
IMPLEMENTASI METODA PENYIMPANAN DAN PENGAMBILAN ONTOLOGI BERBASIS FILE MENGGUNAKAN JAVA DAN JENA

Mohammad Yani

Politeknik Negeri Indramayu

Jl. Raya Lohbener Lama No. 08, Lohbener, Indramayu, 45252, Telpon (0234) 5746464

e-mail: myani0703@gmail.com / mohammad.yani@polindra.ac.id

Abstrak

Perkembangan teknologi web semakin pesat baik teknologi web berbasis klien ataupun server. Perkembangan tersebut tidak lain bertujuan untuk meningkatkan mutu informasi yang dikelolanya. Web semantik adalah salah satu solusi untuk meningkatkan mutu informasi dari sebuah web. Karena dengan konsep ini informasi disajikan dalam bentuk struktur orisinal sehingga informasi dapat dipahami oleh manusia dan mesin. Yang dimaksud dengan informasi dapat dipahami oleh manusia dan mesin adalah bahwa keduanya dapat memahami arti semantik dari informasi yang dikandung di dalamnya.

Pada penelitian yang dilakukan oleh Yinghui Huang dijelaskan bahwa metoda penyimpanan dibagi menjadi tiga, yaitu: metoda berbasis memori; metoda berbasis file; dan metoda berbasis relasi basis data. Namun pada penelitian tersebut hanya menitikberatkan pada definisi, model dan metoda penyimpanan saja. Pada penelitian ini peneliti akan mengimplementasikan bagaimana melakukan proses penyimpanan dan pengambilan ontologi menggunakan metoda berbasis file. Penelitian ini dibagi menjadi tiga tahap yaitu: perancangan ontologi; proses render, dan penyimpanan dan pengambilan Triple; dan evaluasi. Luaran yang diharapkan dari penelitian ini adalah menghasilkan cara atau metoda untuk mengimplementasikan proses penyimpanan dan pengambilan ontologi berbasis file menggunakan Java dan Jena.

Kata kunci: File, Jena, ontologi, penyimpanan, query, semantik, triple, web

Abstract

Development of web technology is increasing rapidly for both client and server technology. This development is to provide and increase the quality of information. The semantic web is one of the many solutions to improve the quality of information of a web. It means that both of human and machine can understand the meaning of semantic of informations are contained in it.

According to the Yinghui Huang stated that there are three kinds of ontology storage methods: memory-based method; file-base method; and relational database-based method. But, the topic of that paper is just emphasize to definition, model, and storage method. In this research, researcher will implement how to store and retrieve an ontology by using file-based method. This study is divided into three phases: ontology design; rendering process, and storing and retrieving the Triple; and evaluation. Expected outcomes of this research is to get a way or method to implement ontology storing and retrieving by using Java and Jena.

Keywords: File, Jena, ontology, storing, query, semantic, triple, web

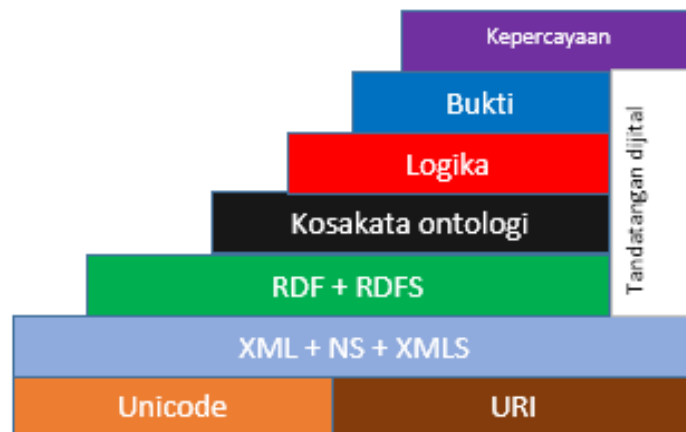
1. Pendahuluan

Perkembangan teknologi web semakin pesat baik teknologi web berbasis klien ataupun server. Perkembangan tersebut tidak lain bertujuan untuk memberikan kepuasan mutu informasi yang dikelolanya. Sayangnya tidak sedikit halaman-halaman web yang ada masih menggunakan konsep pengelolaan informasi yang lama atau konvensional dalam pengelolaan isinya. Web tersebut hanya melayani kebutuhan informasi yang diperlukan dan dapat dipahami oleh manusia saja, akan tetapi belum mampu untuk melayani kebutuhan informasi yang diperlukan dan dapat dipahami oleh mesin. Saat ini

