DESAIN *RULES* UNTUK TRANSFORMASI SINTAKS RDF (TEXT) KE RDF BERBASIS GRAFIK DALAM SEMANTIC WEB

Aditya Prapanca
FMIPA Universitas Negeri Surabaya
Kampus Ketintang, Surabaya 60231
E-mail: aditya_p5@unesa.ac.id

***ABSTRACT**: Semantic Web is a branch of field that provides infrastructure for the exchange of information from various resources based on the combined meaning contained in the information exchanged and the relationship between one particular meaning to another. In the semantic web, there are semantics, which connect a set of data that has a meaning which allows occurrence of the exchange of information. The aim of this study is to make rules used in the process of transformation from the RDF document to be RDF graph, the data model so that it can provide semantic meaning in a web of visual content (graphic). By using these rules, a case can be made based graphical tool that can help the designer in created the data model. The results are models form of graphical data input from text-based documents RDF. For the transformation process, starting with a separate document, then taken sequentially the object, and then take the attributes of the object, and the last search for the relationship of each object that is.*

***Keywords**: Rules, Case Tools, Data Model, graphic notation, Semantic Web*

PENDAHULUAN

Hingga hari ini perkembangan dunia web (*World Wide Web*) sangatlah pesat. Berbagai web dengan *content* yang bermacam-macam bermunculan, baik itu berupa teks, gambar, audio ataupun video streaming telah berkembang pesat dalam dunia web.

World Wide Web juga merupakan tempat penyimpanan informasi terbesar yang pernah dibuat, terutama dengan semakin bertumbuhnya *content* dalam berbagai ragam bahasa dan berbagai ragam bidang pengetahuan. Dengan semakin besarnya *content* dari *World Wide Web* ini, akan semakin sulit bagi *user* untuk melakukan pencarian *content* yang dinginkan secara tepat.

Mesin pencari (*search engine*) mungkin bisa membantu dalam mencari *content* yang berisi kata-kata khusus yang di inginkan, namun juga terbuka peluang bahwa *content* yang ditemukan dari hasil pencarian ini tidak tepat seperti apa yang diinginkan. Biasanya yang muncul dari hasil pencarian sangatlah banyak. Penggunalah yang harus aktif berperan dalam memilih topik yang tepat yang diperlukan, dengan memilih satu-persatu *list* yang ditampilkan oleh *search engine* tersebut. Hal ini dikarenakan pencarian konvensional berdasarkan pada *keywords* yang mungkin muncul pada *content* dari halaman web dan bukannya berdasarkan pada *semantic meaning* (arti yang terkandung) atau konteks dari *content* halaman tersebut.

Semantic Web memungkinkan dilakukannya pendataan *content* yang ada di web, mendeskripsikan informasi yang dikandungnya dan memberikan *semantic meaning* bagi *content* tersebut. Dengan hal ini, mesin pencari menjadi lebih efektif dalam melakukan tugasnya dibandingkan sebelummya, sehingga user mendapatkan informasi yang tepat sesuai dengan yang dia inginkan 1

Teknologi *Semantic web* menawarkan pendekatan baru untuk memecahkan beberapa permasalahan yang timbul dari proses pencarian berbasis kata kunci secara konvensional. Pendekatan ini lebih dikenal dengan istilah metode temu-kembali berbasiskan RDF/RDFS. Pendekatan berbasis RDF/ RDFS diterapkan baik pada proses pemuatan data maupun pada proses temu-kembali[8].

Resource Description Framework (RDF) adalah bahasa yang digunakan untuk merepresentasikan informasi tentang suatu *resource* dan untuk mempertukarkannya di Web. RDF merupakan bahasa yang digunakan untuk memodelkan informasi dari sekumpulan data. Sedangkan data model merupakan suatu tata bahasa yang digunakan untuk mengartikan sekumpulan data untuk menjadi suatu informasi.

Pada *Semantic Web* memungkinkan dilakukannya pendataan seluruh *content* yang ada di web, mendeskripsikan informasi yang dikandungnya dan memberikan *semantic meaning* keseluruh *content* tersebut. Pemberian *semantic meaning* keseluruh *content* merupakan kegiatan penyusunan suatu Query dengan aturan tertentu, menggunakan rule dalam RDF[6].

