



UNIVERSITAS INDONESIA

**IMPLEMENTASI ONTOLOGI WEB DAN APLIKASI SEMANTIK
UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA**

SKRIPSI

**NURIANA AYUNINGTYAS
0706199735**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK - UNIVERSITAS INDONESIA
DEPOK
JUNI 2009**



UNIVERSITAS INDONESIA

**IMPLEMENTASI ONTOLOGI WEB DAN APLIKASI SEMANTIK
UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA**

SKRIPSI

**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**NURIANA AYUNINGTYAS
0706199735**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK - UNIVERSITAS INDONESIA
JUNI 2009**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar**

Nama : Nuriana Ayuningtyas

NPM : 0706199735

Tanda Tangan :

Tanggal : 17 Juni 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Nuriana Ayuningtyas

NPM : 0706199735

Program Studi : Teknik Elektro

Judul Skripsi : Implementasi Ontologi Web dan Aplikasi Semantik untuk Sistem
Sitasi Jurnal Elektronik Indonesia

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir. Riri Fitri Sari M.Sc, MM ()

Penguji : Prof. Dr. Ir. Bagio Budiardjo M.Sc ()

Penguji : Dr. Ir. Kalamullah Ramli M. Eng ()

Ditetapkan di : Depok

Tanggal : 24 Juni 2009

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada :

Dr. Ir. Riri Fitri Sari M.Sc, MM, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juni 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Nuriana Ayuningtyas

NPM : 0706199735

Program Studi : Teknik Elektro

Departemen : Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-Exclusive Royalti-Free Right*)** atas karya ilmiah saya yang berjudul:

**IMPLEMENTASI ONTOLOGI WEB DAN APLIKASI SEMANTIK
UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 17 Juni 2009

Yang menyatakan

(Nuriana Ayuningtyas)

ABSTRAK

Nama : Nuriana Ayuningtyas
Program Studi : Teknik Elektro
Judul : Implementasi Ontologi Web dan Aplikasi Semantik untuk Sistem
Sitasi Jurnal Elektronik Indonesia.

Skripsi ini membangun ontologi *web* sebagai pengembangan ontologi dan aplikasi semantik. Ontologi adalah sebuah struktur hirarki yang menjelaskan sebuah domain yang digunakan sebagai landasan *knowledge base*. Komponen utama dari ontologi adalah *class*, *slot* dan *instance*. Sistem ini ditujukan untuk membangun sistem berbasis web pada aplikasi sitasi jurnal elektronik Indonesia. Tujuan dari aplikasi semantik adalah mengklasifikasikan informasi dari artikel-artikel pada jurnal elektronik dan menghubungkan dengan artikel yang diacunya. Dengan membuka halaman *web* dapat dilakukan pencarian judul dan penulis dalam beberapa kategori yang sudah dimodelkan dalam ontologi yaitu *Art and Design*, *Economy*, *Civil Engineering*, *Electrical Engineering*, *Industrial Engineering* dan *Information Science*. Pencarian tersebut juga akan melakukan hubungan antar artikel dan jumlah artikel yang diacu oleh artikel lainnya.

Implementasi ontologi menggunakan *tool* Protégé 3.4 sebagai editor ontologi dan *plug-in* TGVizTab untuk visualisasi hasil ontologi. Implementasi *web* menggunakan XAMPP yang terdiri dari *web server* Apache, PHP dan MySQL.

Pengujian dilakukan dengan pengujian fungsionalitas sistem pada *tool* Protégé 3.4 oleh *domain expert* atau pustakawan dan pengujian oleh pengguna web dalam melakukan pencarian sitasi jurnal elektronik. Pengujian dilakukan oleh sepuluh orang pengguna yang berperan sebagai pustakawan dan pengguna web dengan mengisi kuesioner. Dari hasil kuesioner pengujian *tool* Protégé 3.4, hasil yang didapat adalah sebagian besar tidak familiar dengan Protégé tetapi setelah diberikan pengarahan, pemakaian Protégé 3.4 sebagai *tool* pemodelan ontologi yang mudah dan efisien. Rata-rata nilai pada skala 2.94 dari skala 4. Dari hasil kuesioner pengujian *web*, hasil yang didapat adalah rata-rata memiliki nilai berskala 3 sehingga web melakukan fungsinya sebagai media pencarian, informasi yang ditampilkan bermanfaat dan sistem terbukti berjalan dengan baik.

Kata kunci :
Ontologi, Semantik, Protégé, Sitasi, XML.

ABSTRACT

Name : Nuriana Ayuningtyas
Study Program : Electrical Engineering
Title : Implementation of Web Ontology and Semantic
Application for Indonesian Electronic Journal Citation
System.

This final project is to create web ontology and semantic application for indonesia's electronic journal citation system. An ontology describes basic concepts in a domain and defines among them. Basic building blocks of ontology design include class, slot and instance. The aim of the semantic application is to classify information from an article in electronic journal and relate it with articles cited in it. Web allow user to search the relation among articles in electronic journal cited by particular article.

The system has been implemented by ontology tool i.e, Protege and web developed by PHP and MySQL. To entry the update, Protege should be installed and used by domain expert or librarian. The ontology model result is XML file. Librarian can submit the result through web based system to upload and parse XML file into database. User through the web based system could find the citation for an article they searched for.

Testing has been performed by ten users to be able to find out the effectiveness in case of the user for the domain expert or librarian and by the web users. The results are Protege 3.4 is an ontology editor that provides a suite of tools to construct domain models and web ontology is an user interface that provides information can work fine.

Keyword :
Ontology, semantic web, Protégé, citation, XML.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PENGESAHAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR SINGKATAN	xv
BAB 1. PENDAHULUAN	1
I.1. LATAR BELAKANG	1
I.2. TUJUAN	2
I.3. PERUMUSAN MASALAH	2
I.4. BATASAN MASALAH	2
I.5. METODOLOGI PENELITIAN	2
I.6. SISTEMATIKA PENULISAN	3
BAB 2. TEORI PENUNJANG	4
2.1. WEB	4
2.1.1. Perkembangan Web	6
2.2. INTEGRASI INFORMASI	7
2.3. ONTOLOGI	8
2.3.1. Pengertian Ontologi	8
2.3.2. Alasan Menggunakan Ontologi	8
2.3.3. Komponen Ontologi	9
2.3.4. Pembangunan Ontologi	12
2.4. SEMANTIK	15
2.4.1. Pengertian Semantik	15

2.4.2. Komponen-Komponen dalam Semantik Web	16
2.4.3. Keuntungan <i>Semantic</i>	18
2.5. TOOL ONTOLOGI	18
2.5.1. PROTEGE	18
2.5.2. Powl	20
2.5.2.1. Arsitektur Powl	20
2.5.2.2. Powl <i>Store</i>	21
2.5.2.3. RDFAPI	22
2.5.2.4. RDFSAPI	22
2.5.2.5. RDQL Query Builder	23
BAB 3. PERANCANGAN ONTOLOGI WEB DAN APLIKASI	24
SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK	
INDONESIA	
3.1. PEMODELAN RANCANGAN SISTEM	24
3.1.1. Persyaratan Sistem	24
3.1.2. Pemodelan Sistem	24
3.2. PERANCANGAN ONTOLOGI	27
3.3. PERANCANGAN PORTAL	31
3.4. DIAGRAM ALIR SISTEM	32
BAB 4. IMPLEMENTASI ONTOLOGI WEB DAN APLIKASI	35
SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK	
INDONESIA	
4.1. IMPLEMENTASI ONTOLOGI	35
4.1.1. Tahapan Implementasi Ontologi	35
4.1.2. Visualisasi Ontologi	49
4.2. IMPLEMENTASI WEB	52
4.2.1. Implementasi Parsing Model Ontologi	52
4.2.2. Implementasi <i>Domain Expert Interface</i>	56
4.2.3. Implementasi <i>User Interface</i>	56
BAB 5. PENGUJIAN ONTOLOGI WEB DAN APLIKASI	57
SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK	

INDONESIA	
5.1. PENGUJIAN FUNGSIONALITAS	57
5.1.1. Pengujian Fungsionalitas Sistem	57
5.1.1.1. Pengujian Fungsionalitas Sistem <i>Domain Expert</i>	57
5.1.1.2. Pengujian Fungsionalitas Sistem Pengguna	59
5.1.2. Pengujian Pengguna	63
5.1.2.1. Pengguna Sebagai <i>Domain Expert</i> atau pustakawan	63
5.1.2.2. Pengguna Yang Melakukan Pencarian	68
5.2. PENGEMBANGAN MASA DEPAN	69
BAB 6. KESIMPULAN	70
DAFTAR ACUAN	71

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Cara kerja web yang diakses	5
Gambar 2.2 Representasi <i>instance</i> atau individual	9
Gambar 2.3 Representasi <i>class</i> yang terdiri dari <i>instance</i>	10
Gambar 2.4 Contoh <i>Functional Properties</i>	10
Gambar 2.5 Contoh <i>Inverse Functional Properties</i>	11
Gambar 2.6 Contoh <i>Transitive Properties</i>	11
Gambar 2.7 Contoh <i>Symetric Properties</i>	12
Gambar 3.1 Skema Ontologi Web dan Aplikasi Semantik	25
Gambar 3.2 <i>Use case</i> diagram <i>e-journal</i>	26
Gambar 3.3 <i>Class Abstract</i>	28
Gambar 3.4 <i>Class Author</i>	28
Gambar 3.5 <i>Class Journal</i>	29
Gambar 3.6 <i>Class Title</i>	29
Gambar 3.7 <i>Class URL</i>	30
Gambar 3.8 <i>Class Volume</i>	30
Gambar 3.9 <i>Class Year</i>	30
Gambar 3.10 Rancangan halaman awal	31
Gambar 3.11 Diagram alir <i>domain expert</i> ke MySQL	32
Gambar 3.12 Diagram alir <i>user</i> melakukan pencarian	33
Gambar 4.1 Halaman Protege 3.4	35
Gambar 4.2 <i>Class</i> utama	37
Gambar 4.3 <i>Superclass</i> dan <i>subclassnya</i>	38
Gambar 4.4 <i>Superclass abstract</i> dan <i>subclassnya</i>	39
Gambar 4.5 <i>Superclass author</i> dan <i>subclassnya</i>	40
Gambar 4.6 <i>Superclass Journal</i> dan <i>subclassnya</i>	41
Gambar 4.7 <i>Superclass Title</i> dan <i>subclassnya</i>	42
Gambar 4.8 <i>Superclass URL</i> dan <i>subclassnya</i>	43
Gambar 4.9 <i>Slot</i> pada ontologi	44
Gambar 4.10 <i>Instance</i> pada <i>ElectricalAuthor</i>	48

Gambar 4.11 <i>Class Abstract</i>	49
Gambar 4.12 <i>Class Author</i>	49
Gambar 4.13 <i>Class Journal</i>	50
Gambar 4.14 <i>Class Title</i>	50
Gambar 4.15 <i>Class URL</i>	51
Gambar 4.16 <i>Class Volume</i>	51
Gambar 4.17 <i>Class URL</i>	52
Gambar 4.18 <i>Database dan Relasi Antar Tabel</i>	55
Gambar 5.1 Tampilan <i>Login</i> Sistem	58
Gambar 5.2 Tampilan <i>Upload</i> dan <i>Parsing</i> Ontologi	59
Gambar 5.3 Tampilan Halaman Pencarian	60
Gambar 5.4 Hasil Pencarian Kata “desain” pada Kategori <i>Art and Design</i>	61
Gambar 5.5 Hasil Pencarian Kata “david” pada Kategori <i>Civil Engineering</i>	62
Gambar 5.6 Hasil Sitasi <i>Alternate Pricing Strategies in Construction</i>	62
Gambar 5.7 Pencarian Tidak Ditemukan	63
Gambar 5.8 Grafik Tanggapan Penggunaan Protégé Sebagai <i>Tool</i> Ontologi.	65
Gambar 5.9 Grafik Tanggapan Penggunaan Web	69

DAFTAR TABEL

	Halaman
Tabel 2.1 Database pada Powl	21
Tabel 5.1 Hasil Pengujian Pengguna Sebagai Pustakawan	64
Tabel 5.2 Hasil Perhitungan Tanggapan Pengguna Sebagai Pustakawan	65
Tabel 5.3 Hasil Pengujian oleh Pengguna Web	67
Tabel 5.4 Hasil Perhitungan Tanggapan Pengguna Web	68

DAFTAR SINGKATAN

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
TCP/IP	Transmission Control Protocol / Internet Protocol
ASP	Active Server Pages
JSP	Java Server Pages
PHP	Personal Homepage Processor
CFM	ColdFussion
RDF	Resource Description Framework
OWL	Ontology Web Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
RSS	RDF Site Summary
FOAO	Friend of a Friend
SIOC	Semantically-Interlinked Online Communities
OKBC	Open Knowledge Base Connectivity Protocol
SWRL	Semantic Web Rule Language
API	Application Programming Interface
RDFS	RDF Schema
RDQL	RDF Data Query Language

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Web adalah suatu ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi oleh pengenalan global yang disebut *Uniform Resource Identifier* (URL). *Website* saat ini menjadi suatu kebutuhan bagi masyarakat modern baik digunakan untuk melakukan transaksi, penyebaran informasi maupun pencarian informasi. *Website* yang memiliki mesin pencari informasi seperti *google* kini menjadi alternatif utama dalam mencari informasi. Namun walaupun mesin pencari tersebut dapat memberikan informasi yang dibutuhkan, seringkali ketepatan dalam mencari informasi tersebut dipertanyakan. Hal ini sebenarnya merupakan fenomena yang wajar mengingat teknologi informasi di dunia internet belum memiliki mekanisme pengorganisasian secara teratur.

Ungkapan lain seperti *World Wide Web* yang menyerupai peta geografi yang kurang bagus.. Untuk memetakan daerah tersebut dengan lebih akurat, perangkat lunak membutuhkan deskripsi isi dan kemampuan dari *resource* yang diakses web yang dapat dibaca oleh mesin. Ontologi adalah konsep ideal untuk menggambarkan struktur dan semantik dari tipe dokumen. Visi untuk masa depan dimana informasi diberikan secara eksplisit yang membuat mesin atau komputer dapat mengerti dan mampu memproses informasi itu secara otomatis dan mampu mengintegrasikan informasi yang tersedia di web. Komputer diharapkan mampu melakukan proses penalaran (*reasoning*) sebagaimana yang dilakukan manusia.

Fungsi *website* sebagai mesin pencari seringkali digunakan oleh masyarakat modern untuk mencari literatur dalam pembuatan suatu karya ilmiah. Literatur tersebut dapat berbentuk artikel pada suatu jurnal yang nantinya akan menjadi acuan atau referensi terhadap riset yang sedang ditulis karena dari artikel yang dikutip memiliki informasi yang relevan. Semakin sering sebuah artikel dikutip, maka semakin besarlah artikel tersebut memberikan kontribusi informasi. Untuk mengetahui keterkaitan suatu artikel yang dijadikan referensi oleh artikel lainnya dapat dilakukan dengan menggunakan ontologi yang akan menggambarkan struktur dan semantik dari domain *e-journal*.

1.2 TUJUAN

Tujuan dalam penyusunan skripsi adalah implementasi Ontologi Web dan Aplikasi Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia dalam domain *electronic journal (e-journal)* sehingga dapat mengetahui keterkaitan suatu artikel dengan artikel lain yang dijadikan referensi oleh artikel tersebut.

1.3 PERUMUSAN MASALAH

Dalam penyusunan skripsi ini memiliki beberapa permasalahan yaitu mengumpulkan data berupa artikel-artikel dari jurnal yang saling memiliki keterkaitan dalam judul literatur yang dijadikan referensi, merancang ontologi dalam domain *e-journal*, membuat portal yang menjadi perantara bagi user untuk menyusun *query* ke dalam sistem dan menghubungkan portal dengan ontologi yang telah dibangun.

1.4 BATASAN MASALAH

Pembatasan masalah pada penyusunan skripsi ini adalah :

1. Ontologi yang dibangun berdasarkan domain *e-journal*. Artikel-artikel didapat dari jurnal universitas di internet.
2. Pemodelan ontologi dilakukan secara manual dengan menggunakan *tool* Protege 3.4.
3. Pemodelan dilakukan secara statis, setiap ada data yang baru akan ditambahkan ke struktur model yang telah dibuat.
4. Sistem berbasis web.
5. Bentuk pencarian berupa pencarian penulis dan judul artikel dari beberapa kategori. Dari hasil pencarian tersebut dapat dilihat keterkaitan antara penulis sumber dengan penulis yang mensitasi serta judul artikel sumber dengan judul artikel yang disitasi.

1.5 METODOLOGI PENELITIAN

Metode penulisan yang digunakan adalah sebagai berikut :

1. Studi Pustaka

Yaitu dengan membaca dan mempelajari bahan – bahan tentang web portal, *semantic*, ontologi dan rekayasa perangkat lunak.

2. Perancangan Sistem

Yaitu melakukan proses penggunaan berbagai teknik dan prinsip yang didapat dari studi pustaka untuk tujuan mendefinisikan proses atau sistem secara detail.

1.6 SISTEMATIKA PENULISAN

Bab 1, Pendahuluan yang menjelaskan latar belakang pemilihan tugas akhir, tujuan penulisan, perumusan masalah, batasan masalah, metodologi penulisan dan sistematika penulisan. Bab 2, Teori Penunjang yang menjelaskan konsep ontologi dan semantik berupa pengertian, komponen, kegunaan dan lain – lain. Konsep – konsep ini akan digunakan sebagai bahan acuan untuk merancang sistem. Bab 3, Perancangan Ontologi Web dan Aplikasi Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia yang menjelaskan mengenai operasi sistem secara keseluruhan. Bab 4, Implementasi Ontologi Web dan Aplikasi Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia yang menjelaskan mengenai proses implementasi ontologi, implementasi koneksi antara model ontologi dengan database dan implementasi web sebagai *user interface*. Bab 5, Pengujian Ontologi Web dan Aplikasi Semantik untuk Sistem Sitasi Jurnal Elektronik Indonesia yang menjelaskan pengujian fungsionalitas sistem yang dilakukan oleh *user* sebagai *domain expert* atau *librarian* dan *user* yang melakukan pencarian judul artikel dan penulis artikel. Bab 6, Kesimpulan dari seluruh penyusunan skripsi.

BAB 2

TEORI PENUNJANG

2.1. WEB

Web merupakan suatu ruang informasi di mana sumber-sumber daya yang berguna diidentifikasi oleh pengenalan global yang disebut *Uniform Resource Identifier* (URI). URL dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri atas [1] :

- Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi.
- Nama komputer (*server*) dimana informasi tersebut berada.
- Jalur atau *path* serta nama *file* dari suatu informasi.

Format umum dari URL adalah sebagai berikut :

Protokol_transfer://nama_host/path/nama_file.

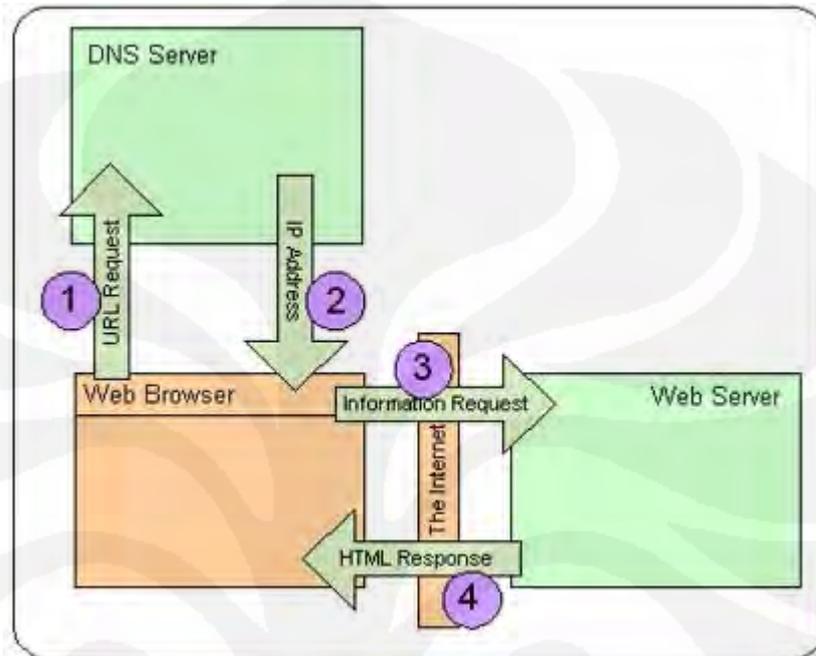
Contoh: <http://www.mine.com/e-journal/index.html>

Dari contoh tersebut dapat disimpulkan bahwa :

- http adalah protokol yang digunakan.
- www.mine.com adalah nama *host* atau *server* komputer dimana informasi yang dicari berada.
- *e-journal* adalah jalur atau *path* dari informasi yang dicari.
- Index.html adalah nama *file* dimana informasi tersebut berada.

Sebuah halaman web diakses dengan menggunakan web *browser* dengan menuliskan URL nya atau mengikuti *link* yang menuju kepadanya. *Uniform Resource Locator* (URL) akan menunjukkan lokasi dokumen yang dikelola oleh sebuah web *server*. URL diubah menjadi alamat IP *server* web yang bersangkutan. *Browser* kemudian mengirimkan request *Hypertext Transfer Protocol* (HTTP) ke web *server* dan web *server* akan menjawab dokumen yang diminta dalam format *Hypertext Markup Language* (HTML). HTTP adalah suatu protokol yang menentukan aturan yang perlu diikuti oleh web *browser* dalam meminta atau mengambil suatu dokumen dan oleh web *server* dalam menyediakan dokumen yang diminta web *browser*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses dokumen HTML. Dalam komunikasi jaringan komputer diatur dengan protokol yang memungkinkan

beragam jaringan komputer untuk berkomunikasi. Protokol ini secara resmi dikenal dengan *Transmission Control Protocol (TCP/IP)* yang merupakan cara untuk mempacketkan sinyal elektronik sehingga data tersebut dapat dikirim ke komputer lain. Gambar 2.1. berikut menggambarkan cara kerja web yang diakses.



Gambar 2.1. Cara kerja web yang diakses [16]

Web dapat dikategorikan menjadi dua yaitu web statis dan web dinamis atau interaktif. Web statis adalah web yang menampilkan informasi-informasi yang sifatnya statis atau tetap, sedangkan web dinamis adalah web yang menampilkan informasi serta dapat berinteraksi dengan user yang sifatnya dinamis. Untuk membuat web dinamis dibutuhkan pemrograman web yang mempunyai dua kategori, yaitu [1] :

- *Server – side Programming*
Perintah-perintah program atau *script* dijalankan di web *server*, kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML.
- *Client – side Programming*
Perintah program dijalankan di web *browser* sehingga ketika *client* meminta dokumen yang mengandung *script*, maka *script* tersebut akan *download* dari *server*nya kemudian dijalankan di *browser* yang bersangkutan.

