

PENERAPAN TEKNOLOGI *SEMANTIC WEB* PADA ENSIKLOPEDIA ALAM

Muhammad Arief Rahman¹, Faisal Rahutomo²

Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Malang

rahman.malang@gmail.com, faisal.polinema@gmail.com

Abstrak

Alam merupakan segala sesuatu materi yang hidup dan materi yang bukan hidup. Dalam kata lain alam adalah lingkungan yang tanpa kegiatan manusia. Informasi mengenai alam di internet terletak terpisah satu sama lain sehingga muncul permasalahan bagaimana mencari informasi tentang alam, dan bagaimana mengklasifikasikan informasi tentang alam. Terdapat metode *semantic* yang dapat merangkai informasi tersebut sehingga lebih mudah untuk dimengerti. *Semantic* yang cocok digunakan sebagai solusi permasalahan ini adalah ontologi. Ontologi nanti akan memodelkan dan menghubungkan antara entitas alam dengan beberapa entitas lainnya seperti makhluk hidup, penggolongan makhluk hidup, dan energi. Berdasarkan pengujian, aplikasi ini mampu mencari informasi tentang alam berdasarkan entitas yang dimiliki seperti makhluk hidup, penggolongan makhluk hidup, dan energi. Ketika pengguna memasukkan kata kunci, sistem dapat melakukan klasifikasi pada alam berdasarkan informasi yang sudah dimiliki.

Kata kunci : alam, ontologi, pencarian semantik

1. Pendahuluan

Alam merupakan segala sesuatu materi yang hidup dan materi yang bukan hidup. Dalam kata lain alam adalah lingkungan yang tanpa kegiatan manusia. Di alam telah tersedia banyak hal seperti kehidupan, energi, sumber daya, dan lain-lain.

Perkembangan teknologi saat ini kian hari kian pesat, hampir semua bidang teknologi sekarang berbasis internet. Salah satu teknologi yang saat ini kita rasakan manfaatnya adalah world wide *web* atau dikenal dengan nama *web*. Peneliti dan pengembang *web* mengusulkan tambahan informasi bagi konten *web* yang sering diistilahkan dengan metadata. Istilah metadata ini sering dipakai dalam *semantic web*, yang menggambarkan pendekatan dalam menangani dan menyimpan dokumen. Saat ini dokumen mulai disajikan dalam format *XML* (*Extensible Markup Language*). Bahasa lain yang digunakan dalam mendukung visi *Semantic Web* selain *XML* adalah *RDF/OWL*. Selanjutnya *RDF/OWL* dengan kemampuan dan fasilitas yang dimilikinya mampu digunakan untuk merepresentasikan makna istilah dalam *web* hingga dapat diproses dalam mesin (Nurkamid, 2009).

Informasi tentang alam di internet telah banyak dan terpisah satu sama lain. Sehingga membuat pengguna harus mencari informasi dari beberapa *web*. Informasi tentang alam yang digunakan adalah informasi dari buku Rangkuman Pengetahuan Alam Lengkap (RPAL) dengan pemodelan ontologi sebagai penghubung antara satu entitas dengan entitas yang lain.

2. Tinjauan Pustaka

2.1 Alam

Alam merupakan segala sesuatu materi yang hidup dan materi yang bukan hidup. Dalam kata lain alam adalah lingkungan yang tanpa kegiatan manusia. Di alam telah tersedia banyak hal seperti kehidupan, energi, sumber daya, dan lain-lain.

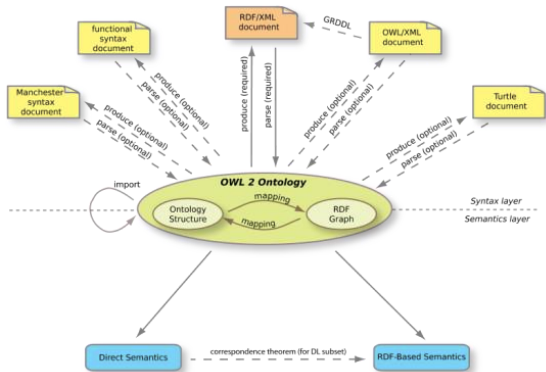
2.2 Ontologi

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*) Sachs (2006).

