

IMPLEMENTASI WEB SEMANTIK DENGAN APACHE JENA

Bernard Renaldy Suteja

Teknik Informatika, Universitas Kristen Maranatha

Jalan Surya Sumantri 65 Bandung

E-Mail : bernard.rs@it.maranatha.edu

ABSTRACT

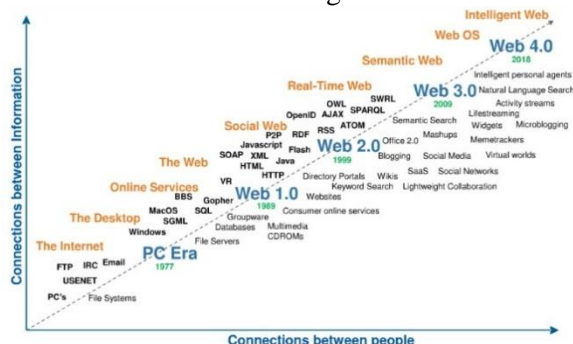
The information available on the Web can be anything (hypermedia) and about anything. Anybody can be a suspect emergence of false or misleading information. The web site is currently present with no central point of control and can not be used as a central repository. Apache Jena was present to address that fact and can be used in the implementation of semantic web that is easily recognizable by the machine without losing the good things from the web itself as well as the change from a decentralized platform for presentation to a decentralized distribution platform for the distribution of knowledge. In this paper delivered JENA implementation in developing semantic web applications in basic.

Keywords : Web, Semantic, JENA

PENDAHULUAN

Perkembangan teknologi web yang pesat hingga saat ini telah mendorong perubahan fungsi dari hanya sekedar penyajian informasi yang terformat secara menarik, efisien hingga menghubungkan informasi-informasi yang tersaji dalam web. Web-web yang ada dapat terkoneksi dengan sebuah *hyperlink*, berpindah dari satu web ke web lainnya dan menjadi referensi antar informasi yang ada pada web. Informasi yang tersedia pada web dapat berupa apa saja (*hypermedia*) dan tentang apa saja. Siapa saja dapat menjadi pelaku munculnya informasi yang benar atau menyesatkan. Sehingga web yang ada saat ini hadir tanpa adanya *central point of control* serta tidak dapat digunakan sebagai *central repository*.

Banyaknya informasi yang tersebar dalam web site tersebut dapat digunakan sebagai sebuah basis pengetahuan (*knowledge base*). Pengetahuan diolah oleh mesin dengan suatu pendekatan-pendekatan tertentu. Sebuah web page secara praktis dibuat oleh manusia namun sulit dimengerti oleh mesin.



Gambar 1. Perkembangan teknologi web

Permasalahan inilah yang ingin diatasi oleh web semantik. Web semantik hadir untuk menghasilkan web data yang mudah dikenali oleh mesin tanpa kehilangan hal baik dari web itu sendiri serta mengubah dari desentralisasi platform untuk distribusi tampilan (*presentation*) menuju desentralisasi platform untuk distribusi pengetahuan (*knowledge*).

Melalui penelitian ini mencoba dilakukan implementasi web semantik untuk menghadirkan informasi yang saling terhubung secara konsisten menjadi sebuah basis pengetahuan dengan menggunakan Apache JENA.

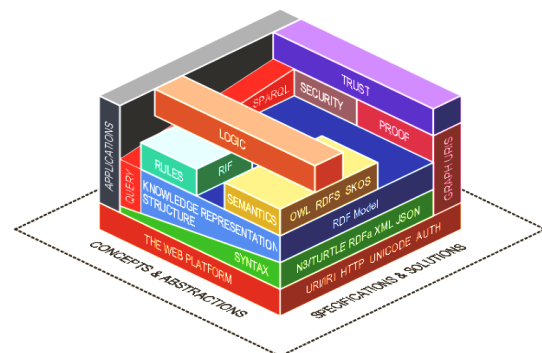
Web Semantik

Penjelasan web semantik dapat diuraikan dari arti katanya. Web merupakan informasi yang terformat dengan baik dan bisa diakses secara online. Semantik adalah makna relasi antar tanda, kata dan sesuatu (*thing*) yang dapat berupa dokumen, orang, tempat kejadian, organisasi, konsep dan lain-lain (Antonio, G., Groth, P., Harmelen, F.V., & Hoekstra, R., 2012). Sebagai ilustrasi dapat diperhatikan gambar 2, yang menjelaskan dua kalimat (*syntax*) memiliki makna (*semantic*) yang sama.



