

# IMPLEMENTASI TEKNOLOGI WEB SEMANTIK DALAM APLIKASI PENCARIAN KATALOG *ONLINE* PERPUSTAKAAN UNIVERSITAS BINA DARMA

Andri

Dosen Universitas Bina Darma  
Jalan Jenderal Ahmad Yani No.12 Palembang  
Pos-el: andri@mail.binadarma.ac.id

---

**Abstract:** *The online catalog is a catalog of the library system that uses a computer, a catalog can provide a collection of bibliographic information. Catalogs are typically designed to simplify the user. In this research will be made a prototype online library catalog search application that utilizes semantic web technologies. Semantic web technologies allow describing objects and repositories in the form of ontology. Ontology represents semantic levels for knowledge ontology containing semantic entities instead of words. The application was built using JSP (Java Server Page) web based programming. Application of semantic web technologies in the study was able to provide results to the user with the information presented richer.*

**Keywords:** *Semantic Web, Catalog, JSP*

**Abstrak:** *Katalog online merupakan sistem katalog perpustakaan yang menggunakan komputer, sebuah katalog dapat memberikan informasi bibliografis suatu koleksi. Katalog biasanya dirancang untuk mempermudah pengguna. Dalam penelitian ini akan dibuat prototipe aplikasi pencarian katalog online perpustakaan yang memanfaatkan teknologi web semantik. Teknologi web semantik memungkinkan menggambarkan objek dan repositori dalam bentuk ontologi. Ontologi merepresentasikan pengetahuan pada level semantik karena ontologi berisikan entitas semantik sebagai pengganti kata. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman berbasis web JSP (Java Server Page). Penerapan teknologi web semantik pada penelitian ini mampu memberikan hasil kepada pengguna dengan informasi yang disajikan lebih kaya.*

**Kata kunci:** *Semantic Web, Katalog, JSP*

---

## 1. PENDAHULUAN

Aplikasi web dalam dunia pendidikan bermanfaat sebagai sarana yang digunakan oleh pihak perguruan tinggi untuk lebih meningkatkan mutu pendidikannya. Perpustakaan merupakan sarana utama dalam usaha pengembangan serta peningkatan pengetahuan bagi siapapun. Perpustakaan memegang peranan yang sangat besar dalam rangkaian penyebaran informasi. Perguruan tinggi sebagai suatu lembaga yang nantinya akan menghasilkan lulusan yang berkualitas, salah satu usaha yang dilakukan adalah memanfaatkan perpustakaan.

Web semantik merupakan teknologi baru dalam dunia internet, teknologi web semantik dapat diterapkan pada berbagai bidang kehidupan. Menurut Lee, dkk (2001) web semantik merupakan perluasan dari web saat ini, dimana informasi memiliki arti yang terdefinisi lebih baik, sehingga memungkinkan manusia dan komputer dapat bekerjasama lebih optimal dalam pengolahan dan penyajian informasi.

Teknologi web semantik menggunakan ontologi yang digunakan untuk mempresentasikan pengetahuan. Ontologi merupakan cara untuk merepresentasikan pengetahuan tentang makna objek, properti dari suatu objek, serta relasi antar objek tersebut yang

mungkin terjadi dalam domain pengetahuan (Chandrasekaran dan Josehson, 1999). Untuk membuat sebuah ontologi dapat menggunakan bahasa yang disebut dengan *OWL (Ontology Web Language)*.

Berdasar latar belakang, penulis mencoba untuk meneliti lebih jauh mengenai bagaimana merancang sebuah sistem pencarian katalog perpustakaan berbasis teknologi web semantik. Teknologi semantik memungkinkan menggambarkan objek dan repositori dalam bentuk ontologi. Ontologi merepresentasikan pengetahuan pada level semantik karena ontologi berisikan entitas semantik (*concept, relation* dan *instance*) sebagai pengganti kata. Selain itu ontologi memungkinkan untuk menspesifikasikan hubungan semantik antar entitas dan juga untuk menyimpan fakta dan aksioma tentang domain pengetahuan (Salton, 1993).