Namun menyusun Query dengan memberikan *semantic meaning* keseluruh *content* bukanlah pe-

kerjaan yang mudah. Pengguna harus mempelajari sintaks model penyusunan dari Query tersebut, kemudian menyusun sintaks tersebut menggunakan *rule-rule* penyusunan sintaks dari RDF (misal: `Resource`, `Class`, `Property`, `Literal`, `type`, `subClassOf`, `subPropertyOf`, `range`, dan `domain`) 10.

Hingga saat ini belum ada *tools* berbasiskan notasi grafis untuk memodelkan data model dalam penyusunan sintaks dari RDF. Beberapa *tools* yang ada masih berbasiskan teks, misalnya *protege* dan RDF Editor [2,5]. Karena itu pada penelitian ini dibuat *rules* yang digunakan dalam proses transformasi dari dokumen RDF hingga menjadi grafik RDF, juga digunakan dalam proses validasi dari data model yang dibuat oleh perancang, sehingga data model tersebut dapat memberikan *semantic meaning* pada suatu *content web* secara visual (grafis). Dengan menggunakan rules ini, dapat dibuat suatu *Case Tool* yang berbasiskan grafis yang dapat membantu perancang dalam pemodelan data.

PERUMUSAN MASALAH

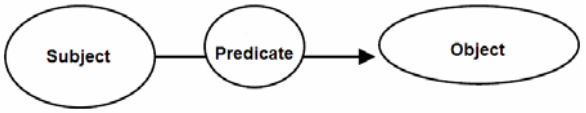
Permasalahan dari penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan parsing dari dokumen RDF berbasiskan text ini (berformat XML), supaya dapat digunakan untuk membuat data model berbasiskan grafis RDF.
2. Bagaimana melakukan query terhadap data model tersebut, sehingga dapat memperoleh desain data model yang sesuai dengan referensi pada paper 2
3. Bagaimana cara mengisikan atribut dari masing-masing *object* data model, dari dokumen RDF yang berbasiskan text.
4. Bagaimana membuat relasi antar *object* data model, yang sesuai dengan rekomendasi dari W3C (*W3C Recommendation*).

METODE PENELITIAN

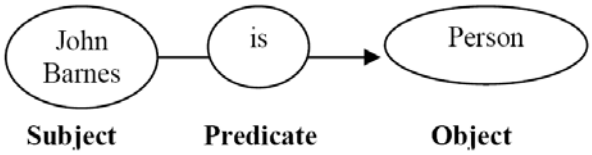
Memparsing Dokumen RDF

Dokumen RDF merupakan dokumen yang berbasiskan text, dengan menggunakan format XML (*eXtensible Markup Language*). RDF memiliki beberapa istilah untuk menjelaskan bagian-bagian dari kalimat antara lain: *subject* – bagian dari kalimat yang menjelaskan tentang sesuatu yang menjadi pusat kalimat; *predicate* – bagian dari kalimat yang menjelaskan *property* atau karakteristik dari *subject* yang dibicarakan yang berisikan relationship resource; dan *object* – bagian dari kalimat yang menjelaskan *value* dari *property*. Sehingga jika digambarkan dalam bentuk RDF Graph 9:



Gambar 1. Graph Data Model

Dimana: *Subject* mempunyai suatu **nilai** (*object*) untuk suatu sebutan (*predicate*) tertentu. Contoh dengan single RDF: John Barnes is *Person*. Maka dalam RDF graph 2:



Gambar 2. RDF Graph

Parser melakukan pemisahan bagian-bagian kalimat dari suatu dokumen RDF, dimana dalam dukumen RDF direpresentasikan dalam bentuk object *class, property, datatype,* dan *literal*. Object *class, property, datatype,* memiliki beberapa atribut yaitu, *label, about, comment,* dan *isDefinedBy*.

Label, berisikan fragmen dari suatu URI *reference* (URIref) yang biasanya dipisahkan oleh karakter "#", misalnya : http://www.w3.org/1999/02/22-rdf-syntax-ns#nil, maka labelnya adalah : nil.