Gambar 2. Contoh pemaknaan kata

Web semantik merupakan buah pikir dari Sir Tim Berners-Lee, yang sekaligus penemu WWW, URI, HTTP, dan HTML. Sir Tim Berners-Lee mendefinisikan Web semantik sebagai pengembangan dari web yang ada saat ini dengan informasi yang memiliki makna yang didefinisikan dengan baik (*welldefined meaning*), sehingga manusia dan mesin dapat bekerja sama. Web semantik yang lebih dikenal dengan istilah web 3.0 adalah cara efisien untuk merepresentasikan data pada World Wide Web, atau sebagai database yang terhubung secara global (Abadi, D.J., Marcus, A., Madden, S.R., & Hollenbach, K., 2009).



Gambar 3. Layer-layer teknologi web semantik

Pembuatan web semantik melibatkan sekumpulan standar yang dikoordinasi oleh World Wide Web Consortium (W3C). Standar dalam rancangan pembuatannya konsep web semantik terbagi menjadi beberapa layer teknologi seperti pada gambar

3, Teknologi ini terdiri atas (Daconta, M.C., Obrst, L.J., dan Smith, K.T):

- **Web Platform** meliputi Unicode dan URI.
Unicode adalah standard representasi karakter komputer. URI (Uniform Resources Identifier) merupakan standard untuk lokasi dan identitas suatu resource (misalnya web page).
- **Syntax** meliputi N3, N-Triple, Turtle (Terse RDF Triple Language), RDFa, XML, dan JSON : N3 (Notation 3), N-Triple, Turtle (Terse RDF Triple Language) merupakan *serialization syntax for RDF* atau Representasi RDF dalam bentuk triple. RDFa (Resource Description Framework in Attributes) merupakan penambahan atribut pada XHTML yang berbasis pada RDF *subject-predicate-object expressions*. XML (Extensible Markup Language) dan Namespace, merupakan aturan sintaks yang berfungsi untuk menyajikan struktur data pada web, namun tidak menekankan batasan pada makna dari dokumen tersebut. XML Schema merupakan bahasa untuk membatasi struktur XML dan memperkaya XML dengan datatype. JSON (JavaScript Object Notation) merupakan bentuk sederhana berbasis teks untuk pertukaran data.
- **Knowledge Representation Structure** merupakan RDF (*Resource Description Framework*) sebuah model untuk representasi pengetahuan secara sederhana dalam format triple yang dapat direpresentasikan dalam bentuk graph untuk menjelaskan resource dan relasinya.
- **Semantic** merupakan representasi pengetahuan tahap lanjutan dengan menjelaskan property dan kelas dari resource yang terdapat dalam RDF dikenal dengan RDF Schema (RDFS) serta bahasa ontologi (*Ontology vocabulary*) yang direkomendasikan oleh W3C pada 10 Februari 2004 dikenal dengan Web Ontology Language (OWL). OWL merupakan bahasa yang lebih kaya (perluasan dari RDF) dan kompleks untuk mendeskripsikan resource. OWL menggunakan format triple, sama seperti RDF.

- **Rules** merupakan *Rule Interchange Format* (RIF) yang dapat digunakan untuk melakukan reasoning pada ontologi sehingga dapat disimpulkan apakah suatu resource memenuhi syarat tertentu.
- **Query** diimplementasikan dengan menggunakan SPARQL. SPARQL merupakan bahasa query untuk RDF.
- **Security** diimplementasikan dengan melakukan autentifikasi protocol SSL/TLS.
- **Proof** bagian ini menentukan validasi sebuah web semantik termasuk didalamnya interoperabilitas dan skalabilitas. Bagian ini ditunjukkan dengan dimungkinkan terjadinya multiple RDF graf.
- **Trust** bagian ini memungkinkan pengguna web semantik untuk mempercayai suatu informasi pengetahuan.
- **Application** bagian teratas dari layer ini merupakan bentuk pemanfaatan web semantik dalam bentuk aplikasi web.