Sistem pencarian pada katalog *online* perpustakaan berbasis web semantik yang akan dikembangkan menyediakan fungsi pencarian koleksi perpustakaan berdasarkan informasi yang ada pada koleksi seperti judul, nama pengarang dan kata kunci yang berhubungan dengan koleksi yang tersedia.

Dalam sebuah aplikasi web semantik ontologi digunakan sebagai basis pengetahuan atau metadata yang menerapkan konsep semantik.

## 2. METODOLOGI PENELITIAN

### 2.1 *Semantic Web*

*Semantic web* merupakan perluasan dari web saat ini, dimana informasi memiliki arti yang terdefinisi secara lebih baik dengan mengupayakan persamaan persepsi antara konsep-konsep yang ada, sehingga memungkinkan manusia dan komputer untuk bekerjasama secara lebih optimal (Berners-Lee., 2001).

*W3C (World Wide Web Consortium)* memberikan suatu visi dari semantic web yaitu gagasan untuk memiliki data di web yang didefinisikan serta dihubungkan sedemikian rupa sehingga bisa digunakan oleh mesin, bukan hanya untuk ditampilkan tetapi juga untuk tujuan automasi, integrasi dan penggunaan kembali data antar berbagai aplikasi (*W3C*, 2001).

*Semantic web* mengindikasikan bahwa makna data pada web dapat dipahami, baik oleh manusia maupun oleh computer (Passin, 2004). Agar dapat diproses oleh mesin, dokumen web dianotasikan dengan metadata.

### 2.2 *Arsitektur Semantic Web*

Menurut *World Wide Web Consortium (W3C)*, arsitektur dari *semantic web* terdiri dari beberapa Layer yang ditunjukkan oleh gambar 1.

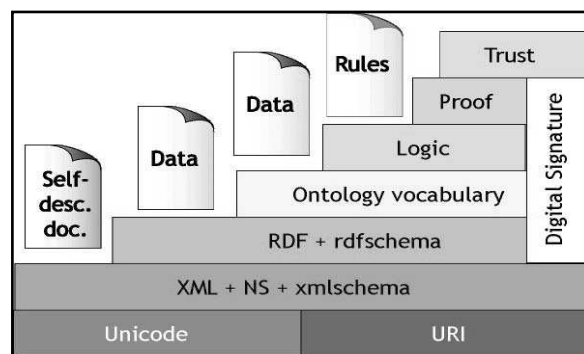
- 1) *Layer Unicode* dan *URI. Uniform Resource Identifiers (URI)* memastikan penggunaan sekumpulan karakter yang telah disepakati secara internasional dan menyediakan alat untuk mengidentifikasi obyek di semantic

web. Jenis *URI* yang terkenal adalah *URL* (*Uniform Resource Locator*) yang akan memberitahu komputer dimana letak suatu *resource*.

- 2) *Layer XML, Namespace, XML Schema*. Layer ini mengintegrasikan definisi *Semantic web* dengan dokumen *XML* (*Extensible Markup Language*) lain yang sesuai standar. *XML* merupakan format standar untuk dokumen terstruktur dan sebagai cara paling fleksibel untuk menciptakan standar bagi format informasi dan kemudian menyediakan format tersebut beserta datanya di web. *XML Schema* menggambarkan struktur dan batasan dari isi dokumen *XML*, *namespace* merupakan standar yang digunakan untuk menentukan label unik kepada sekumpulan nama elemen yang didefinisikan oleh *XML Schema*.
- 3) *RDF dan RDF Schema*. *RDF* (*Resource Description Framework*) dan *RDF Schema* memungkinkan pengguna untuk membuat pernyataan tentang obyek dan *URI* serta mendefinisikan kosakata yang bisa diacu dengan *URI* tersebut. Layer inilah yang menentukan tipe dari *resource* dan *link*. *RDF Schema* mendeklarasikan keberadaan kelas dan properti, termasuk subkelas, subproperti, domain dan range.
- 4) *Ontology vocabulary*. *Ontology* mendukung perkembangan kosakata karena pada *Layer RDF* dapat ditentukan relasi antar konsep yang berbeda.
- 5) *Logic*. *Layer logic* menyediakan *framework* untuk menulis aksioma dari aturan dasar sistem. Layer ini digunakan untuk meningkatkan bahasa ontologi dan

memungkinkan penulisan aplikasi pengetahuan deklaratif khusus.

