



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1. Social Media

Social media site dapat didefinisikan sebagai *website* yang isinya merupakan *user generated content* atau berasal dari kontributor atau pengguna, bukan dari pemiliknya. *Website* yang tergolong *social media site* cukup luas ruang lingkup aplikasinya. Berikut adalah beberapa kategori aplikasi *social media site* dan contohnya (Sukarto, 2009) :

1. *Blogs* : Wordpress, Blogger, Typepad, dan lain-lain.
2. *Microblogs* : Twitter dan Pownce.
3. *Social news site* : Digg, Mixx, Reddit, dan lain-lain.
4. *Social bookmarking* : Del.icio.us, StumbleUpon, dan lain-lain.
5. *Social networking* : Facebook, Friendster, LinkedIn, Myspace, Orkut dan lain-lain.
6. *Social network aggregation* : FriendFeed, Youmeo, dan lain-lain.
7. *Wikis* : Wikipedia, PBWiki, dan lain-lain.
8. *Photo sharing* : Flickr, Photobucket, dan lain-lain.
9. *Video sharing* : YouTube, Vimeo, dan lain-lain.
10. *Opinion site* : Epinions, Yelp, dan lain-lain.
11. *Online gaming* : World of Warcraft.
12. *Virtual worlds* : Second life.

Tetapi ada pendapat lain datang dari Stephen Wilson (2010), yang mendefinisikan *social media* sebagai sebuah *tools online* dimana memungkinkan orang untuk dengan ketertarikan yang sama dalam menyebarkan informasi, belajar dari orang lain atau berhubungan dengan jaringan yang terbuka untuk umum. Informasi yang ditemukan dalam *website* biasanya merupakan *user generated content* yang berarti dimana siapapun bisa melakukan *posting* ke dalam *website* dengan pembatasan yang minimal. Setiap pengguna mampu menyediakan konten dalam berbagai bentuk seperti tulisan atau *post*, foto, video, musik asli atau *bookmarked links*. Hal ini dapat berbentuk *unstructured* seperti foto keluarga di Facebook, atau sangat mendetail dan formal seperti artikel di Wikipedia.

Sejak konsep *social media* dianggap penting, penggunaannya pun diubah menjadi gratis. Dengan layanan gratis, maka *social media* telah memberikan kontribusi lebih jauh ke dalam kehidupan umat manusia yang lebih baik. Berikut beberapa catatan penting bahwa orang-orang sudah bisa menikmati hidup yang lebih baik dalam penggunaan *social media* (Brown, 2008) :

1. Dengan *social media*, dimana terdapat akses bebas untuk menghibur khususnya dengan menggunakan situs Youtube. Situs *social media* mampu menjembatani dunia hiburan dari seluruh belahan dunia.
2. Dengan *social media*, dimana terdapat akses bebas untuk informasi yang layak yang tidak mungkin dilakukan pada tahun-tahun sebelumnya. *Wiki's* yang menyimpan semua bentuk potongan naskah yang relevan dan penting dari informasi yang dibisa diakses oleh semua orang yang memiliki koneksi internet.

3. Dengan *social media*, dimana terdapat akses bebas untuk membangun pertemanan dan hubungan. Tren terbaru dimana orang-orang mulai “mencandu” adalah situs *social media* Friendster. Situs ini mengizinkan penggunanya untuk berhubungan dengan pengguna lainnya dan berhubungan kembali dengan orang lain tanpa membayar sepeserpun.

2.2. Web 2.0

World Wide Web telah mengubah cara orang-orang dalam berbisnis, berkomunikasi dan menyebarkan informasi dengan orang lain. Sejak awal tahun 1990-an, *world wide web* telah berkembang dari koleksi situs *web* yang berisi halaman statis atau yang jarang mengubah teks, gambar dan *hyperlinks* untuk men-*support* aplikasi web yang interaktif untuk melakukan transaksi bisnis, berbagi dan menampilkan foto dan video, *chatting*, dan kolaborasi. Nama web 2.0 diberikan setelah banyak penggunaan baru yang muncul sejak awal dekade kedua. Web 2.0 memiliki karakteristik seperti aplikasi interaktif yang mengizinkan penggunanya untuk berpartisipasi dan berkontribusi, mengorganisir dan membuat konten (Shelly, 2010).