URI dan Name Space

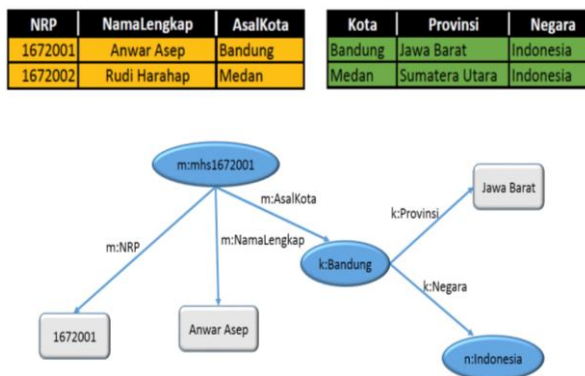
Sebuah resource dalam web semantik merupakan sumber ide yang abstrak atau kongkret. Resource dapat berupa apapun seperti: manusia, bangunan, cuaca, musik, dan sebagainya. Potensi resource memiliki beberapa nama yang akan berakibat pada asumsi dan ambiguitas. Menghindari hal tersebut maka digunakan penamaan yang unik yaitu URI. URI singkatan dari Uniform Resource Identifier. Bagian yang tidak terpisahkan dari URI adalah URN dan URL (Nowack, B., 2009). URN (Uniform Resource Name) merupakan identifikasi resource dengan memberikan nama kode, contohnya: isbn:9789791153362. Sedangkan URL merupakan identifikasi sumber dengan memberikan alamat web site, contohnya: <http://www.book.com/isbn/9789791153362.pdf>

Name Spaces (NS) adalah prefix (awalan pendek) yang juga merupakan qualified name (qnames). Hadirnya NS dapat digunakan untuk mengatasi permasalahan dalam penggunaan URI yang terlalu panjang. Representasi resource yang panjang diwakilkan pada sebuah prefix. Keuntungan penggunaan NS adalah lebih mudah dibaca dan hemat ruang penyimpanan. Sebagai

contoh terdapat 3 resource dengan URI sebagai berikut :
<http://www.book.com/bernard/book1>,
<http://www.book.com/bernard/book2>, dan
<http://www.book.com/bernard/book3>. Maka dapat didefinisikan sebuah NS yaitu bnad: untuk meningkatkan <http://www.book.com/bernard/>, sehingga penamaan URI tersebut menjadi bnad:book1, bnad:book2, dan bnad:book3.

Resource Description Framework (RDF)

Dalam web semantik diperlukan metode untuk mengurai pengetahuan dan merepresntasikannya kedalam bagian-bagian yang lebih kecil sesuai dengan maknanya (decompose knowledge). RDF digunakan sebagai metode untuk menggambarkan fakta dalam bentuk bagian-bagian yang sederhana. Dalam RDF ini di deskripsikan resource dan relasinya. RDF direpresentasikan dalam bentuk graph (gambar 4) serta dapat diserialisasi notasi dalam bentuk text (Suteja, B Renaldy, 2016).



Gambar 4. Representasi Graph

Serialisasi notasi dalam bentuk text meliputi RDF/XML, TRIX, N-triples, Turtle, dan JSON. RDF/XML serta TRIX memiliki representasi dalam bentuk dokumen XML sehingga dapat digunakan secara lintas platform, cukup dibutuhkan adanya XML parser untuk memperoleh nilai nodenya. N-triples dan Turtle dipresentasikan dalam bentuk statement yang tersusun secara subyek, predikat dan obyek. JSON merupakan representasi yang sangat ringan dan bersih serta dapat digunakan lintas platform. Contoh hasil luaran untuk RDF/XML dapat dilihat pada gambar 5 (Suteja, B Renaldy, 20156).

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:n="http://www.somewhere.com/negara#"
  xmlns:m="http://www.universitas.com/mahasiswa#"
  xmlns:k="http://www.somewhere.com/kota#"
  <rdf:Description rdf:about="http://www.universitas.com/mahasiswa#mhs1672001">
    <m:AsalKota>
      <rdf:Description rdf:about="http://www.somewhere.com/kota#Bandung">
        <n:Negara rdf:resource="http://www.somewhere.com/negara#Indonesia"/>
        <k:Provinsi>Jawa Barat</k:Provinsi>
      </rdf:Description>
    </m:AsalKota>
    <m:NamaLengkap>Anwar Asep</m:NamaLengkap>
    <m:NRP>1672001</m:NRP>
  </rdf:Description>
</rdf:RDF>
```

Gambar 5. Serialisasi notasi RDF/XML

Berikut adalah contoh hasil luaran untuk Turtle dapat dilihat pada gambar 6 (Suteja B Renaldy, 2016).