sedang hangat dibicarakan tentang bagaimana mengelola informasi dalam web yang kita miliki kemudian dapat disajikan dalam bentuk informasi yang dapat dipahami oleh manusia dan mesin. Istilah tersebut adalah yang disebut dengan konsep web berbasis semantik. Pada penelitian yang dilakukan oleh Yinghui Huang[6] dijelaskan bahwa metoda penyimpanan dibagi menjadi tiga, yaitu: metoda berbasis memori; metoda berbasis file; dan metoda berbasis relasi basis data. Namun pada penelitian tersebut hanya menitikberatkan pada definisi, model dan metoda penyimpanan saja. Pada penelitian ini peneliti akan mengimplementasikan bagaimana melakukan proses penyimpanan dan pengambilan ontologi menggunakan metoda berbasis file. Penelitian ini dibagi menjadi tiga tahap yaitu: perancangan ontologi; penyimpanan Triple dan pengambilan Triple; dan evaluasi. Luaran yang diharapkan dari penelitian ini adalah menghasilkan cara atau metoda untuk mengimplementasikan proses penyimpanan dan pengambilan ontologi berbasis file menggunakan Java dan Jena.

1.1. Web Semantik

Web semantik adalah sebuah konsep dalam teknologi web yang merujuk pada kemampuan komputer untuk lebih memahami bahasa manusia, sehingga sebuah web dapat saling berbagi informasi dengan web lain [2].



Gambar 1. Arsitektur web semantik [9]

1.2. Ontologi, RDF, dan RDF Scheme

Dalam web semantik, ontologi adalah sebuah spesifikasi dan format eksplisit dari sebuah konsep perancangan instan dan propertinya sesuai dengan relasi logika yang digunakannya [3]. Pada dasarnya Teknik ontologi menekankan pada membangun sebuah ontologi berdasarkan standar dan norma tertentu [8]. RDF adalah sebuah model data yang direpresentasikan dalam bentuk Triple yang disebut dengan pernyataan atau statement. RDF juga disebut sebagai bangunan blok dari web semantik []. Sedangkan RDF Scheme (RDFS) adalah merupakan skema dari RDF yang digunakan untuk mendefinisikan sebuah terminologi [3].

Table 1. Perbandingan basis data dan komponen ontologi [5]

Basis Data Relasi	Ontologi
Tabel Relasi	Kelas Properti Obyek
Kolom Tuple	Properti Data Individu

1.3. Query SPARQL

SPARQL adalah merupakan bahasa query yang digunakan untuk melakukan pemrosesan Triple pada database TDBJena [9]. Tidak seperti bahasa query lainnya, SPARQL tidak didesain untuk hubungan relasi data akan tetapi untuk query dalam bentuk model RDF [1].

1.4. Jena

Jena adalah *framework opensource* web semantik untuk Java. Pada Jena ada yang disebut dengan TDBJena, yaitu basis data Jena yang digunakan untuk menyimpan file RDF dan query. TDBJena pada prinsipnya menyimpan *Triple* (Subyek, Predikat, dan Obyek) [7].

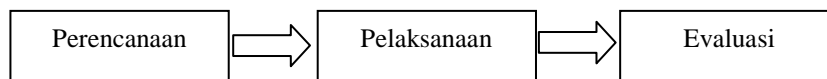
1.5. Protégé

Perangkat lunak ini adalah suatu aplikasi alat bantu berbasis IDE (Integrated Development Environment) yang digunakan untuk membuat atau merancang ontologi dari sebuah rancangan web berbasis semantik. Perangkat lunak ini dilengkapi dengan antar muka yang sederhana, sehingga dapat dengan mudah dipahami dan digunakan.

2. Metoda Penelitian

Penelitian ini dilakukan dalam tiga tahap, yaitu:

1. Perencanaan
Perencanaan dalam penelitian ini meliputi perancangan desain ontologi, dalam hal ini mengambil contoh kasus sistem informasi akademik Politeknik Negeri Indramayu pada modul daftar pengampu mata kuliah [4].
2. Pelaksanaan
Tahapan berikutnya adalah proses *render* ontologi, yang dilanjutkan dengan pembuatan fungsi-fungsi untuk menyimpan dan mengambil file dari TDBJena menggunakan Java. Dan diakhiri dengan proses pengujian fungsi-fungsi tersebut menggunakan metoda *Black Box*.
3. Evaluasi
Tahapan ini adalah untuk mengevaluasi peubah yang akan digunakan dalam analisis dan menarik kesimpulan.