Program web yang tergolong dalam server side seperti CGI/Perl. *Active Server Pages (ASP)*, *Java Server Pages (JSP)*, PHP dan *ColdFussion (CFM)*.

Sedangkan yang tergolong *client side* seperti Javascript, VBScript dan HTML. Teknologi *server side* yang dilakukan dalam aplikasi ini adalah PHP. PHP termasuk dalam produk *open source* sehingga *source code* dapat diubah dan didistribusikan secara bebas. PHP dapat berjalan diberbagai web *server* misalnya IIS, Apache dan PWS. Adapun kelebihan-kelebihan PHP adalah sebagai berikut :

- PHP mudah dibuat dengan kecepatan akses tinggi.
- PHP dapat berjalan dalam web *server* yang berbeda dan dalam sistem operasi yang berbeda pula.
- PHP diterbitkan secara gratis.
- PHP adalah termasuk bahasa yang *embedded* (bisa diletakkan dalam tag HTML).

2.1.1. Perkembangan Web

Dalam perkembangan teknologi web, banyak praktisi yang memberi label perkembangan web dengan Web 1.0, Web 2.0 dan Web 3.0. Sebenarnya tidak ada kesepakatan adanya versi dalam aplikasi web tetapi hanya untuk memudahkan perkembangannya saja.

- Web 1.0

Web 1.0 memiliki sifat yang sedikit interaktif dan dikembangkan untuk pengaksesan informasi. Sifat dari Web 1.0 adalah *read* karena *user* hanya akan membaca informasi yang ditampilkan web.

- Web 2.0

Menurut Tim O'Really, Web 2.0 dapat didefinisikan sebagai berikut : "Web 2.0 adalah revolusi bisnis di industri komputer yang disebabkan oleh penggunaan internet sebagai *platform* dan merupakan suatu percobaan untuk memahami berbagai aturan untuk mencapai keberhasilan pada *platform* baru tersebut. Salah satu aturan utama adalah membangun aplikasi yang mengeksplorasi efek jaringan untuk mendapatkan lebih banyak lagi pengguna aplikasi tersebut" [2].

Kemudahan interaksi antara *user* dengan sistem merupakan tujuan dibangunnya teknologi web 2.0. Sifat dari web 2.0 adalah *read* dan *write*. Perkembangan web 2.0 lebih menekankan pada perubahan cara berpikir dalam

menyajikan konten dan tampilan di dalam sebuah *website*. Web 2.0 diaplikasikan sebagai bentuk penyajian halaman web yang bersifat sebagai program dekstop pada umumnya seperti windows. Implementasi dapat dilihat pada aplikasi *spreadsheet* pada Google yang merupakan aplikasi untuk operasi mengolah angka seperti MS Excel. Aplikasi tersebut dapat diakses secara online tanpa user harus menginstalnya terlebih dahulu. Web 2.0 pada umumnya adalah suatu teknologi yang gratis atau lebih dikenal dengan sebutan *Open Source* dan sangat memudahkan untuk *share*, *upload* dan *download* data.

- Web 3.0

Web 3.0 adalah generasi ketiga dari layanan Internet berbasis web. Menurut Tim Berners Lee, Web 3.0 sebagai sebuah sarana bagi mesin untuk membaca halaman-halaman web [2]. Hal ini berarti bahwa mesin akan memiliki kemampuan yang sama dengan manusia dalam membaca web. Web 3.0 berhubungan dengan konsep web semantik yang memungkinkan manusia dapat berkomunikasi dengan mesin pencari yang juga mampu menyediakan keterangan – keterangan yang relevan tentang informasi yang dicari. Web 3.0 memiliki beberapa standar operasional agar dapat menjalankan fungsinya dalam menampung metadata, yaitu RDF (*Resource Description Framework*), dan OWL (*Ontology Web language*).

Dalam perkembangan web harus diimbangi dengan kecepatan untuk mengakses karena faktor yang menentukan kinerja aplikasi adalah kecepatan akses jaringan dan Internet. Oleh karena itu diperlukan *bandwidth* yang cukup dalam menjalankan suatu aplikasi.

2.2. INTEGRASI INFORMASI

Saat ini sangat banyak informasi yang tersedia *online* dan seringkali informasi tersebut tidak terstruktur. Integrasi informasi merupakan *database* komunitas untuk menyediakan informasi yang dibutuhkan dari beberapa sumber *online*. Tujuan integrasi informasi adalah menyediakan sumber data, membuat sebuah *interface* atau skema untuk mengintegrasikan data serta menentukan jawaban atas suatu pertanyaan (*query*) yang diajukan oleh *user*. Hal yang harus

diperhatikan juga adalah bagaimana mendapatkan informasi yang konsisten dan menentukan aliran informasi tersebut secara semantik atau memiliki silsilah.

Salah satu cara untuk membuat informasi secara semantik adalah dengan menggunakan ontologi yang dapat menggambarkan struktur data sehingga dapat diketahui garis keturunannya.

2.3. ONTOLOGI

2.3.1. Pengertian Ontologi

Dalam literatur kecerdasan buatan terdapat beberapa pengertian ontologi. Ontologi adalah istilah yang dipinjam dari filosofi yang mengacu kepada ilmu untuk menggambarkan jenis-jenis entitas di dunia dan bagaimana mereka berhubungan. Menurut Barnaras pada proyek Kactus [3] memberikan definisi ontologi yaitu : "Penjelasan secara eksplisit dari konsep terhadap representasi pengetahuan pada *knowledge base*". Proyek Sensus [3] juga memberikan definisi : "Sebuah ontologi adalah sebuah struktur hirarki dari istilah untuk menjelaskan sebuah domain yang dapat digunakan sebagai landasan untuk sebuah *knowledge base*"[4]. Pengertian lain mengemukakan ontologi adalah sebuah uraian formal yang menjelaskan tentang sebuah konsep dalam suatu domain tertentu (*classes*, terkadang disebut *concepts*), properti dari setiap konsep yang menjelaskan bermacam-macam fitur dan atribut sebuah *concepts* (*slots*, terkadang disebut *roles* atau *properties*) dan batasan pada slots (*facets*, terkadang disebut *role restriction*). Sebuah ontologi bersama dengan seperangkat *instances* (menyatakan objek pada suatu domain) dari *class* membentuk sebuah *knowledge base*. Pendapat lain mengatakan bahwa ontologi terbentuk oleh 4 *tuple* (C,R,I,A), C adalah *concept*, R adalah *relation*, I adalah *instance*, dan A adalah *axiom*. *Axiom* digunakan untuk menyediakan informasi mengenai *class* dan *properties*, sebagai contoh batasan pada *properties*.

2.3.2. Alasan Menggunakan Ontologi

Terdapat beberapa alasan untuk menggunakan ontologi, yaitu [4] :

- Menjelaskan suatu domain secara eksplisit.

Memberikan struktur hirarki dari istilah untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.

- Berbagi pemahaman dari informasi yang terstruktur.

Sebagai contoh beberapa *website* yang berbeda mempunyai informasi medis atau menyediakan servis medis *e-commerce*. Jika *website* tersebut dipakai bersama dan dipublikasikan dengan dasar ontologi yang sama maka perangkat lunak dapat mengekstrak dan mengumpulkan informasi dari *site* yang berbeda. Perangkat lunak tersebut dapat menggunakan kumpulan informasi tersebut untuk menjawab permintaan *user* atau sebagai data input untuk aplikasi lainnya.

- Penggunaan ulang domain pengetahuan.

Apabila ingin membangun ontologi yang luas dapat mengintegrasikan dengan beberapa ontologi yang sudah ada.

2.3.3. Komponen Ontologi

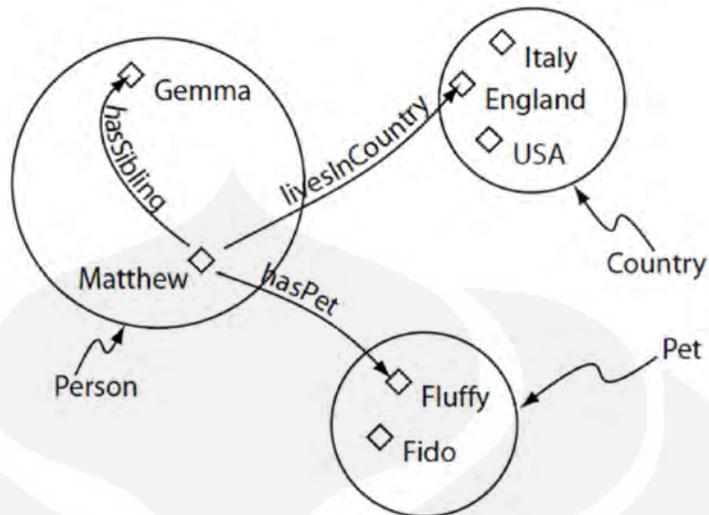
Ontologi memiliki beberapa komponen yang dapat menjelaskan ontologi tersebut, yaitu :

- **Instance** atau **individual** digunakan untuk merepresentasikan elemen pada suatu domain. Contoh *instance* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Representasi *instance* atau individual [5]

- **Class** merupakan titik pusat ontologi. **Class** menjelaskan sebuah konsep dalam suatu domain yang terdiri dari beberapa *instance* atau individual. **Class** juga dikenal sebagai **concept**, **object** dan **categories**. Sebuah **class** memiliki **subclasses** yang menyatakan **concept** yang lebih spesifik dari **superclass**. Contoh class dapat dilihat pada Gambar 2.3.



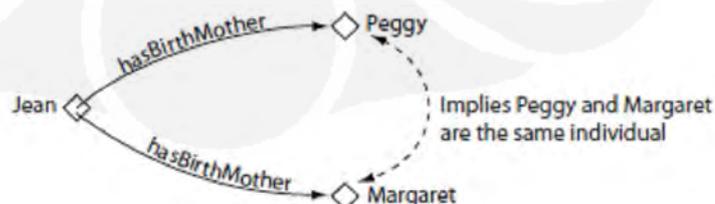
Gambar 2.3. Representasi *class* yang terdiri dari *instance* [5]

Dari gambar tersebut dapat dilihat bahwa *class* **Person** memiliki *instance* atau *individual* Gemma dan Matthew.

- **Properties** atau *slot*. *Properties* atau *slot* terdiri dari dua jenis, yaitu *object properties* dan *datatype properties*. *Object properties* akan menghubungkan *instance* dengan *instance* sedangkan *datatype properties* akan menghubungkan *instance* dengan *datatype value* seperti *text string* atau *number*. Pada *object properties*, terdapat beberapa jenis *properties* [5], yaitu :

- *Functional Properties*

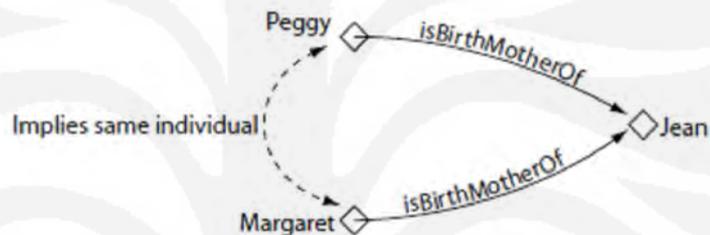
Functional properties adalah sebuah individu yang berhubungan hanya dengan satu individu. *Functional property* disebut juga sebagai *single valued property* atau *feature*. Sebagai contoh pada Gambar 2.4 menunjukkan individu **Jean** *hasBirthMother* **Peggy** dan individu **Jean** *hasBirthMother* **Margaret**, *hasBirthMother* merupakan *functional property*, dapat disimpulkan bahwa Peggy dan Margaret adalah individu yang sama karena Jean hanya memiliki satu *BirthMother*.



Gambar 2.4. Contoh *Functional Properties* [5]

○ *Inverse Functional Properties*

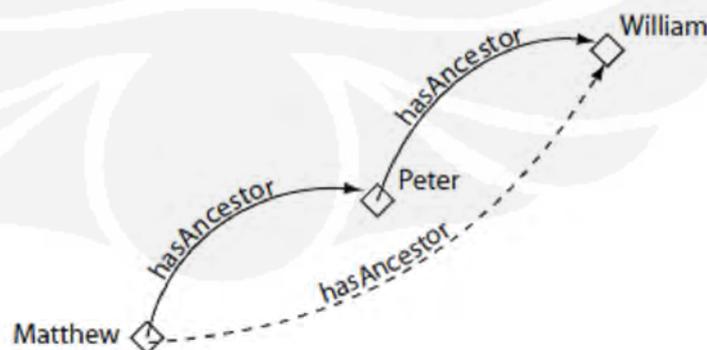
Jika sebuah properti adalah *inverse functional*, maka *inverse property* tersebut adalah *functional*, sebuah individu berhubungan hanya dengan satu individu. Sebagai contoh pada Gambar 2.5 menunjukkan *inverse functional property* **isBirthMotherOf** yang merupakan *inverse property* dari **hasBirthMother**, **hasBirthMother** adalah *functional*, **isBirthMother** adalah *inverse functional*. Jika dikatakan bahwa Peggy merupakan *birth mother* dari Jean, dapat dikatakan juga bahwa Margaret merupakan *birth mother* dari Jean. Sehingga dapat disimpulkan bahwa Peggy dan Margaret adalah individu yang sama.



Gambar 2.5. Contoh *Inverse Functional Properties* [5]

○ *Transitive Properties*

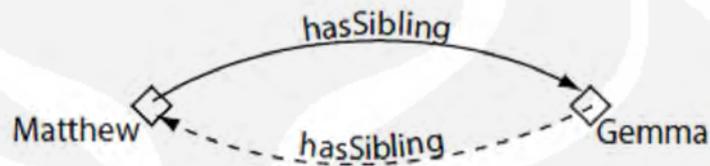
Jika sebuah properti *transitive*, properti menghubungkan individu A dengan individu B serta menghubungkan individu B dengan individu C, maka dapat disimpulkan bahwa individu A berhubungan dengan individu C melalui properti P. Sebagai contoh pada Gambar 2.6 menunjukkan *transitive property* **hasAncestor**. Jika individu Matthew mempunyai *ancestor* Peter, Peter mempunyai *ancestor* William, maka dapat disimpulkan bahwa Mathew mempunyai *ancestor* William.



Gambar 2.6. Contoh *Transitive Properties* [5]

o *Symetric Properties*

Properti P adalah *symetric* jika properti tersebut menghubungkan A ke individu B kemudian menghubungkan B ke individu A. Sebagai contoh pada Gambar 2.7, individu Mathew berhubungan dengan individu Gema melalui **hasSibling** property. Dengan kata lain, jika Mathew bersaudara dengan Gema maka dapat disimpulkan bahwa Gema bersaudara dengan Mathew.



Gambar 2.7. Contoh *Symetric Properties* [5]

2.3.4. Pembangunan Ontologi

Terdapat beberapa tahapan yang dilakukan dalam pengembangan ontologi yaitu [6] :

- Menentukan domain dan batasan ontologi

Dalam mengembangkan ontologi dimulai dengan mendefinisikan domain dan batasan dengan menjawab pertanyaan berikut :

- o Domain apa yang akan melingkupi ontologi?
- o Mengapa ontologi digunakan?
- o Apa jenis pertanyaan terhadap informasi dalam ontologi sehingga perlu menyediakan jawaban?

Salah satu cara dalam menentukan batasan dalam ontologi adalah dengan membuat daftar pertanyaan yang harus dapat dijawab oleh *knowledge base* atau yang biasa disebut *competency questions* [6].

- Mempertimbangkan penggunaan ontologi yang sudah ada.

Ontologi yang sudah ada dapat diperhalus dan diperluas untuk domain dan *task* yang akan dibuat. Penggunaan ontologi yang sudah ada merupakan persyaratan apabila sistem yang akan dibuat akan berinteraksi dengan aplikasi lainnya yang telah dilakukan pada suatu ontologi atau

perbendaharaan kata yang dikontrol. Banyak ontologi yang telah tersedia dan dapat dimasukkan dalam pengembangan ontologi yang dilakukan.

- Menentukan istilah penting dalam ontologi

Menulis daftar istilah yang akan dijelaskan ke *user*. Terdapat beberapa pertanyaan yang dapat membantu penentuan istilah, yaitu:

- Istilah apa saja yang akan diperbincangkan?
- Apa yang akan menjadi jawaban mengenai istilah – istilah tersebut?
- Properti apa saja yang dimiliki istilah tersebut?

Sebagai contoh pada ontologi minuman anggur, istilah yang penting meliputi minuman anggur, anggur, tempat membuat anggur, warna minuman anggur, bentuk, cita rasa dan kadar gula. Pada dasarnya, hal ini dibutuhkan untuk mendapat daftar istilah yang menyeluruh tanpa khawatir tumpang tindih antara *concept*, hubungan antara istilah dan properti dari *concept* atau *concept* tersebut termasuk *class* atau *slot*.

- Mendefinisikan *class* ontologi dan menyusun class dalam hirarki taksonomi (*subclass – superclass*).

Terdapat beberapa pendekatan dalam pengembangan hirarki *class* [5], yaitu :

- Proses pengembangan ***top-down*** dimulai dengan mendefinisikan *concept* umum dalam domain dilanjutkan dengan *concept* yang lebih spesifik.
- Proses pengembangan ***bottom-up*** dimulai dengan mendefinisikan *class* yang paling spesifik kemudian dikelompokkan menjadi *class* dengan konsep yang lebih umum.
- Proses pengembangan ***combination*** adalah sebuah kombinasi antara pendekatan *top-down* dan *bottom up*. Mendefinisikan konsep yang menonjol terlebih dahulu kemudian menggeneralisasi dan mengkhususkan konsep tersebut.

- Mendefinisikan *slot* atau *properties* dan menjabarkan nilai dari *slot* tersebut.
- Mendefinisikan *facets* pada *slots*.

Slot dapat memiliki *facet* yang berbeda dalam menggambarkan tipe nilai, nilai tersebut dapat berupa *cardinality* (sejumlah nilai) dan fitur lainnya.

Berikut merupakan beberapa *facet* yang umum digunakan :

- *Slot cardinality*

Slot cardinality mendefinisikan sejumlah nilai yang dimiliki oleh *slot*. Beberapa sistem hanya membedakan *single cardinality* (mempunyai satu nilai) dengan *multiple cardinality* (mempunyai beberapa nilai).

○ *Slot value type*

Sebuah tipe *facet* menggambarkan tipe dari nilai yang dapat mengisi *slot*.

Berikut merupakan tipe nilai yang umum digunakan :

1. *String*, tipe nilai yang paling sederhana yang biasa digunakan untuk *slot*.
2. *Number*, terkadang tipe nilai yang lebih spesifik seperti *Float* dan *Integer* yang digunakan dalam penggambaran *slot* dengan nilai numerik.
3. Boolean
Pada Boolean *slot* mempunyai *yes-no flag*.
4. *Enumerated*
Enumerated slot merinci daftar dari spesifik nilai yang diperbolehkan untuk *slot*.
5. *Instance*
Instance membuat hubungan antara individual.

● Membuat *instances*

Langkah terakhir adalah membuat individual atau *instances* pada *class* dalam suatu hirarki. Dalam mendefinisikan individual *instances* pada *class* harus diperhatikan persyaratan berikut :

- Pemilihan *class*.
- Pembuatan individual *instances* pada *class* tersebut.

Knowledge base selanjutnya dapat dibuat dengan mendefinisikan individual *instances* dari *class* yang terisi pada nilai spesifik *slot* dan *slot* batasan tambahan (*facets*).

- Mengisi nilai *slot*.

2.4. SEMANTIK

2.4.1. Pengertian Semantik

Semantik (Bahasa Yunani : semantikos, memberikan tanda, penting, dari kata *sema*, tanda) adalah cabang linguistik yang mempelajari makna yang terkandung pada suatu bahasa, kode atau jenis representasi lain [7]. Semantik biasanya dikontraskan dengan ekspresi makna sintaksis yaitu pembentukan simbol kompleks dari simbol yang lebih sederhana.

Web Semantik merupakan pengembangan dari *world wide web* dimana *content* web yang ditampilkan tidak hanya dalam bahasa format manusia yang umum (*natural language*) tetapi juga dalam format yang dapat dibaca dan digunakan oleh mesin (*software*). Web Semantik memiliki informasi yang dimiliki oleh mesin yang memiliki kecerdasan buatan sehingga mampu menemukan dan mengintegrasikan informasi dengan mudah. Dengan demikian fungsi web menjadi wadah bagi pertukaran data, informasi dan pengetahuan melalui kecerdasan buatan sehingga mengerti keinginan user dimana dapat diinstruksikan untuk mengambil informasi sesuai kriteria tertentu. Tujuan dari web semantik adalah mengatur informasi dan prosedur. Fundamental dalam pembangunan *semantic* web adalah kreasi dan *semantic* metadata [7].

Metadata terdiri dari dua bagian, yaitu :

- Penggambaran sebuah dokumen.
Contohnya adalah halaman web atau bagian dari suatu dokumen seperti sebuah paragraf.
- Penggambaran entitas didalam suatu dokumen.
Contohnya adalah seseorang atau sebuah perusahaan.

Pada semua kasus, yang terpenting bahwa metadata adalah semantik, yang menggambarkan semua isi dari dokumen tersebut.

Saat membuat aplikasi semantik, sebenarnya terdapat dua variabel yang dibangun secara sekaligus. Variabel pertama adalah web terdiri dari protocol komunikasi dan format web. Terdapat standar web semantik yang direkomendasikan W3C seperti RDF, OWL dan SPARQL. Variabel lainnya adalah semantik merepresentasikan makna dari web data.

2.4.2. Komponen – Komponen dalam *Semantic Web*

Pembuatan *semantic web* dimungkinkan dengan adanya sekumpulan standar yang dikoordinasi oleh *World Wide Web Consortium* (W3C). Standar yang paling penting dalam pembangunan *semantic web* adalah XML, XML Schema, RDF, RDF Schema dan OWL. Komponen – komponen dalam *semantic web* ini yang memungkinkan komunikasi dan interaksi pada level mesin [2].

- XML dan XML Schema

Extensible Markup Language (XML) merupakan bahasa *markup* yang didesain untuk menjadi sarana yang mudah dalam mengirimkan dokumen melalui web. Berbeda dengan *Hypertext Markup Language* (HTML), XML memungkinkan penggunanya untuk mendefinisikan *custom tag*.

XML Schema merupakan bahasa yang digunakan untuk mendefinisikan sekumpulan aturan (*schema*) yang harus dipatuhi oleh dokumen XML. Struktur dari dokumen XML yang dibuat harus sesuai dengan *schema* yang telah didefinisikan tersebut.