2.3 Pencarian Semantik

Pencarian semantik merupakan pencarian informasi berdasarkan maksud dari pencari dan konteks pencarian, bukan tergantung pada arti kata itu sendiri. Pencarian semantik kini sudah memasuki ranah alam. Salah satu teknologi pada semantik yaitu ontologi. Ontologi diharapkan mampu menggabungkan pecahan informasi dari suatu alam sehingga mampu terkait satu sama lain dan menjadi lebih mudah untuk dikelola (R. Guha, Rob McCool, and Eric Miller. 2003).

2.4 RDF / OWL



Gambar 1. Gambaran umum RDF / OWL (Bao, J. et al, 2009).

RDF (*Resource Description Framework*) merupakan bahasa yang digunakan untuk merepresentasikan *metadata*. RDF mendukung interoperabilitas antar aplikasi yang mempertukarkan informasi yang bersifat *machine-understandable* di *web*. *Semantic web* terdiri dari data yang ditulis dalam bahasa yang dapat dimengerti oleh mesin, seperti RDF. RDF menggunakan graf untuk merepresentasikan kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas.

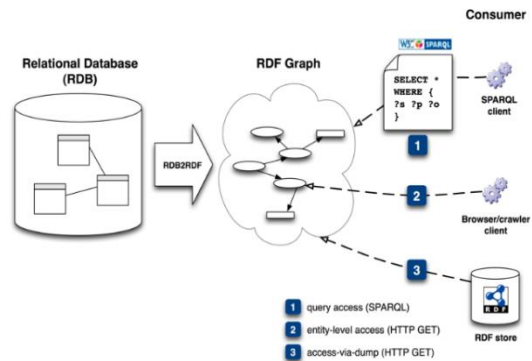
```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdfsyntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person
    rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric
    Miller</contact:fullName>
    <contact:mailbox
    rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

Gambar 2. Contoh Sintaks RDF (Manola dan Miller, 2004).

```
<rdf:Description rdf:about="#Continent">
<rdf:type
  rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

Gambar 3. Sintaks RDF / XML yang sama arti dengan sintaks *instance* pada OWL (Bechhofer. et al, 2004)

2.5 SPARQL



Gambar 4. Gambaran umum SPARQL (Prud'hommeaux dan Hausenblas, 2010.)

SPARQL merupakan bahasa *query* untuk RDF/OWL. SPARQL menyediakan fasilitas untuk mengekstrak informasi dalam bentuk *URI*, *blank node* dan *literal*, mengekstrak subgraf RDF, dan membangun graf RDF baru berdasar pada informasi dari graf yang di-query (Prud'hommeaux dan Seaborne, 2005). *Query SPARQL* didasarkan pada pencocokan pola graf (Karsanti, 2006). Pola graf yang paling sederhana adalah *triple pattern* yang mirip dengan *RDF triple*, hanya saja pola pada *query* dimungkinkan pemberian nama diluar terminologi RDF pada posisi subyek, predikat dan obyek.

```
PREFIX
dc:<http://www.anakmalank.org/res/bab1.owl#>
SELECT
DISTINCT ?Nama ?GolMakan ?GolGerak ?Gol
Habitat ?GolPenutupTubuh
?GolJenisAkar ?GolJenisBatang ?GolJe
nisDaun ?TerkaitMakan ?TerkaitGerak
WHERE {
?x dc:Name ?Nama.
OPTIONAL {?x dc:Description ?Deskripsi}
OPTIONAL {?x dc:OtherName ?NamaLain}
OPTIONAL {?x
dc:hasGroupMeal ?GolMakanP.
OPTIONAL {?GolMakanP
dc:Name ?GolMakan} }
OPTIONAL {?x
dc:isGroupMealOf ?GolMakanP.
OPTIONAL {?GolMakanP
dc:Name ?GolMakan} }
OPTIONAL {?GolMakanP
dc:isGroupMealOf ?TerkaitMakanP.
OPTIONAL {?TerkaitMakanP
dc:Name ?TerkaitMakan} .
FILTER (?TerkaitMakan != ?Nama ) }
```