About, berisikan URIref secara lengkap yaitu URI beserta fragmennya,yaitu http://www.w3.org/1999/02/22-rdf-syntax-ns#nil.

isDefinedBy, berisikan URIref tanpa label, dibatasi oleh karakter "#",yaitu http://www.w3.org/1999/02/ 22-rdf-syntax-ns#.[4]

Penyusunan RDF Rules12 :

RDF Rules merupakan rules standar yang digunakan dalam dokumen RDF, dimana dalam penyusunan dokumen RDF adalah berdasarkan pada rules RDF ini. Berikut ini RDF rules-nya :

1. *Resource rule*:
   $\forall r, l, p \in RDF, \exists l \wedge p \notin r$
   Semua *resource* dalam RDF adalah *instance* dari rdfs:Resource, kecuali literal dan property. RDF *resource* ini dinamakan juga *Resource*
2. *Property rule* :
   $\forall p \quad p \in \tau(s, p, o), \exists p \notin r$
   Semua *property* p, yang merupakan bagian dari *triple*, tidak termasuk dalam *Resource*.
3. *Subproperty rule* :
   $\forall s, p_1, o, p_2 \quad \tau(s, p_1, o) \wedge \tau(p_1, rdfs : sub\ \mathrm{Pr}\ operty Of, p_2) \Rightarrow \tau(s, p_2, o)$

$\forall p_1, p_2, p_3 \ \tau(p_1, rdfs : sub\ \text{Pr}\ opertyOf\ , p_2)$
$\wedge \tau(p_2, rdfs : sub\ \text{Pr}\ opertyOf\ , p_3)$
$\Rightarrow \tau(p_1, rdf : sub\ \text{Pr}\ opertyOf\ , p_3)$

Jika suatu *property* p_1 adalah *subproperty* dari *property* p_2 dan *property* p_2 adalah *subproperty* dari *property* p_3, maka *property* p_1 adalah *subproperty* dari *property* p_3

4. *Class rule* :
$\forall c \ c \in RDFS$
$\forall i, c_1, c_2 \ \tau(i, rdf : type, c_1) \wedge \tau(c_1, rdfs : subClassOf, c_2) \Rightarrow \tau(i, rdf : type, c_2)$
Jika i merupakan rdf:type *class* c_1 dan *class* c_1 adalah *subclass* dari *class* c_2, maka i juga merupakan rdf:type *class* c_2
5. *Subclass rule* :
$\forall c_1, c_2, c_3 \ \tau(c_1, rdfs : subClassOf, c_2)$
$\wedge \tau(c_2, rdfs : subClassOf, c_3)$
$\Rightarrow \tau(c_1, rdf : subClassOf, c_3)$
Jika suatu *class* c_1 adalah *subclass* dari *class* c_2 dan *class* c_2 adalah *subclass* dari *class* c_3, maka *class* c_1 adalah *subclass* dari *class* c_3
6. *Domain rule* :
$\forall d, p, s, o \ \tau(s, p, o) \wedge \tau(p, rdfs : domain, d) \Rightarrow \tau(s, rdfs : domain, d)$
Domain rule. Jika suatu *property* p adalah suatu *domain* D dan *subject* s adalah suatu *triple* dengan *property* p, maka *subject* s adalah *domain* dari D
7. *Range rule* :
$\forall p, r_1, r_2 \ \tau(p, rdfs : range, r_1) \wedge r_1 \neq r_2 \Rightarrow \neg\tau(p, rdfs : range, r_2)$
Jika suatu *property* p memiliki *range* R dan o adalah *object* dari suatu *triple* dengan *property* p, maka *object* o adalah *range* R
8. *Literals rule* :
$\forall l \in Strings, \exists l \notin (s \vee p)$
Literals (*constant values*) merupakan sekumpulan karakter *strings*, dengan tujuan untuk merepresentasikan *property values* dan tidak boleh digunakan sebagai *subject* atau *predicate* dalam kalimat RDF
9. *Datatype rule :*
$\forall D \in l, \exists D \notin (s \vee p \vee o)$
Datatype merupakan elemen dari *Literals*, dimana *Datatype* bukan bagian dari *subject*, *predicate* dan *object*. *Datatype* merupakan *subClass* dari *Literals*, untuk merepresentasikan tipe data dari *literal*, apakah termasuk XMLLiteral, Literal (*string*), ataupun Literal (*integer*).