Secara tradisional, untuk menggali lebih jauh arsitektur web 2.0 secara umum, akan digunakan metode untuk menggali berdasarkan contoh, maka akan dapat menangkap pengetahuan dan membangun model dan juga arsitektur berdasarkan kesamaan yang terdiri aturan arsitektur desain. Dalam MDA (*Model Driven Architecture*) konteks, desain arsitektur sistem didapat dari model dari sebuah sistem. MDA dikenal sebagai pendekatan dari *layer* untuk pemodelan dari aturan

desain arsitektur dan menggunakan pola desain untuk meningkatkan kualitas dari sistem *software* dan untuk memasukkan keamanan ke dalam sistem *software*, pola keamanan diperkenalkan dan ditawarkan di sisi level arsitektur (Das, 2011).

Istilah dari web 2.0 terdiri dari 3 area teknologi masing-masing AJAX dan RSS, *license* dan *social software*. Dari persepektif teknologi perspektif fasilitas penciptaan dan publikasi konten, itulah mengapa Coates menyebutkan bahwa web 2.0 sebagai “The Mass Amateurisation of (Nearly) Everything”. Mengubah *website* harian tidak lagi menjadi aktivitas harian yang dilakukan oleh para profesional (atau seseorang yang memiliki hobi dan *passion*) yang dapat menyelesaikannya. Sekarang hal ini terbuka bagi semua orang, tanpa biaya dan usaha yang sangat sedikit. Untuk satu hal, teknologi web 2.0 telah mengalami penyurutan menjadi *background* untuk desain antar muka pengguna yang intuitif yang bahkan pengguna yang paling teknis paham dan dapat membuat dan mempublikasikan sumber informasi (Peters, 2010).

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 terutama adalah membangun aplikasi yang mengeksploitasi efek jaringan untuk mendapatkan lebih banyak lagi pengguna aplikasi tersebut. Pada umumnya, *website* yang dibangun dengan menggunakan teknologi web 2.0 memiliki fitur-fitur sebagai berikut (Ibrahim, 2007):

1. CSS (*Cascading Style Sheet*).
2. Aplikasi *Rich Internet* atau berbasis AJAX.

3. Markup XHTML.
4. Sindikasi dan agregasi data menggunakan RSS/atom.
5. URL yang valid.
6. *Folksonomies*.
7. Aplikasi wiki pada sebagian atau seluruh website.
8. XML berbasis Web-Service API.

2.3. Application Programming Interface

Application Programming Interface atau biasa disingkat API merupakan satu set rutinitas dan *protocols* dimana sebuah program bisa melaluinya dan berkomunikasi dengan program yang lain (Horak, 2006). Dalam konteks web 2.0, sebuah pengembangan dalam *web services* disebut dengan *web API* dimana penekanan berpindah dari layanan berbasis *Simple Object Access Protocol* (SOAP) menjadi gaya komunikasi *Representational State Transfers* (REST). *Web API* mengijinkan kombinasi dari beberapa *web service* ke dalam aplikasi yang baru yang dikenal sebagai *mashups* (Prasad, 2011).

Tujuan dari web API adalah untuk mengaktifkan integrasi aplikasi. Secara lebih spesifik, sebuah web API memungkinkan untuk mengintegrasikan aplikasi yang heterogen. Penggunaan *web service* untuk banyak tujuan yang berbeda. Penggunaan tersebut dapat diimplementasikan aplikasi proyek integrasi *internal point to point*. Selain itu, *web service* dapat mengkonsolidasikan usaha pengembangan dan mengurangi ketergantungan antar aplikasi. Lalu yang terakhir,

penggunaan *web service* untuk implementasi tujuan umum integrasi untuk aplikasi internal (Manes, 2003).