- RDF dan RDF Schema

Resource Description Framework (RDF) adalah spesifikasi yang dibuat oleh W3C sebagai metode umum untuk memodelkan informasi dengan menggunakan sekumpulan format sintaks. Ide dasar dari RDF adalah bagaimana kita dapat membuat pertanyaan mengenai sebuah *resource web* dalam bentuk ekspresi subjek (S), predikat (P), objek (O). Dalam terminology RDF, SPO ini seringkali disebut dengan istilah N-triple. Subjek mengacu pada *resource* yang ingin dideskripsikan. Predikat merupakan komposisi yang menerangkan sudut pandang dari subjek yang dijelaskan objek, sementara subjek dan objek merupakan entitas. Objek di dalam RDF dapat menjadi subjek yang diterangkan oleh objek lainnya. Dengan inilah objek dapat berupa masukan yang dapat diterangkan secara jelas dan detail, sesuai dengan keinginan pengguna yang memberikan masukan. Dengan menggunakan RDF, *website* dapat menyimpan dan melakukan pertukaran informasi antar web. RDF telah digunakan pada aplikasi-aplikasi berikut :

- *RDF Site Summary (RSS)*
RSS memberikan informasi update sebuah *website* tanpa pengunjung perlu mengunjungi *website* tersebut.
- *Friend of a Friend (FOAO)*
Didesain untuk mendeskripsikan orang-orang, ketertarikan dan hubungan mereka.
- *Semantically-Interlinked Online Communities (SIOC)*
Menerangkan komunitas *online* dan menciptakan koneksi antara diskusi berbasis internet seperti *message board*, *blog* maupun *mailing list*. *RDF schema* dapat dipandang sebagai kamus data atau *vocabulary* untuk mendeskripsikan *properties* dan *classes* dari *resources* *RDF*.
- *Ontology Web Language (OWL)*
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. Ontologi sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah tersebut dengan cara yang lebih mudah atau dengan pengertian lain adalah representasi istilah beserta hubungannya. Ketika informasi yang ada dalam dokumen perlu untuk diproses oleh aplikasi atau mesin, OWL dapat digunakan untuk merepresentasikan makna suatu istilah secara eksplisit sekaligus hubungan antara istilah-istilah tersebut.
Dengan menggunakan OWL, kita dapat menambah *vocabulary* tambahan disamping *semantic* formal yang telah dibuat sebelumnya menggunakan XML, RDF dan RDF Schema. Hal ini sangat membantu penginterpretasian mesin yang lebih baik terhadap isi web. Untuk mendeskripsikan *properties* dan *classes*, OWL menambahkan *vocabulary* seperti :
 - “*among others*”
 - *Relasi antar classes (misalnya : “disjointness”)*
 - *Kardinalitas*
 - *Kesamaan (equality)*
 - *Karakteristik property*
 - *Enumerated classes*

OWL memiliki tiga *sub-language* yaitu :

○ OWL *Lite*

Mendukung pengguna yang memerlukan klasifikasi hirarki dan dalam batasan yang sederhana.

○ OWL DL

Mendukung konstruksi seluruh OWL, tetapi hanya dapat digunakan pada batasan tertentu.

○ OWL *Full*

Diperuntukkan bagi pengguna yang menginginkan maksimum penggunaan dan kebebasan sintaksis.

2.4.3. Keuntungan *Semantic*

Keuntungan yang dimiliki oleh *semantic* web adalah sebagai berikut :

- Waktu yang diperlukan untuk mendapatkan informasi yang dicari lebih singkat.
- Pekerjaan pencarian yang dilakukan manusia dapat digantikan oleh mesin.

2.5. TOOL ONTOLOGI

2.5.1. PROTEGE

Protege merupakan *tool* ontologi dengan *platform open source* untuk membangun domain model dan aplikasi *knowledge based*. Protege mengimplementasikan struktur pemodelan *knowledge* dan dapat memvisualisasikan hasil ontologi dalam berbagai format. Sebuah ontologi menggambarkan *concept* dan hubungan-hubungan yang penting dalam domain yang khusus, yang menyediakan kosa kata dalam domain tersebut. Dalam beberapa tahun terakhir, ontologi telah diadopsi dalam bisnis dan komunitas ilmiah seperti *scientific knowledge* portal, manajemen informasi, integrasi informasi, *electronic commerce* dan *semantic web service*.

Pada Protege terdapat dua cara dalam pemodelan ontologi, yaitu :

- Protege Frame editor

User dapat membangun ontologi dalam frame-based dengan kesepakatan dengan *Open Knowledge Base Connectivity Protocol* (OKBC). Pada model

ini, sebuah ontologi terdiri dari seperangkat *class* yang terorganisir pada suatu hirarki yang merepresentasikan sebuah domain, seperangkat *slot* yang berhubungan dengan *class* serta *instance* dari tiap *class* tersebut.

- Protege OWL

Protege-OWL editor merupakan kelanjutan dari Protege yang mendukung *Ontology Web Language* (OWL). OWL merupakan pengembangan mutakhir standar bahasa ontologi yang disahkan oleh *World Wide Web Consortium* (W3C) untuk mempopulerkan *semantic web vision*. Protege-OWL editor memungkinkan user untuk :

- Mengambil dan menyimpan OWL dan RDF ontologi.
- Mengubah dan memvisualisaikan class, properties dan *Semantic Web Rule Language* (SWRL).
- Menjabarkan karakteristik class secara logis sebagai ekspresi OWL.
- Mengeksekusi penalaran seperti *description logic classifier*.
- Mengubah OWL individual untuk web semantik.

Protege-OWL berhubungan secara erat dengan Jena dan mempunyai *open source* Java API untuk pengembangan *Semantic Web Service*. Jena merupakan *framework* berbasis Java untuk mengkonstruksi aplikasi semantik web. *Framework* ini menyediakan lingkungan pemrograman RDF, RDF Schema OWL dan SPARQL.

Protege-OWL *Application Programming Interface* (API) adalah sebuah *Java Library open source* untuk *Web Ontology Language* (OWL) dan RDF. API menyediakan metode untuk mengambil dan menyimpan file OWL, menanyakan OWL data model, menjalankan penalaran berdasarkan *Description Logic engines* dan sebagai *graphical user interface*. API didesain untuk dapat digunakan dalam dua konteks, yaitu :

- Pengembangan komponen yang dieksekusi dalam Protege-OWL editor *user interface*.
- Pengembangan aplikasi stand-alone (contoh : aplikasi Swing, Servlet dan *plug-in* Eclipse).

2.5.2. Powl

Powl merupakan tool ontologi berbasis web. Kemampuan yang dimiliki Powl pada *knowledge base* RDF dan OWL meliputi :

- *Parsing*
- *Storing*
- *Querying*
- *Manipulating*
- *Versioning*
- *Serving*
- *Serializing*

Bagi *semantic web researcher*, Powl menyediakan *test bed* untuk implementasi yang cepat. Bagi developer web aplikasi, Powl menyediakan *object-oriented application programming interface* untuk pengembangan perangkat lunak *semantic web*. Bagi *knowledge engineer*, Powl merupakan web *interface* untuk pengembangan kolaboratif ontologi dalam web *environment*.

Powl diimplementasikan dengan menggunakan pemrograman PHP. Pengembangan dengan PHP memiliki berbagai keuntungan, diantaranya adalah [8] :

- Sebagai *platform* yang mandiri, Powl dapat dimodifikasi dan diperluas oleh peneliti dan web *developer*.
- PHP sebagai bahasa pemrograman yang luas yang digunakan untuk aplikasi web dengan estimasi sebanyak 35% dari semua website yang ada.

Sejak *semantic web* dipandang sebagai pengembangan dari web saat ini, diyakinkan paradigma *semantic web* dapat berjalan sukses apabila didukung oleh aplikasi dan tool yang tersedia berinteraksi dengan PHP.

2.5.2.1. Arsitektur Powl

Arsitektur Powl terdiri dari empat tingkatan, mencoba untuk meminimalkan ketergantungan dan mensuplai *clean interface* antara tingkatan. Tingkatan tersebut adalah :

- *Powl store* - kompatibel dengan *database SQL*.

- RDFAPI, RDFSAPI, OWLAPI – layer API untuk menangani RDF, RDF *schema* (RDFS) dan OWL.
- Powl API – terdiri dari *class-class* dan fungsi-fungsi lainnya untuk membangun web aplikasi.
- *User interface* – seperangkat halaman PHP dikombinasikan dengan widget yang disediakan oleh Powl API untuk mengakses (*browsing, viewing, editing*) model data yang tersimpan di Powl *store*.

2.5.2.2.Powl Store

SQL *compatible relational database* disupport oleh AdoDB, yang merupakan database abstraction layer yang digunakan oleh Powl, yaitu Powl store. *Database* dibawah ini digunakan untuk meyimpan semua informasi yang berhubungan dengan ontologi dan pengembangannya :

Table 2.1 *Database* pada Powl

Tabel	Deskripsi
models	Menyediakan informasi mengenai model dalam tempat penyimpanan.
statement	Terdiri dari semua model statement dalam tempat penyimpanan.
log_actions	Menahan informasi mengenai perubahan (<i>edit</i>) pada model.
log_statements	Penambahan dan pemindahan statement pada setiap tindakan

Powl *store* menggunakan skema denormalisasi *database*, dimana semua *resource* ditulis secara lengkap pada sebuah tabel dan setiap barisnya (*row*) merepresentasikan RDF *statement*. Pengujian dilakukan oleh developer RDFAPI bahwa dua sampai tiga kali lebih cepat dibandingkan dengan skema *database* normalisasi dimana *resource* disimpan secara terpisah. Powl *store* diakses oleh RDFAPI.

2.5.2.3.RDFAPI

RAP – RDFAPI untuk PHP merupakan sebuah proyek mandiri oleh Chris Bizer, Radoslaw Oldakowski yang menyediakan fungsi-fungsi pada Powl :

- Parser, penyambungan untuk RDF *serialization* yang berbeda (XML, N3, N-Triple).
- RDQL *declarative query backend*.
- *Class-class* dan metode untuk bekerja dengan RDF model.
- NetAPI untuk mempublikasikan pemodelan pada web.

2.5.2.4.RDFSAPI

RDFSAPI merupakan perluasan dari *class* RDFAPI oleh RDF-Schema spesifik *class*. Saat *resource* (*class*, *instance* dan *property*) diminta, sebagai parameter fungsi, pilihan dibawah ini merepresentasikan *resource* yang tersedia :

- RDFS *Resource Object*
- *Local name* (sebagai *string*, contohnya : "Article")
- URL (sebagai *string*, contohnya: "<http://purl.org/net/nkouf/ns/bibtex#Article>")
- *Namespace prefix* dan *local name* (sebagai *string*, contohnya "bibtex : Article")

Sebagai pengganti file konfigurasi atau tabel database untuk kebutuhan khusus, otorisasi pengguna dan pemilihan manajemen dalam user translation purpose dan integrasi modul, Powl menggunakan sistem ontologi untuk menyimpan *knowledge base*. Sistem ontologi tersebut dapat diubah dan diatur oleh Powl sendiri. Tiga contoh dalam sistem ontologi adalah sebagai berikut :

- *Instance* dari *class* "Label" sistem ontologi terdiri dari terjemahan dari semua teks yang ditampilkan pada user interface.
- *Instance* dari *class* "User" dan "Group" bertanggung jawab terhadap otorisasi dan penyimpanan dari setiap pilihan.
- *Class* yang lainnya digunakan untuk menyimpan konfigurasi data dari modul Powl dan *widget plugin*. *Widget* untuk merepresentasikan dan mengubah literal data yang telah dipilih dan konfigurasi bergantung pada *context* yang sebenarnya digunakan (misalnya user yang sedang *login*).

User interface diatur pada tab-tab, setiap tab memiliki fungsi-fungsi yang berbeda terhadap *knowledge base*. Tab berikut terdapat pada Powl :

- *Models* – menyediakan tampilan model pada Powl store
- *Triples* – menampilkan list yang dapat dicari dari *triple* pada ontologi yang dipilih.
- *Classes* – *Class* yang di susun secara hirarki dan dapat dilihat dan diedit sesuai definisinya.
- *Properties* – Properti yang disusun secara hirarki dan dapat dilihat dan diedit sesuai definisinya.
- *Instance* – menampilkan *instance* yang bervariasi dari *class-class* yang terpisah.
- *Version* – menyediakan akses untuk informasi mengenai pengembangan ontologi.

2.5.2.5.RDQL query builder

RDQL merupakan implementasi dari SQL , yaitu sebagai *query* untuk RDF. Dengan mengacu RDF sebagai data dan menyediakan *query* melalui *triple pattern* dan *constraint* dari single RDF model. Powl memungkinkan *user* untuk membuat *query* untuk menjawab pertanyaan *knowledge base*. Hal itu dibatasi *user* mengacu pada RDQL *query* pada *resource* yang tersedia.

BAB 3

PERANCANGAN ONTOLOGI WEB DAN APLIKASI SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA

3.1. PEMODELAN RANCANGAN SISTEM

3.1.1. Persyaratan Sistem

Web Ontologi dan aplikasi semantik memiliki persyaratan pembangunan yang menjadi acuan dalam perancangan. Sistem akan dibangun pada server dan informasi dapat diakses oleh pengguna melalui *web browser*.

3.1.2. Pemodelan Sistem

Sebagai bagian dari persyaratan sistem dan kegiatan perancangan, web terdiri dari beberapa komponen yang berkaitan, yaitu :

- Komponen untuk merancang model ontologi.

Komponen ini digunakan untuk merancang model ontologi berdasarkan domain *expert*. Dalam pemodelan ontologi dimanfaatkan sebuah *tool open source* yang dikenal dengan Protege. Hasil rancangan berupa model ontologi yang tersusun dalam hirarki taksonomi. Proses perancangan model hanya dapat dilakukan oleh seorang domain *expert* atau dengan kata lain orang yang benar-benar menguasai suatu domain. Peranan domain *expert* adalah membuat hirarki taksonomi (*classes – subclasses*), *property* dan *instance* dengan menggunakan Protege. Setelah model ontologi selesai dibuat, maka model ontologi tersebut disimpan dalam file berekstensi XML

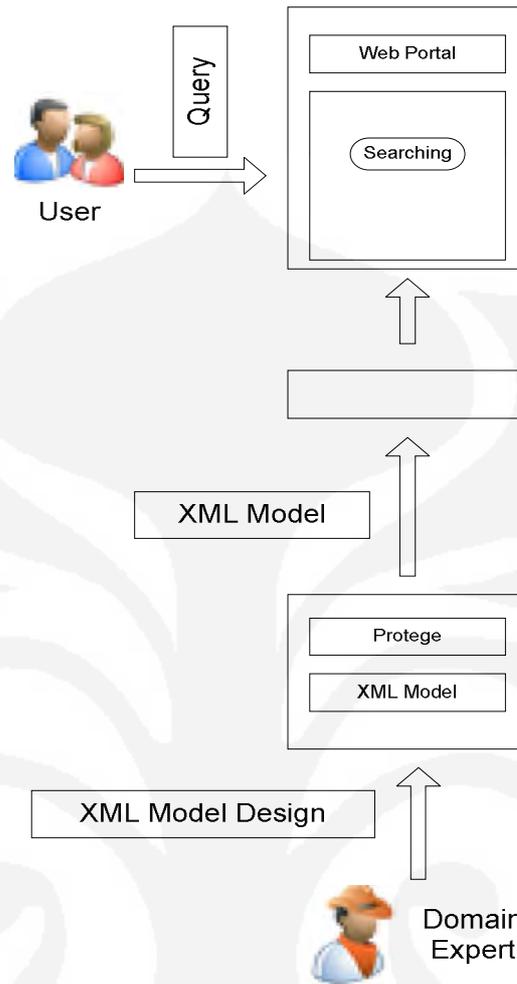
- Komponen *interface*

Komponen ini digunakan bagi seorang domain *expert* atau *librarian* untuk *upload* dan *parsing file* XML tersebut agar dapat dibaca oleh *database* MySQL.

- Komponen portal.

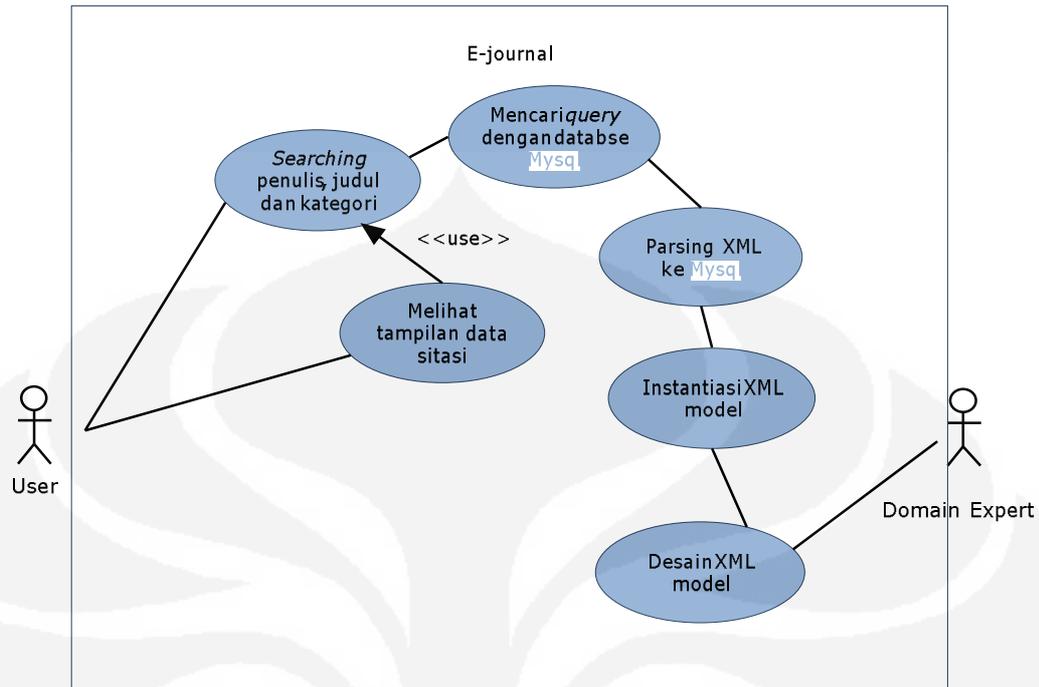
Portal ini dibuat dengan tujuan sebagai perantara bagi *user* untuk menyusun *query* kedalam sistem untuk mendapatkan informasi yang diinginkan..

Gambar 3.1. merupakan skema sistem secara keseluruhan.



Gambar 3.1. Skema Ontologi Web dan Aplikasi Semantik

Pembangunan aplikasi dimulai dengan perancangan model XML dan mengimplementasikannya ke dalam *class*, *properties* dan *instance* dengan menggunakan Protege oleh *domain expert*. Dilanjutkan pembuatan perantara antara model XML dengan database MySQL, dalam hal ini *domain expert* atau *librarian* akan *upload* dan *parsing* file XML ke dalam *database*. Selanjutnya pembuatan perantara *user* dalam hal ini pembuatan portal web semantik. *User* melakukan *searching* dengan menyusun *query* yang kemudian *query* tersebut akan dikirimkan ke *database*. Setelah *query* yang dikirimkan ke *database* menemukan informasi yang dicari maka informasi tersebut akan ditampilkan. Pembangunan aplikasi secara lengkap dapat dilihat dengan menggunakan *use case diagram* pada Gambar 3.2 berikut.



Gambar 3.2. Use case diagram e-journal

Pada *use case* diagram terdapat *use case text* yang akan menjelaskan secara naratif bagaimana sistem akan digunakan.

1. Pencarian Sitasi

- *Objective* : Pencarian keterkaitan literatur yang telah disitasi oleh artikel lainnya pada suatu jurnal.
- *Actors* : *User*
- *Pre-condition* : *User* menginginkan suatu informasi mengenai keterkaitan suatu artikel atau literatur lainnya yang telah disitasi oleh suatu artikel pada sebuah jurnal.
- *Normal Flow* :
 - *User* membuka halaman web portal.
 - *User* menyusun *query* pencarian.
 - *User* memulai pencarian.
 - *User* mendapatkan informasi mengenai penulis dan judul serta keterkaitannya dengan artikel yang telah disitasi.
- *Alternate Flow* : *User* tidak mendapatkan informasi yang diinginkan atau informasi tersebut tidak terdapat dalam ontologi model yang sudah dibangun.

- *Post Condition* : *User* memperoleh hasil pencarian.

2. Pemodelan ontologi

- *Objective* : Membangun desain ontologi dalam domain *e-journal* ke dalam *class*, *properties* dan *instance* dengan menggunakan *tool* Protege.
- *Actor* : *Domain expert*
- *Normal Flow* :
 - *Domain expert* melakukan pemodelan terhadap rancangan ontologi.
 - Model XML dapat diintegrasikan dengan database MySQL.
- *Result* : Model ontologi telah diintegrasikan dengan *portal* web.

3.2. PERANCANGAN ONTOLOGI

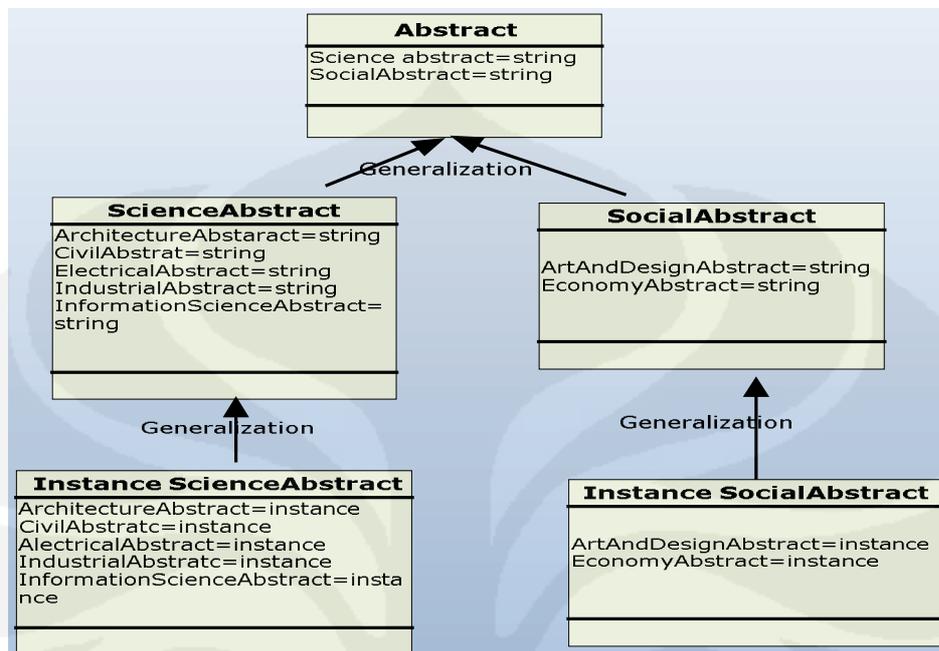
Perancangan ontologi memberikan informasi mengenai tahapan- tahapan dalam pembangunan ontologi dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi.

Tahapan-tahapan dalam pembangunan ontologi adalah sebagai berikut :

- Penentuan domain. Domain yang melingkupi ontologi adalah *e-journal*
- Mendefinisikan *class* ontologi dan menyusun *class* tersebut dalam hirarki taksonomi (*subclass-superclass*) dengan menggunakan proses pengembangan *top-down* dimulai dengan mendefinisikan *concept* umum dalam domain dilanjutkan dengan *concept* yang lebih spesifik.
- Mendefinisikan *slot* atau *property*
- Mendefinisikan facets pada slot atau axiom pada *properties*.
Properties memiliki domain dan *range* yang spesifik. *Properties* menghubungkan individu pada domain dan individu pada *range*.
- Membuat *instance*.
- Mengisi nilai *slot* pada *instance*.