Notasi dalam RDF Graph

Notasi yang digunakan dalam *Case Tool* ini merupakan pengembangan dari referensi paper 2. Gambar 3 adalah notasi visual (grafik) yang digunakan:

Gambar 3. Notasi dalam RDF Graph

Keterangan :

Class : Merupakan *subClass* dari *Resource*, yang memiliki tipe *Class*. Dapat merepresentasikan sebagai *Subject* (merupakan domain dari suatu properti) maupun *Object* (merupakan *range* dari suatu properti) dari suatu kalimat.

Properties : Memiliki type *Property*, dan merupakan predikat dari suatu kalimat, yang menjelaskan *value* (*range*) dari suatu pokok permasalahan (*domain*).

Datatype : Merupakan *subClass* dari *Literal*, yang memiliki atribut label XMLLiteral, Literal (*string* dan *integer*).*subClass*, *subProperties*, *Domain*, *type*, dan *range*: menunjukkan relasi yang terjadi antar *object*, yaitu menurut pada *rules* RDF

Literal : Setiap sesuatu yang bukan termasuk *resource* adalah *instance* dari rdfs:Literal

Proses Transformasi dari Dokumen RDF ke Grafik RDF

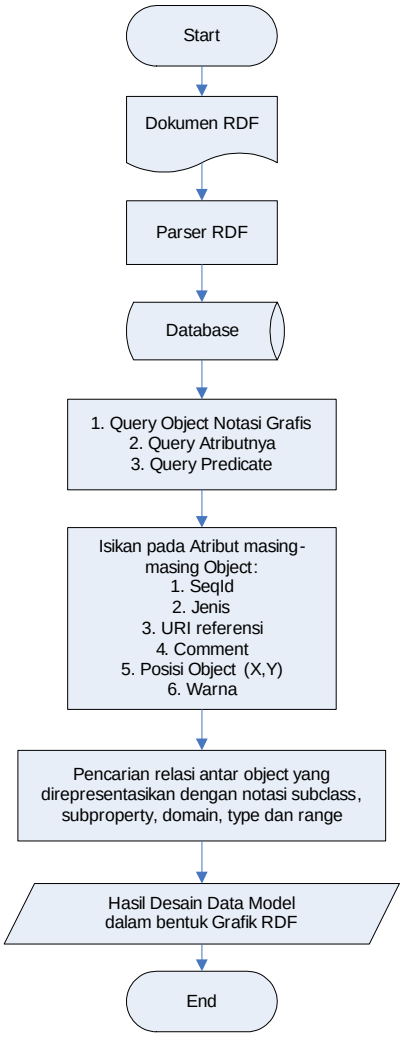
Dimulai dari suatu dokumen RDF berformat XML, yang berbasiskan text. Dokumen ini berisikan data model yang mengartikan sekumpulan data untuk menjadi suatu informasi.

Setelah itu dilakukan parser pada dokumen ini, yaitu dengan memisah-misahkan kata pada masing-masing *object* dari data model, memisah-misahkan kata yang merupakan atribut dari masing-masing *object*, dan memisah-misahkan kata yang merupakan relasi yang terdapat pada masing-masing *object*, untuk kemudian disimpan pada suatu database[7].

Data pada database ini di Query untuk mengambil data *object*, atribut kemudian relasinya. Query untuk mengambil data *object* yang pertamakali dilakukan adalah mengambil object *Class*, kemudian *Properties*, baru kemudian *Datatype*. Hal ini sesuai dengan *rules* dari grafik RDF.

Untuk atribut, dilakukan *query* yang pertama adalah atribut *About*, yang kedua adalah atribut label dan yang terakhir adalah meng-Query atribut

Comment. Setelah itu meng-Query relasi, dalam meng-Query relasi urut-urutannya adalah meng-Query *SubClass*, kemudian meng-Query *Domain* dan yang terakhir meng-Query *Range*.