2.4. Semantic Web

Semantic web didefinisikan sebagai ekstensi dari web yang sudah ada sekarang ini dimana informasi yang didapat diberikan arti yang baik, lebih membuat manusia dan komputer menjadi lebih bekerja sama. *semantic web* adalah sebuah visi, ide dimana data yang ada di dalam web dapat didefinisikan dan dihubungkan dengan cara yang dimengerti oleh mesin tidak hanya untuk menampilkan tujuan, namun juga otomatisasi, integrasi dan penggunaan kembali data dari berbagai aplikasi. Tujuan utama dari *semantic web* merupakan perluasan dari web, untuk menyediakan media untuk bertukar data (Kashyap, 2008).

Semantic web atau web 3.0 secara fundamental mengenai penggunaan teknologi yang membantu untuk *re-mix*, menggunakan kembali, dan kembali ke tujuan awal dari data di dalam web dengan cara yang baru. Web 3.0 memiliki beberapa karakteristik sebagai berikut (Pollock, 2009).

1. *Ubiquitous networking*

Web 3.0 membutuhkan data yang terhubung dan terjalin tanpa memperhatikan lokasi fisiknya. Perangkat dan jalur akses diasumsikan memiliki akses *web* atau protokol yang mampu mengakomodasi *bandwidth* rendah atau periode *downtime*.

2. *Open everything*

Web 3.0 bergantung pada tingkat otomatisasi dan kecerdasan. Akibatnya, banyak bagian dari jaringan harus tetap terbuka dan tidak boleh tertutup. *Open data*, *open service*, dan *open identity* adalah semua bagian dari visi web 3.0.

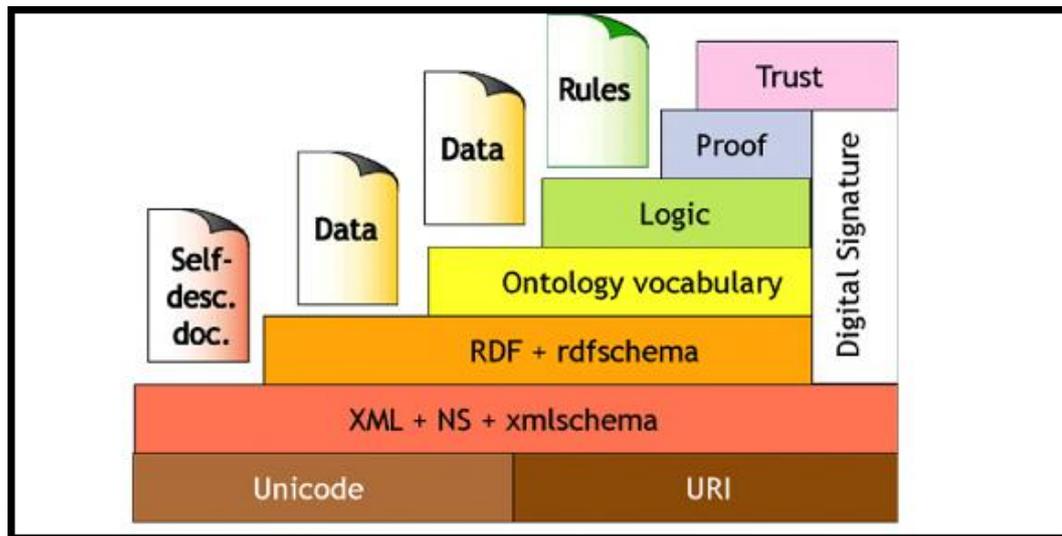
3. *Adaptative information*

Web 3.0 telah digambarkan sebagai Web data dan juga sebagai *executable* Web. Kedua label akurat. Menggunakan analogi pengolah kata, web 1.0 adalah seseorang yang mengubah dokumen, web 2.0 adalah sekelompok orang mengubah dokumen, dan web 3.0 adalah sekelompok orang menciptakan *bit data* dari luar dokumen sama sekali.