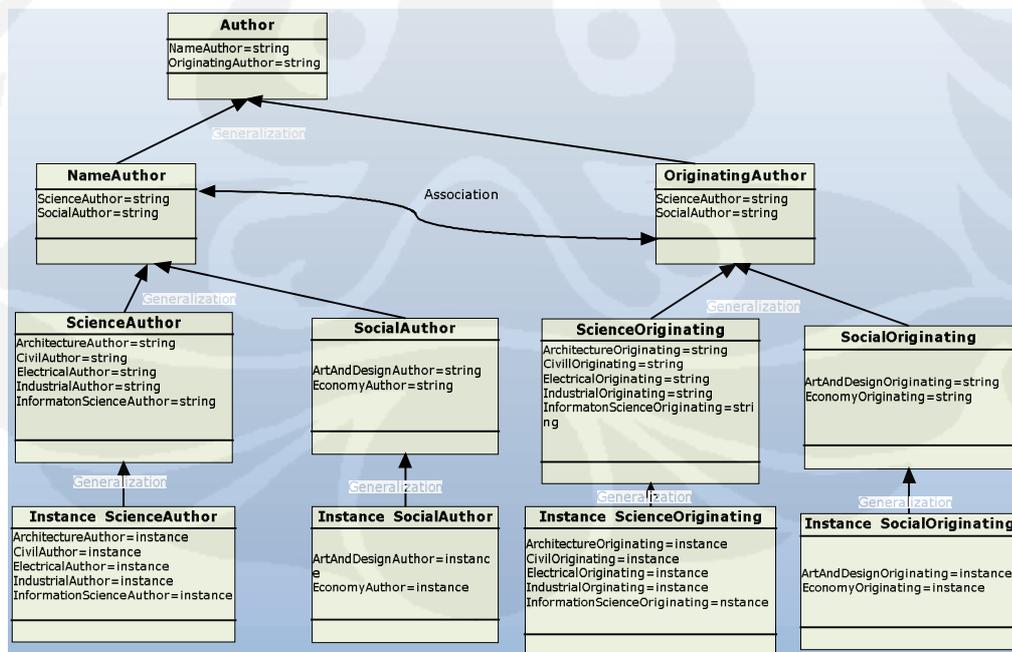
Perancangan ontologi ditunjukkan pada Gambar 3.3 hingga Gambar 3.9 berikut. *Class Abstract* yang terdiri dari *Subclass ScienceAbstract* dan

SocialAbstract. Setiap masing-masing *subclass* tersebut terdiri dari *instance* seperti pada Gambar 3.3.



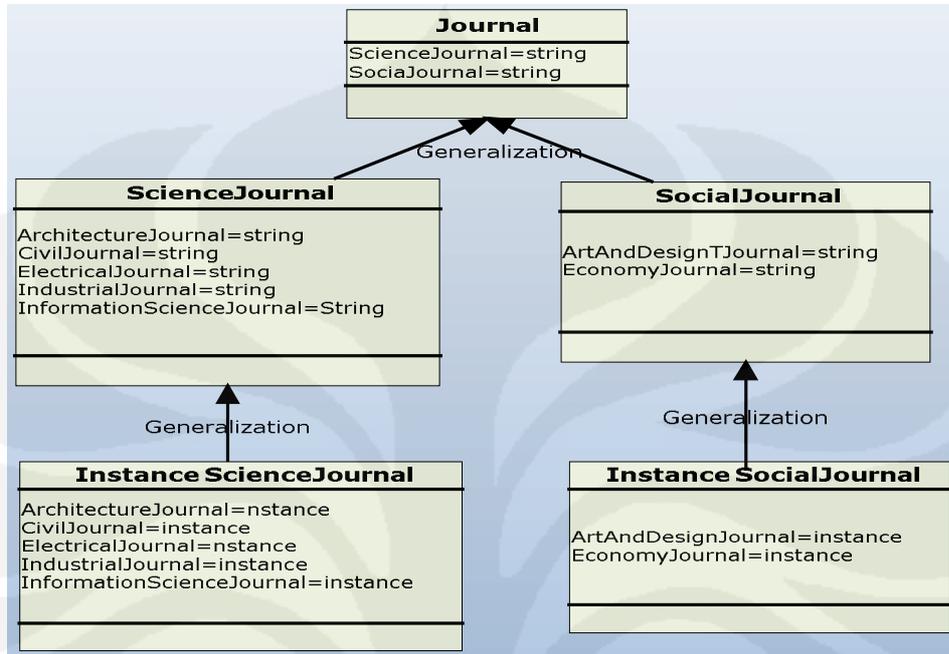
Gambar 3.3 Class Abstract

Class Author yang terdiri dari *Subclass NameAuthor* dan *OriginatingAuthor*. Setiap masing-masing *subclass* tersebut terdiri dari *instance* seperti pada Gambar 3.4.



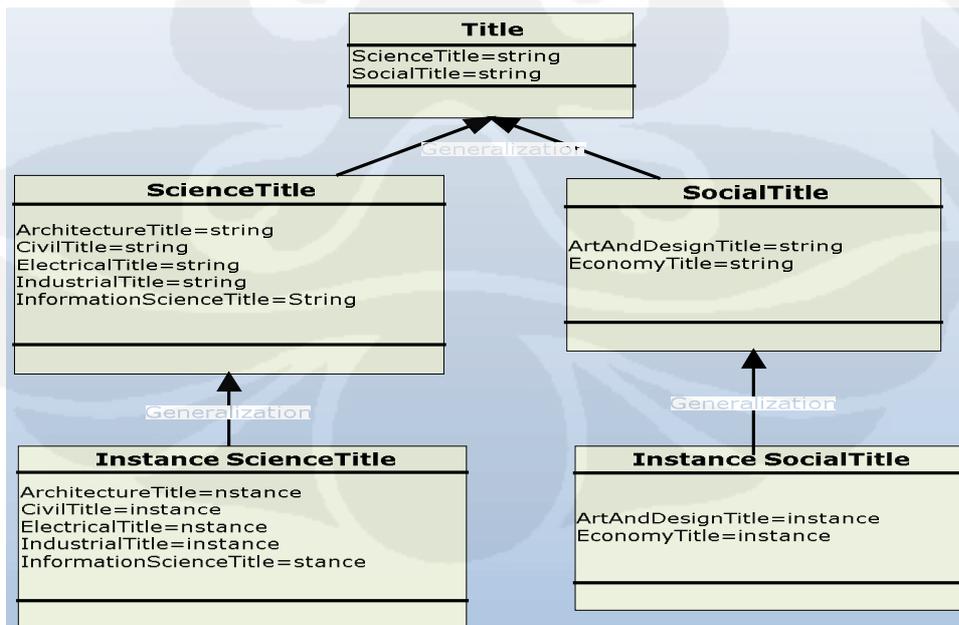
Gambar 3.4 Class Author

Class Journal yang terdiri dari *Subclass ScienceJournal* dan *SocialJournal*. Setiap masing-masing *subclass* tersebut terdiri dari *instance* seperti pada Gambar 3.5.



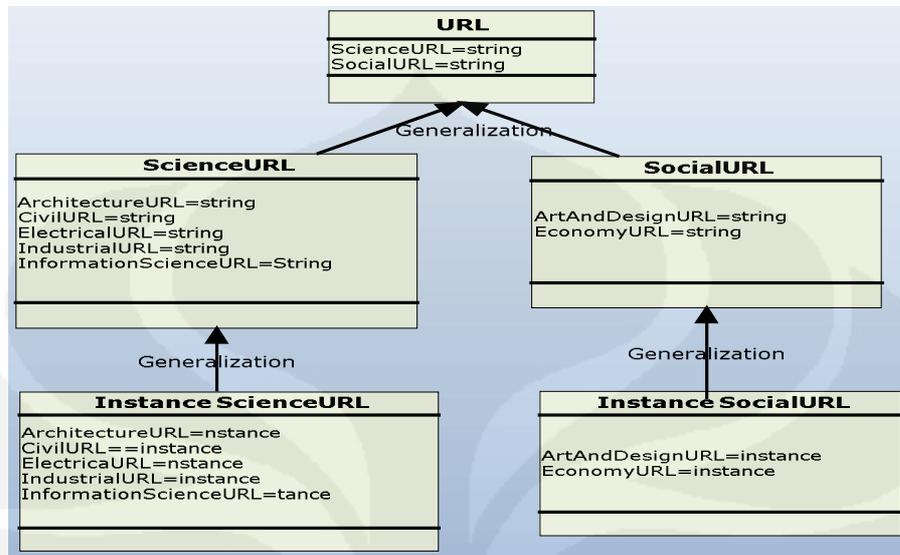
Gambar 3.5 Class Journal

Class Title yang terdiri dari *Subclass ScienceTitle* dan *SocialTitle*. Setiap masing-masing *subclass* tersebut terdiri dari *instance* seperti pada Gambar 3.6.



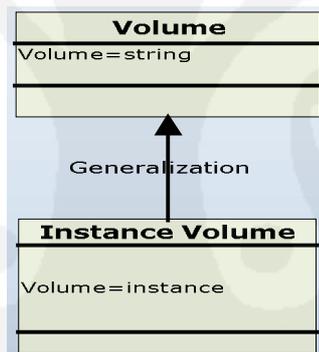
Gambar 3.6 Class Title

Class URL yang terdiri dari *Subclass ScienceURL* dan *SocialURL*. Setiap masing-masing *subclass* tersebut terdiri dari *instance* seperti pada Gambar 3.7.



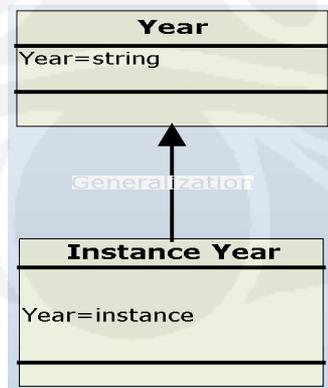
Gambar 3.7 *Class URL*

Class Volume yang terdiri dari *instance* seperti pada Gambar 3.8.



Gambar 3.8 *Class Volume*

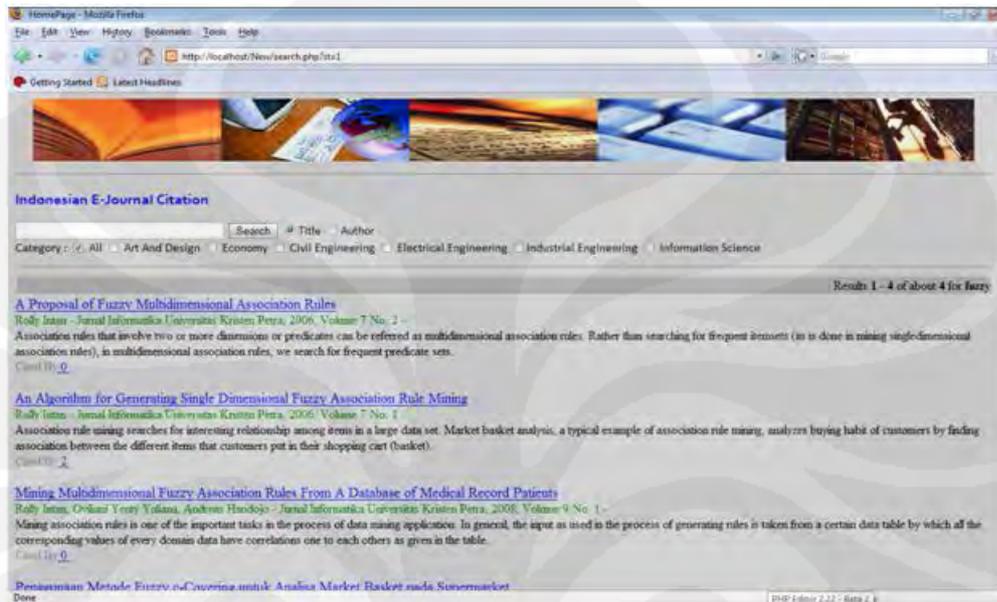
Class Year yang terdiri dari *instance* seperti pada Gambar 3.9.



Gambar 3.9 *Class Year*

3.3. PERANCANGAN PORTAL

Portal web adalah situs web yang menyediakan kemampuan tertentu yang dibuat sedemikian rupa mencoba menuruti selera para pengunjungnya. Perancangan portal memberikan informasi tentang komponen apa saja yang tersedia pada suatu halaman seperti yang ditunjukkan pada Gambar 3.10 berikut.



Gambar 3.10. Rancangan halaman awal (*Homepage*)

Pada halaman *Homepage*, *user* akan menemukan *text box* dan tombol *search* untuk melakukan pencarian. Pencarian tersebut didukung oleh menu pilihan *radio button Title* dan *Author* dengan beberapa pilihan kategori dalam *checkbox* yaitu *All*, *Art and Design*, *Economy*, *Civil Engineering*, *Electrical Engineering* dan *Information Science*. Saat pengguna memilih *radio button Title*, *query* akan diarahkan kepada tabel *hasTitle* yang merupakan hasil *parsing* dari file XML. Saat pengguna memilih *radio button Author*, *query* akan diarahkan kepada tabel *isArticleOf*. *Checkbox* pada tampilan akan mengarahkan *query* pada *class Title* pada ontologi yang terdapat pada tabel *hasTitle*. Apabila dipilih *checkbox Art and Design* maka hanya akan ditampilkan artikel dari kategori jurnal *Art and Design*, hal tersebut juga berlaku bagi *checkbox* lainnya.

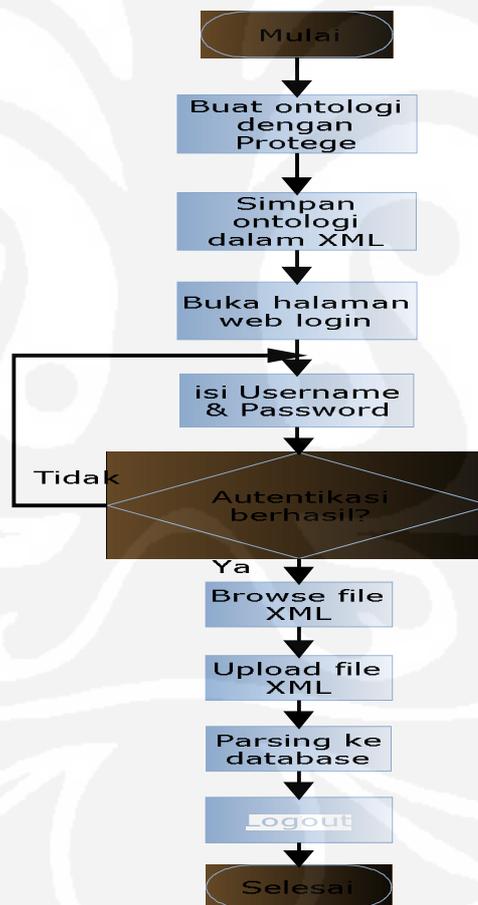
Informasi hasil pencarian yang akan ditampilkan pada halaman web yaitu:

- Pada baris pertama akan ditampilkan judul artikel hasil pencarian.
- Pada baris kedua akan ditampilkan penulis dari artikel, jurnal dari artikel tersebut, tahun dan volume artikel diterbitkan.

- Pada baris ketiga akan ditampilkan abstraksi dari artikel.
- Pada baris keempat akan ditampilkan jumlah artikel yang mensitasi dari artikel pada baris pertama. Apabila jumlah tersebut ditekan maka akan ditampilkan artikel yang yang mensitasi tersebut.

3.4. DIAGRAM ALIR SISTEM

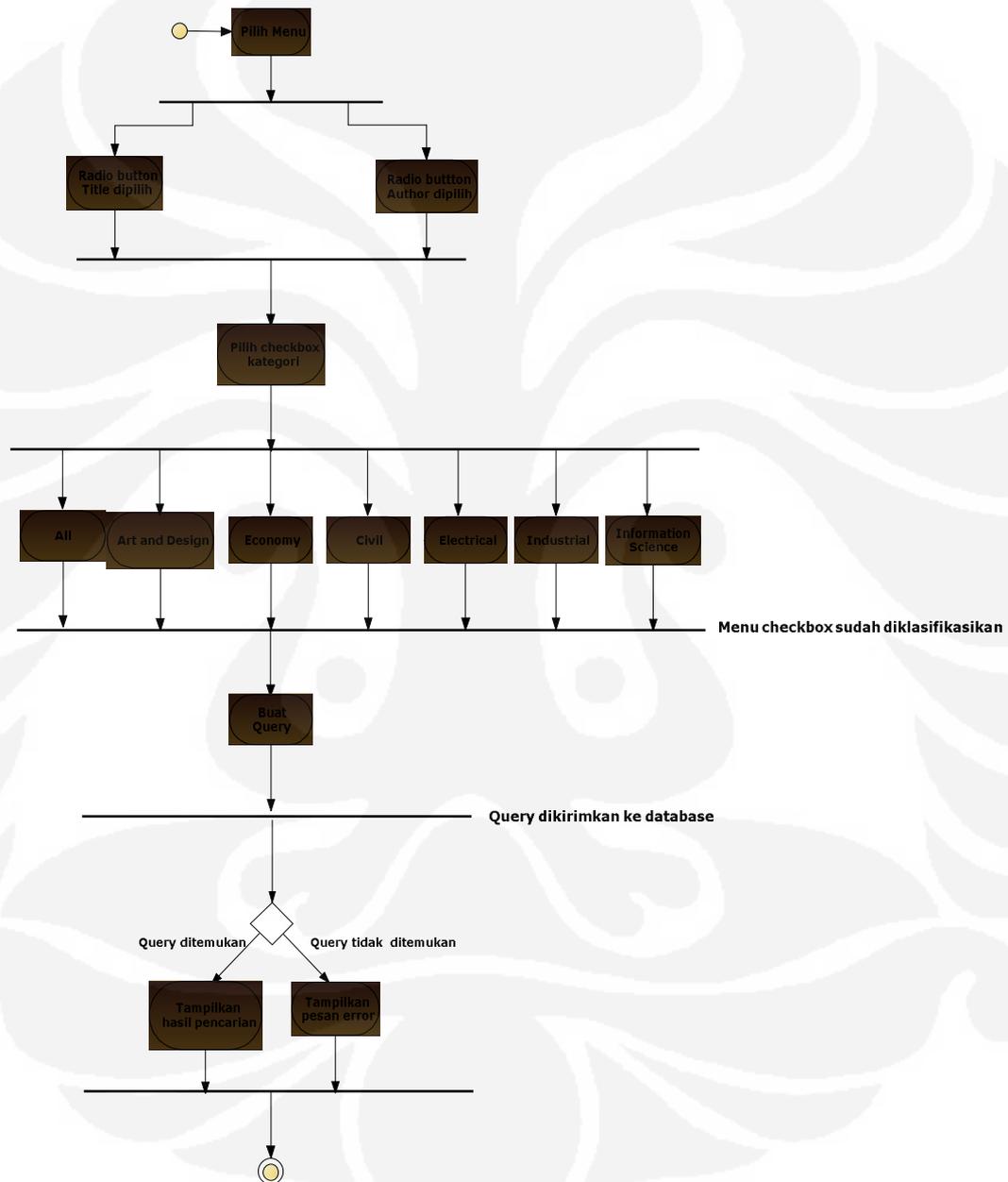
Diagram alir sistem menunjukkan bagaimana sistem melakukan proses eksekusi dan pengguna memberikan masukan dan sistem merespon masukan tersebut. Secara lengkap diagram alir sistem direpresentasikan pada Gambar 3.11 dan Gambar 3.12 . Gambar 3.11 merupakan diagram alir dari *domain expert* atau pustakawan ke *web server* untuk melakukan *upload dan parsing* sedangkan Gambar 3.12 merupakan diagram alir pengguna dalam melakukan pencarian.



Gambar 3.11 .Diagram alir *domain expert* ke MySQL

Setelah pemodelan dibuat pada Protege, akan dilakukan proses *upload* ke *directory server* dan *parsing* model ontologi ke dalam database MySQL, tetapi

sebelumnya pengguna dalam hal ini adalah *domain expert* atau *librarian* harus melakukan proses autentikasi terlebih dahulu sehingga hanya pengguna tertentu yang berhak melakukan *update* ke database MySQL. Pengguna akan membuka halaman web login dan memasukkan *username* dan *password*, apabila berhasil pengguna dapat membuka halaman web untuk melakukan *upload* dan *parsing* model ontologi.



Gambar 3.12 Diagram alir pengguna melakukan pencarian

Pada halaman web terdapat beberapa menu utama pencarian yaitu pencarian berdasarkan judul artikel yang ditunjukkan dengan *radio button Title* dan pencarian berdasarkan penulis artikel yang ditunjukkan dengan *radio button Author*. Sebelum melakukan pencarian pengguna harus memilih salah satu *radio button* tersebut.

Setelah pemilihan *radio button* dilakukan, pengguna dihadapkan pada pilihan kategori yang ditunjukkan dengan *checkbox*. Kategori tersebut terdiri dari *All*, dimana pengguna melakukan pencarian terhadap semua jenis kategori. Kategori *Art and Design* dikhususkan untuk melakukan pencarian pada judul atau penulis dari kategori *Art and Design*. Hal tersebut juga berlaku untuk kategori *Economy, Civil Engineering, Electrical Engineering, Industrial Engineering* dan *Information Science*.

Pengguna yang akan melakukan pencarian akan menempatkan kata pencarian pada *textbox* yang tersedia dan menekan tombol *Search*. Apabila pencarian ditemukan maka akan muncul informasi yang dicari.