- 6) *Proof*. *Layer proof* mengeksekusi aturan dari *Layer logic*. *Layer proof* melibatkan proses deduktif serta representasi proof dalam bahasa web dan validasi proof.
- 7) *Trust*. *Layer trust* mengevaluasi apakah hasil dari *Layer proof* bisa dipercaya. Layer ini akan muncul melalui penggunaan digital signature berdasarkan rekomendasi yang diberikan oleh agen yang terpercaya.
- 8) *Digital Signature*. *Digital signature* mendefinisikan blok dari data yang terenkripsi yang akan dimanfaatkan oleh komputer dan agen untuk memastikan apakah suatu informasi yang disediakan oleh sumber yang terpercaya serta mendeteksi adanya perubahan pada dokumen.



**Gambar 1. Arsitektur Semantic Web (Berners-Lee, 2001)**

### 2.3 Ontology

*Semantic web* memanfaatkan ontologi untuk merepresentasikan basis pengetahuan dan sumberdaya *web*. Ontologi menghubungkan simbol-simbol yang dipahami manusia dengan bentuknya yang dapat diproses oleh mesin,

dengan demikian ontologi menjadi jembatan antara manusia dan mesin (Davies dkk., 2006).

Ontologi bermanfaat untuk meningkatkan akurasi dalam proses pencarian informasi di web. Mesin pencari dapat mencari halaman yang merujuk pada konsep yang tepat dalam sebuah ontologi. Mesin pencari dapat menemukan dokumen yang relevan, mesin pencari dapat menyarankan pengguna untuk memberikan query yang lebih umum (Nurkhamid, 2009). Jika terlalu banyak dokumen yang ditemukan, mesin pencari dapat menyarankan *query* yang lebih spesifik (Antoniou & van Harmelen, 2008).

### 2.3.1 Konsep Ontologi

Antoniou dan Van Harmelen (2008), mengemukakan bahwa sebuah ontologi didefinisikan sebagai sebuah spesifikasi formal dan eksplisit dari sebuah konseptual. Makna konseptual merujuk pada model abstrak dari sesuatu hal. Eksplisit mengindikasikan bahwa elemen-elemen konseptual harus didefinisikan dengan jelas, dan formal berarti bahwa spesifikasi tersebut harus dapat diproses oleh mesin. Dalam pandangan Gruber ontologi merupakan representasi pengetahuan dari sebuah domain, dengan sekumpulan objek dan relasi dideskripsikan oleh sebuah *vocabulary*.

W3C menyebutkan bahwa ontologi adalah sebuah istilah yang diambil dari ilmu filsafat yang merujuk pada bidang ilmu yang mendeskripsikan berbagai entitas dalam dunia dan bagaimana entitas-entitas tersebut saling berelasi (McGuinness & van Harmelen, 2004). Ontologi menyediakan deskripsi untuk elemen kelas-kelas (*classes*) dalam berbagai domain, relasi (*relations*) antar kelas-kelas, dan properti

(*property*) yang dimiliki oleh kelas-kelas tersebut.

Ontologi digunakan untuk bidang kecerdasan buatan, representasi pengetahuan, pemrosesan bahasa alami, web semantik, rekayasa perangkat lunak, dan banyak bidang lainnya. Dalam sistem informasi, ontologi adalah spesifikasi yang jelas tentang serangkaian konsep yang menjelaskan sebuah wilayah pengetahuan tertentu yang dipakai bersama oleh para pengguna sistem yang bersangkutan.