Gambar 5. Contoh query SPARQL

3. Design dan Implementasi

3.1 Analisis

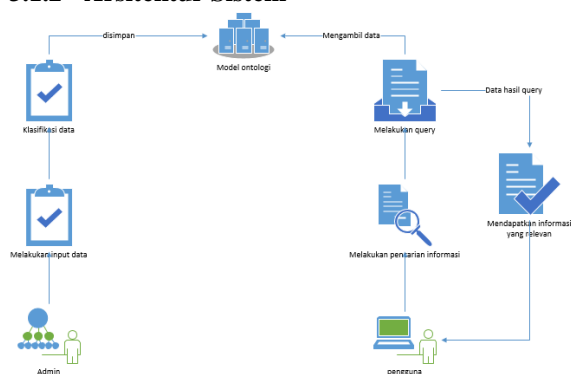
Pada bagian ini akan ditunjukkan design model dan struktur implementasi dari solusi yang ditawarkan.

3.1.1 Deskripsi Umum

Aplikasi yang dibangun adalah aplikasi pencarian informasi alam berbasis semantik. Aplikasi ini mampu menangani pencarian informasi berdasarkan konteks. Kemudian aplikasi menampilkan data yang relevan disesuaikan dengan konteks yang sudah diinputkan sebelumnya.

Setiap informasi alam cenderung memiliki karakteristik yang berbeda-beda. Karakteristik yang berbeda tadi dapat dikelompokkan menjadi beberapa entitas yang mampu mempermudah pengguna untuk mencari informasi yang diinginkan. Tentu saja diperlukan admin untuk memasukkan data informasi tentang alam yang lengkap beserta konteks yang sesuai.

3.1.2 Arsitektur Sistem



Gambar 6. Arsitektur sistem

Secara umum sistem akan menerima konteks yang dikirimkan pada *web service*. Pada *web service* akan dilakukan proses pencarian yang menggunakan teknologi semantik. *Server* konteks akan menjadi kata kunci untuk melakukan query pada basis data yang ada pada berkas *OWL*. Kemudian hasil yang didapat dikirimkan pada pengguna. Proses keseluruhan dari sistem digambarkan pada Gambar 6.

3.2 Perancangan Data

Pada bagian ini dijelaskan mengenai data yang akan digunakan pada sistem. Pada sistem *semantic* yang akan dibangun menggunakan ontologi.

3.2.1 Pembangunan Ontologi

Terdapat beberapa tahapan yang dilakukan dalam pengembangan ontologi (Noy, 2000) yaitu :

3.2.1.1 Menentukan domain dan batasan ontology

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

3.2.1.1.1 Domain apa yang akan melingkupi ontology ?

3.2.1.1.2 Mengapa ontologi digunakan ?

3.2.1.1.3 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* (Gruninger dan Fox, 1995).

3.2.1.2 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.

3.2.1.3 Menentukan istilah penting dalam ontologi

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

3.2.1.3.1 Istilah apa saja yang akan diperbincangkan ?

3.2.1.3.2 Apa yang akan menjadi jawaban mengenai istilah – istilah tersebut ?

3.2.1.3.3 Properti apa saja yang dimiliki istilah tersebut ?

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

Terdapat beberapa pendekatan dalam pengembangan hirarki *class* (Uchold dan Gruninger, 2004), yaitu :

3.2.1.4.1 Proses pengembangan *top-down* dimulai dengan mendefinisikan *concept* umum dalam domain dilanjutkan dengan *concept* yang lebih spesifik.

3.2.1.4.2 Proses pengembangan *bottom-up* dimulai dengan mendefinisikan *class* yang paling spesifik kemudian dikelompokkan menjadi *class* dengan konsep yang lebih umum.

3.2.1.4.3 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.

3.2.1.5 Mendefinisikan slot atau properties dan menjabarkan nilai dari slot tersebut.

3.2.1.6 Mendefinisikan facets pada slots.

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

3.2.1.6.1 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).