Gambar 4. Blok Diagram Proses Transformasi Dokumen RDF

Setelah Query dilakukan, hasil dari query ini dibuat *object-object* yang berbasiskan notasi grafis yang berupa *Class*, *Properties* ataupun *Datatype*. Setelah *object-object* yang berbasiskan notasi grafis dibuat, langkah selanjutnya adalah mengisikan atribut masing-masing *object* yang menggunakan struktur data *record pointer* untuk grafis RDF. Dan langkah terakhir adalah membuat notasi-notasi *subClass*, *subProperty*, *Domain*, *type* dan *range*, dimana notasi-notasi ini merupakan notasi relasi dari data model yang dibuat.

Hasil akhir dari proses ini adalah terbentuknya data model yang berbasiskan grafik RDF, lengkap dengan atribut juga relasi antar *object* dari data model tersebut.

Transformation Rules dari Dokumen RDF Ke Grafik RDF

Transformasi rules ini digunakan dalam proses konversi dari dokumen RDF/XML yang berbasiskan text, hingga menjadi suatu data model yang berbasiskan notasi grafis RDF. Rules tersebut adalah :

a. Dokumen yang akan dikonversikan harus merupakan dokumen RDF berformat XML, yang berbasiskan text, dengan memenuhi kriteria yang terdapat pada RDF rules seperti yang disebutkan dalam subbab diatas, yaitu subbab 3.2
b. Pertamakali yang diambil dari data model dalam dokumen RDF tersebut adalah yang bertype *Class*, kemudian *Properties* dan yang terakhir adalah *Datatype*. Ketiga type ini merupakan object pada notasi grafis.
c. Langkah kedua adalah mengambil atribut dari dokumen RDF. Atribut ini akan diisikan pada atribut masing-masing object yang telah terbentuk. Atribut ini adalah atribut *About*, atribut *label* dan atribut *Comment*. Sedangkan atribut *isDefineBy* adalah berisikan URIref tanpa *label*, dibatasi oleh karakter "#", yang diambil dari atribut *About*.
d. Langkah terakhir adalah mengambil relasi dari dokumen RDF. Relasi ini diambil setelah semua *object* dan atribut dari notasi grafis terbentuk. Relasi pada dokumen RDF adalah berupa *subClass*, *subProperty*, *Domain*, *type* dan *range*.

HASIL PENELITIAN

Untuk uji coba digunakan dokumen RDF berbasiskan text yang diambil dari referensi *case tool Protégé* yang berbasiskan text. *Case tool* ini merupakan salah satu *case tool* yang menggunakan *rule* RDF yang direkomendasikan oleh W3C (*W3C Recommendation*).

Dokumen RDF ini menjelaskan suatu data model mengenai kendaraan bermotor, yang dimodelkan sebagai *class* abstrak. Kendaraan bermotor ini merupakan *subClass* dari *resource*, yang memiliki beberapa *subClass* dan merupakan domain dari beberapa *properties*. Untuk lebih jelasnya, berikut ini ditampilkan suatu dokumen RDF yang berbasiskan text (data model mengenai kendaraan bermotor) dan data model RDF berbentuk grafis, yang merupakan hasil transformasi menggunakan *rules* dari penelitian ini.

Sumber Dokumen RDF yang akan ditransformasikan ke dalam bentuk grafik RDF:

