

BAB II

LANDASAN TEORI

2.1 PENGERTIAN SKRIPSI

Definisi skripsi sebagaimana ditulis dalam wikipedia” *Skripsi adalah istilah yang digunakan di Indonesia untuk mengilustrasikan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian sarjana S1 yang membahas suatu permasalahan/ fenomena dalam bidang ilmu tertentu dengan menggunakan kaidah-kaidah yang berlaku. Skripsi bertujuan agar mahasiswa mampu menyusun dan menulis suatu karya ilmiah, sesuai dengan bidang ilmunya. Mahasiswa yang mampu menulis skripsi dianggap mampu memadukan pengetahuan dan ketrampilannya dalam memahami, menganalisis, menggambarkan dan menjelaskan masalah yang berhubungan dengan bidang keilmuan yang diambilnya. Skripsi merupakan persyaratan untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) yang ada di Indonesia.”[2]*

Tugas skripsi di prodi teknik informatika UMG dapat diambil setelah dinyatakan lulus tugas khusus atau yang biasa disebut KP(Kerja Praktek) . Waktu pengambilan tugas skripsi dibagi menjadi 2 periode dalam 1 tahun, sehingga dilakukan 2 kali wisuda setiap tahunnya.

2.2 TEKNIK ONTOLOGI

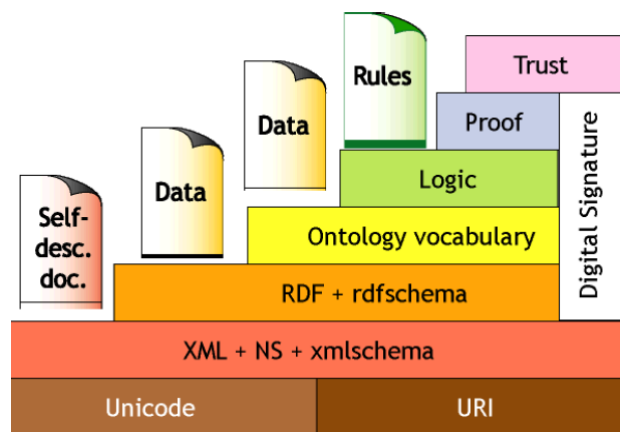
“Istilah ontologi sebenarnya berasal dari istilah filosofi “ontology” yang artinya sesuatu yang sesungguhnya ada dan bagaimana menggambarkannya. Dalam dunia komputer ontologi digunakan untuk menspesifikasikan suatu konseptualisasi. Dalam istilah lain ontologi dijelaskan sebagai suatu representasi dari domain pengetahuan tertentu yang berisi istilah-istilah dalam domain tersebut beserta hubungan antara istilah-istilah yang ada.

Ontologi saat ini banyak digunakan terutama untuk mendukung web semantik, yaitu teknologi web yang diarahkan dapat memahami makna suatu

kata atau kalimat yang diberikan oleh pengguna. Membuat komputer mengerti seperti manusia adalah suatu hal yang sepertinya mustahil, namun visi ini terus diupayakan dengan menyediakan seperangkat alat sehingga membuat mesin atau komputer dengan mudah dapat memproses informasi dan mengerti informasi yang diinginkan oleh pengguna.

Tidak ada standar khusus untuk membangun suatu ontologi dan tidak ada justifikasi bahwa ontologi yang dikembangkan oleh seseorang adalah salah atau benar. Kualitas ontologi dapat dilihat dari aplikasi yang dibangun berdasarkan ontologi ini. Ketika aplikasi yang dibangun dapat memenuhi kebutuhan pengguna dan menjawab permasalahan yang ada maka ontologi yang digunakan termasuk ontologi yang berkualitas.” [3]

Layer ontologi :



Gambar 2.1 Layer ontologi [2]

“Setiap layer memiliki fungsi masing-masing :

- XML menyimpan halaman web
- RDF untuk merepresentasikan semantik dari isi halaman tersebut.
- Ontologi layer untuk menjelaskan vocabulary dari domain
- Logic layer memungkinkan untuk mengambil data yang diinginkan.”[2]

2.2.1 KOMPONEN ONTOLOGI

Ontologi mempunyai beberapa komponen antara lain [3] :

1. Konsep

Konsep dapat berupa sesuatu yang dikatakan. Konsep juga dikenal sebagai classes, object dan categories.