### 2.3.2 Manfaat Ontologi

Beberapa manfaat menggunakan ontologi (Antoniou dan Van Harmelen, 2008), yaitu: 1) Ontologi dapat membagi pemahaman atau definisi tentang konsep-konsep dalam sebuah *domain* (*sharing* informasi); 2) Ontologi menyediakan cara untuk menggunakan kembali domain pengetahuan (*knowledge domain reusable*); 3) Ontologi membuat asumsi eksplisit sebuah *domain*; 4) Ontologi bersama dengan bahasa deskripsi (seperti *RDF Schema*), menyediakan cara untuk mengkodekan pengetahuan dan semantik seperti *machine-understand*; 5) Ontologi memungkinkan pemrosesan mesin otomatis dalam skala besar.

### 2.3.3 OWL (*Ontology Web Language*)

OWL (*Ontology Web Language*) merupakan suatu bahasa ontologi yang digunakan untuk mendeskripsikan kelas-kelas, properti-properti dan relasi antar objek-objek dalam suatu cara yang dapat diinterpretasi oleh mesin (Breitman dkk., 2007).

OWL merupakan sebuah *vocabulary* namun dengan tingkatan semantik yang lebih

tinggi dibandingkan dengan *RDF* dan *RDF Schema*. *OWL* menyediakan tiga sub bahasa yang berbeda tingkatan bahasanya yang dirancang untuk berbagai kebutuhan tertentu dari pengguna, antara lain (Breitman dkk., 2007):

- 1) *OWL Lite*: *OWL Lite* menyediakan pendefinisian hirarki kelas dan properti dengan batasan-batasan (*constraints*) yang sederhana. Jenis ini digunakan jika pengguna hanya membutuhkan hirarkis kelas yang sederhana dengan batasan yang sederhana pula.
- 2) *OWL DL (Description Logic)*: *OWL DL* mendukung pengguna yang menginginkan ekspresi maksimum tanpa kehilangan perhitungan yang lengkap dan ketepatan, *OWL DL* meliputi semua bahasa konstruksi dalam *OWL* dengan batasan tertentu. *OWL DL* dapat menghasilkan hirarkis klasifikasi secara otomatis dan mampu mengecek konsisten dalam suatu ontologi karena *OWL DL* mendukung reasoning.
- 3) *OWL Full*; *OWL Full* berguna untuk pengguna yang menginginkan ekspresi maksimum dan kebebasan sintaksis dari *RDF* tanpa ada jaminan perhitungan. *OWL Full* memperbolehkan ontologi untuk meningkatkan arti dari kosakata yang belum digambarkan (*RDF* atau *OWL*). *OWL Full* diperuntukkan bagi user yang menginginkan sub bahasa yang sangat ekspresif dan secara sintaks lepas dari *RDF* tanpa jaminan komputasional.

### 2.3.4 Struktur Dokumen OWL

- 1) *Namespace*. Sebelum kita dapat menggunakan kumpulan istilah (*term*) dalam

ontologi, kita perlu terlebih dahulu mendefinisikan secara tepat kosakata apa yang akan digunakan. Komponena awal dari standar ontologi adalah mendeklarasikan *namespace*. *Namespace* menyediakan sarana untuk pengidentifikasi dan membuat ontologi lebih mudah dibaca. Berikut ini contoh cuplikan *script OWL* untuk mendeklarasikan sebuah *namespace*

```
<rdf:RDF
  xmlns
    ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xmlns:vin
    ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xml:base
    ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"

  xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
  xmlns:owl
    ="http://www.w3.org/2002/07/owl#"
  "
  xmlns:rdf
    ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd
    ="http://www.w3.org/2001/XMLSchema#">
```

- 2) *Header*. Dokumen *OWL* biasanya disebut juga *OWL ontology*, memiliki elemen *root* berupa *tag owl: Ontology*. Berikut ini adalah contoh cuplikan *script header* dalam dokumen *OWL*:

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.w3.org/TR/2003/PR-owl-guide-20031215/wine"/>
  <owl:imports
    rdf:resource="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food"/>
  <rdfs:label>Wine
  Ontology</rdfs:label>
  ...
```

- 3) Elemen *Class*. *OWL* menyediakan konsep untuk mendefinisikan kelas beserta dengan batasan-batasan (*constraints*) dan aksiomanya (*axiom*). Kelas-kelas (*classes*) dalam dokumen *OWL* didefinisikan dengan menggunakan *owl:Class*. Setiap kelas dalam *OWL* merupakan subkelas dari *owl:Thing*. Berikut ini adalah cuplikan dari *script OWL* untuk mendefinisikan sebuah kelas dan subkelas dalam ontologi:

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Publication"/>

<owl:Class rdf:ID="Person">
  <rdfs:subClassOf
rdf:resource="#Thing" />
  ...
</owl:Class>
```