BAB 4

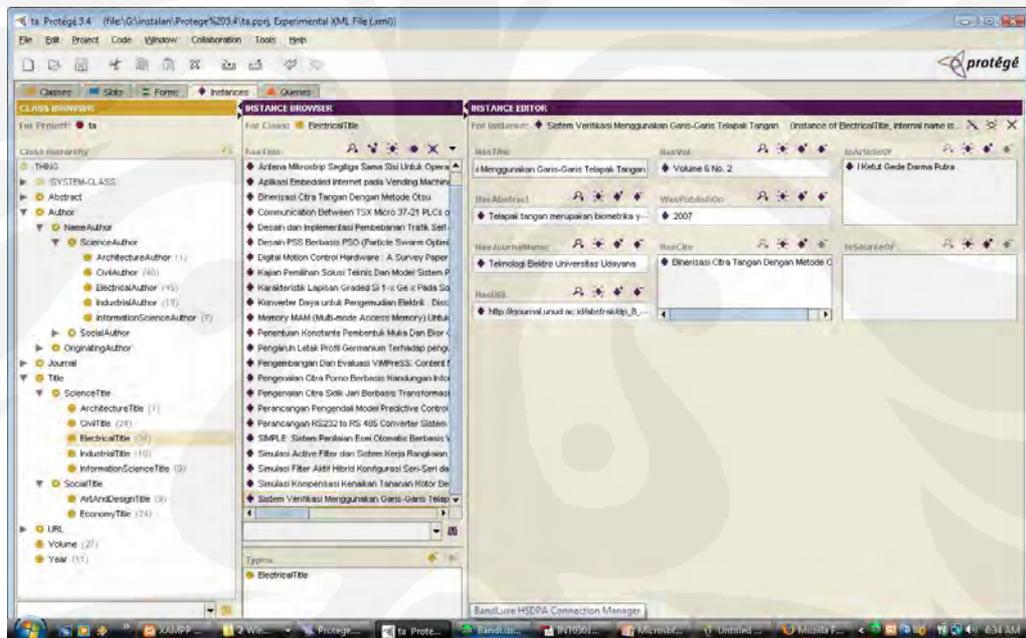
IMPLEMENTASI ONTOLOGI WEB DAN APLIKASI SEMANTIK UNTUK SITASI JURNAL ELEKTRONIK INDONESIA

4.1. IMPLEMENTASI ONTOLOGI

Implementasi ontologi menggunakan perangkat lunak sebagai berikut :

1. Protégé 3.4

Protégé 3.4 merupakan *tool* ontologi yang menyediakan komponen-komponen yang menjelaskan ontologi suatu domain. Komponen-komponen tersebut yaitu *class*, *properties* dan *instance*. Gambar 4.1 menunjukkan halaman pada Protégé.



Gambar 4.1. Halaman Protégé

2. TGVizTab

TGVizTab merupakan *plug-in* yang dapat memvisualisasikan ontologi yang telah dibangun.

4.1.1. Tahapan Implementasi Ontologi

Implementasi ontologi dilakukan melalui tahapan sebagai berikut :

1. Menentukan domain ontologi.

Untuk domain jurnal elektronik Indonesia, situs yang akan melingkupi ontologi adalah beberapa situs yang menyediakan jurnal elektronik. Situs-situs tersebut adalah sebagai berikut :

- <http://ejournal.unud.ac.id>
- <http://puslit2.petra.ac.id/ejournal>
- <http://journal.ui.ac.id>

Pada jurnal elektronik tersebut diambil beberapa artikel dalam beberapa kategori, yaitu *Art and Design* (Kesenian dan Desain), *Economy* (Ekonomi), *Civil Engineering* (Teknik Sipil), *Electrical Engineering* (Teknik Elektro), *Industrial Engineering* (Teknik Industri) dan *Information Science* (Ilmu Informasi).

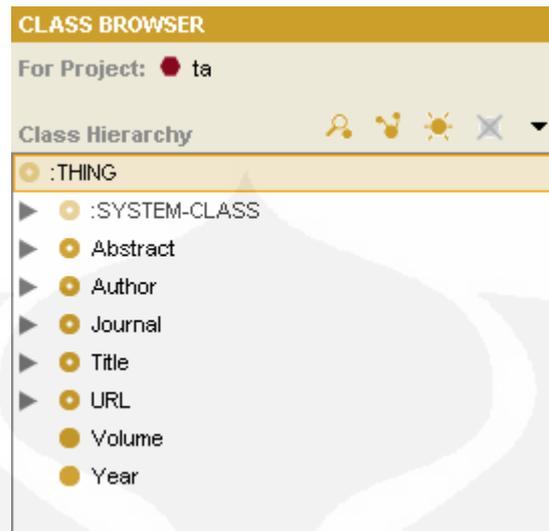
2. Menentukan batasan ontologi.

Penentuan batasan dalam ontologi dilakukan dengan membuat daftar pertanyaan yang harus dapat dijawab oleh *knowledge base* atau yang biasa disebut *competency questions*. Pertanyaan-pertanyaan yang membatasi ontologi adalah sebagai berikut :

- Suatu artikel pada jurnal terdiri dari apa saja?
Pertanyaan tersebut memiliki jawaban : judul, penulis, tahun terbit, volume terbit, jurnal yang menerbitkan artikel, referensi, abstrak.
- Referensi artikel terdiri dari apa saja?
Pertanyaan tersebut memiliki jawaban : penulis, tahun terbit, volume terbit, jurnal yang menerbitkan artikel.

3. Mendefinisikan ontologi dengan menyusun hirarki taksonomi (*superclass-subclass*).

Implementasi ontologi dilakukan dengan pendekatan *top-down* yaitu dengan mendefinisikan *class* atau *concept* umum dilanjutkan dengan *class* atau *concept* yang lebih spesifik. Untuk implementasi pada Protégé pertama kali dilakukan penentuan *class* umum untuk domain sitasi jurnal elektronik. *Class* tersebut adalah sebagai berikut seperti ditunjukkan pada Gambar 4.2 :



Gambar 4.2. *Class* utama

- *Abstract*, pengelompokkan ringkasan artikel.
- *Author*, pengelompokkan penulis artikel.
- *Journal*, pengelompokkan jurnal artikel diterbitkan.
- *Title*, pengelompokkan judul artikel.
- *URL*, pengelompokkan URL artikel.
- *Volume*, pengelompokkan volume artikel diterbitkan.
- *Year*, pengelompokkan tahun artikel diterbitkan.

Tahap implementasi selanjutnya adalah penentuan *subclass* yang berfungsi untuk mengkhususkan *class* umum. *Subclass* tersebut adalah sebagai berikut seperti ditunjukkan pada gambar 4.3 :

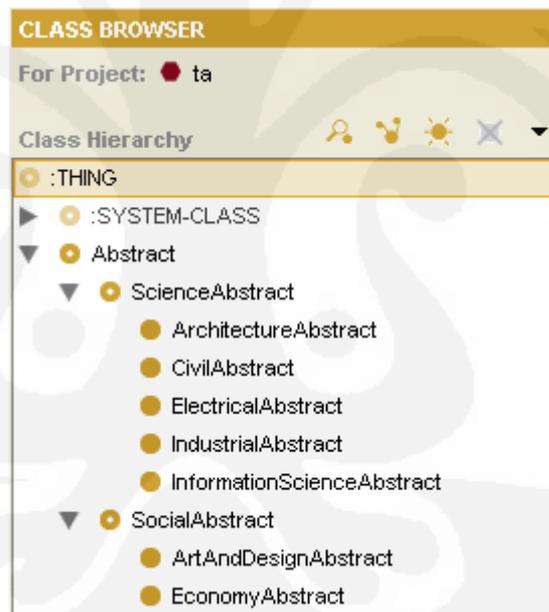


Gambar 4.3. *Superclass* dan *subclassnya*

- *Class Abstract*
 - *Subclass ScienceAbstract*, pengelompokkan ringkasan artikel dalam kategori *science*.
 - *Subclass SocialAbstract*, pengelompokkan ringkasan artikel dalam kategori *Social*.
- *Class Author*
 - *Subclass NameAuthor*, pengelompokkan nama penulis artikel.
 - *Subclass OriginatingAuthor*, pengelompokkan instansi penulis artikel berasal.
- *Class Journal*
 - *Subclass ScienceJournal*, pengelompokkan jurnal dari kategori *Science*.
 - *Subclass SocialJournal*, pengelompokkan jurnal dari kategori *Social*.

- *Class Title*
 - *Subclass ScienceTitle*, pengelompokkan judul artikel dari kategori *Science*.
 - *Subclass SocialTitle*, pengelompokkan judul artikel dari kategori *Social*.
- *Class URL*
 - *Subclass ScienceURL*, pengelompokkan URL artikel dari kategori *Science*.
 - *Subclass SocialURL*, pengelompokkan URL artikel dari kategori *Social*.

Dari tiap *subclass*, masing-masing memiliki *subclass* sehingga *subclass* sebelumnya menjadi *superclass*. *Subclass* selanjutnya adalah sebagai berikut seperti ditunjukkan pada Gambar 4.4 hingga Gambar :



Gambar 4.4. *Superclass Abstract* dan *subclassnya*

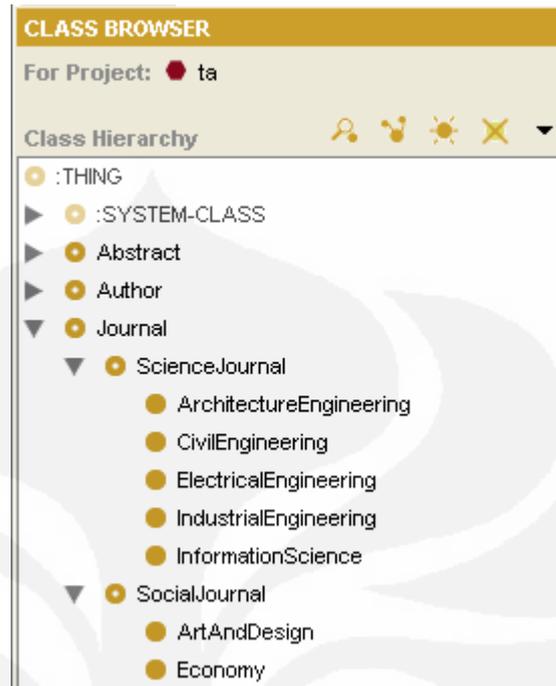
- *Class ScienceAbstract*
 - *Subclass ArchitectureAbstract*, pengelompokkan ringkasan artikel dari kategori teknik arsitektur.
 - *Subclass CivilAbstract*, pengelompokkan ringkasan artikel dari kategori teknik sipil.
 - *Subclass ElectricalAbstract*, pengelompokkan ringkasan artikel dari kategori teknik elektro.

- *Subclass IndustrialAbstract*, pengelompokkan ringkasan artikel dari kategori teknik industri.
- *Subclass InformationScience*, pengelompokkan ringkasan artikel dari kategori Ilmu Informasi.
- *Class SocialAbstract*
 - *Subclass ArtAndDesign*, pengelompokkan ringkasan artikel dari kategori Kesenian dan Desain.
 - *Subclass Economy*, pengelompokkan ringkasan artikel dari kategori Ekonomi.



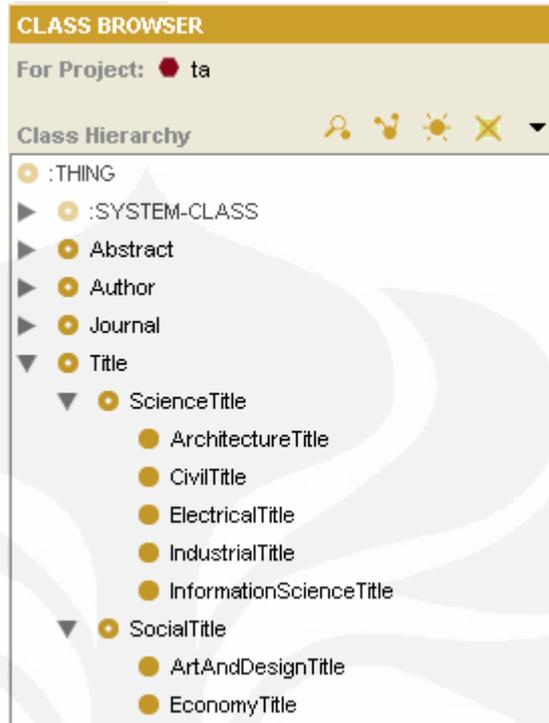
Gambar 4.5. *Superclass Author* dan *subclassnya*

- *Class NameAuthor*
 - *Subclass ScienceAuthor*, pengelompokkan penulis artikel dari kategori *Science*.
 - *Subclass SocialAuthor*, pengelompokkan penulis artikel dari kategori *Social*.
- *Class OriginatingAuthor*
 - *Subclass ScienceOriginating*, pengelompokkan instansi penulis artikel berasal dari kategori *Science*.
 - *Subclass SocialOriginating*, pengelompokkan instansi penulis artikel berasal dari kategori *Social*.



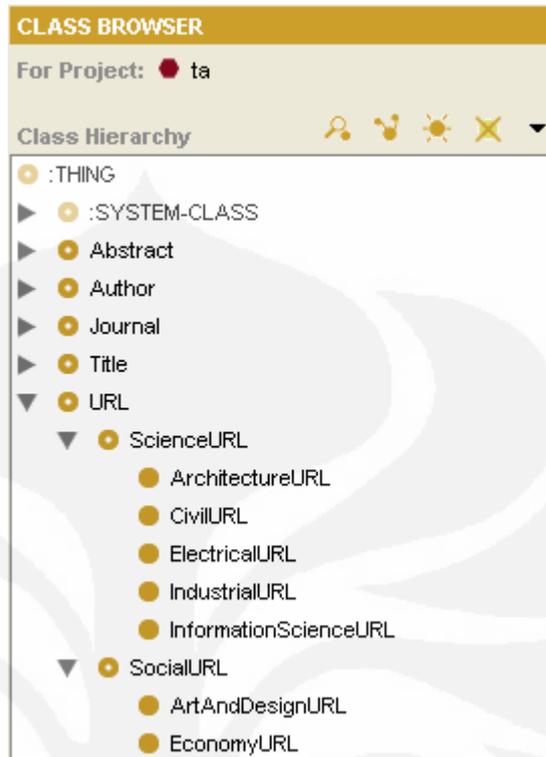
Gambar 4.6. Superclass Journal dan subclassnya

- *Class ScienceJournal*
 - *Subclass ArchitectureEngineering*, pengelompokkan jurnal dari kategori teknik arsitektur.
 - *Subclass CivilEngineering*, pengelompokkan jurnal dari kategori teknik sipil.
 - *Subclass ElectricalEngineering*, pengelompokkan jurnal dari kategori teknik elektro.
 - *Subclass IndustrialEngineering*, pengelompokkan jurnal dari kategori teknik industri.
 - *Subclass InformationScience*, pengelompokkan dari kategori ilmu informasi.
- *Class SocialJournal*
 - *Subclass ArtAndDesign*, pengelompokkan jurnal dari kategori kesenian dan desain.
 - *Subclass Economy*, pengelompokkan jurnal dari kategori ekonomi.



Gambar 4.7. Superclass *Title* dan subclassnya

- *Class ScienceTitle*
 - *Subclass ArchitectureTitle*, pengelompokkan judul artikel dari kategori teknik arsitektur.
 - *Subclass CivilTitle*, pengelompokkan judul artikel dari kategori teknik sipil.
 - *Subclass ElectricalTitle*, pengelompokkan judul artikel dari kategori teknik elektro.
 - *Subclass IndustrialTitle*, pengelompokkan judul artikel dari kategori teknik industri.
 - *Subclass InformationScienceTitle*, pengelompokkan judul artikel dari kategori ilmu informasi.
- *Class SocialTitle*
 - *Subclass ArtAndDesignTitle*, pengelompokkan judul artikel dari kategori kesenian dan desain.
 - *Subclass EconomyTitle*, pengelompokkan judul artikel dari kategori ekonomi

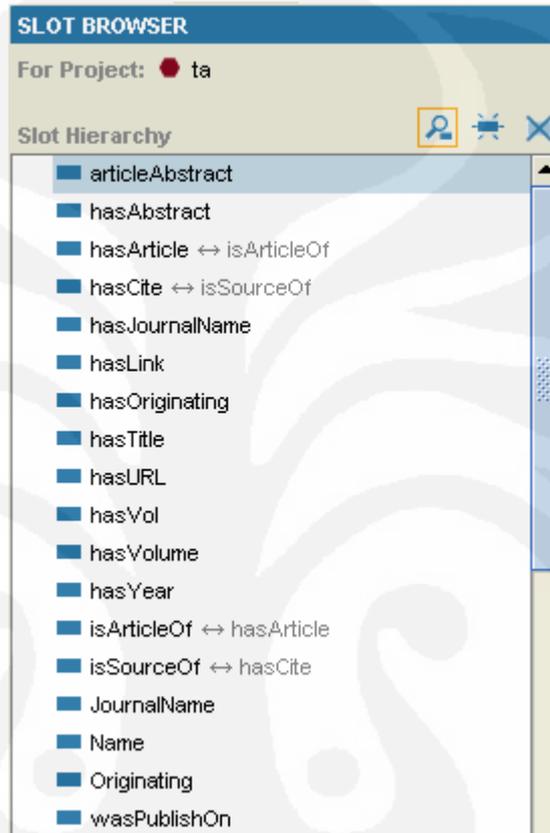


Gambar 4.8. Superclass URL dan subclassnya

- *Class ScienceURL*
 - *Subclass ArchitectureURL*, pengelompokkan URL artikel dari kategori teknik arsitektur.
 - *Subclass CivilURL*, pengelompokkan URL artikel dari kategori teknik sipil.
 - *Subclass ElectricalURL*, pengelompokkan URL artikel dari kategori teknik elektro.
 - *Subclass IndustrialURL*, pengelompokkan URL artikel dari kategori teknik industri.
 - *Subclass InformationScienceURL*, pengelompokkan URL artikel dari kategori ilmu informasi.
- *Class SocialURL*
 - *Subclass ArtAndDesignURL*, pengelompokkan URL artikel dari kategori kesenian dan desain.
 - *Subclass EconomyURL*, pengelompokkan URL artikel dari kategori ekonomi.

4. Mendefinisikan slot atau properties.

Slot atau *properties* yang diimplementasikan adalah *object properties* yang akan menghubungkan *class* dengan *instance* dan hubungan antar *instance*. *Slot* yang terdapat pada ontologi jurnal elektronik adalah sebagai berikut seperti pada Gambar 4.9 :



Gambar 4.9. *Slot* pada ontologi

- *Slot articleAbstract*
 - Menghubungkan *subclass ScienceAbstract* dengan *instance* pada *subclass ScienceAbstract*.
 - Menghubungkan *subclass SocialAbstract* dengan *instance* pada *subclass SocialAbstract*.

Pada *slot articleAbstract* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.
- *Slot hasArticle*
 - Menghubungkan *subclass ScienceAuthor* dengan *instance* pada *subclass ScienceTitle*.

- Menghubungkan *subclass SocialAuthor* dengan *instance* pada *subclass SocialTitle*.

Pada *slot hasArticle* ditentukan *cardinality* bernilai *multiple* sehingga *instance* yang dihubungkan dapat lebih dari satu. *hasArticle* merupakan *inverse slot* dari *isArticleOf*.

- *Slot hasOriginating*

- Menghubungkan *instance* pada *subclass ScienceAuthor* dengan *instance* pada *subclass ScienceOriginating*.
- Menghubungkan *instance* pada *subclass SocialAuthor* dengan *instance* pada *subclass SocialOriginating*.

Pada *slot hasOriginating* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot Name*

- Menghubungkan *subclass ScienceAuthor* dengan *instance* pada *subclass ScienceAuthor*.
- Menghubungkan *subclass SocialAuthor* dengan *instance* pada *subclass SocialAuthor*.

Pada *slot Name* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot JournalName*

- Menghubungkan *subclass ScienceJournal* dengan *instance* pada *subclass ScienceJournal*.
- Menghubungkan *subclass SocialJournal* dengan *instance* pada *subclass SocialJournal*.

Pada *slot JournalName* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasAbstract*

- Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceAbstract*.
- Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *subclass SocialAbstract*.

Pada slot *hasAbstract* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasCite*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceTitle* atau *SocialTitle*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *subclass ScienceTitle* atau *SocialTitle*.

Pada slot *hasCite* ditentukan *cardinality* bernilai *multiple* sehingga *instance* yang dihubungkan dapat lebih dari satu. *Slot hasCite* merupakan *inverse slot* dari *isSourceOf*.

- *Slot hasJournalName*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceJournal*.
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceJournal*.

Pada slot *hasJournalName* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasTitle*
 - Menghubungkan *subclass ScienceTitle* dengan *instance* pada *subclass ScienceTitle*.
 - Menghubungkan *subclass SocialTitle* dengan *instance* pada *subclass SocialTitle*.

Pada slot *hasTitle* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasURL*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceURL*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *subclass SocialURL*.

Pada slot *hasURL* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasVol*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *class Volume*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *class Volume*.

Pada *slot hasURL* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.
- *Slot isArticleOf*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceAuthor* atau *SocialAuthor*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *subclass ScienceAuthor* atau *SocialAuthor*.

Pada *slot isArticleOf* ditentukan *cardinality* bernilai *multiple* sehingga *instance* yang dihubungkan dapat lebih dari satu. *Slot isArticleOf* merupakan *inverse slot* dari *hasArticle*.
- *Slot isSourceOf*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *subclass ScienceTitle* atau *SocialTitle*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *subclass ScienceTitle* atau *SocialTitle*.

Pada *slot isSourceOf* ditentukan *cardinality* bernilai *multiple* sehingga *instance* yang dihubungkan dapat lebih dari satu. *Slot isSourceOf* merupakan *inverse slot* dari *hasCite*.
- *Slot wasPublishOn*
 - Menghubungkan *instance* pada *subclass ScienceTitle* dengan *instance* pada *class Year*.
 - Menghubungkan *instance* pada *subclass SocialTitle* dengan *instance* pada *class Year*.

Pada *slot wasPublishOn* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

- *Slot hasLink*
hasLink Menghubungkan *Class Year* dengan *instance* pada *class Year*. Pada *slot hasLink* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.
- *Slot hasVolume*
hasVolume menghubungkan *Class Volume* dengan *instance* pada *class Volume*. Pada *slot hasVolume* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.
- *Slot hasYear*
hasYear Menghubungkan *Class Year* dengan *instance* pada *class Year*. Pada *slot hasYear* ditentukan *cardinality* bernilai *single* sehingga hanya sebuah *instance* yang dapat dihubungkan.

5. Membuat instance

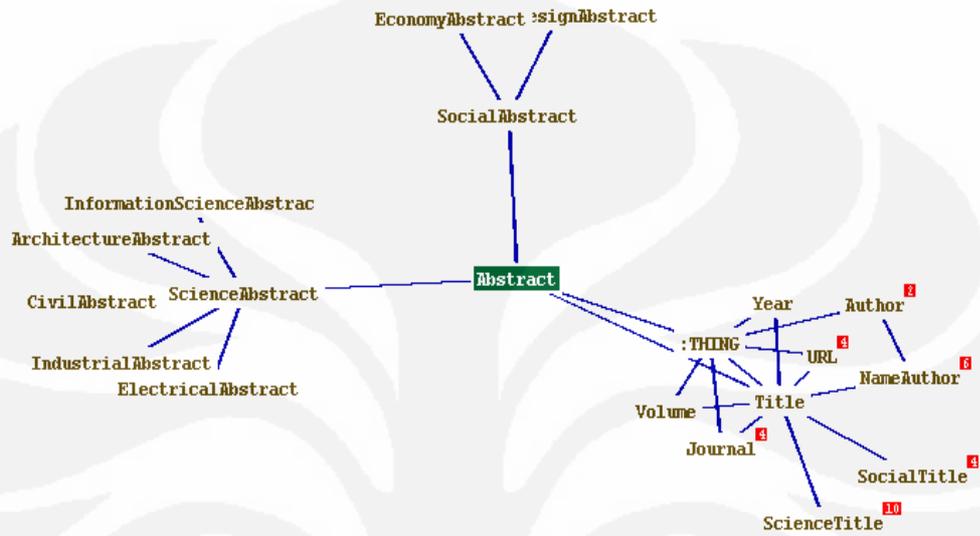
Tahapan terakhir adalah membuat individual atau *instances* pada *class* dalam suatu hirarki. Gambar 4.10 berikut menunjukkan *instance* pada *class EconomyAuthor*.



