crawler dalam bibtex

KESIMPULAN

WebCrawler menggabungkan metode full-text dengan metadata yang ada di web. Sehingga menghasilkan informasi yang diperoleh berdasarkan pencarian queri yang dilakukan. Dengan menggunakan WPT dengan struktur HTML memungkinkan web crawler untuk men-generate RDF semantic web pada suatu halaman web. Metadata RDF itulah yang kemudian disimpan dalam repositori mesin pencari berbasis semantik.

DAFTAR PUSTAKA

[1] Brain Pinkerton (2000). *WebCrawler:Finding what people want.* Doctor of Philosophy. University of Washington.

[2] Matrix Information and Directory Services, Inc. *Matrix.Net Home* http://www.mids.org/

[3] Shkapenyuk, V. and Suel, T. (2002). Design and implementation of a high performance distributed web crawler. In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357-368, San Jose, California. IEEE CS Press.

[4] http://en.wikipedia.org/wiki/Website_Parse_Template

[5] http://en.wikipedia.org/wiki/Web_crawler

[6] Castillo, C., (2004). "*Effective Web Crawling*"., Departement of Computer Science – University of Chile, November 2004.

Gambar 3. Implementasi