Bechhofer, dkk (2004) mengatakan kelas *OWL* pada prinsipnya sama dengan kelas yang didefinisikan di *RDF*. *OWL* menyediakan mekanisme untuk mengklasifikasikan sejumlah *resource* dengan karakteristik tertentu. Setiap kelas *OWL* yang didefinisikan dapat berhubungan dengan sejumlah individu yang disebut perluasan kelas (*class extension*).

### 2.3.5 *RDF*

*Resource Description Framework (RDF)* merupakan sebuah model data yang sederhana dan fleksibel untuk mendeskripsikan hubungan antara sumberdaya-sumberdaya web dalam bentuk *RDF statement* (Breitman dkk., 2007). *RDF* mendukung interoperabilitas antar aplikasi yang melakukan pertukaran informasi dan bersifat machine-understandable di web. *RDF* menggunakan graf untuk merepresentasikan

kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas. *RDF* didasarkan pada gagasan dimana hal-hal yang sedang diuraikan memiliki properti yang didalamnya mempunyai nilai-nilai dan *resource* yang dapat diuraikan dengan pembuatan statement (Manola dan Miller, 2004).

*RDF* menggunakan istilah tertentu untuk menguraikan suatu statement. Bagian yang mengidentifikasi dalam statement dapat disebut *subject*, karakteristik (*property*) dari *subject* disebut sebagai *predicate*, sedangkan nilai dari *property* disebut sebagai *object*.

Lassila dan Swick (1999) menyatakan model data *RDF* terdiri atas tiga objek tipe: 1) *Resource*, segala sesuatu yang digambarkan dengan *RDF* disebut *resource*. *Resource* bisa berupa keseluruhan atau bagian dari sebuah halaman web. *Resource* ini biasanya diberi nama menggunakan *URI (Uniform Resource Identifier)*. *URI* bersifat bisa diperluas maka *URI* bisa digunakan sebagai pengenalan bagi berbagai macam entitas; 2) *Properti (Property)*, properti merupakan aspek atau karakteristik, atribut, serta relasi khusus yang digunakan untuk menggambarkan sebuah *resource*. Setiap properti memiliki arti khusus, mendefinisikan nilai yang mungkin, tipe *resource* yang digambarkan dan relasinya dengan properti lain. *Pernyataan (Statement)*, suatu *resource* bersama dengan properti dan nilai dari suatu properti untuk *resource* membentuk suatu pernyataan *RDF*. Ketiga bagian ini disebut subjek, predikat dan obyek, yang membentuk *RDF triple*. Obyek dapat berupa *resource* lain, atau berupa literal

(string sederhana atau tipe data primitif lain yang didefinisikan oleh XML).

### 2.3.6 Perancangan Ontologi Koleksi Perpustakaan

Langkah awal dalam perancangan ontologi untuk koleksi perpustakaan adalah menentukan konsep dan domain yang akan dibangun

- 1) Penentuan Konsep dan Domain: Domain dari penelitian ini adalah dokumen atau koleksi digital perpustakaan Universitas Bina Darma Palembang.
- 2) Penentuan Daftar Terminologi: Tahap penentuan daftar terminologi sebelumnya dapat terlebih dahulu melihat kemungkinan penggunaan ontologi yang sudah ada. Pemanfaatan ontologi yang sudah ada, dapat mempercepat waktu pengembangan perangkat lunak tanpa harus memulai dari awal.
- 3) Definisi Properti Kelas (*Slot*).

Pada Tabel 1 didefinisikan property untuk setiap kelas Lecturer dalam domain ontologi koleksi perpustakaan, kelas *Lecturer* digunakan untuk menyimpan data dosen.