Gambar 4.10 *Instance* pada *EconomyAuthor*

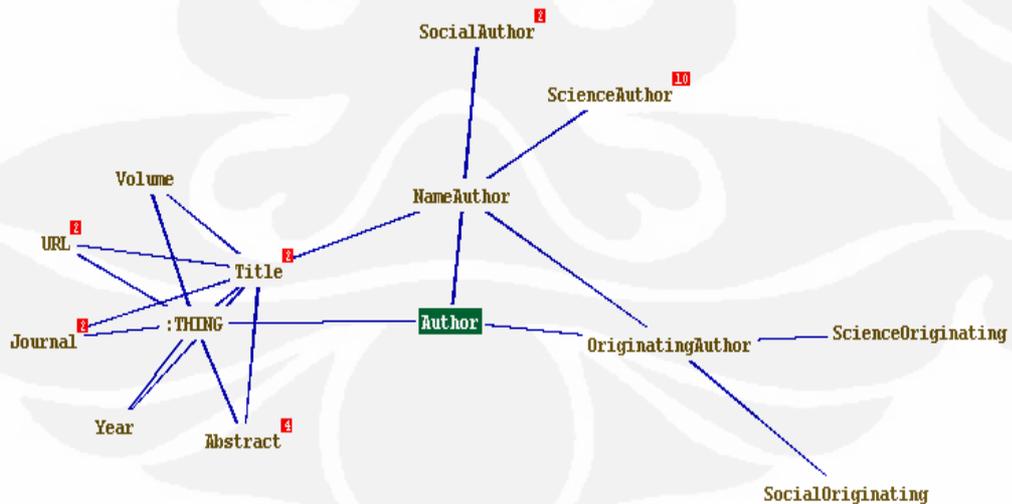
4.1.2. Visualisasi Ontologi

1. Visualisasi hubungan *class Abstract* dengan *class* lainnya serta dengan *subclassnya*.



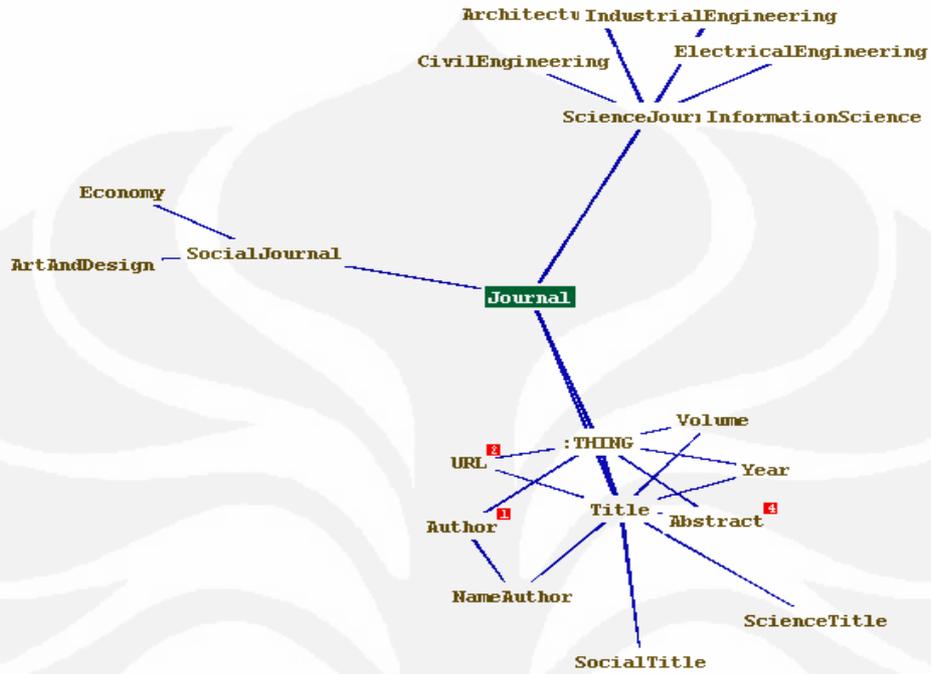
Gambar 4.11. *Class Abstract*

2. Visualisasi hubungan *class Abstract* dengan *class* lainnya serta dengan *subclassnya*.



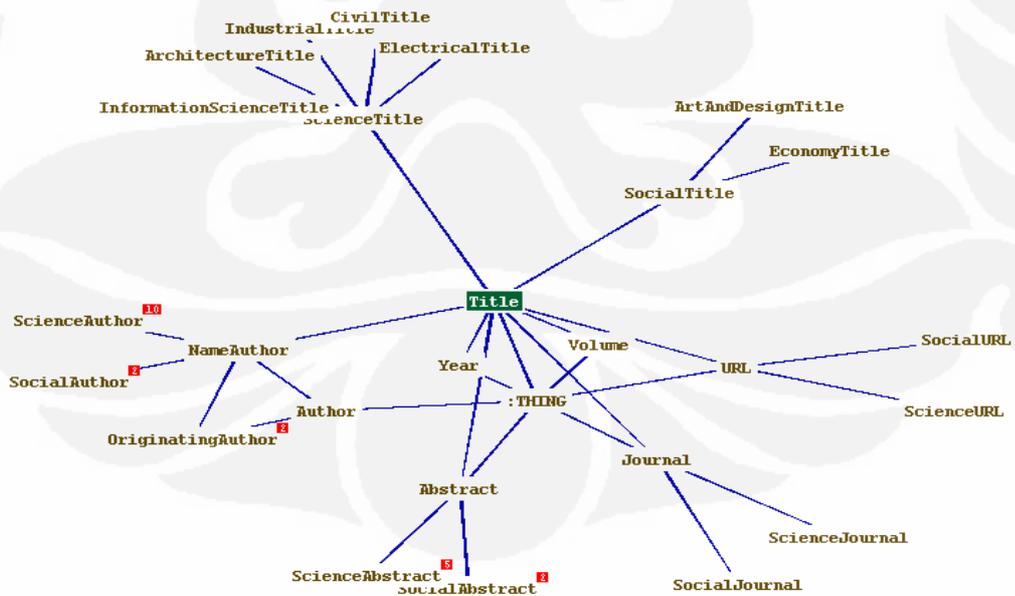
Gambar 4.12. *Class Author*

3. Visualisasi hubungan *class Journal* dengan *class* lainnya serta dengan *subclassnya*.



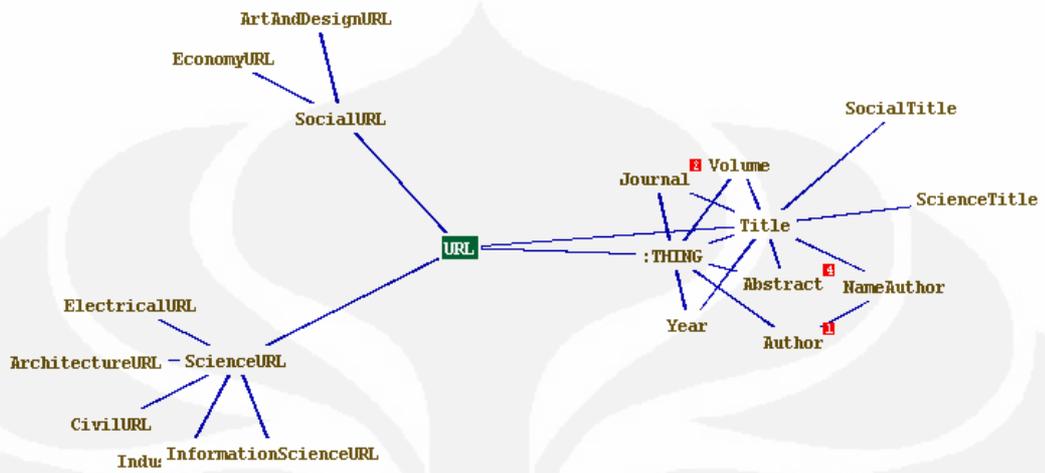
Gambar 4.13. *Class Journal*

4. Visualisasi hubungan *class Title* dengan *class* lainnya serta dengan *subclassnya*.



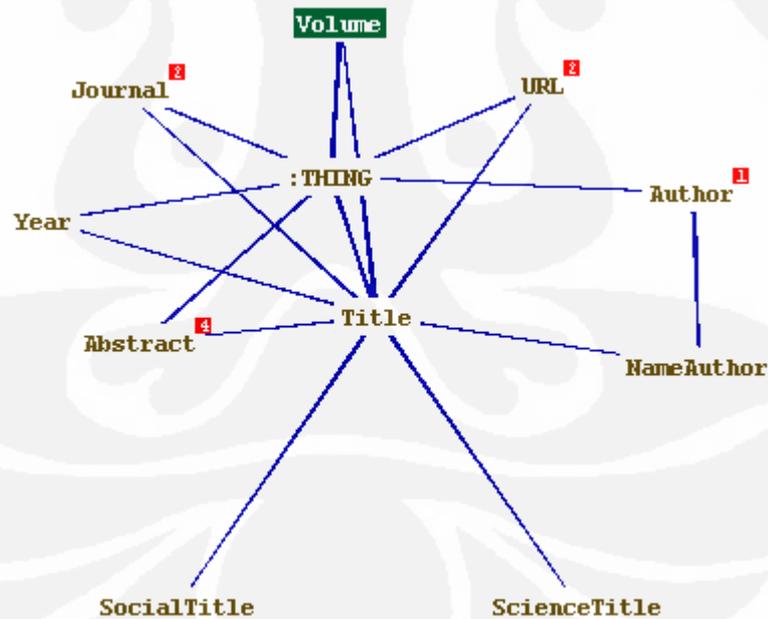
Gambar 4.14. *Class Title*

5. Visualisasi hubungan *class* URL dengan *class* lainnya serta dengan *subclassnya*.



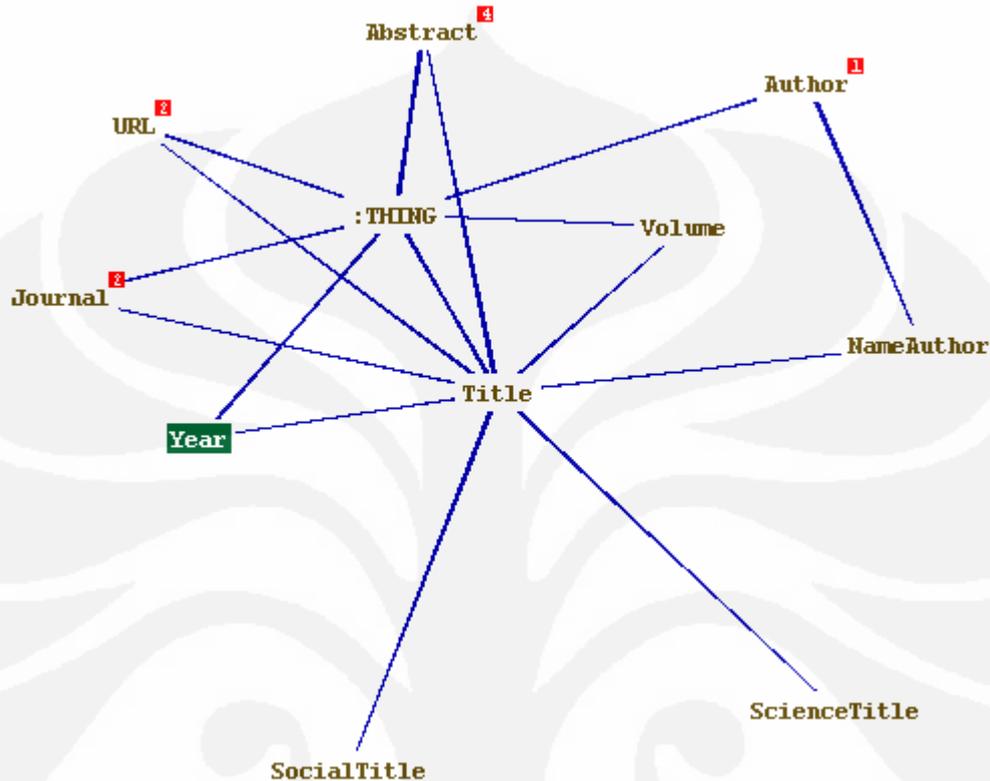
Gambar 4.15. *Class URL*

6. Visualisasi hubungan *class* Volume dengan *class* lainnya serta dengan *subclassnya*.



Gambar 4.16. *Class Volume*

7. Visualisasi hubungan *class Year* dengan *class* lainnya serta dengan *subclassnya*.



Gambar 4.17. *Class Year*

4.2. IMPLEMENTASI WEB

Implementasi web menggunakan perangkat lunak XAMPP yang terdiri dari Apache sebagai server, PHP sebagai bahasa pemrograman dan MySQL sebagai *database*.

4.2.1. Implementasi *Parsing Model Ontologi*

Setelah model ontologi selesai dibuat dengan Protégé, selanjutnya ontologi disimpan dalam file berekstensi XML. File XML tersebut merupakan data mentah yang akan diproses melalui tahapan *parsing* agar dapat disimpan ke dalam *database* MySQL. *Parsing* dijalankan dengan menggunakan script PHP sehingga model ontologi dapat disesuaikan dengan kebutuhan sistem sitasi jurnal elektronik Indonesia.

Berikut ini merupakan salah satu kutipan model ontologi dalam bentuk XML dalam domain jurnal elektronik :

```

<class>
  <name>ScienceAbstract</name>
  <type>:STANDARD-CLASS</type>
  <own_slot_value>
    <slot_reference>:ROLE</slot_reference>
    <value value_type="string">Abstract</value>
  </own_slot_value>
  <superclass>Abstract</superclass>
</class>

<class>
  <name>CivilAbstract</name>
  <type>:STANDARD-CLASS</type>
  <own_slot_value>
    <slot_reference>:ROLE</slot_reference>
    <value value_type="string">Concrete</value>
  </own_slot_value>
  <superclass>ScienceAbstract</superclass>
</class>

<slot>
  <name>hasAbstract</name>
  <type>:STANDARD-SLOT</type>
  <own_slot_value>
    <slot_reference>:SLOT-MAXIMUM-
CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>
  <own_slot_value>
    <slot_reference>:SLOT-VALUE-TYPE</slot_reference>
    <value value_type="string">Instance</value>
    <value value_type="class">Abstract</value>
  </own_slot_value>
</slot>

<simple_instance>
  <name>ta_Class30103</name>
  <type>CivilTitle</type>
  <own_slot_value>
    <slot_reference>hasAbstract</slot_reference>
    <value value_type="simple_instance">ta_Class30110</value>
  </own_slot_value>
  <own_slot_value>
    <slot_reference>isArticleOf</slot_reference>
    <value value_type="simple_instance">ta_Class30104</value>
  </own_slot_value>
  <own_slot_value>
    <slot_reference>hasJournalName</slot_reference>
    <value value_type="simple_instance">ta_Class23</value>
  </own_slot_value>
  <own_slot_value>
    <slot_reference>hasTitle</slot_reference>

```

```

        <value value_type="string">Application of Packing Theory on
Grading Design for Porous Asphalt Mixtures</value>
    </own_slot_value>
    <own_slot_value>
        <slot_reference>hasURL</slot_reference>
        <value value_type="simple_instance">ta_Class30109</value>
    </own_slot_value>
    <own_slot_value>
        <slot_reference>hasVol</slot_reference>
        <value value_type="simple_instance">ta_Class30038</value>
    </own_slot_value>
    <own_slot_value>
        <slot_reference>isSourceOf</slot_reference>
        <value value_type="simple_instance">ta_Class30100</value>
    </own_slot_value>
    <own_slot_value>
        <slot_reference>wasPublishOn</slot_reference>
        <value value_type="simple_instance">ta_Class20036</value>
    </own_slot_value>
</simple_instance>

```

Serangkaian tag XML diatas menggambarkan :

1. *Instance* pada *class CivilTitle* memiliki *name* ta_Class30103 yang didefinisikan oleh *slot hasTitle* memiliki judul artikel Application of Packing Theory on Grading Design for Porous Asphalt Mixtures.
2. ta_Class30103 memiliki artikel abstrak dengan *name* ta_Class30110 yang dihubungkan oleh *slot hasAbstract*.
3. ta_Class30103 memiliki penulis artikel dengan *name* ta_Class30104 yang dihubungkan oleh *slot isArticleOf*.
4. ta_Class30103 memiliki jurnal dengan *name* ta_Class23 yang dihubungkan oleh *slot hasJournalName*.
5. ta_Class30103 memiliki URL dengan *name* ta_Class30109 yang dihubungkan oleh *slot hasURL*.
6. ta_Class30103 memiliki volume terbit dengan *name* ta_Class30038 yang dihubungkan oleh *slot hasVol*.
7. ta_Class30103 memiliki artikel yang mensitasi *name* ta_Class30103 dengan *name* ta_Class30100 yang dihubungkan oleh *slot isSourceOf*.
8. ta_Class30103 memiliki tahun terbit dengan *name* ta_Class20036 yang dihubungkan oleh *slot wasPublishOn*.

Adapun pendefinisian dari *instance* yang masih dalam bentuk *name* ke dalam bentuk *string* dapat digambarkan pada model ontologi berikut :

```
<simple_instance>
```

```

<name>ta_Class30110</name>
<type>CivilAbstract</type>
<own_slot_value>
  <slot_reference>articleAbstract</slot_reference>
  <value value_type="string">The design life of porous asphalt is
shorter than dense mix as a.....</value>
</own_slot_value>
</simple_instance>

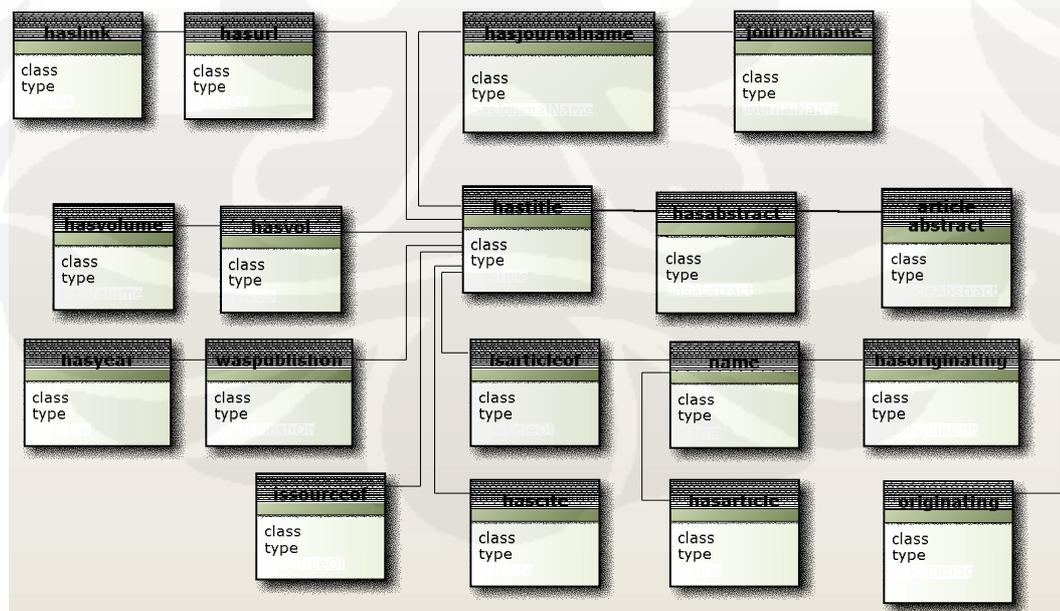
```

Serangkaian tag XML diatas menggambarkan *instance* dengan *name* ta_Class30110 merupakan *instance* dari class CivilAbstract yang memiliki nilai The design life of porous asphalt is shorter than dense mix as a.....

Sehingga apabila serangkaian tag XML diatas dihubungkan dengan serangkaian tag XML yang pertama, akan didapat suatu hubungan Application of Packing Theory on Grading Design for Porous Asphalt Mixtures memiliki artikel abstrak The The design life of porous asphalt is shorter than dense mix as a.....

Dari model ontologi XML dibuat fungsi *parsing* yang dapat mengakomodir hubungan *class*, *slot* dan *instancenya*. Hasil dari *parsing* tersebut adalah sebuah *database* yang terdiri dari 18 tabel dimana tabel-tabel tersebut diekstrak dari setiap *slotnya*. Setiap table terdiri dari 3 kolom, yaitu kolom *class* yang berasal dari tag *name*, kolom *type* yang diekstrak dari tag *type* dan kolom *slot reference* yang diekstrak dari tiap *value slot reference*.

Database hasil *parsing* beserta relasi antar tabel digambarkan pada Gambar 4.18 berikut.



Gambar 4.18 Database dan relasi antar tabel

4.2.2. Implementasi *Domain Expert Interface*

Sebelum proses *upload* ke *directory server* dan *parsing* model ontologi, dilakukan proses autentikasi sehingga hanya administrator yang berhak dalam melakukan update ke database MySQL.

4.2.3. Implementasi *User Interface*

Setelah *parsing* file XML berhasil dilakukan, selanjutnya diimplementasikan halaman web sebagai interface bagi user untuk melakukan pencarian. Dalam pembuatannya digunakan PHP.

Pada halaman web terdapat beberapa menu utama pencarian yaitu pencarian berdasarkan judul artikel yang ditunjukkan dengan *radio button Title* dan pencarian berdasarkan penulis artikel yang ditunjukkan dengan *radio button Author*. Sebelum melakukan pencarian pengguna harus memilih salah satu *radio button* tersebut.

Setelah pemilihan *radio button* dilakukan, pengguna dihadapkan pada pilihan kategori yang ditunjukkan dengan *checkbox*. Kategori tersebut terdiri dari *All*, dimana user melakukan pencarian terhadap semua jenis kategori. Kategori *Art and Design* dikhususkan untuk melakukan pencarian pada judul atau penulis dari kategori *Art and Design*. Hal tersebut juga berlaku untuk kategori *Economy*, *Civil Engineering*, *Electrical Engineering*, *Industrial Engineering* dan *Information Science*.

Pengguna yang akan melakukan pencarian akan menempatkan kata pencarian pada *textbox* yang tersedia dan menekan tombol *Search*. Apabila pencarian ditemukan maka akan muncul informasi pencarian sebagai berikut :

- Pada baris pertama akan ditampilkan judul artikel hasil pencarian.
- Pada baris kedua akan ditampilkan penulis dari artikel, jurnal dari artikel tersebut, tahun dan volume artikel diterbitkan.
- Pada baris ketiga akan ditampilkan abstraksi dari artikel.
- Pada baris keempat akan ditampilkan jumlah artikel yang mensitasi dari artikel pada baris pertama. Apabila jumlah tersebut ditekan maka akan ditampilkan artikel yang mensitasi tersebut.