2. Relasi

Merupakan representasi sebuah tipe dari interaksi antara konsep dari sebuah domain.

Contoh : SubclassOf, Connected-to

3. Fungsi

Adalah sebuah relasi khusus dimana elemen ke n dari relasi adalah unik untuk elemen ke $n-1$. $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$.

Contohnya adalah *Mother-of*.

4. Aksiom

Digunakan untuk memodelkan sebuah *sentence* yang selalu benar.

5. Instance

Digunakan untuk merepresentasikan elemen.

2.2.2 BAHASA ONTOLOGI

Beberapa struktur bahasa yang menyusun ontologi antara lain [3] :

2.2.2.1 XML (EXTENSIBLE MARKUP LANGUAGE) DAN XMLS (XML SCHEMA)

“XML adalah sebuah format teks yang sederhana yang berdasarkan SGML(ISO 8879), yang didesain untuk mempertemukan berbagai macam sumber informasi dalam dunia web. XML dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998.

XML Schema (XMLS) adalah sebuah definisi dari XML untuk memberikan aturan untuk sebuah dokumen XML, seperti analoginya pada database konvensional, sebuah XMLS bagaikan data dictionary sementara XML adalah data d X-basenya.

Definisi XML dan XMLS secara sederhana dapat dijelaskan sebagai berikut,

XML Schema :

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Mahasiswa" type="Mahasiswa"/>
<xs:complexType name="Mahasiswa">
<xs:sequence>
<xs:element name="NRP" type="xs:number"/>
<xs:element name="Nama" type="xs:string"/>
<xs:element name="Angkatan" type="xs:number"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

XML :

```
<Mahasiswa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mahasiswa.xsd">
<NRP>5106100611</NRP>
<Nama>M.Awaludin</Nama>
<Angkatan>2003</Angkatan>
</Mahasiswa>
```

➤ **Sintaksis dan elemen XML**

Sintak dokumen XML yang sederhana terdiri dari deklarasi XML dan elemen puncak. Deklarasi XML merupakan tempat untuk menyatakan Versi dari XML dan encoding untuk dokumen tersebut. Untuk dokumen XML standar versi yang tersedia adalah versi "1.0" dan menggunakan ISO-8859-1 (Latin-1/West European) sebagai encodingnya.

Bagian yang selanjutnya adalah elemen-elemen yang menyusun dokumen XML tersebut. Setiap elemen tersebut memiliki tags penutup. Tags dalam XML memperhatikan penggunaan huruf atau dalam arti Case Sensitive.

Dalam dokumen XML setiap elemen harus tersusun dengan benar. Dalam arti setiap elemen harus benar-benar terkurung. Sebuah dokumen XML harus memiliki 9 elemen utama. Sedangkan setiap nilai atribut dari elemen tersebut harus menggunakan tanda kutip dua (" ").