```
@prefix k: <http://www.somewhere.com/kota#> .
@prefix m: <http://www.universitas.com/mahasiswa#> .
@prefix n: <http://www.somewhere.com/negara#> .

m:mhs1672001 m:AsalKota k:Bandung ;
             m:NRP      "1672001" ;
             m:NamaLengkap "Anwar Asep" .

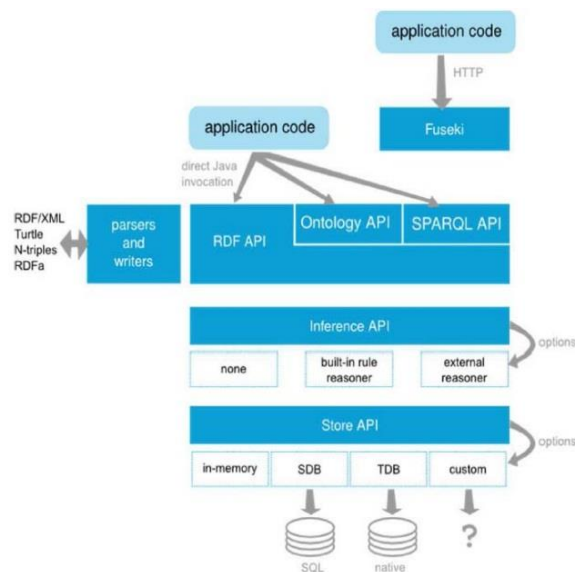
k:Bandung k:Provinsi "Jawa Barat" ;
          n:Negara   n:Indonesia .
```

Gambar 6. Serialisasi notasi Turtle

Apache Jena

Apache Jena adalah framework opensource berbasis Java untuk membangun aplikasi web semantik. Apache Jena menyediakan sekumpulan tool dan library Java untuk membantu membangun aplikasi berbasis pada linked data. Apache Jena framework terdiri dari (Yu, L., 2011):

- API untuk membaca, memproses, menulis RDF data dalam format XML, N-triples, dan Turtle.
- API Ontologi untuk menangani ontologi pada OWL dan RDFS.
- Rule-based inference engine untuk reasoning dengan RDF dan OWL data source.
- Menyimpan RDF triple dalam jumlah besar secara efisien pada disk.
- Query engine dengan spesifikasi SPARQL terbaru.



Gambar 7. Layer-layer pada framework Jena

Implementasi statement dalam Jena dilakukan dengan import `org.apache.jena.rdf.model.*`. Selanjutnya melakukan pendefinisian URI graph diikuti dengan definisi modelnya. URI yang sudah didefinisikan dapat dibuatkan Name Space sebagai prefix qnames. Pada tahap selanjutnya dilakukan pendefinisian statement yang meliputi subyek (resource), predikat (property) dan obyek. Obyek dapat berupa resource yang lainnya atau Literal. Literal tersedia dalam 2 bentuk yaitu tidak bertipe dan bertipe. Untuk yang bertipe harus import `org.apache.jena.datatypes.xsd.XSDDatatype`.

Tabel 1. berikut merupakan type data XSD yang didukung dalam Jena serta berbandingannya dengan type data pada Java.

Tabel 1. Type data dalam Jena

Java class	XSD type
Float	float
Double	double
Integer	int
Long	long
Short	short
Byte	byte
BigInteger	integer
BigDecimal	decimal
Boolean	boolean
String	string

Implementasi Jena dalam kode program secara lengkap untuk contoh pada gambar 4 adalah sebagai berikut (Suteja, B Renaldy, 2016) :

```
<%@ page import="org.apache.jena.rdf.model.*" %>
<%@ page import="org.apache.jena.vocabulary.*" %>
<%@ page contentType="text/xml" pageEncoding="UTF-8"%>

<%
String mahasiswaURI = "http://www.universitas.com/mahasiswa#";
String kotaURI = "http://www.somewhere.com/kota#";
String negaraURI = "http://www.somewhere.com/negara#";

Model model = ModelFactory.createDefaultModel();
model.setNsPrefix("m", mahasiswaURI);
model.setNsPrefix("k", kotaURI);
model.setNsPrefix("n", negaraURI);