**Tabel 1. Properti Kelas Lecturer**

Properties	Tipe	Kardinalitas	Keterangan
employeeId	String	Single	Identitas dosen
email	String	Multiple	Email dosen
fullName	String	Single	Nama lengkap dosen

Kelas *Student* digunakan untuk menyimpan data-data mahasiswa. Tabel 2 menunjukkan beberapa properti yang dimiliki oleh kelas

*Student* diantaranya, *studentNumber*, *year*, *email* dan *fullName*.

Kelas *TextBook* digunakan untuk menyimpan koleksi perpustakaan berupa buku. Tabel 3 menunjukkan beberapa properti yang dimiliki oleh kelas buku diantaranya *author*, *edition*, *isbn*, *publisher*, *referTo*, *sinopsis*, *category* dan *title*.

**Tabel 2. Properti Kelas Student**

Nama	Tipe	Kardinalitas	Keterangan
Student Number	String	Single	Identitas mahasiswa
year	String	Single	Tahun Masuk mahasiswa
email	String	Multiple	Email mahasiswa
fullName	String	Single	Nama lengkap mahasiswa

**Tabel 3. Properti Kelas TextBook**

Nama	Tipe	Kardinalitas	Keterangan
Author (**)	String	Multiple person	Pencipta buku
edition	Date	Single	Tanggal/tahun penciptaan buku
Isbn	String	Single	Standar international nomor buku
publisher	String	Single	Badan yang mempublikasikan buku
referTo (**)	String	Multiple Publikasi	Rujukan ke sumber asal suatu buku
Synopsis	String	Single	Keterangan suatu isi dari buku
Title	String	Single	Judul dari buku
Category	String	Single	Kategori publikasi

(\*\*) Object properties

Kelas *FinalProject* digunakan untuk menyimpan koleksi dari tugas akhir mahasiswa diploma. Tabel 4 menunjukkan

beberapa properti yang dimiliki oleh kelas *FinalProject*.

Kelas *BachelorThesis* digunakan untuk menyimpan koleksi dari tugas akhir mahasiswa berupa skripsi. Tabel 5 menunjukkan beberapa properti yang dimiliki oleh kelas *BachelorThesis* diantaranya, *adviser*, *sinopsis*, *author*, *category*, *edition*, *publisher*, *referTo* dan *title*.

**Tabel 4. Properti Kelas *FinalProject***

Nama	Tipe	Kardinalitas	Keterangan
adviser(**)	String	Multiple Lecturer	Pembimbing tugas akhir
Abstrak	String	Single	Keterangan suatu isi dari tugas akhir
Author	String	Multiple Student	Pencipta tugas akhir
category	String	Single	Kategori publikasi
Edition	Date	Single	Tanggal/tahun penciptaan tugas akhir
publisher	String	Single	Perguruan tinggi/sekolah
referTo(**)	String	Multiple Publikasi	Rujukan ke sumber asal suatu tugas akhir
Title	String	Single	Judul dari tugas akhir

(\*\*) Object properties

**Tabel 5. Properti Kelas *BachelorThesis***

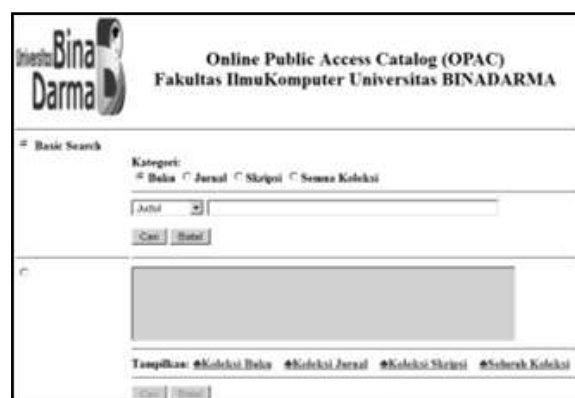
Nama	Tipe	Kardinalitas	Keterangan
adviser(**)	String	Multiple Lecturer	Pembimbing skripsi
sinopsis	String	Single	Keterangan suatu isi dari skripsi
Author	String	Multiple Student	Pencipta skripsi
category	String	Single	Kategori publikasi
Edition	Date	Single	Tanggal/tahun penciptaan skripsi
publisher	String	Single	Perguruan tinggi/sekolah
referTo(**)	String	Multiple	Rujukan ke