Berikut ini adalah contoh sintak XML yang sederhana :

```
<?xml version="1.0"?> <note date=12/11/99>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

2.2.2.2 RDF (RESOURCE DESCRIPTION FRAMEWORK) DAN RDFS (RDF SCHEMA)

Resource Description Framework (RDF) adalah Model atau framework yang menjelaskan atau menggambarkan resource serta relasi antar resource. RDF menggambarkan resource dan relasi antar resource dengan menggunakan konsep triple (Subyek-Predikat-Obyek).

Dalam perkembangannya, RDF membutuhkan suatu fasilitas untuk menjelaskan / mendefinisikan sesuatu dengan pasti tentang jenis resource secara spesifik. Untuk pustaka contohnya, membutuhkan atribut 'penulis', 'judul', dll. Untuk mendeklarasikan secara spesifik arti atribut-atribut tersebut dan hubungannya dengan semantic dibutuhkan RDF Schema. RDF Schema juga dapat digunakan untuk mendefinisikan masing-masing class dan property yang akan digunakan pada suatu dokumen RDF.

RDFS menyediakan kerangka untuk mendeskripsikan class-class dan property selayaknya class dalam Object Oriented Programming. Ini mengijinkan setiap resource dideskripsikan dalam class atau subclass dari suatu class.

Skema RDF yang merupakan kosakata penjelasan memperbolehkan perancang kosakata untuk merepresentasikan classess dan properties dalam World Wide Web. Contoh di bawah ini menggambarkan penggunaan kosakata skema RDF untuk menjelaskan classes dan properties, dan hubungan ke data pada tingkat aplikasi.

RDF Schema merupakan gabungan antara RDF triples (SPO) dan RDF vocabulary.

Elemen Inti RDFS:

- + Core classes: *rdfs:Resource, rdf:Property, rdfs:Class, rdfs:datatype*
- + Core properties: *rdfs:subClassOf, rdfs:subPropertyOf*
- + Core constraints: *rdfs:range, rdfs:domain,*

Perlu diperhatikan bahwa “resource” di dunia RDF Schema mempunyai maksud yang sama dengan “class”, jadi dua kata ini dapat dipertukarkan. Jika subClassOf tidak ada, maka diasumsikan termasuk subClassOf rdfs:Resource, karena ini merupakan kelas dasar (root kelas) dari semua kelas yang ada. Penulisan 'Class' dalam penamaan rdfs, sedang untuk 'Property' dalam penamaan rdf.”[3]

Contoh :

RDFS

```
<rdf:Property rdf:about="mengajar_di">
  <rdfs:domain rdf:resource="#Dosen"/>
  <rdfs:domain rdf:resource="#Universitas"/>
  //untuk pemanggilan ke resource
  <rdfs:range rdf:literal="#UMG"/>
</rdf:Property>
```

RDF