Resource s = model.createResource(mahasiswaURI+"mhs1672001");
Property p = model.createProperty(mahasiswaURI + "NRP");
Literal o = model.createLiteral("1672001");
model.add(s,p,o);

p = model.createProperty(mahasiswaURI + "NamaLengkap");
model.add(s,p,"Anwar Asep");

Resource s1 = model.createResource(kotaURI+"Bandung");
p = model.createProperty(mahasiswaURI + "AsalKota");
model.add(s,p,s1);

p = model.createProperty(kotaURI + "Provinsi");
model.add(s1,p,"Jawa Barat");

Resource s2 = model.createResource(negaraURI+"Indonesia");
p = model.createProperty(negaraURI + "Negara");
model.add(s1,p,s2);

model.write(out,"RDF/XML-ABBREV");
%>
```

Simple Protocol And RDF Query Language (SPARQL)

SPARQL merupakan bahasa query untuk RDF. Graph RDF terdiri dari triple statement yang terbentuk dari Subyek, Predikat dan Objek[6]. Dalam hal ini RDF dapat di definisikan pada RDF Konsep dan Abstrak Sintaks. Triple dapat datang dari berbagai macam sumber resource. Untuk Instance dapat diperoleh secara langsung dari dokumen RDF. Ekspresi RDF dapat disimpan dalam format lain seperti XML dan Database Relational.

Query SPARQL digunakan untuk memperoleh informasi dari Graph RDF. Fasilitas yang disediakan oleh SPARQL adalah sebagai berikut :

- Mengekstrak informasi dalam bentuk URI, Blank Node dan Literal
- Mengekstrak RDF Subgraph
- Membangun Graph RDF baru berdasarkan query graph

Sebagai bahasa akses data maka SPARQL sangat sesuai jika digunakan untuk local maupun remote.

Bentuk umum dari syntax penulisan SPARQL adalah sebagai berikut :

```
PREFIX (Namespace Prefixes)
PREFIX
m:<http://www.universitas.com/mahasiswa
#>
SELECT (Result Set)
SELECT ?nama
FROM and FROM NAMED (Dataset)
FROM <http://bernard.com/info.rdf>
WHERE (Query Triple Pattern)
WHERE { m:mhs1672001 m>NamaLengkap
?nama }
ORDER BY, DISTINCT, HAVING,
LIMIT, dst. (Modifiers)
ORDER BY ?nama
```

Penggunaan SPARQL dalam Jena dapat dilakukan dengan melakukan import org.apache.jena.query.* terlebih dahulu. Berikut adalah contoh implementasi SPARQL dalam Jena (Suteja, B Renaldy, 2016):

```
String perintahSPARQL = "PREFIX
m:<http://www.universitas.com/mahasiswa#> SELECT ?nama WHERE
{ m:mhs1672001 m>NamaLengkap ?nama .}";
Query query = QueryFactory.create(perintahSPARQL);
QueryExecution qe = QueryExecutionFactory.create(query, mode
ResultSet hasil = qe.execSelect();

String output = ResultSetFormatter.asText(hasil);
out.println(output);
```

Hasil luaran dari implementasi SPARQL pada Jena adalah sebagai berikut :

```
-----
| nama |
=====
| "Anwar Asep" |
-----
```

KESIMPULAN

Jena menyediakan integrasi rekomendasi implementasi dari W3C Web Semantik yang fokus pada RDF Graph. Jena juga hadir dengan dilengkapi kemampuan query informasi dengan menggunakan

SPARQL sehingga dapat digunakan untuk menghasilkan informasi dan membantu dalam proses inferensi.

DAFTAR PUSTAKA

- Antonio, G., Groth, P., Harmelen, F.V., dan Hoekstra, R., 2012, *A Semantic Web Primer (Information Systems)*, The MIT Press.
- Abadi, D.J., Marcus, A., Madden, S.R., dan Hollenbach, K., 2009, SW-Store: a vertically partitioned DBMS for SemanticWeb data management, The VLDB Journal Springer-Verlag, vol 18, 385–406.
- Daconta, M.C., Obrst, L.J., dan Smith, K.T., The Semantic Web, Wiley Publishing, Inc.
- Nowack, B., The Semantic Web - Not a piece of cake..., <http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>, diakses 20 April 2015.
- Suteja B Renaldy, 2016, Diktat Web Semantik , UK. Maranatha.
- Yu, L., 2011, A Developer's Guide to the Semantic Web, Springer.