Gambar 2. Tahapan penelitian

2.1. Perencanaan

Pada tahapan ini dilakukan proses perancangan ontologi yang meliputi: perancangan kelas; perancangan slots (slot adalah rincian properti dari kelas); dan perancangan instan/individu. Proses perancangan ini menggunakan perangkat lunak Protégé.

a. Perancangan Kelas

Kelas yang dirancang dalam penelitian ini adalah meliputi kelas utama **Kurikulum**, dan sub-kelas **Dosen**, dan **MataKuliah**.

b. Perancangan Slots

Perancangan slots ini digunakan untuk menjelaskan rincian properti dan hubungan/relasi antar kelas, sehingga dapat diketahui mana yang menjadi subyek, predikat, dan obyeknya. Pada penelitian ini, sampel yang digunakan pada perancangan slots ini adalah pada kelas **Dosen**, dan **MataKuliah**.

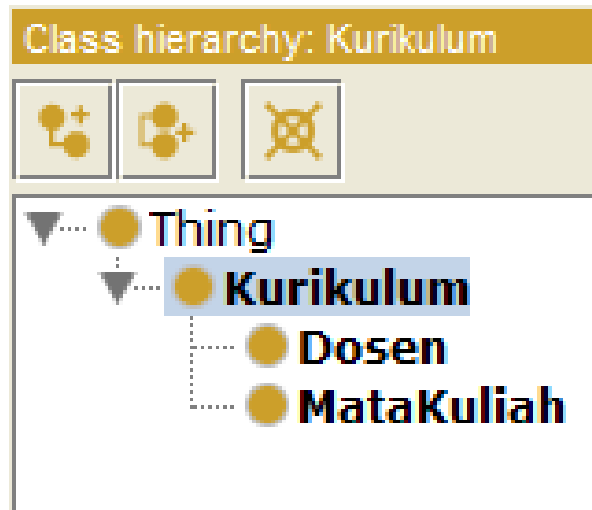
Table 2. Rancangan *slots* kelas **Dosen**

Properti	Batas	Nilai yang diizinkan	Tipe
NIDN	String	Instan	Properti Data
Nama	String	Instan	Properti Data
BidangKeahlian	String	Instan	Properti Data
Mengampu	Multiple	Instan	Properti Obyek

MataKuliah

Table 3. Rancangan *slots* kelas **MataKuliah**

Properti	Batas	Nilai yang diizinkan	Tipe
KodeMK	String	Instan	Properti Data
Deskripsi	String	Instan	Properti Data
SKS	Int	Instan	Properti Data
DiampuOleh	Multiple Dosen	Instan	Properti Obyek



Gambar 3. Hirarki kelas

c. Perancangan Instan/Individu

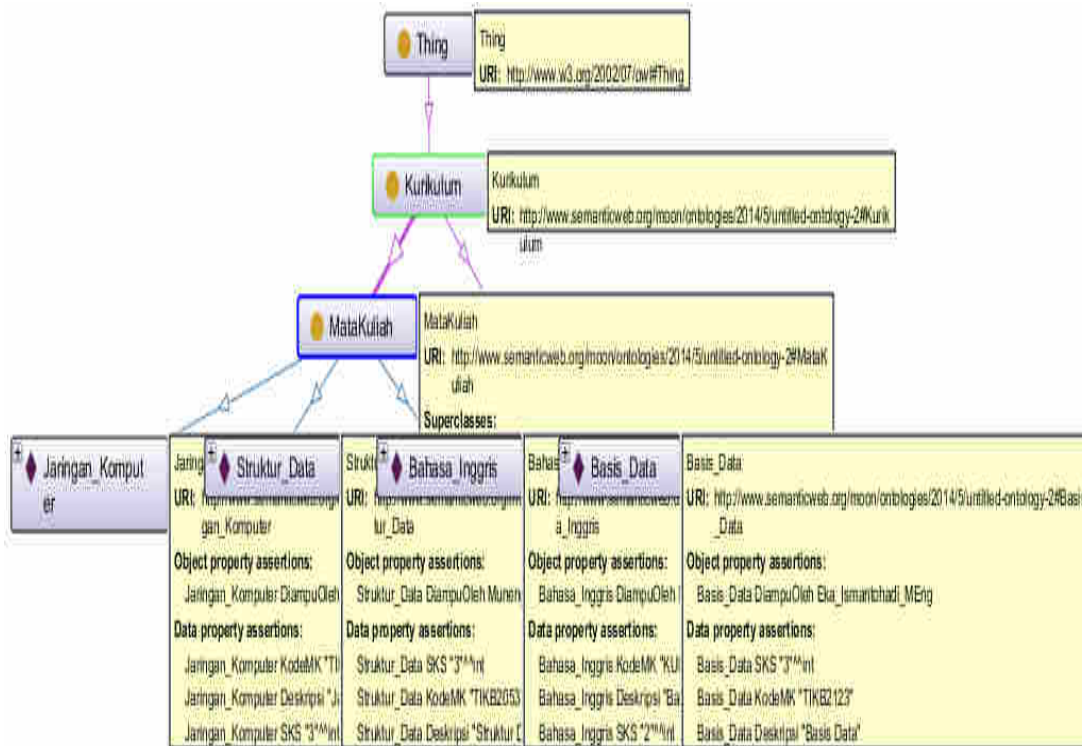
Perancangan instan/individu ini adalah untuk mengimplementasikan hubungan antar kelas yang ada pada Slots ke dalam instan/individu yang akan dibuat. Instan/individu ini bisa juga disebut dengan kerangka konten data yang akan disajikan dalam sistem kita. Pada penelitian ini perancangan dilakukan pada instan /individu **MataKuliah** dan **Dosen**.