3.2.1.6.2 Slot value type

Sebuah tipe facet menggambarkan tipe dari nilai yang dapat mengisi slot. Berikut merupakan tipe nilai yang umum digunakan :

3.2.1.6.2.1 *String*, tipe nilai yang paling sederhana yang biasa digunakan untuk slot.

3.2.1.6.2.2 *Number*, terkadang tipe nilai yang lebih spesifik seperti *Float* dan *Integer* yang digunakan dalam penggambaran slot dengan nilai numerik.

3.2.1.6.2.3 *Boolean*, pada Boolean slot mempunyai *yes-no flag*.

3.2.1.6.2.4 *Enumerated*, *Enumerated slot* merinci daftar dari spesifik nilai yang diperbolehkan untuk slot.

3.2.1.6.2.5 Instance, Instance membuat hubungan antara individual.

3.2.1.7 Membuat instances

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

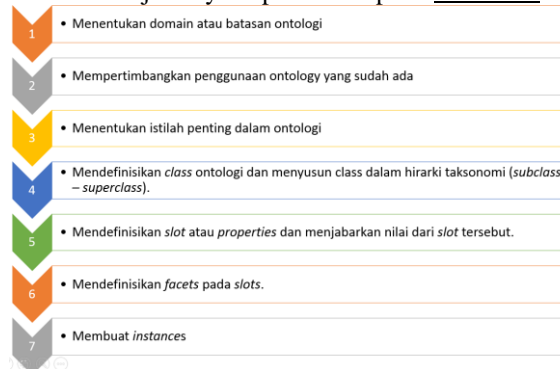
3.2.1.7.1 Pemilihan class.

3.2.1.7.2 Pembuatan individual instances pada class tersebut.

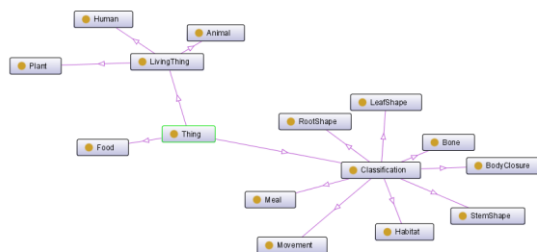
Knowledge base selanjutnya dapat dibuat dengan mendefinisikan individual instances dari class yang terisi pada nilai spesifik slot dan slot batasan tambahan (facets).

3.2.1.7.3 Mengisi nilai slot.

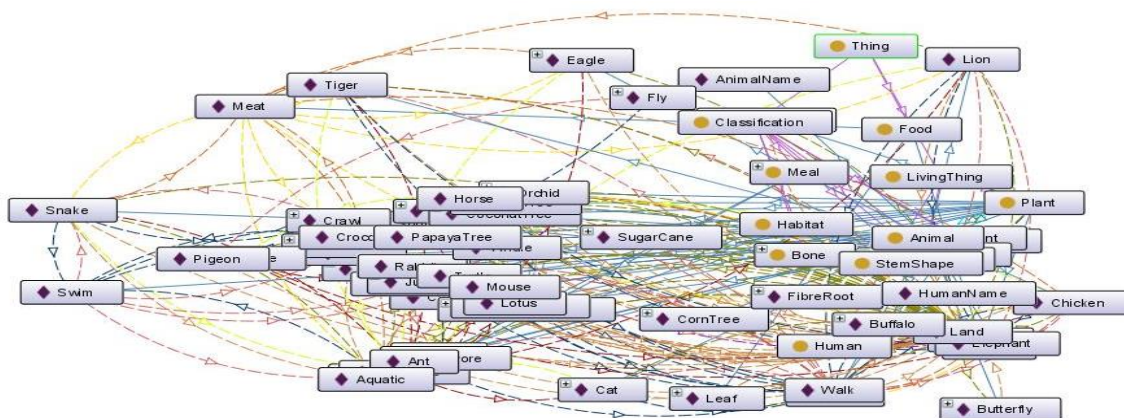
Untuk lebih jelasnya dapat dilihat pada Gambar 7.