```
<foaf:Dosen rdf:about="#DosenTI">
  <foaf:nama> Susilo Bambang Y </foaf:nama>
  <rdfs:label> sby </rdfs:label>
  <foaf:mengajar_di rdf:resource="#UMG"/>
</foaf:Dosen>
```

➤ **Beberapa kosakata dasar dalam RDF**

“Tabel di bawah ini adalah sebagian dari kosakata dasar dalam RDF, digunakan bersama kosakata-kosakata tambahan dari model RDF dan sintaksis khusus untuk class dan properties.”[2]

Tabel 2.1 Property

Nama property	Pengertian
rdfs:Resource	Class sumber daya
rdfs:Class	Konsep Class
rdf:Property	Konsep dari property
rdfs:Literal	Merepresentasikan bagian dari nilai literal
rdf:Statement	Class dari statement RDF
rdfs:Container	Merepresentasikan penyimpanan
Rdf:Bag	Koleksi yang tidak terstruktur
Rdf:Seq	Koleksi yang terstruktur
Rdf:Alt	Koleksi alternative

Tabel 2.2 Classes

Nama property	Pengertian	Domain	Range
rdfs:isDefinedBy	Nama space dari sumber	rdfs:Resource	rdfs:Resource
rdf:subject	Subjek dari statement RDF	rdf:Statement	rdfs:Resource
rdf:predicate	Predikat dari statement RDF	rdf:Statement	rdf:Property

2.2.2.3 OWL (WEB ONTOLOGY LANGUAGE)

“OWL adalah bahasa ontologi yang baru untuk sebuah web semantik, dikembangkan oleh World Wide Web Consortium (W3C) kelompok kerja web ontologi.

Pada mulanya OWL dirancang untuk merepresentasikan informasi tentang kategori dari sebuah objek dan bagaimana objek tersebut berhubungan. OWL dapat juga menyediakan informasi tentang objek itu sendiri.

OWL menyediakan tiga sub bahasa yang berbeda tingkatan bahasanya yang dirancang untuk berbagai kebutuhan tertentu dari pengguna, antara lain :

1. OWL Lite

Mendukung bagi pengguna yang memerlukan hirarki klasifikasi dan batasan yang sederhana. Sebagai contoh, hanya mendukung batasan Cardinality, dengan nilai untuk Cardinality sebesar 0 atau 1. Haruslah lebih sederhana untuk menyediakan alat pendukung untuk OWL Lite dibanding yang lebih ekspresif, dan

OWL Lite menyediakan suatu alur migrasi cepat untuk thesauri dan taksonomi lain.

OWL Lite juga mempunyai suatu kompleksitas formal yang lebih rendah dibanding OWL DL. OWL Lite didesain untuk kemudahan penerapan dan untuk memberikan user dengan fungsi subset yang akan membawa mereka ke permulaan dalam penggunaan OWL.

2. OWL DL

Mendukung pengguna yang menginginkan ekspresi yang maksimum di saat menahan perhitungan yang lengkap (semua kesimpulan dijamin menjadi dapat diperhitungkan) dan ketepatan (semua perhitungan akan diselesaikan waktu tertentu). OWL DL meliputi semua bahasa konstruksi dalam OWL, tetapi mereka dapat digunakan hanya di bawah batasan tertentu (sebagai contoh, selama suatu class menjadi suatu subclass dari banyak kelas, suatu kelas tidak bisa menjadi suatu instance dari kelas yang lain).

OWL DL di beri nama dalam kaitan dengan Description Logic, suatu bidang riset yang telah belajar logika yang membentuk pondasi bagi yang formal pada OWL. OWL DL memang dirancang untuk mendukung deskripsi logikal dari segmen bisnis dan untuk menyediakan bahasa subset yang memberikan property untuk perhitungan dalam sebuah sistem.

3. OWL Full

Sangat berguna untuk pengguna yang menginginkan ekspresi maksimum dan kebebasan sintaksis dari RDF tanpa ada jaminan perhitungan. Sebagai contoh, dalam OWL Full sebuah class dapat diperlakukan berturut-turut sebagai kumpulan individu dan sebagai individu dengan haknya sendiri. OWL Full memperbolehkan ontologi untuk meningkatkan arti dari kosa kata yang belum digambarkan (RDF atau OWL). Tidaklah mungkin untuk semua pembuat software untuk mendukung secara penuh kemampuan yang ada pada OWL Full. “[3].

Contoh :