BAB 5

PENGUJIAN ONTOLOGI WEB DAN APLIKASI SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA

5.1. PENGUJIAN FUNGSIONALITAS

5.1.1. Pengujian Fungsionalitas Sistem

Untuk mengetahui apakah sistem telah berjalan sesuai tujuan awal, maka dilakukan pengujian fungsionalitas sistem. Pengujian ini dilakukan pada aplikasi *domain expert* atau pustakawan dan pada aplikasi *user* yang sedang melakukan pencarian.

5.1.1.1. Pengujian Fungsionalitas Sistem *Domain Expert*

Pada aplikasi *domain expert* pengujian dilakukan saat *update* pemodelan ontologi dan saat koneksi ke server untuk *upload* dan *parsing* model ontologi ke dalam *database*. Dengan menggunakan Protégé 3.4, pemodelan ontologi dapat dilakukan dalam berbagai platform mulai dari windows, Linux bahkan dalam Mac OS. Protégé 3.4 juga menyediakan berbagai *plug-in* yang dapat memudahkan pustakawan melakukan pemodelan, salah satu *plug-in* yang digunakan adalah TGVizTab dimana pustakawan dapat memvisualisasikan pemodelan dari radius terkecil (0) yang hanya menampilkan sebuah *class* hingga radius terbesar (10) yang dapat menampilkan pemodelan ontologi secara keseluruhan.

Update data pada ontologi dilakukan secara manual oleh pustakawan yang diawali dengan membuka situs yang menjadi acuan dalam melakukan pemodelan, yaitu :

- <http://ejournal.unud.ac.id>
- <http://puslit2.petra.ac.id/ejournal>
- <http://journal.ui.ac.id>

Pada situs-situs tersebut, dipilih jurnal-jurnal yang sesuai dengan kategori pemodelan. Dari jurnal tersebut, tiap artikel dalam bentuk PDF di buka dan didapat informasi berupa abstrak, penulis, nama jurnal, judul, URL, volume dan tahun terbit. Informasi tersebut dimasukkan dalam Protégé melalui *instance browser* bertipe *string* sesuai dengan *class* masing-masing.

Instance bertipe *string* yang sudah dimasukkan akan menjadi acuan untuk *slot* lainnya dalam menghubungkan antar *instance* yaitu dengan melakukan pemilihan pada *instance* yang dimaksud tanpa melakukan pengetikan ulang. Hal tersebut dapat mempersingkat waktu dalam melakukan pengisian data.

Pemodelan ontologi memberikan ekspresi makna sintaksis dari simbol *class*, *slot* dan *instance* sehingga ontologi menjadi suatu aplikasi semantik karena dapat menggambarkan semua isi dari dokumen tersebut. Pemodelan ontologi dalam bentuk XML yang dapat dikatakan sebagai aturan sintaks untuk membuat bahasa *markup* yang lebih semantik pada domain tertentu. Dengan menggunakan ontologi, pustakawan akan lebih mudah dalam pengisian data karena klasifikasi dan pembatasan yang jelas.

Setelah pengisian data selesai dibuat, pustakawan menyimpan pemodelan ontologi dalam file XML. File XML ini yang nantinya akan di *upload* ke *directory server* dengan proses autentikasi terlebih dahulu. Proses autentikasi dimulai dengan menyetikkan URL *login.htm* pada web *browser*. Bila koneksi terhubung maka akan muncul tampilan *login* seperti pada Gambar 5.1 berikut :



Gambar 5.1 Tampilan halaman *login*

Pada tampilan *login* inilah terjadi proses autentikasi pengguna. Pengguna diminta memasukkan *username* dan *password* untuk kemudian diverifikasi apakah data yang dimasukkan sesuai dengan *login* pada *database*. Jika sesuai,

maka akan muncul tampilan halaman untuk melakukan *parsing* ontologi XML ke dalam *database* seperti pada Gambar 5.2.

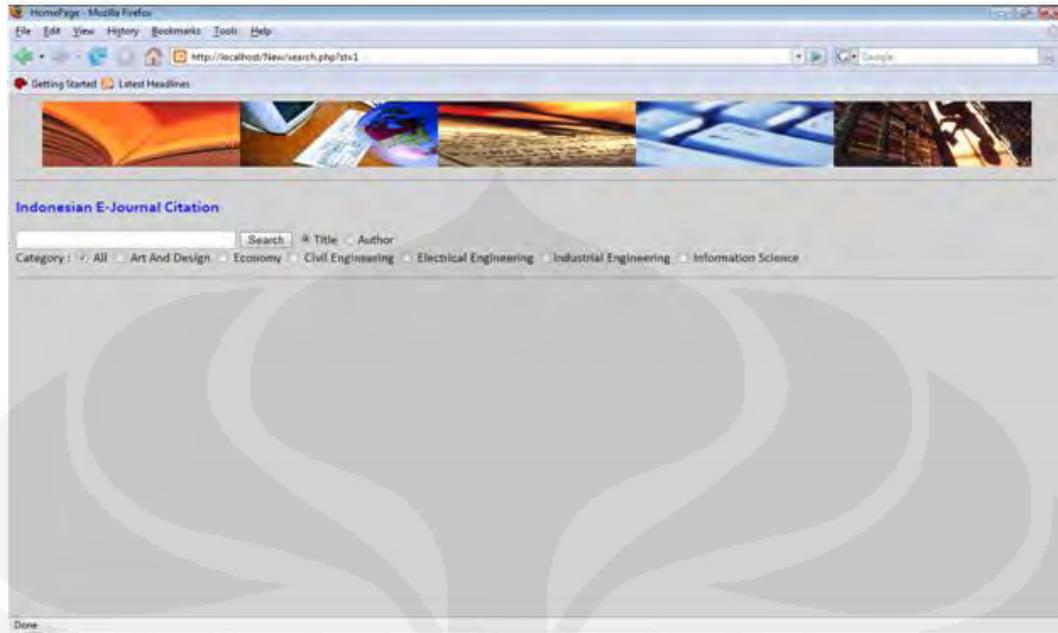


Gambar 5.2 Tampilan halaman *upload* dan *parsing* ontologi

Pada halaman *upload* dan *parsing* ini, pustakawan dapat meng-*upload* file berekstensi XML serta melakukan *parsing* ke dalam *database* agar pemodelan ontologi yang telah dilakukan dapat diakomodir untuk sistem sitasi jurnal elektronik Indonesia.

5.1.1.2. Pengujian Fungsionalitas Sistem Pengguna

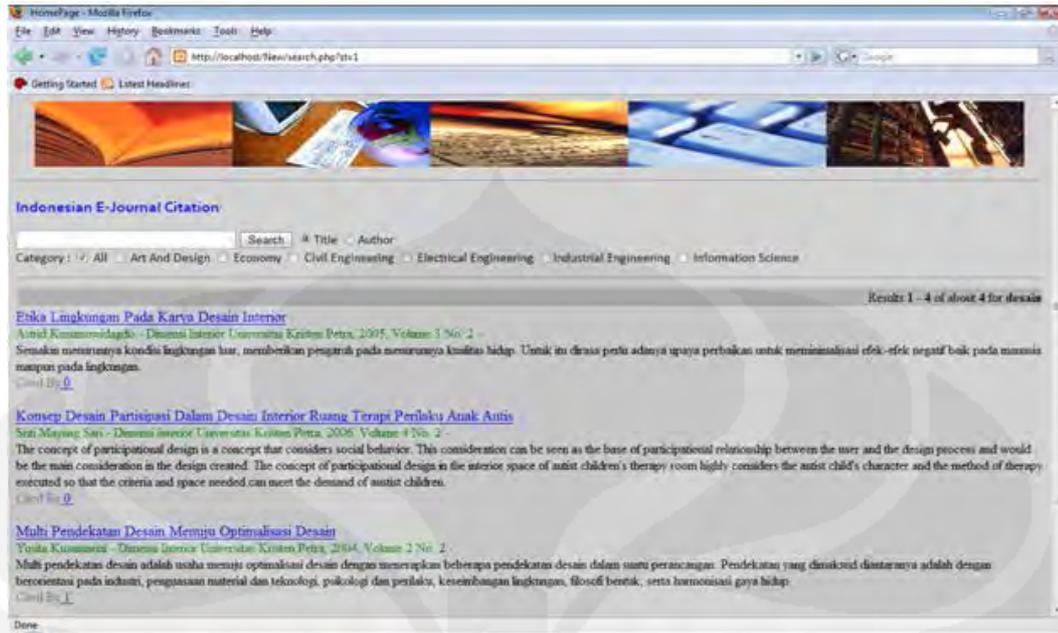
Pada sisi pengguna, proses pencarian diawali dengan menyetikkan URL pada *web browser*. Bila koneksi dengan server berhasil, maka akan muncul tampilan seperti pada Gambar 5.3.



Gambar 5.3 Tampilan halaman pencarian

User melakukan pencarian berdasarkan pilihan pada *radio button* yaitu *Title* dan *Author*. Selanjutnya pengguna dihadapkan oleh beberapa pilihan *checkbox* yaitu *All* (semua kategori), *Art and Design* (kesenian dan desain), *Economy* (ekonomi), *Civil Engineering* (teknik sipil), *Electrical Engineering* (teknik elektro), *Industrial Engineering* (teknik industri) dan *Information Science* (ilmu informasi). Selanjutnya *user* menuliskan kata yang akan dilakukan pencarian pada *text box* yang telah disediakan. Ketika *user* menekan tombol *Search* maka akan dilakukan penelusuran terhadap *radio button* dan *checkbox* yang telah dipilih sebelumnya.

Contoh kasus adalah seorang *user* melakukan pencarian terhadap kata “desain” dengan menggunakan pilihan *radio button Title* dan *checkbox Art and Design*. Hasil yang didapat dari pencarian tersebut seperti pada Gambar 5.4.

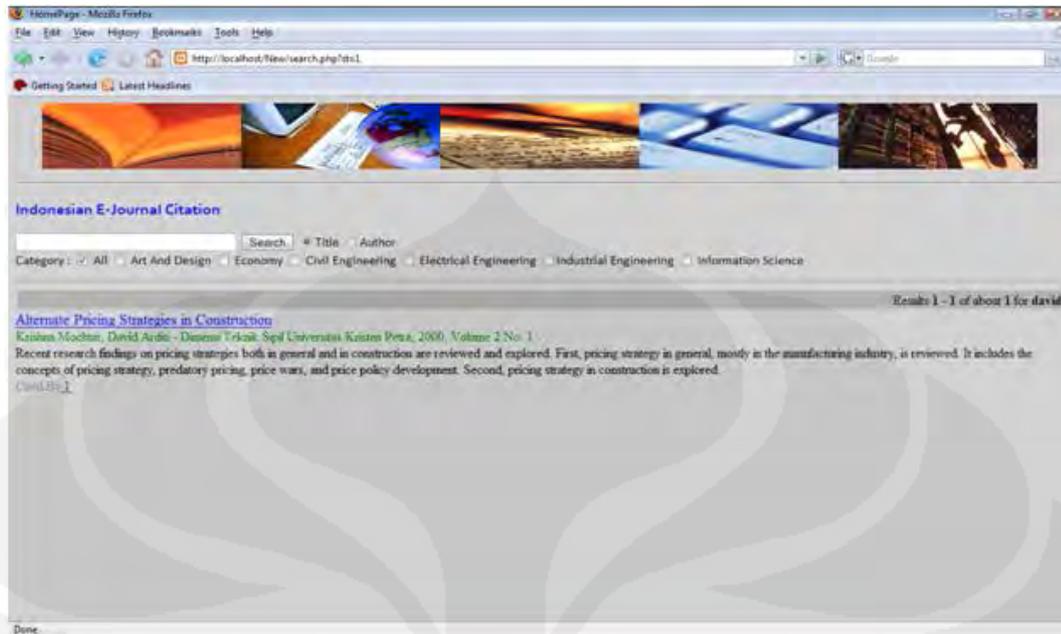


Gambar 5.4 Hasil pencarian kata “desain” pada kategori *Art and Design*
 Dari hasil pencarian kata “desain” didapat 3 artikel yang mengandung kata desain.
 Artikel-artikel tersebut adalah :

1. Etika Lingkungan Pada Karya **Desain** Interior
2. Konsep Desain Partisipasi Dalam **Desain** Interior Ruang Terapi Perilaku Anak Autis.
3. Multi Pendekatan Desain Menuju Optimalisasi **Desain** (Interior)

Ketiga artikel tersebut dilengkapi dengan informasi nama penulis, jurnal yang menerbitkan, tahun dan volume diterbitkan, abstraksi, jumlah artikel tersebut disitasi dan *hyperlink* judul artikel ke server tempat penyimpanan artikel dalam bentuk PDF. Ketiga artikel diatas belum pernah disitasi oleh artikel lainnya sehingga keterangan Cited By bernilai 0.

Contoh kasus lain adalah user melakukan pencarian terhadap kata “david” dengan menggunakan pilihan *radio button Author* dan *checkbox Civil Engineering*. Hasil yang didapat dari pencarian tersebut seperti pada Gambar 5.5.



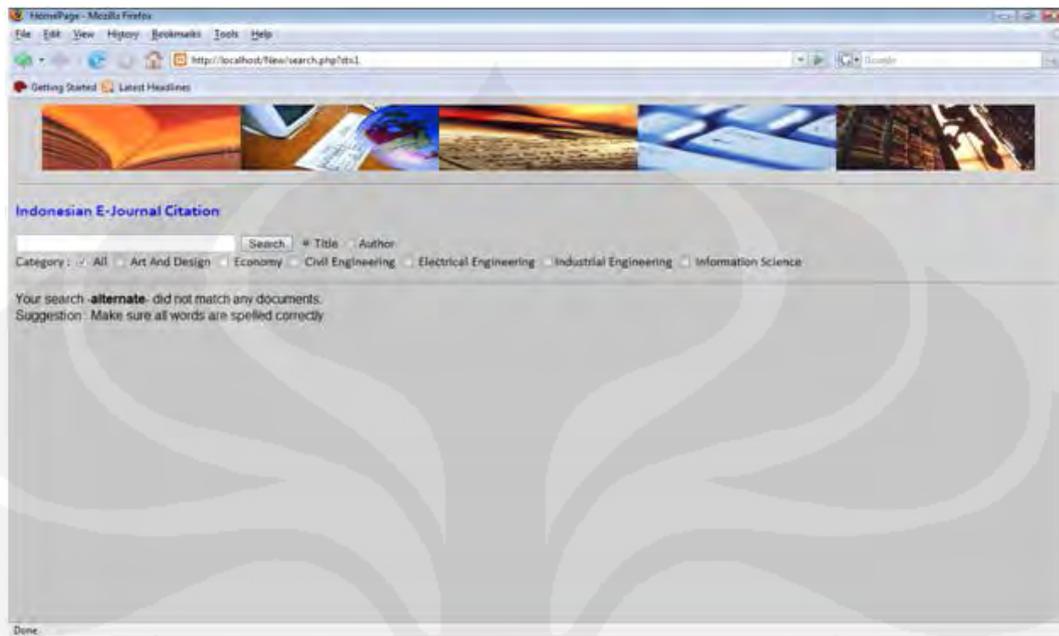
Gambar 5.5 Hasil pencarian kata “david” pada kategori *Civil Engineering*

Dari hasil pencarian kata “david” didapat 1 artikel yang mengandung kata david. Artikel tersebut adalah *Alternate Pricing Strategies in Construction* yang dilengkapi dengan informasi nama penulis, jurnal yang menerbitkan, tahun dan volume diterbitkan, abstraksi serta jumlah artikel tersebut disitasi. Artikel disitasi sebanyak 1 kali. Apabila angka 1 ditekan, maka akan muncul artikel yang mensitasi *Alternate Pricing Strategies in Construction*. Hasil setelah penekanan *Cited By* seperti pada Gambar 5.6.



Gambar 5.6 Hasil sitasi *Alternate Pricing Strategies in Construction*

Apabila pencarian tidak sesuai maka akan muncul pernyataan bahwa kata tidak ditemukan seperti pada Gambar 5.7.



Gambar 5.7 Pencarian tidak ditemukan

Pencarian diatas dilakukan terhadap kata *alternate* pada radio button *Title* dan kategori *Art and Design*. Dari hasil pengujian fungsionalitas, hasil pencarian sesuai dengan pemodelan ontologi yang telah dilakukan.

5.1.2. Pengujian Pengguna

Pengujian pengguna dilakukan melalui survey kepada pengguna dengan memberikan kuesioner. Kuesioner adalah kumpulan pertanyaan yang kaitannya dengan performansi produk sistem. Ada dua jenis pengguna yaitu pengguna sebagai pustakawan dan pengguna yang melakukan pencarian. Pengukuran ini menggunakan skala 1 – 4 terhadap pernyataan yang diberikan yang merepresentasikan 1 = sangat tidak setuju, 2 = tidak setuju, 3 = setuju, 4 = sangat setuju . *Survey* dilakukan melalui kuesioner yang diberikan kepada 10 orang.

5.1.2.1. Pengguna Sebagai *Domain Expert* atau pustakawan

Pengujian dilakukan untuk mengetahui apakah *Protege* dapat digunakan dengan mudah sebagai *tool* ontologi dan dapat diimplementasikan dalam sistem sitasi jurnal elektronik Indonesia.

Hasil pengujian pengguna sebagai pustakawan yang menggunakan Protege ditunjukkan pada Tabel 5.1.

Tabel 5.1 Hasil Pengujian Pengguna Sebagai Pustakawan

No	Pertanyaan	Jumlah Responden			
		1	2	3	4
1	Saya familiar dengan Protégé.	3	6	1	
2.	Tab - tab dalam editor ontologi perwujudan konsep ontologi mudah untuk dipahami.		2	8	
3	Pengoperasian Protégé mudah untuk dijalankan.		1	8	1
4	Protégé efisien dalam melakukan <i>input</i> data.		2	6	2
5	Sistem mudah serta membantu proses penyimpanan.		1	6	3
6	Cara memasukkan informasi suatu artikel cukup mudah.		3	7	
7	Cara <i>upload</i> ke direktori server dan <i>parsing</i> ke database cukup mudah.		1	9	

Dari data tersebut, diperhitungkan nilai rata-rata jawaban dan nilai standar deviasi. Nilai rata-rata didapat berdasarkan hasil perkalian antara bobot nilai (1 sampai 4) dengan banyaknya responden yang memberikan jawaban. Perhitungan 95% *confidence* interval menggunakan formula sebagai berikut :

$$\text{Stdev} = \frac{\sqrt{\sum (\text{nilai} - \text{mean})^2}}{\text{Populasi} - 1} \dots\dots\dots 1)$$

$$\text{Rata-rata} \pm \frac{1,96 * \text{Stdev}}{\sqrt{\text{Populasi}}} \dots\dots\dots 2)$$

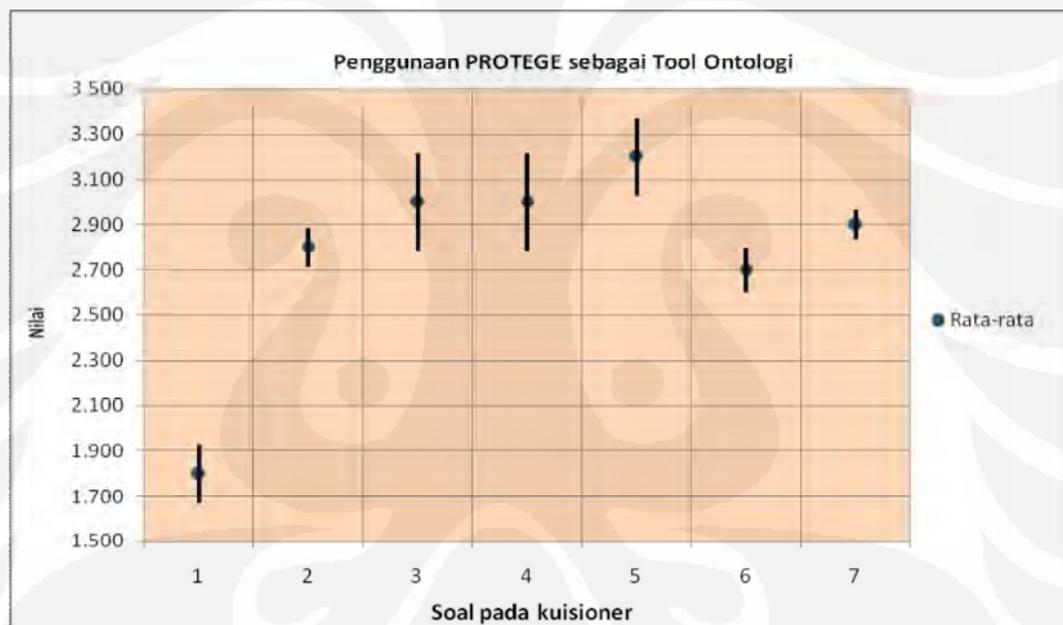
Stdev = Standar deviasi
 Populasi = Banyaknya responden

Hasil perhitungan ini ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil Perhitungan Tanggapan Pengguna Sebagai Pustakawan

No Pertanyaan	Standar Deviasi	Minimum	Rata-rata	Maksimum
1	0.210	1.670	1.800	1.930
2	0.140	2.713	2.800	2.887
3	0.351	2.782	3.000	3.218
4	0.351	2.782	3.000	3.218
5	0.272	3.031	3.200	3.369
6	0.161	2.600	2.700	2.800
7	0.105	2.835	2.900	2.965

Dari perhitungan di atas digambarkan tanggapan terhadap penggunaan Protégé ke dalam grafik yang ditunjukkan oleh Gambar 5.8.



Gambar 5.8 Grafik Tanggapan Penggunaan Protégé Sebagai *Tool* Ontologi

Dari data tanggapan pada pertanyaan pertama mengenai kefamiliaran terhadap Protégé, terlihat sangat sedikit orang yang mengenal Protégé sebagai tool ontologi. Rata-rata yang didapatkan pada pertanyaan yang pertama hanya pada skala 1.8. Hal ini terjadi karena belum banyak pengembangan ontologi yang dilakukan di Indonesia. Kecenderungan penyimpanan informasi tidak dilakukan secara semantik. Untuk mendukung penyimpanan informasi yang lebih bermakna secara semantik atau jika informasi yang disimpan dapat dipandang sebagai pengetahuan yang lebih konkrit dapat dimanfaatkan pemodelan yang berbasis

ontologi. Salah satu tool yang dapat melakukan fungsi-fungsi ontologi adalah Protégé. Dengan menggunakan Protégé, model pengetahuan jurnal elektronik disusun ke dalam bentuk ontologi hirarki *class*, *slot* dan *instance*. Model ontologi yang disusun mampu mendeskripsikan informasi artikel pada jurnal elektronik secara semantik.

Pertanyaan kedua mengacu kepada pengenalan *tab-tab* pada Protégé sebagai perwujudan komponen-komponen ontologi yaitu *class*, *slot* dan *instance*. *Tab-tab* terdiri dari *tab class*, *tab slot* dan *tab instance*. Rata-rata yang didapat dari pertanyaan *tab-tab* dalam editor ontologi mudah dipahami adalah pada skala 2.8. Sebanyak 80% dari responden berpendapat bahwa *tab-tab* dalam editor Protégé mudah dipahami, namun sebanyak 20% responden berpendapat bahwa *tab-tab* tersebut sulit untuk dipahami, hal ini disebabkan oleh pengguna belum memahami keterhubungan antara komponen-komponen tersebut.