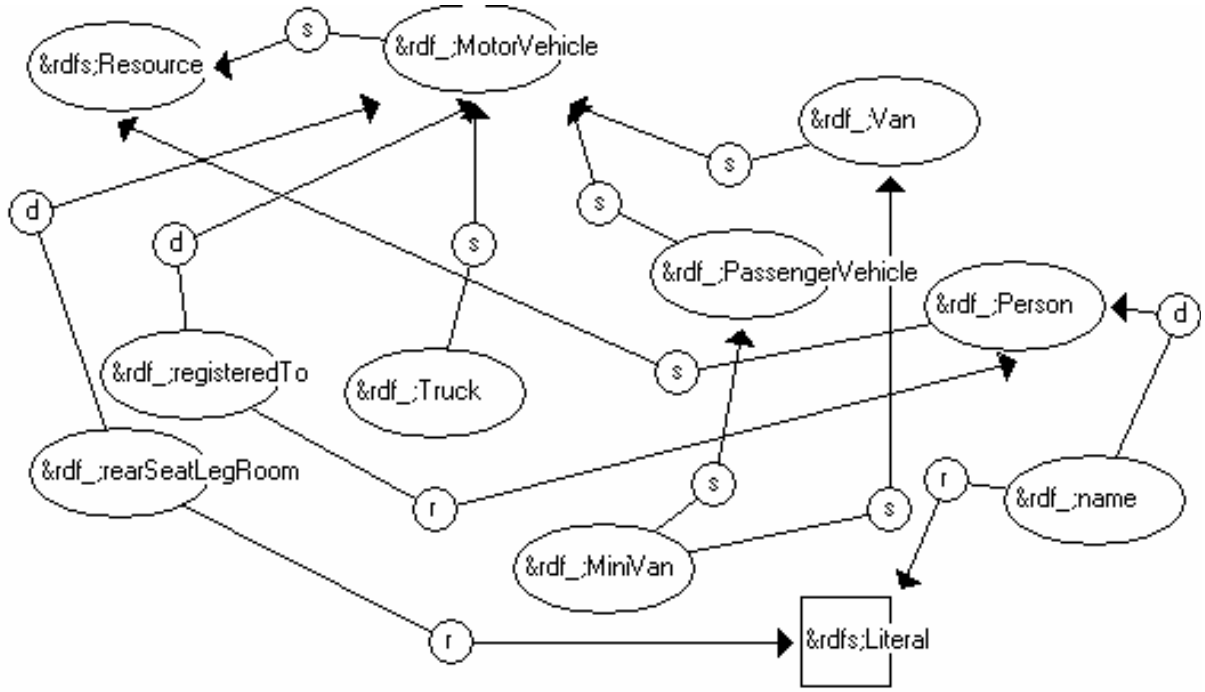
```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
```

```
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-
rdf-syntax-ns#'>
    <!ENTITY rdf_
'http://protege.stanford.edu/rdf'>
    <!ENTITY rdfs
'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:rdf_="&rdf_;"
    xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&rdf_;MiniVan"
    rdfs:label="MiniVan">
   <rdfs:subClassOf
rdf:resource="&rdf_;PassengerVehicle"/>
   <rdfs:subClassOf rdf:resource="&rdf_;Van"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;MotorVehicle"
    rdfs:label="MotorVehicle">
   <rdfs:subClassOf
rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;PassengerVehicle"
    rdfs:label="PassengerVehicle">
   <rdfs:subClassOf
rdf:resource="&rdf_;MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;Person"
    rdfs:label="Person">
   <rdfs:subClassOf
rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;Truck"
    rdfs:label="Truck">
   <rdfs:subClassOf
rdf:resource="&rdf_;MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdf_;Van"
    rdfs:label="Van">
   <rdfs:subClassOf
rdf:resource="&rdf_;MotorVehicle"/>
</rdfs:Class>
<rdf:Property rdf:about="&rdf_;name"
    rdfs:label="name">
   <rdfs:domain rdf:resource="&rdf_;Person"/>
   <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property
rdf:about="&rdf_;rearSeatLegRoom"
    rdfs:label="rearSeatLegRoom">
   <rdfs:domain
rdf:resource="&rdf_;MotorVehicle"/>
   <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&rdf_;registeredTo"
    rdfs:label="registeredTo">
   <rdfs:domain
rdf:resource="&rdf_;MotorVehicle"/>
   <rdfs:range rdf:resource="&rdf_;Person"/>
</rdf:Property>
</rdf:RDF>
```

Dari dokumen RDF diatas terdapat beberapa langkah yang dilakukan untuk proses transformasi kedalam bentuk data model berbasiskan grafis RDF, yaitu :