4. *Adaptive service clouds*

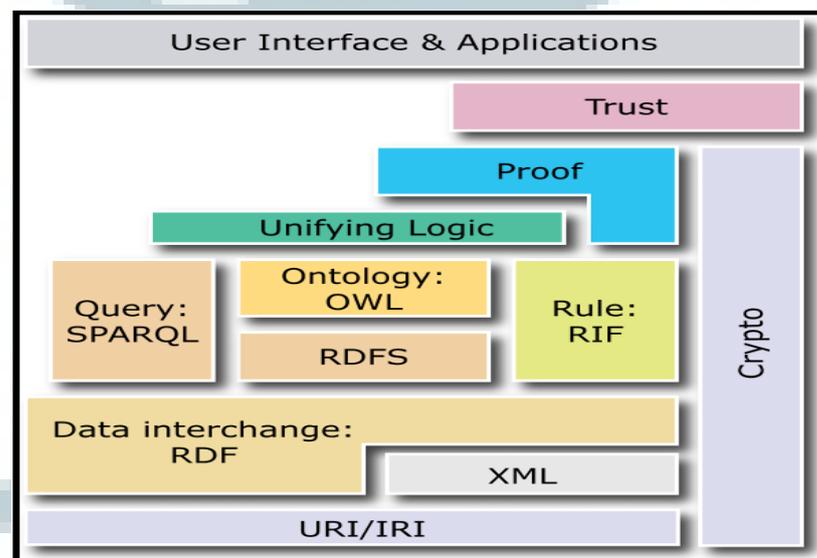
Dengan web 3.0, data merupakan sebuah *service*. Malahan menjadi layanan perangkat lunak yang mengerti tentang perilaku dan pemrograman antarmuka. Web 3.0 dan pergerakan *semantic* web yang memungkinkan publikasi dan konsumsi data dan model data sebagai layanan di dalam *cloud computing system* (aplikasi perangkat lunak yang disediakan via web protokol dan layanan).

Teknologi dan standar perlu didefinisikan tidak hanya untuk representasi sintaks untuk dokumen (seperti HTML), tetapi juga untuk konten *semantic*. *Semantic* interoperabilitas adalah sebuah fasilitas dari standarisasi W3C, terutama XML/XML *schema*, RDF/RDF *schema* dan OWL. Bentuk layer yang dibuat berdasarkan ketentuan W3C adalah sebagai berikut.



Gambar 2.1. Semantic web layer
(Semantic Web Services, 2007)

Semantic web layer sekarang telah mengalami evolusi yang lebih tepat dari segi efektifitas dan fungsional yang tidak hanya akan melayani arsitektur yang fundamental, tetapi juga sebagai rangkaian yang visioner untuk mengembangkan *semantic web* (Fensel, 2008).



Gambar 2.2. Semantic web layer
(Implementing Semantic Web Services: The Sesa Framework, 2008)

2.5. Resource Description Framework

Resource Description Framework adalah salah satu standar baru yang paling menarik yang muncul dalam *World Wide Web Consortium*. RDF adalah segala sesuatu tentang metadata dan membahas masalah dengan menggunakan format XML. Misalnya orang menggunakan kata *author* ketika orang lain menggunakan kata *writer* dan yang lainnya. Kedua bahasan tersebut bertujuan untuk menunjuk kepada objek yang sama namun dengan menggunakan tag yang berbeda untuk hal yang sama dan dapat membuat bingung mesin ketika ingin berbagi data dengan yang lain. Sebaliknya mencegah masalah fleksibilitas dan ekstensibilitas akan menghasilkan hasil yang kurang memadai dari *resource description*. Oleh karena itu, harus ada kerangka umum yang dapat menjembatani *gap* diantara beberapa jumlah skema. Ini adalah tahap dimana RDF muncul sebagai sebuah skema (Lingras, 2010).