Pertanyaan ketiga mengenai pengoperasian pada Protégé yang dilakukan terhadap pengguna yang berperan sebagai pustakawan. Sebelum melakukan pengujian, pengguna telah diberikan pengarahan dalam pengoperasian. Pengguna melanjutkan pengisian data artikel baru yang terdapat pada jurnal elektronik dengan menggunakan model ontologi yang telah dibuat. Rata-rata yang didapat adalah pada skala 3, yang berarti pengguna setuju bahwa Protégé mudah untuk dijalankan. Kemudahan yang didapat dikarenakan *user interface* yang baik dan adanya fasilitas bantuan sehingga pengguna tidak perlu mengikuti training khusus ataupun waktu yang lama untuk mahir terhadap sistem ini.

Pertanyaan keempat sangat berkaitan dengan pertanyaan ketiga dan keenam, yaitu pengujian terhadap keefisienan dalam melakukan pengisian data. Pengisian data dilakukan secara manual, tetapi hal tersebut tidak menyulitkan pengguna karena form pengisian dibuat sesederhana mungkin sehingga pengguna tidak perlu melakukan pengisian berulang-ulang tetapi tetap pada klasifikasi yang jelas di tiap *class* dan *subclass* serta tidak adanya data yang duplikat walaupun data tersebut digunakan berkali-kali. Rata-rata yang didapat adalah pada skala 3 yang menandakan pengguna setuju dengan metode pengisian data melalui Protégé.

Pertanyaan kelima mengenai penyimpanan model ontologi. Rata-rata yang didapat adalah pada skala 3.2 yang menyatakan Protégé membantu dalam proses penyimpanan yang akan digunakan untuk sistem sitasi. Pemodelan dalam aplikasi ini disimpan dalam dokumen XML. Selain XML, berbagai format penyimpanan disediakan oleh Protégé, yaitu OWL, RDF dan HTML. Protégé menyediakan kemudahan *plug and play* yang membuatnya fleksibel untuk pengembangan ontologi.

Pertanyaan keenam, sebanyak 70% dari responden setuju bahwa memasukkan informasi suatu artikel cukup mudah dan 30% dari responden berpendapat sulit dalam memasukkan informasi baru karena pengguna dituntut untuk teliti dan memperhatikan bagian referensi secara detail apakah suatu artikel mensitasi artikel lain dalam jurnal elektronik. Rata-rata pertanyaan ini memiliki skala 2.7.

Pengujian terakhir yang dilakukan pada aplikasi pustakawan adalah pustakawan melakukan *upload* dan *parsing* yang terdapat dalam halaman *interface* dokumen XML dengan *database*. Aktivitas yang dilakukan adalah *upload* dari *folder* PC pustakawan ke direktori *server* dan melakukan penerusan (*parsing*) ke dalam *database*. Rata-rata yang didapat adalah pada 2.9 yang berarti sistem mudah untuk dipahami dan sangat membantu dalam melanjutkan ke aplikasi web *interface* dalam melakukan pencarian.

5.1.2.2. Pengguna Yang Melakukan Pencarian

Pengujian dilakukan untuk mengetahui web berjalan dengan baik dan sesuai dengan fungsi-fungsi yang ada pada web.

Hasil pengujian pengguna yang melakukan pencarian ditunjukkan pada Tabel 5.3.

Tabel 5.3 Hasil Pengujian oleh Pengguna Web

No	Pertanyaan	Jumlah Responden			
		1	2	3	4
1	<i>Interface</i> (tampilan) <i>website</i> menarik.		3	6	1
2.	Bermanfaat untuk mendapatkan informasi.			5	5
3	Informasi yang didapatkan jelas			7	3
4	Penilaian <i>website</i> secara keseluruhan baik.			9	1

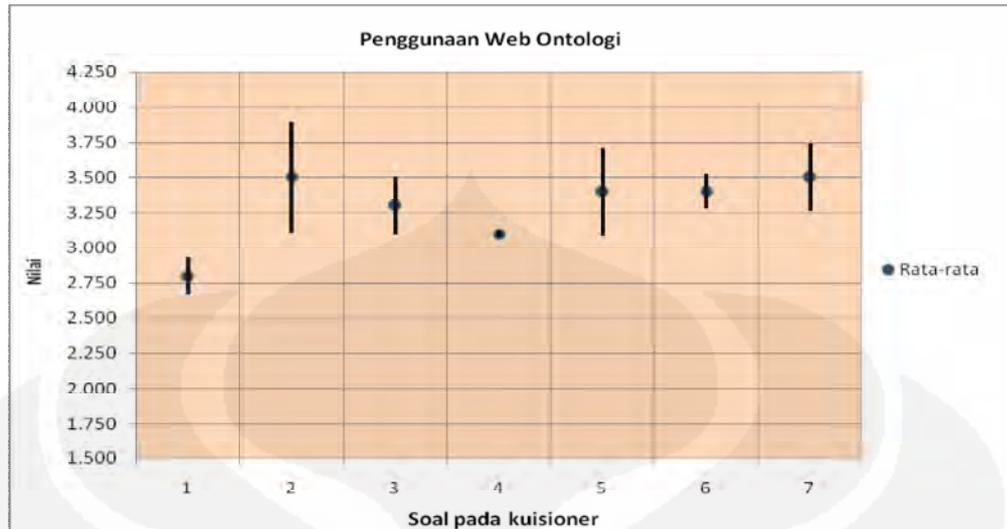
No	Pertanyaan	Jumlah Responden			
		1	2	3	4
5	Sistem dapat membantu pencarian jumlah sitasi suatu artikel.			6	4
6	Sistem dapat membantu pencarian berdasarkan judul artikel.		1	4	5
7	Sistem dapat membantu pencarian berdasarkan penulis artikel.			5	5

Dari data tersebut, diperhitungkan nilai rata-rata jawaban dan nilai standar deviasi. Perhitungan ini ditunjukkan dalam Tabel 5.4.

Tabel 5.4 Hasil Perhitungan Tanggapan Pengguna Web

No Pertanyaan	Standar Deviasi	Minimum	Rata-rata	Maksimum
1	0.21	2.670	2.800	2.930
2	0.63	3.110	3.500	3.890
3	0.33	3.095	3.300	3.505
4	0.03	3.081	3.100	3.119
5	0.5	3.090	3.400	3.710
6	0.2	3.276	3.400	3.524
7	0.39	3.258	3.500	3.742

Dari perhitungan di atas digambarkan tanggapan terhadap penggunaan web sebagai *interface user* dalam melakukan pencarian ke dalam grafik yang ditunjukkan oleh Gambar 5.9.



Gambar 5.9 Grafik Tanggapan Penggunaan Web

Pada bagian kedua, pertanyaan yang diberikan pada sekitar penggunaan web dan aplikasinya dalam sitasi jurnal elektronik Indonesia. Pertanyaan yang diberikan digunakan untuk mengevaluasi kepuasan pelanggan terhadap aplikasi yang dibuat. Untuk hasil aplikasi ini, rata-rata yang dihasilkan adalah lebih dari skala 3. Hasil dapat diartikan bahwa pengguna sudah cukup puas dengan aplikasi yang dibuat dan web dapat melakukan fungsinya sebagai media dalam melakukan pencarian artikel berdasarkan judul, penulis dan kategori yang telah dimodelkan dalam ontologi. Hasil pencarian yang didapat sangat bermanfaat bagi pengguna dan informasi yang ditampilkan jelas sehingga pemodelan ontologi yang dibuat sebelumnya memenuhi persyaratan dalam pembangunan ontologi yaitu menjawab pertanyaan-pertanyaan dari suatu domain.

5.2. PENGEMBANGAN MASA DEPAN

Dari hasil pengujian dan analisa yang dilakukan maka untuk selanjutnya Protege dapat dimanfaatkan sebagai *tool* pemodelan ontologi untuk *semantic web*. *Class*, *slot* dan *instance* yang dibuat secara manual di Protégé dapat dimanfaatkan sebagai kata kunci untuk pencarian pada halaman web. Dalam pengembangan sistem sitasi jurnal elektronik berbasis *semantic web* ini perlu didukung oleh *tool* lain untuk mengekstraksi halaman PDF karena kemampuan Protégé yang hanya dapat melakukan proses penalaran pada halaman web misalnya dengan *plug-in reasoner* Pellet [13].

BAB 6

KESIMPULAN

Berdasarkan implementasi dan pengujian didapatkan beberapa kesimpulan yaitu :

1. Pembangunan ontologi web dan aplikasi semantik telah dapat diimplementasikan sesuai dengan perancangan.
2. Pemodelan ontologi yang dibangun ini dapat digunakan sebagai basis data sistem sitasi jurnal elektronik Indonesia.
3. Pemodelan dengan menggunakan ontologi merupakan pendekatan semantik. Pemodelan ontologi memberikan ekspresi makna sintaksis dari simbol *class*, *slot* dan *instance* sehingga ontologi menjadi suatu aplikasi semantik karena dapat menggambarkan semua isi dari dokumen.
4. Keuntungan semantik adalah waktu yang diperlukan untuk mendapatkan informasi yang dicari lebih singkat.
5. Pemilihan Protégé sebagai tool ontologi didasari oleh produk *open source*, penyimpanan dalam bentuk dokumen XML, adanya buku manual dan *plug-in* visualisasi TGViz yang tersedia.
6. Protégé dinilai sebagai tool ontologi yang dapat digunakan dengan cukup mudah dan dapat diimplementasikan dalam sitasi jurnal elektronik Indonesia walaupun sebelumnya pengguna tidak familiar dengan tool tersebut. Rata – rata nilai pada skala 2,94 dari skala 4 dan memiliki persentase sebesar 73.5 %.
7. Web sebagai *user interface* dalam melakukan pencarian dinilai pengguna memiliki aplikasi yang baik. Nilai rata-rata yang didapat mempunyai skala di atas 3.
8. Dari hasil sitasi, ditemukan 90 artikel dari semua kategori pada jurnal Universitas Udayana dan Universitas Kristen Petra yang memiliki hubungan antar artikel. Hasil sitasi tersebut rata-rata memiliki penulis artikel yang sama sebagai pengembangan dari karya ilmiah sebelumnya.

DAFTAR ACUAN

- [1] Sutarman, *Membangun Aplikasi Web dengan PHP & MySQL*, Penerbit Graha Ilmu, 2007.
- [2] Niko Ibrahim, Pengembangan Aplikasi Semantic Web Untuk Membangun Web Yang Lebih Cerdas, *Jurnal Informatika* Vol. 3 No.1, 2007, http://www.itmaranatha.org/jurnal/jurnal.informatika/Jurnal/Juni2007/artikel/artikelpdf/juni07_3.pdf. Diakses 19 Februari 2008.
- [3] I Wayan Simri Wicaksana, Lintang Y. Banowosari, Kris Triyantio, *Pengujian tool ontologi engineering*, hal. 2, <http://ftp.gunadarma.ac.id/research/WorkGroupInformationSystem/LaTeX/ExampleLaTeX/NakertransS2/ContohBahasa2Kolom/ContohBahasa2Kolom.pdf>. Diakses 19 Februari 2009, dari website Gunadarma.
- [4] Eliza Sachs (2006), *Getting Started with Protégé-Frames*, <http://protege.stanford.edu>. Diakses 10 Maret 2009. dari website Stanford University
- [5] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe, (2004) *A Practical Guide to Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0*, <http://protege.stanford.edu> Diakses 8 Maret 2009 dari website Stanford University
- [6] Natalya F. Noy, Deborah L. McGuinness, *Ontology Development 101 : A Guide to Creating Your First Ontology.*, hal. 1, <http://protege.stanford.edu>. Diakses 17 Mei 2009 dari website Stanford University
- [7] Semantic Web, www.wikipedia.com diakses tanggal 5 April 2009.
- [8] Soren Auer, *Powl – A Web Based Platform for Collaborative Semantic Web Development*, hal. 2, www.powl.com Diakses 20 Mei 2009, dari website Powl.
- [9] *DL Query Tab*, hal. 1, <http://protegewiki.stanford.edu/index.php/DLQueryTab> Diakses 20 Mei 2009, dari website protegewiki.
- [10] Leigh Dodds, *Introducing SPARQL: Querying the Semantic Web.*, hal. 1, <http://www.xml.com>. Diakses 17 Mei 2009 dari website XML.
- [11] Dave Beckett, Jeen Broekstra, *SPARQL Query Results XML Format*, hal. 1, <http://www.w3.org>. Diakses 15 Mei 2009 dari website W3C.

- [12] Radoslaw Oldakowski, Christian Bizer, Daniel Westphal, *RAP: RDF API for PHP*, hal. 1, <http://www.w3.org>. Diakses 15 Mei 2009 dari website W3C.
- [13] Holger Knublauch, Olivier Dameron, Mark A. Musen, *Weaving the Biomedical Semantic Web with the Protégé OWL Plugin*, hal. 1, <http://protege.stanford.edu>. Diakses 15 Mei 2009 dari website Stanford University.
- [14] Robert Baumgartner, Nicola Henze, Markus Herzog, *The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for The Semantic Web*, 2005.
- [15] OWL, Web Ontology Language, W3C Recommendation, hal. 1, <http://www.w3.org>. Diakses 15 Mei 2009.
- [16] Yusuf Yudi Prayudi, *Cara Kerja Web*, hal. 1, <http://salhaya.net/2008/08/cara-kerja-web/>. Diakses 15 Mei 2009.

Lampiran-1 *Script PHP* untuk aplikasi *parsing file XML ke database MySQL*

```
<?php
//Function object2array//
function object2array($object)
{
    $return = NULL;

    if(is_array($object))
    {
        foreach($object as $key => $value)
            $return[$key] = object2array($value);
    }
    else
    {
        $svar = get_object_vars($object);

        if($svar)
        {
            foreach($svar as $key => $value)
                $return[$key] = object2array($value);
        }
        else
            return $object;
    }

    return $return;
}

// Load xml file using simplexml element object //
$TugasAkhir = simplexml_load_file('ta.xml');

$row = get_object_vars($TugasAkhir);

// Database initalitation //
$mysql_host = "localhost";
$mysql_user = "nuriana";
$mysql_password = "mignon";
$mysql_dbname = "ta";

// setting up mysql connection //
if (!mysql_connect($mysql_host, $mysql_user, $mysql_password)) {
    echo 'Could not connect to mysql';
    exit;
}

//

// Check database //

$sql = "DROP DATABASE if exisTS " . $mysql_dbname;
$res = mysql_query($sql);
if (!$res) { die('Invalid query: ' . mysql_error()); }else{
//echo ' Database DELETED!';
echo '<br>';
}
}
```

```

$sql = "CREATE DATABASE IF NOT EXISTS " . $mysql_dbname;
$res = mysql_query($sql);
if (!$res) { die('Invalid query: ' . mysql_error()); }else{
echo ' Database Created!';
echo '<br>';
}

$sql = "USE " . $mysql_dbname;
$res = mysql_query($sql);
if (!$res) { die('Invalid query: ' . mysql_error()); }else{
//echo ' Accessing ' . $mysql_dbname . ' ...';
//echo '<br>';
}

//Debuging //
//echo 'debug';
//echo '<hr>';

// Object2array process //
$arr = object2array($TugasAakhir);
$n = count($arr);

foreach($arr as $key => $value){

    if($key == 'simple_instance'){
        $n_simple_instance = count($arr[$key]);
        $arr_simple_instance = $arr[$key];
    }
}

//echo '<pre>';

$i=0;
foreach($arr_simple_instance as $key => $value){
    $arr = $value;
    if(is_array($arr) == true){
        if($arr["type"]){
            if(is_array($arr["own_slot_value"]) == true){
                if(array_key_exists('slot_reference', $arr["own_slot_value"])){

/*
echo 'Checking mysql table ...';
echo '<br>';
*/

                $sql = 'SHOW TABLES FROM ' . $mysql_dbname . ' LIKE "' .
                $arr["own_slot_value"]["slot_reference"] . "'';
                $res = mysql_query($sql);
                $num = mysql_num_rows($res);
                if (!$res) { die('Invalid query: ' . mysql_error()); }

                if($num == 0){
                    $sql = 'CREATE TABLE '.$arr["own_slot_value"]["slot_reference"].' (class TEXT,type TEXT,
                    '.$arr["own_slot_value"]["slot_reference"].' TEXT)';
                    $res = mysql_query($sql);
                    if (!$res) { die('Invalid query: ' . mysql_error()); }
                }elseif($num == 1){
/*

```

```

echo ' Table OK!';
echo '<br>';
*/
}

$ssql = 'INSERT INTO '.$sarr["own_slot_value"]["slot_reference"].' (class,type,
'.$sarr["own_slot_value"]["slot_reference"].')
VALUES("'.$sarr_simple_instance[$i]["name"]."', "'.$sarr_simple_instance[$i]["type"]."',
"'.addslashes($sarr_simple_instance[$i]["own_slot_value"]["value"]).'");
$res = mysql_query($ssql);
if (!$res) { die('Invalid query: ' . mysql_error()); }

} else{
foreach($sarr["own_slot_value"] as $key => $value){
$sarr_own_slot_value = $value;

if(is_array($sarr_own_slot_value["value"]) == true){
foreach($sarr_own_slot_value["value"] as $key => $value){
$ssql = 'SHOW TABLES FROM ' . $mysql_dbname . ' LIKE "' .
$sarr_own_slot_value["slot_reference"] . "'';
$res = mysql_query($ssql);
$num = mysql_num_rows($res);
if (!$res) { die('Invalid query: ' . mysql_error()); }
if($num == 0){
$ssql = 'CREATE TABLE '.$sarr_own_slot_value["slot_reference"].' (class TEXT,type TEXT,
'.$sarr_own_slot_value["slot_reference"].' TEXT);
$res = mysql_query($ssql);
if (!$res) { die('Invalid query: ' . mysql_error()); }
}elseif($num == 1){
/*
echo ' Table OK!';
echo '<br>';
*/
}

$ssql = 'INSERT INTO '.$sarr_own_slot_value["slot_reference"].' (class,type,
'.$sarr_own_slot_value["slot_reference"].')
VALUES("'.$sarr_simple_instance[$i]["name"]."', "'.$sarr_simple_instance[$i]["type"]."',
"'.addslashes($value).'");
$res = mysql_query($ssql);
if (!$res) { die('Invalid query: ' . mysql_error()); }
}
}else{
/*
echo 'Checking mysql table ...';
echo '<br>';
*/
$ssql = 'SHOW TABLES FROM ' . $mysql_dbname . ' LIKE "' .
$sarr_own_slot_value["slot_reference"] . "'';
$res = mysql_query($ssql);
$num = mysql_num_rows($res);
if (!$res) { die('Invalid query: ' . mysql_error()); }
if($num == 0){
$ssql = 'CREATE TABLE `'.$sarr_own_slot_value["slot_reference"].'` (class TEXT,type TEXT,
`'.$sarr_own_slot_value["slot_reference"].'` TEXT);
$res = mysql_query($ssql);
if (!$res) { die('Invalid query: ' . mysql_error()); }
}elseif($num == 1){

```


SURVEY UNTUK PENGGUNAAN PROTÉGÉ SEBAGAI TOOL ONTOLOGI UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA.

Dalam rangka mendapatkan masukan tentang penggunaan PROTÉGÉ sebagai tool ontologi yang berfungsi sebagai sarana dalam pemodelan dan penyimpanan ontologi, dengan rendah hati memohon kesediaan rekan-rekan untuk dapat memberikan masukan dengan mengisi survey di bawah ini.

Ontologi adalah sebuah struktur hirarki untuk menjelaskan sebuah domain yang digunakan sebagai landasan *knowledge base*.

PROTÉGÉ adalah sebuah alat bantu yang berbentuk perangkat lunak yang digunakan untuk pengembangan *knowledge base system*. PROTÉGÉ menyediakan seperangkat komponen ontologi yaitu class, slot dan instance. Tab class dalam editor ontologi untuk mendefinisikan class. Tab slot untuk membuat hubungan suatu class dengan instance atau instance dengan instance. Tab instance merupakan elemen yang merepresentasikan suatu individu.

Aplikasi yang dikembangkan adalah sitasi jurnal elektronik Indonesia, ahli bidang atau pustakawan akan melanjutkan pengisian data artikel baru yang terdapat dalam jurnal elektronik dengan menggunakan model ontologi yang telah dibuat.

Mohon untuk dapat dipilih jawaban yang menurut pendapat anda paling tepat.

Pertanyaan	1	2	3	4
(1 = sangat tidak setuju, 2 = tidak setuju, 3 = setuju, 4=sangat setuju)				
1. Saya familiar dengan Protégé				
2. Tab dalam editor ontologi mudah untuk dipahami.				
3. Pengoperasian Protégé mudah untuk dijalankan				
4. Protégé efisien dalam melakukan input data				
5. Sistem mudah serta membantu proses penyimpanan				
6. Cara memasukkan informasi pada suatu artikel cukup mudah.				
7. Cara upload ke direktori <i>server</i> dan <i>parsing</i> ke database cukup mudah.				

Lampiran-3 **Kuesioner untuk pengguna yang melakukan pencarian**

SURVEY UNTUK PENGGUNAAN WEB ONTOLOGI DAN APLIKSI SEMANTIK UNTUK SISTEM SITASI JURNAL ELEKTRONIK INDONESIA.

Dalam rangka mendapatkan masukan tentang penggunaan Web Ontologi dan Aplikasi Semantik sebagai sarana dalam pencarian sitasi jurnal elektronik, dengan rendah hati memohon kesediaan rekan-rekan untuk dapat memberikan masukan dengan mengisi survey di bawah ini.

Ontologi adalah sebuah struktur hirarki untuk menjelaskan sebuah domain yang digunakan sebagai landasan *knowledge base*.

Sistem akan bekerja pada sebuah sistem berbasis web dan pengguna dapat memulai dengan mengetikkan URL pada web browser. Dengan membuka halaman web dapat dilakukan pencarian hubungan antar artikel dan jumlah artikel yang diacu oleh artikel lainnya.

Mohon untuk dapat dipilih jawaban yang menurut pendapat anda paling tepat.

Pertanyaan	1	2	3	4
(1 = sangat tidak setuju, 2 = tidak setuju, 3 = setuju, 4=sangat setuju)				
1. <i>Interface</i> (tampilan) website.				
2. Bermanfaat untuk mendapatkan informasi.				
3. Informasi yang didapatkan jelas.				
4. Penilaian <i>website</i> secara keseluruhan baik				
5. Sistem dapat membantu pencarian jumlah sitasi suatu artikel.				
6. Sistem dapat membantu pencarian berdasarkan judul makalah.				
7. Sistem dapat membantu pencarian berdasarkan penulis makalah.				
8. Saran yang diberikan untuk <i>website</i> .				