Gambar 7. Tahapan membangun ontology.



Gambar 8. Graph Ontologi kelas.



Gambar 9. Graph Ontologi kelas dan individu beserta property

Gambar 8 menunjukkan ontology yang telah dibuat hanya menampilkan kelas-kelas. Gambar 9 menunjukkan ontology yang telah dibangun dengan menampilkan kelas-kelas, individu beserta properti.

3.3 Perancangan Proses

Pada bagian ini adalah proses utama yang ada pada aplikasi yang dibangun. Pada aplikasi ini terdapat dua proses utama, yaitu proses klasifikasi kategori informasi alam, dan proses mencari informasi berdasarkan konteks.

3.3.1 Proses klasifikasi berdasarkan kategori informasi alam

Proses klasifikasi kategori informasi alam mampu mengklasifikasikan informasi berdasarkan informasi yang dimiliki. Proses ini menggunakan reasoning yang ada pada ontologi. Reasoning adalah proses terpenting dalam klasifikasi data alam. Model yang sudah ada sebelumnya akan dicocokkan dengan rule yang telah kita buat. Hasil reasoning baru kemudian ditulis pada file *OWL*.

3.3.2 Proses mencari informasi berdasarkan konteks.

Proses ini mencari data alam yang memiliki hubungan dengan konteks. Semua atribut yang dimiliki oleh alam dicocokkan dengan konteks.

4. Uji Coba dan Evaluasi.

4.1 Spesifikasi Hardware

RAM : 2GB
 Harddisk : 80GB
 Processor : Intel Core i3 2.13Ghz
 VGA : Intel HD Graphics

4.2 Spesifikasi Software

Sistem Operasi : Windows 8.1 Professional
 Browser : Google Chrome v41
 Web Server : Apache dan Tomcat.
 Editor : Notepad++
 Tool OWL : Protégé 3.5

4.3 Uji Coba SPARQL

```
PREFIX
dc:<http://www.anakmalank.org/res/bab1.owl#>
select ?Binatang ?Kelas
WHERE{
?x dc:isGroupMealOf ?BinatangP.
?BinatangP dc:Name ?Binatang.
?BinatangP dc:hasGroupMeal ?KelasP.
?KelasP dc:Name ?Kelas.
}
```

Gambar 10. Query dengan bahasa SPARQL

| Binatang | Kelas |
|----------------|-----------|
| Kura-kura | Omnivora |
| Tikus | Omnivora |
| Semut | Omnivora |
| Katak | Omnivora |
| Ular | Karnivora |
| Burung Elang | Karnivora |
| Buaya | Karnivora |
| Ikan Hiu | Karnivora |
| Kucing | Karnivora |
| Harimau | Karnivora |
| Singa | Karnivora |
| Kerbau | Herbivora |
| Kuda | Herbivora |
| Kupu-kupu | Herbivora |
| Gajah | Herbivora |
| Ayam | Herbivora |
| Kelinci | Herbivora |
| Sapi | Herbivora |
| Bebek | Herbivora |
| Burung Merpati | Herbivora |
| Kambing | Herbivora |

Gambar 11. Result Query Sparql.

Gambar 10 menunjukkan query menggunakan bahasa SPARQL untuk mendapatkan nama binatang dan kelas (keluarga) dari binatang tersebut berdasarkan jenis makanan.

Gambar 11 menunjukkan hasil dari query menggunakan bahasa SPARQL yaitu nama binatang dan kelas (keluarga) dari binatang tersebut berdasarkan jenis makanan.

4.4 Uji Coba Web



Gambar 12. Halaman awal



Gambar 13. Hasil pencarian “sapi”.

Gambar 12 menunjukkan halaman awal dari aplikasi ini. Yaitu tersedia textbox untuk memasukkan kata kunci yang akan dicari informasinya. Gambar 13 menunjukkan hasil pencarian dari kata kunci “sapi” yaitu deskripsi singkat dari “sapi” beserta gambar dari kata kunci yang diberikan. Serta artikel yang terkait dengan kata kunci yaitu *sapi*. Baik itu binatang yang sejenis dengan keluarga yang dimiliki *sapi* maupun keluarga lain yang terkait dengan keluarga milik *sapi*.