Resource Description Framework merupakan basis standar dalam memproses *metadata*. RDF terdiri dari model untuk representasi dari nama *property* dan *value property*. Model dasar untuk RDF terdiri dari 3 tipe objek:

1. *Resources*

Setiap objek yang dideskripsikan oleh ekspresi RDF disebut *resources* dan dinamakan dengan *Uniform Resource Identifiers* (URI). Penggunaan URI *schema* (contohnya : http, ftp, nfs), setiap tipe dari *resource* bisa diidentifikasi dengan cara yang sama.

2. *Properties*

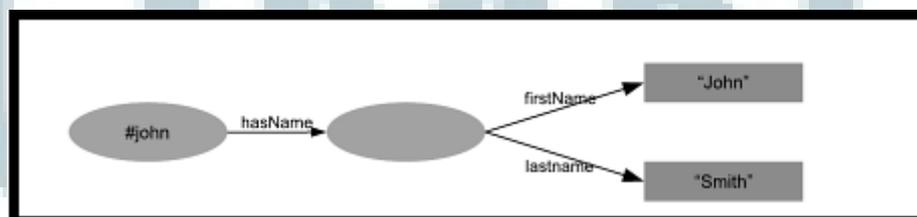
Sebuah *property* adalah aspek yang spesifik, karakteristik, atribut dan relasi yang digunakan untuk mendeskripsikan sebuah pernyataan. Setiap *property* memiliki makna yang spesifik, mendefinisikan *value* yang diijinkan, tipe dari *resources* yang bisa dispesifikasi, dan hubungan dengan *property* lain (via *RDF schema*).

3. *Statements*

Spesifik *resources* yang bersama dengan nama *property*, ditambah dengan *value* dari *property* untuk *resource* adalah sebuah *RDF statement*. Tiga bagian individual dari *statement* disebut dengan subyek, predikat, obyek. Obyek dari *statement* bisa dari *resource* atau *literal*.

Model data RDF menyediakan sebuah abstrak, *conceptual framework* untuk mendefinisikan *metadata* (Grigoras, 2002).

RDF memiliki tripel subyek predikat obyek, biasanya ditulis sebagai P(S,O), sebagai model data dasar. Sebuah obyek bisa dibalik sehingga menjadi subyek untuk obyek yang lain. Sebagai contoh *RDF graph* berikut (Lausen, 2007).



Gambar 2.3. RDF Graph Data
(Enabling Semantic Web Services , 2007)