```
<owl:ontologi rdf:about="" "/>
<owl:class rdf:ID="DosenTI">
<rdfs:subclassOf rdf:ID="DosenUMG">
</rdfs:subclassOf>
</owl:class>
<owl:objectProperty rdf:ID="mengajar_di">
<rdfs:range>
<owl:class>
<owl:unionOf rdf:parseType="Collection">
<owl:class rdf:about="#UMG">
<owl:class rdf:about="#ITS">
<owl:class rdf:about="#ITATS">
</owl:unionOf>
</owl:class>
</owl:range>
</owl:objectProperty>
```

2.2.3 APLIKASI UNTUK MEMBANGUN ONTOLOGI

“Protégé adalah perangkat lunak bantu yang digunakan untuk pengembangan sistem berikut Knowledge Base System. Aplikasi yang dikembangkan oleh protégé digunakan dalam pemecahan masalah dan pembuatan keputusan dalam sebuah domain. Protégé dikembangkan oleh sebuah organisasi yang bernaung di bawah Stanford, yang mengambil spesialisasi di bidang ontologi.

Protégé merupakan sebuah alat yang digunakan untuk membuat sebuah domain ontology, menyesuaikan form untuk entry data, dan memasukkan data. Berbagai format penyimpanannya seperti OWL, RDF, XML dan HTML. Protégé menyediakan kemudahan plug and play yang membuatnya fleksibel untuk pengembangan prototype.

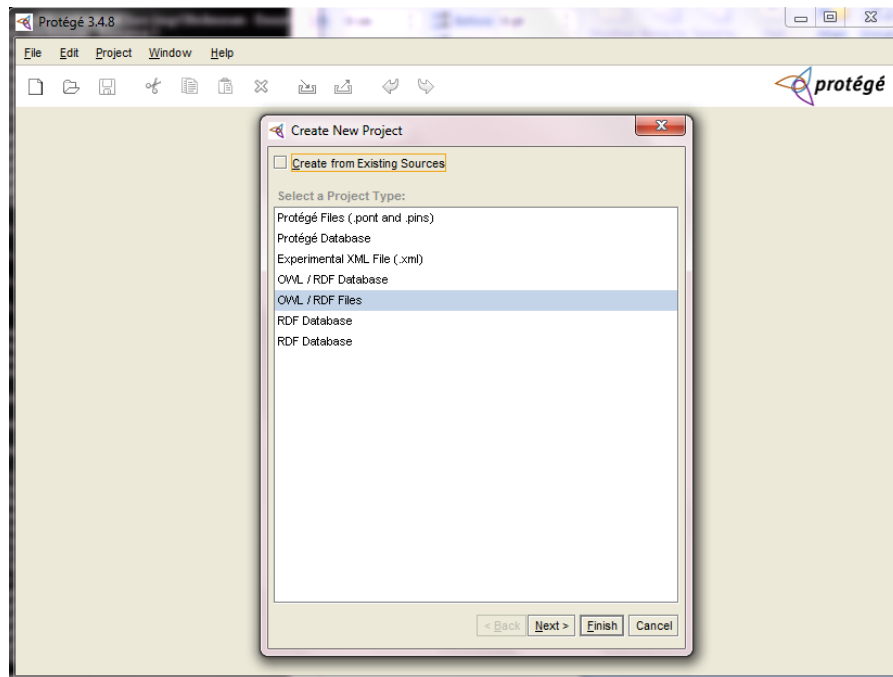
Protégé dibuat dengan menggunakan bahasa pemrograman Java. Semua alat-alat dalam protégé dapat digunakan melalui Graphical User Interface (GUI)

dengan menyediakan Tab untuk masing-masing bagian dan fungsi standar. Class tab dalam editor ontologi berfungsi untuk mendefinisikan class dan hirarki class, property dan nilai property tersebut, relasi antara class dan property dari relasi tersebut .”[4].

“Perangkat lunak protégé menyediakan konsepsi dasar pengetahuan yang terintegrasi, serta mengubah tampilan visual lingkungan dengan memperluas arsitektur sistem untuk membuat pemodelan dasar pengetahuan secara lebih sederhana dan mudah..

Protégé dapat juga digunakan dengan tujuan berikut : membangun ontologi, memodelkan tampilan pengetahuan akuisisi dan memasukkan domain pengetahuan. Protégé memvisualisasikan hubungan sub-class dalam tree, mendukung berbagai penurunan (multiple inheritance) dan root pada hirarki class yang terbentuk adalah class “THING”.

Untuk mendapatkan protégé dapat dilakukan dengan cara mengunduh dari web penyedia tool, alamat web tersebut adalah <http://protege.stanford.edu/>. Ukuran file instalasi untuk Protégé tergantung pada versi yang diinginkan dan juga tergantung dengan ada tidaknya SDK Java. Protégé membutuhkan SDK Java, terdapat dua pilihan yaitu apakah SDK Java termasuk ke dalam file instalasi atau tidak. Protégé dapat berjalan di berbagai platform sistem operasi, antara lain Windows, Mac OS, Solaris, Linux, HP-UX, Unix, AIX. Protégé dapat membuka berbagai macam format file, ada tiga format file umum yang dapat dibuka dengan protégé, yaitu XML, RDF dan OWL. Untuk dapat membuka format file tersebut hal yang perlu dilakukan adalah dengan membuat sebuah project baru pada protégé, project tersebut akan memiliki format file .pprj.”[3].



Gambar 2.2 Halaman awal protege

2.3 TEKNOLOGI WEB