Table 4. Rancangan instan/individu **MataKuliah**

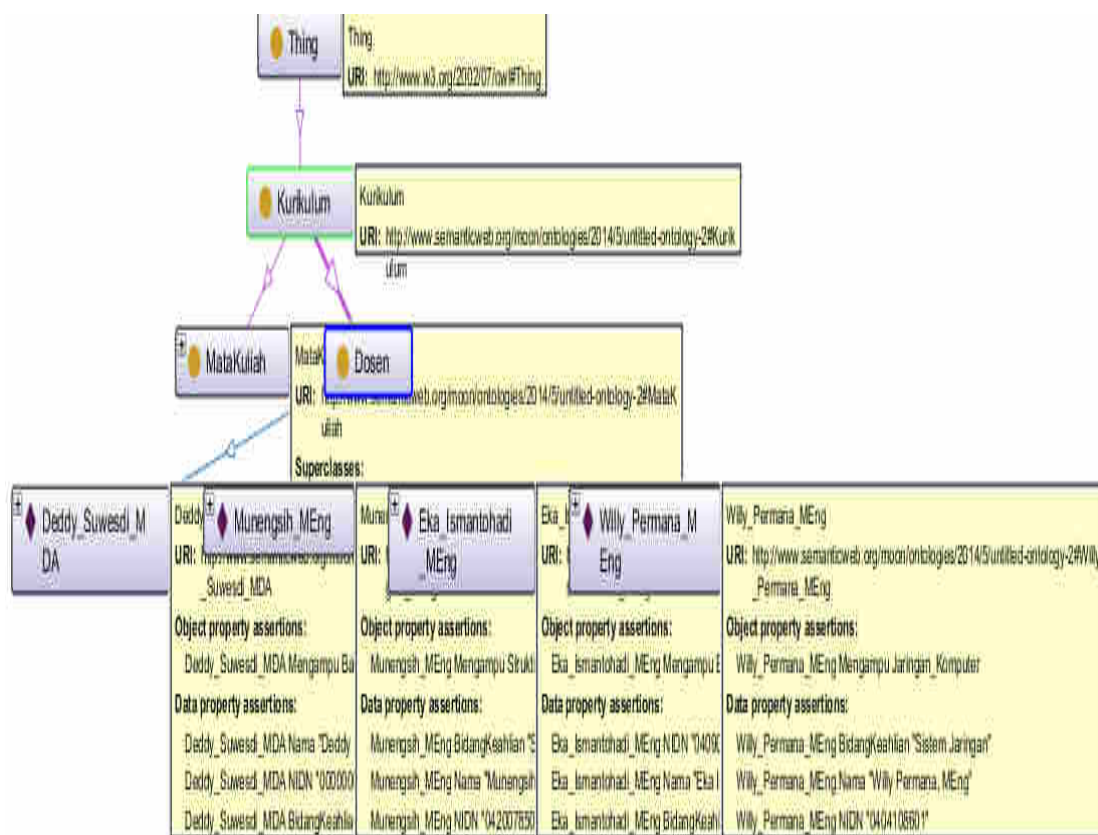
Tipe	Subyek	Predikat	Obyek
MataKuliah	Bahasa_Ingggris	DiampuOleh	Deddy_Suwesdy_MDA
MataKuliah	Basis_Data	DiampuOleh	Eka_Ismantohadi_Meng
MataKuliah	Jaringan_Komputer	DiampuOleh	Willy_Permana_Meng
MataKuliah	Struktur_Data	DiampuOleh	Munengsih_Meng