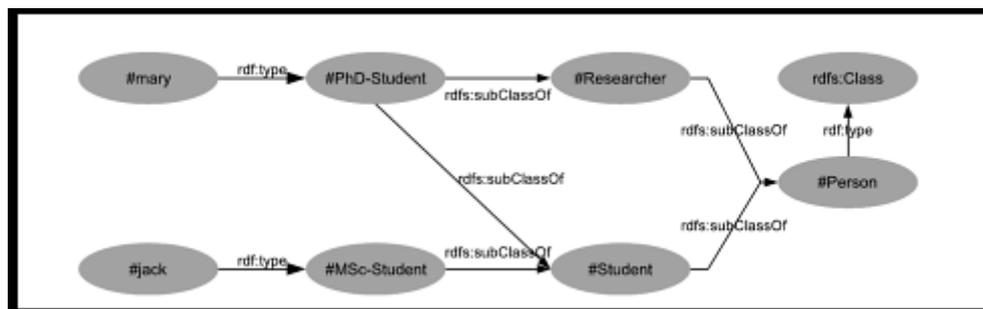
Sedangkan untuk serialisasi RDF sebagai berikut.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/#"
  xml:base="http://example.org/" >
  <rdf:Description about="#john" >
    <ex:hasName rdf:parseType="Resource" >
      <ex:firstName>John</ex:firstName>
      <ex:lastName>Smith</ex:lastName>
    </ex:hasName>
  </rdf:Description >
</rdf:RDF >
```

Gambar 2.4. RDF Serialization
(Enabling Semantic Web Services , 2007)

2.6. RDF Schema

RDF *schema* adalah sebuah ontologi yang ringan untuk mendefinisikan kosakata untuk RDF. Berbeda dengan XML schema, yang mengatur order dan kombinasi dari *tags* (struktur) dalam dokumen XML, RDF *schema* hanya menyediakan informasi tentang interpretasi dari pernyataan yang diberikan dalam model data RDF. RDF *schema* tidak mengatakan apapun tentang wujud sintaksis dari RDF *description*. RDFS bisa dilihat sebagai ekstensi dari RDF dengan kosakata untuk mendefinisikan *classes*, *class hierarchies*, dan *property restrictions*. RDF *classes* dan *properties* dapat insiasi dalam RDF. Sebagai perbandingan antara XML *schema* dan RDF *schema* dapat dilihat dari gambar berikut.



Gambar 2.5. RDFS Graph
(Enabling Semantic Web Services , 2007)

RDFS (yang menunjukkan kombinasi dari RDF dan RDF *schema*) sangat tidak ekspresif dibandingkan dengan ontologi yang lain, dimana hanya mengijinkan representasi dari konsep, konsep taksonomi, dan relasi biner (Fensel, 2007) .

2.7. Ontology Web Language

Ontology web language atau OWL adalah bagian lain dari *semantic web* karena tidak ada jaminan semua hal yang sama dapat dideskripsikan dengan cara yang sama kecuali menggunakan kosakata umum. Penambahan OWL untuk mengaktifkan keseragaman dalam kosakata, juga memfasilitasi hubungan antar istilah. Keseragaman dan integritas membuat pencarian dan agregasi dari produk menjadi lebih mudah, efisien dalam mencapai peningkatan skalaibilitas, interoperabilitas, fleksibilitas, ekstensibilitas, dan meringankan pemrograman. OWL tidak hanya melakukan spesifikasi metode untuk mendefinisikan elemen dan kosakata tetapi juga hubungannya. Terdapat 3 level dalam OWL adalah OWL Lite, OWL DL, dan OWL Full (Sankar, 2009)

Spesifikasi OWL memiliki beberapa persyaratan seperti yang tertulis dalam W3C *Consortium* dalam mendefinisikan deksripsi bahasa ontologi untuk *semantic*

web. Beberapa poin utama untuk *requirements* tersebut adalah sebagai berikut (Breitman, 2007).

1. Desain untuk bahasa sebaiknya kompatibel dengan XML, dalam arti bahwa :
 - a) Sebuah ontologi sebaiknya dalam bentuk sintaks XML.
 - b) Sebuah ontologi sebaiknya menggunakan tipe data skema XML, yang dapat digunakan.
2. Desain untuk bahasa sebaiknya mengikuti logika deskripsi, dalam arti bahwa :
 - a) Bahasa sebaiknya didasarkan pada gagasan konsep atau *classes*, *role* atau *property*, dan individual.
 - b) Bahasa sebaiknya mendukung ekspresi.
3. Desain untuk bahasa sebaiknya mendukung definisi dari kosakata ontologi, dalam arti bahwa :
 - a) Ontologi sebaiknya diidentifikasi oleh referensi URI.
 - b) *Classes*, *property*, dan individual untuk ontologi sebaiknya diidentifikasi oleh referensi URI.
4. Desain untuk bahasa sebaiknya memfasilitasi :
 - a) Pengembangan ontologi untuk gaya distribusi.
 - b) Definisi untuk versi yang berbeda untuk ontologi yang sama.
 - c) Penggunaan kembali ontologi yang telah didefinisikan sebelumnya.

Perlu diketahui bahwa OWL menggunakan RDF dan sintaks XML. Lebih jauh, OWL lebih kuat dalam kapabilitas *reasoning*. Gambar 2.6 merupakan contoh dari kode OWL (Thuraisingham, 2010).