4.5 Uji Coba Fungsionalitas

Tabel 1. Uji coba fungsional

| No | Pengujian | Hasil yang diharapkan | Keberhasilan |
|----|--|---|--------------|
| 1. | Pencarian berdasarkan kata kunci | Menampilkan deskripsi dari kata kunci beserta artikel terkait | berhasil |
| 2. | Pencarian berdasarkan tautan | Menampilkan deskripsi dari tautan beserta artikel terkait | berhasil |
| 3. | Pencarian berdasarkan keluarga (kelas) | Menampilkan deskripsi keluarga (kelas) serta individu yang terkait dengan keluarga tersebut | berhasil |
| 4. | Pencarian binatang dengan SPARQL | Menampilkan hasil binatang | berhasil |
| 5. | Pencarian menggunakan nama alias | Menampilkan binatang yang memiliki nama alias | berhasil |
| 6. | Klasifikasi kategori alam | Makhluk hidup, jenis, sumber daya | berhasil |

Tabel 1 menunjukkan bahwa pengujian fungsional terhadap aplikasi dapat berjalan dengan baik.

5. Kesimpulan dan Saran.

5.1 Kesimpulan

Dari aplikasi ini dapat diperoleh beberapa kesimpulan sebagai berikut :

- d. Solusi yang ditawarkan dapat melakukan klasifikasi berdasarkan kata kunci yang dimasukkan.
- e. Aplikasi yang dibangun pada artikel ini dapat membantu pengguna mencari informasi alam tanpa harus melakukan pencocokan dengan kata kunci pencarian.

5.2 Saran

Kemudian saran untuk pengembangan aplikasi ini adalah sebagai berikut :

- a. Materi yang digunakan pada aplikasi ini masih dalam lingkup Sekolah Dasar dan Sekolah Menengah Pertama. Jadi untuk pengembangannya diharapkan menggunakan materi yang lebih luas.

- b. Ontologi yang telah dibuat dapat digabungkan dengan ontologi lain sehingga dapat memberikan informasi yang lebih banyak.

Daftar Pustaka :

Bao, J. et al. 2009. *OWL 2 Web Ontology Language Document Overview*. Latest version available : <http://www.w3.org/TR/2009/WD-owl2-overview-20090327/>.

Gruninger, M. and Fox, M.S. 1995. *Methodology for the Design and Evaluation of Ontologies in : Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal*.

Karsanti, H. T. 2006. *Penerapan Teknologi Semantic Web pada Sistem Manajemen Perpustakaan Digital*. Yogyakarta : Program Studi Ilmu Komputer FMIPA Universitas Gadjah Mada.

Manola, F dan Miller, E. 2004. *RDF Primer*. Latest version available : <http://www.w3.org/TR/rdf-primer/>.

Nurkamid, Mukhamad. 2009. *Aplikasi Bibliografi Perpustakaan Berbasis Teknologi Web Semantik*.

Noy, N. F. dan McGuinness. D. L. 2000. *Ontology Development 101: A Guide to Creating Your First*, Stanford University, Stanford. Latest version available : http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

Prud'hommeaux, Eric dan Hausenblas, Michael. 2010. *Use Cases and Reuirements for Mapping Relational Database to RDF*. Latest version available : <http://www.w3.org/2001/sw/rdb2rdf/use-cases/>.

R. Guha, Rob McCool, and Eric Miller. 2003. *Semantic Search*, WWW2003.

Sachs, Eliza. 2006. *Getting Started with Protégé-frames*. Latest version available : <http://lpis.csd.auth.gr/mtpx/km/Getting%20Started%20with%20Protege-Frames.pdf>.

Uschold, M. dan Gruninger, M. 1996. *Ontologies: Principles, Methods and Applications*. Latest version available: <http://www.aiai.ed.ac.uk/~oplan/documents/1996/96-ker-intro-ontologies.pdf>.