Title	String	Single	Publikasi sumber asal suatu skripsi Judul dari skripsi
(**) Object properties			

### 3. HASIL

Implementasi dari perancangan aplikasi dalam penelitian ini menggunakan bahasa Pemrograman *Java* dan *JSP (Java Server Page)*. Hasil penelitian ini adalah berupa prototipe aplikasi pencarian katalog perpustakaan berbasis teknologi web semantik. Antarmuka (*interface*) dalam aplikasi ini dibuat dalam bentuk sederhana. Sistem aplikasi ini menyediakan layanan pencarian *basic search* yaitu pencarian berdasarkan judul, pengarang dan kata kunci dari koleksi yang akan dicari dalam basis pengetahuan (ontologi).

Halaman index merupakan halaman utama yang akan ditampilkan pertama kali jika sistem aplikasi pencarian katalog perpustakaan ini dipanggil. Halaman index ini merupakan halaman antar muka dari sistem aplikasi pencarian katalog perpustakaan yang dikembangkan. Gambar 2 merepresentasikan bentuk tampilan halaman utama dari prototipe aplikasi yang dibuat.



**Gambar 2. Tampilan User Interface**



Dalam halaman index ini terdapat beberapa kategori pencarian koleksi yang disediakan yang terdiri dari: 1) Kategori pencarian berdasarkan buku; 2) Kategori pencarian berdasarkan jurnal; 3) Kategori pencarian berdasarkan skripsi; 4) Kategori pencarian berdasarkan semua koleksi.

Dalam aplikasi ini juga disediakan layanan pencarian koleksi yang ada pada masing-masing kategori dengan pencarian berdasarkan judul kategori, pengarang untuk setiap kategori, dan berdasarkan kata kunci setiap kategori.

Gambar 3 dan 4 menunjukkan contoh pencarian koleksi digital berdasarkan kata kunci dan hasil yang didapat dari proses pencarian saat tombol cari diklik oleh *user*.



**Gambar 3. Contoh Pencarian Berdasarkan Kata Kunci**



**Gambar 4. Contoh Hasil Pencarian Berdasarkan Kategori Buku**

#### 4. SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa pencarian koleksi perpustakaan yang menggunakan yang menerapkan teknologi web semantik dalam penelitian ini dapat memberikan hasil yang relevan, karena penerapan teknologi web semantik.

#### DAFTAR RUJUKAN

Antoniou, G., dan van Harmelen, F. 2008. *A Semantic web Primer*. MIT Press.

Berners-Lee. 2001. *The Semantic Web*. The Scientific American.

Breitman, K.K., Casanova, M.A., dan Truszkowski, W. 2007. *Semantic web: Concepts, Technologies and Applications*. Springer. London.

Chandrasekaran, B., Josehson, J. 1999. *What are Ontologies, and Why Do Who Need Them?*. Jurnal. IEEE Intelligent System, vol 14(1), hal 20-26.

Davies, J., Studer, R., dan Warren, P., 2006. *Semantic web Teknologies Trends and Research in Ontology-based Systems*. John Wiley & Sons. Chichester.

Lassila, O. dan Swick, R. R. 1999. *Resource Description Framework (RDF) Model and Syntax Specification, Recommendation*. Online. (World Wide Web Consortium (W3C): <http://www.org/TR/REC-rdf-syntax-19990222> Latest version available: <http://www.w3.org/TR/REC-rdf-syntax/>, diakses 10 September 2012).

Manola, F. dan Miller, E.. 2004. *RDF Primer, Recommendation, World Wide Web Consortium (W3C)*. Online. (<http://www.w3.org/TR/2004/REC-rdf->

primer-20040210/Lasterversion available:  
<http://www.w3.org/TR/rdf-primer/>,  
diakses 10 September 2012).

Passin, T.B. 2004. *Explorer's Guide the Semantic web*. Manning Publications. Greenwich.