```

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.owl-ontologies.com/
Ontology1178660130.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1178660130.
owl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" >
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="AnnualReports">
    <rdfs:subClassOf rdf:resource="#Department" />
  </owl:Class>
  <owl:ObjectProperty rdf:ID="canAccess">
    <rdfs:domain rdf:resource="#Faculty" />
    <rdfs:range rdf:resource="#AnnualReports" />
    <rdfs:subPropertyOf rdf:resource="#isDean" />
  </owl:ObjectProperty>
  <owl:Class rdf:ID="Course">
    <rdfs:subClassOf rdf:resource="#Department" />
  </owl:Class>
  <owl:Class rdf:ID="Department" />
  <owl:Class rdf:ID="Faculty">
    <rdfs:subClassOf rdf:resource="#Department" />
    <owl:disjointWith rdf:resource="#Student" />

```

Gambar 2.6. OWL Code
(Secure Semantic Service – Oriented Systems , 2010)

Selain itu, terdapat ontologi yang telah memenuhi standar W3C dan memiliki dokumentasi, menggunakan provider data yang independen dan didukung oleh beberapa *tools*. Berikut ontologi yang memiliki dokumentasi lengkap tersebut (*Good Ontologies*, 2012).

1. *Dublin Core Ontology*
2. *Friend of A Friend (FOAF)*
3. *Socially Interconnected Online Communities (SIOC)*
4. *Good Relations dan Music Ontology*

2.8. Friend of A Friend

Secara sederhana, *Friend of a Friend* atau FOAF merupakan sebuah kosakata atau ontologi yang di dalamnya terdapat istilah dasar untuk mendeskripsikan data pribadi seseorang, seperti siapa diri anda, apa yang anda lakukan, dan siapa teman anda. FOAF berfungsi sebagai standar untuk semua orang yang ingin meningkatkan kualitas *home pages*-nya dan membalikinya ke dalam sebuah dokumen yang mampu dibaca oleh mesin. FOAF bukan merupakan standar resmi dari W3C, melainkan sebuah proyek *open source* yang dikembangkan oleh *developer*. Hanya saja pengembangan FOAF menggunakan standar dari W3C yakni RDF dan OWL. Pengertian secara spesifik adalah sebagai berikut (Yu, 2011).

1. Ontologi FOAF ditulis dalam OWL.
2. Dokumen FOAF mengikuti standar dokumen RDF.

Berdasarkan esensinya, kosakata FOAF mengizinkan untuk menyerupai *social networks* dengan cara yang terdesentralisasi. Oleh karena itu, setiap pengguna dapat membuat profil FOAF-nya sendiri dalam menentukan data informasi dasar seperti alamat atau pekerjaan dan mengizinkan orang lain tahu hubungan yang ada diantaranya. Gambar 2.7 menunjukkan potongan contoh kode dari data FOAF.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  <foaf:PersonalProfileDocument rdf:about="">
    <foaf:maker rdf:resource="#me"/>
    <foaf:primaryTopic rdf:resource="#me"/>
    <admin:generatorAgent rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  </foaf:PersonalProfileDocument>
  <foaf:Person rdf:ID="me">
    <foaf:name>Hans-Joerg Happel</foaf:name>
    <foaf:depiction rdf:resource="http://www.hjhappel.de/images/site/hj.jpg"/>
    <foaf:phone rdf:resource="tel:+49-(0)-721-9654-814"/>
    <foaf:workplaceHomepage
      rdf:resource="http://www.fzi.de/ipe/eng/mitarbeiter.php?id=418"/>
    <foaf:knows>
      <foaf:Person>
        <foaf:name>Walid Maalej</foaf:name>
        <foaf:mbox
          rdf:resource="mailto:maalejw@in.tum.de"/></foaf:Person></foaf:knows>
      <foaf:knows>
        <foaf:Person>
          <foaf:name>Stefan Seedorf</foaf:name>
          <foaf:mbox rdf:resource="mailto:seedorf@uni-mannheim.de"/>
        </foaf:Person></foaf:knows>
    </foaf:Person>
  </rdf:RDF>