“World Wide Web (WWW) atau yang lebih di kenal dengan web adalah sebuah sistem yang saling terkait dalam sebuah dokumen berformat hypertext yang berisi beragam informasi, baik tulisan, gambar, suara, video, dan informasi multimedia lainnya dan dapat diakses melalui web browser. Web pertama kali ditemukan oleh Sir Timothy John Berners-Lee yang ketika itu bekerja di Laboratorium Fisika Partikel Eropa atau yang lebih dikenal dengan nama CERN (Conseil Européen pour la Recherche Nucléaire) yang berada di Genewa, Swiss.

Tim Berners-Lee mengajukan sebuah proyek yang berbasiskan konsep hypertext untuk memfasilitasi pembagian dan pembaharuan informasi di antara para peneliti. Dengan bantuan dari Robert Cailliau dia menciptakan sistem prototipe bernama Enquire. Situs web pertama yang dibuat Tim Berners-Lee yang juga merupakan situs web pertama di dunia beralamat di <http://info.cern.ch/> dan online untuk pertama kalinya pada tanggal 6 Agustus 1991.” [5].

2.3.1 PERKEMBANGAN WEB

“Teknologi Web sedemikian berkembangnya sehingga para ahli telah memberikan penomoran untuk mengklasifikasikan generasi teknologi Web yang digunakan.

2.3.1.1 WEB 1.0

Web 1.0 merupakan teknologi awal dari website, dimana pembuat sebagai pemberi informasi dan pengguna hanya sebagai pembaca (seperti membaca koran lewat komputer, aktifitasnya hanya searching saja). Bahasa yang digunakan pada web ini masih berupa HTML saja.

2.3.1.2 WEB 2.0

Web 2.0 muncul sekitar tahun 2003 atau 2004, dimana para pengguna website dapat berkomunikasi 2 arah dan memiliki berbagai kelebihan lainnya.

Menurut Tim Berners-Lee sebenarnya tidak ada perbedaan fundamental antara Web 1.0 dengan Web 2.0. Ia menganggap Web 2.0 hanyalah jargon karena teknologinya tetap sama-sama dibangun dengan HyperText Markup Language (HTML) yang merupakan bahasa yang digunakan untuk menyusun konten suatu web.

Tim O'Reilly sebagai orang yang memperkenalkan istilah Web 2.0 sebenarnya juga tidak mengatakan bahwa Web 2.0 berbeda sama sekali. Ia mengakui bahwa Web 2.0 merupakan aplikasi berbasis web yang diperkaya oleh serangkaian aplikasi lain.

Pengertian Web 2.0 yang mulanya berpusat pada konsumen pembaca/pengakses secara personal berkembang dan mulai berpusat pula pada pengguna korporat. Menurut Coach Wei (2006) Web 2.0 yang berpusat pada konsumen ia sebut Consumer 2.0 berkembang menjadi Enterprise 2.0. Aplikasi Web 2.0 yang awal perkembangannya didominasi untuk memungkinkan pembaca

berinteraksi dengan pembuat berita dan pembaca lainnya, dalam Enterprise 2.0 aplikasi tersebut digunakan untuk mendukung operasi perusahaan.

Kelebihan dari Web 2.0 ini adalah sebagai berikut:

- 1. The Web as Platform (Pengerjaan suatu aplikasi/tulisan dapat langsung dikerjakan di media internet tanpa harus mengerjakannya terlebih dahulu di windows desktop)*
- 2. Harnessing Collective Intelligence (Web 2.0 memiliki kinerja untuk memanfaatkan tulisan orang lain untuk mengisi konten web secara kolektif (tidak hanya webmaster yang mengisi konten sendiri), contohnya seperti youtube)*
- 3. Data is the Next Intel Inside (merupakan suatu garansi kepercayaan dari para pemberi data kepada pemilik website bahwa pada era web 2.0 data sangatlah penting dan harus di update setiap waktu)*