a. Dilakukan parsing terhadap dokumen RDF, dengan melakukan pemisahan kata berdasarkan pada format XML (yang dimodelkan dalam bentuk *tree*).
b. Mengambil *object* data model yang bertype *Class* yaitu: *MiniVan, MotorVehicle, PassengerVehicle, Person*, *Truck,* dan *Van.*
c. Mengambil *object* data model yang bertype *Properties,* yaitu: *Name, rearSeatLegRoom,* dan *registeredTo*
d. Mengambil *object* data model yang bertype *Datatype,* namun ternyata tidak ada *object* yang bertype *Datatype* ini.
e. Langkah berikutnya mengambil atribut dari dokumen RDF. Atribut ini akan diisikan pada atribut masing-masing *object* yang telah terbentuk. Atribut ini adalah atribut *About,* misalnya rdf:about="&rdf_;MiniVan", kemudian atribut *label,* misalnya, rdfs:label="MiniVan".
f. Langkah terakhir adalah mengambil relasi dari dokumen RDF. Relasi ini diambil setelah semua *object* dan atribut dari notasi grafis terbentuk. Relasi pada dokumen RDF misalnya : rdfs:subClassOf rdf:resource="&rdf_;PassengerVehicle", dan rdfs:subClassOf rdf:resource="&rdf_;Van", yang terdapat pada *object* data model MiniVan. Sehingga *object* data model MiniVan memiliki arti, *object* data model MiniVan (bertype *Class*) merupakan *subClass* dari *Class* &rdf_;PassengerVehicle dan *Class* &rdf_;Van.

Setelah melalui proses transformasi tersebut, maka dapat dihasilkan data model dalam RDF graph seperti berikut ini:



Gambar 4. Hasil Data Model dari Dokumen RDF MotorVehicle

KESIMPULAN

Dokumen RDF merupakan dokumen berbasiskan text yang berformat XML, yang sesuai dengan rules RDF. RDF (*Resource Description Framework*) adalah bahasa yang digunakan untuk merepresentasikan informasi tentang suatu *resource* dan untuk mempertukarkannya di Web. RDF merupakan bahasa yang digunakan untuk memodelkan informasi dari sekumpulan data. Sedangkan data model merupakan suatu tata bahasa yang digunakan untuk mengartikan sekumpulan data untuk menjadi suatu informasi.

Untuk mentransformasikan suatu Dokumen RDF yang berbasiskan text, menjadi data model berbasiskan grafis RDF, diperlukan suatu *rules* yang dapat mengakomodasikan proses transformasi tersebut. Dimulai dengan memisahkan dokumen RDF tersebut perkata, kemudian diambil secara berurutan yang merupakan bagian *object* dari notasi grafis, lalu mengambil atribut dari *object* tersebut, dan yang terakhir mengambil dan mencari relasi dari masing-masing *object* yang ada.

Hasil dari uji coba ini adalah suatu data model RDF berbasiskan grafis, dengan mengambil data berupa dokumen RDF yang sesuai dengan rekomendasi dari W3C (*W3C Recommendation*). Dokumen RDF tersebut ditransformasikan ke dalam bentuk grafis melalui beberapa tahapan yang sudah dijelaskan sebelumnya. Data model dalam bentuk grafis ini, diharapkan lebih memudahkan user dalam memodelkan suatu data model dalam *Resource Description Framework*.

DAFTAR PUSTAKA

1 Balani, N., 2005. *The future of the Web is Semantic*, IBM developerWorks, Oktober 2005.

2 Bernstein O., 2002. *RDF Editor*, CS Department of Rensselaer Polytechnic Institute, 2002.

3 Champin A. P., 2001. *RDF Tutorial*, recommended by the World Wide Web Consortium (W3C), 2001.

4 Muslimin, A., Wibisono, W., dan Siahaan, D. O., 2006. *Semantic Web Constructin For Image Retrieval System From Internet Sport News*, Informatics Department, Faculty of Information Technology, Sepuluh Nopember Institute of Technology, Surabaya, 2006

5 Protégé, *http://protege.stanford.edu*, Stanford University, 2006.

6 Siahaan, D. O., 2006. *Graphical Notations For Semantic Web Language*, Prosiding ICTS 2006, Informatics Department, Faculty of Information Technology, Sepuluh Nopember Institute of Technology, Surabaya, 2006.

7 Berners-Lee, T. 1998. *Relational Databases on the Semantic Web*.

8 Berners-Lee, T. 1998. *What the Semantic Web can Represent*.

9 W3C Recommendation, *RDF Primer*, 2004.

10 W3C Recommendation, *RDF Vocabulary Description Language 1.0: RDF Schema*, 2004.