```

Gambar 2.7. FOAF Code
(Collaborative Software Engineering, 2010)

2.9. SPARQL

RDF diarahkan ke dalam bentuk berlabel grafik format data untuk mewakili informasi di dalam web. Spesifikasi ini mendefinisikan sintaks dan *semantic* ke dalam bahasa *query* SPARQL untuk RDF. SPARQL dapat digunakan untuk mengeskpresikan *query* untuk sumber data yang beragam baik data RDF yang bersifat *native* maupun data RDF yang dilihat melalui *middleware*. SPARQL memiliki kemampuan untuk melakukan *query* yang diperlukan maupun yang pola grafik opsional bersama dengan konjungsi atau tanpa konjungsi. SPARQL juga mendukung pengujian nilai ekstensibel dan membatasi *query* oleh grafik data RDF. Hasil *query* SPARQL dapat berupa sebuah set atau grafik RDF (SPARQL Query Language of RDF, 2013).

Berikut beberapa contoh SPARQL *query* :

1. *Query* untuk *select* :

```
Query:

SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

Gambar 2.8. SPARQL Select Code
(<http://www.w3.org/TR/rdf-sparql-query/>, 2013)

2. *Query* untuk *insert* :

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
INSERT DATA
{ GRAPH <http://example/bookStore> { <http://example/book1> ns:price 42 } }
```

Gambar 2.9. SPARQL Insert Code
(<http://www.w3.org/TR/sparql11-update/#updateLanguage>, 2013)

3. *Query* untuk *delete* :

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

DELETE DATA
{
  <http://example/book2> dc:title "David Copperfield" ;
  dc:creator "Edmund Wells" .
}
```

Gambar 2.10. SPARQL Delete Code
(<http://www.w3.org/TR/sparql11-update/#updateLanguage>, 2013)

2.10. ARC2

ARC2 dibuat pada tahun 2004 sebagai sistem RDF yang ringan untuk melakukan *parsing* dan melakukan serialisasi terhadap *file* XML atau RDF.

Kemudian ARC2 berkembang menjadi kerangka yang lebih lengkap dengan model penyimpanan dan fungsi query. Pada tahun 2011, ARC2 telah menjadi salah satu *library* RDF yang paling banyak dipakai. Namun demikian, pengembangan kode secara aktif harus dihentikan karena mengalami masalah dalam hal biaya dan ketidakmampuan untuk secara efisien melakukan pengembangan terhadap spesifikasi RDF yang terus berkembang (Nowack, 2012).

Berikut pemakaian ARC2 secara sederhana :

```
include_once("path/to/arc/ARC2.php");

$config = array(
  /* db */
  'db_name' => 'my_db',
  'db_user' => 'user',
  'db_pwd' => 'secret',
  /* store */
  'store_name' => 'arc_tests',
  /* stop after 100 errors */
  'max_errors' => 100,
);
$store = ARC2::getStore($config);
if (!$store->isSetUp()) {
  $store->setUp();
}

/* LOAD will call the Web reader, which will call the
format detector, which in turn triggers the inclusion of an
appropriate parser, etc. until the triples end up in the store. */
$store->query('LOAD <http://example.com/home.html>');

/* list names */
$q = '
PREFIX foaf: <http://xmlns.com/foaf/0.1/> .
SELECT ?person ?name WHERE {
  ?person a foaf:Person ; foaf:name ?name .
}
';
$r = '';
if ($rows = $store->query($q, 'rows')) {
  foreach ($rows as $row) {
    $r .= '<li>' . $row['name'] . '</li>';
  }
}

echo $r ? '<ul>' . $r . '</ul>' : 'no named persons found';
```

Gambar 2.10. ARC2 Sample Code
(<https://github.com/semsol/arc2/wiki/Getting-started-with-ARC2>, 2012)