- 4. End of the Software Release Cycle (pada web 2.0 aplikasi software dapat langsung digunakan lewat internet/internet menjadi platform menjalankan program)*
- 5. Lightweight Programming Models (pembuatan web 2.0 menggunakan bahasa yang ringan dan mendukung pengembangan program).*

2.3.1.3 WEB 3.0 / WEB SEMANTIK

Web 3.0 diyakini sebagai generasi ke tiga dari internet yang akan berkembang pada dekade 2010-2020. Pada Web 3.0 ini, sudah terjadi konvergensi yang sangat dekat antara dunia IT dengan dunia telekomunikasi. Sebenarnya konsep ini sudah diperkenalkan oleh Tim Berners-Lee sejak tahun 2001, namun sampai saat ini masih sangat sedikit website yang menerapkan Web 3.0. Web 3.0 memiliki konsep dimana manusia dapat berkomunikasi dengan mesin pencari. Web 3.0 memiliki beberapa kemampuan seperti berikut:

- 1. Semantic Web. Sebuah web dengan kemampuan membaca situs semudah manusia membacanya. Satu informasi yang dibutuhkan oleh manusia dapat*

dengan mudah tersajikan dengan korelasi informasi yang tepat dan cepat. Teknologi ini dikembangkan berbasis RDF format. Basis data dapat dijadikan acuan untuk pembuatan metadata, namun BibTex file dapat juga dijadikan acuan untuk pembuatan web semantik. Dengan menggunakan format XML yang merupakan dasar pembentukan metadata (RDF), maka berdasarkan format tersebut metadata dapat didistribusikan atau disimpan dalam domain yang beragam, data yang tersimpan berupa BibTex file di konversi ke format RDF, selanjutnya dengan menggunakan sesame metadata di load pada file index, SPARQL selanjutnya melakukan query terhadap metadata yang telah di load untuk ditampilkan di browser.

2. *The 3D Web.* Nuansa web semakin menarik dengan kemampuan visual 3 dimensi. Pengguna internet dapat mengunjungi berbagai tempat di dunia secara virtual dengan kemampuan akses data dan interaksi secara realtime tanpa harus meninggalkan rumah.
3. *The Media-Centric Web.* Keyword bukan lagi satu-satunya cara untuk mendapatkan informasi yang di tuju. Photo, audio, video akan menjadi cara lain untuk mencari informasi yang diinginkan.
4. *The Pervasive Web.* Web akan dengan mudah diakses dengan berbagai cara maupun perangkat. Tidak harus menggunakan komputer atau laptop. Berbagai perangkat mobile pun sekarang sudah sangat banyak yang mendukung.

Ditambah lagi dengan adanya HTML5 yang memungkinkan pemutaran media file tanpa perlu plugin Adobe Flash Player ataupun Microsoft Silverlight. Bahkan bisa membuat grafik 3D yang menawan hanya dengan HTML5, CSS3, Javascripts, dan WebGL. Ini merupakan generasi web yang luar biasa.”[5]

2.4 PENELITIAN TERDAHULU

Penelitian serupa telah banyak ditulis sebelumnya, antara lain “Penerapan Teknologi Semantic Web pada Aplikasi Pencarian Koleksi Perpustakaan” oleh tiga mahasiswa Teknik Informatika UPN “Veteran” Yogyakarta pada tahun 2010. Penelitian ini dilakukan untuk membuat aplikasi pencarian data pustaka seperti buku, jurnal, skripsi, dan laporan kerja praktek.

Penelitian tersebut menggunakan protege 3.4 dan Jena API sebagai frameworknya. Tools yang digunakan pada penelitian ini sama dengan penelitian sebelumnya yaitu protege 3.4.8, sedangkan frameworknya menggunakan Rdf API for PHP (RAP).