Table 5. Rancangan instan/individu **Dosen**

Tipe	Subyek	Predikat	Obyek
Dosen	Deddy_Suwesdy_MDA	Mengampu	Bahasa_Ingggris
Dosen	Eka_Ismantohadi_Meng	Mengampu	Basis_Data
Dosen	Willy_Permana_Meng	Mengampu	Jaringan_Komputer
Dosen	Munengsih_Meng	Mengampu	Struktur_Data



Gambar 4. Hirarki rancangan instan/individu **MataKuliah**



Gambar 5. Hirarki rancangan instan/individu **Dosen**

2.2. Pelaksanaan

Pada tahap ini pertama-tama rancangan ontologi akan di-*render*. Sebagai bahan *trial and error*, maka ontologi di-*render* menjadi file *.owl dan *.rdf. Masing-masing akan diuji pada tahap uji coba untuk mengetahui file mana yang dapat digunakan pada proses pengambilan data.

Selanjutnya adalah membuat fungsi untuk menyimpan data ontologi dari file *.owl atau *.rdf ke dalam TDBJena. Berikut adalah *pseudo code* fungsi untuk menyimpan data ontologi dari file *.owl atau *.rdf.

```
public void storeToTDB(){
define_a_directory_1 = "D:\\your\\path\\of\\your\\storage";
create dataset;
create model;
define_a_directory_2 = "D:\\your\\path\\of\\your\\source\\owl\\or\\rdf";
read model;
close tdb;
close dataset;
}
```

Gambar 6. *Pseudo code* menyimpan data dari file ke TDBJena

Dengan fungsi di atas kedua file *.owl dan *.rdf dapat disimpan dalam TDBJena dengan daftar file seperti Tabel 6.

Table 6. Daftar file yang tersimpan dalam TDBJena

Nama file	File ekstensi
GOSP	dat
GOSP	idn
GPOS	dat
GPOS	idn
GSPO	dat
GSPO	idn
journal	jrnI
node2id	dat
node2id	idn
nodes	dat
OSP	dat
OSP	idn
OSPG	dat
OSPG	idn
POS	dat
POS	idn
POSG	dat
POSG	idn
prefix2id	dat
prefix2id	idn
prefixes	dat
prefixIdx	dat
prefixIdx	idn
SPO	dat
SPO	idn
SPOG	dat
SPOG	idn

Dari hasil percobaan penyimpanan file ontologi ke TDBJena, secara struktur file dapat disimpan dengan baik ke TDBJena, namun secara struktur data belum dapat diketahui apakah nanti dapat diproses pada proses pengambilan data atau tidak. Hasilnya akan diketahui setelah proses percobaan pengambilan data dilakukan.

Tahap selanjutnya adalah membuat fungsi untuk mengambil dan menampilkan data ontologi dari file *.owl dan *.rdf. Untuk proses tersebut digunakan query SPARQL yang dapat dilihat pada Gambar 7.

```
"SELECT ?KodeMK ?Deskripsi ?SKS ?DiampuOleh "+
    "WHERE {"+
    "?lib a:KodeMK ?KodeMK ."+
    "?lib a:Deskripsi ?Deskripsi ."+
    "?lib a:SKS ?SKS ."+
    "?lib a:DiampuOleh ?DiampuOleh ."+
    "?lib a:DiampuOleh ?Dosen }";
```

Gambar 7. Query SPARQL menampilkan data dari file ontologi

Table 7. Tabel percobaan pengambilan data ontologi

File sumber	Keluaran	Keterangan
*.owl	<i>Error</i>	<i>Unresolved prefixed name</i>
*.rdf	<i>Error</i>	<i>Unresolved prefixed name</i>

Dari hasil percobaan di atas proses pengambilan data dari kedua file terjadi *error*. Jika dilihat dari pesan *error*-nya maka kesalahan tersebut disebabkan oleh tidak adanya *prefix* pada query SPARQL di atas. Maka query pada Gambar 7 diperbaiki dengan menambahkan *prefix* menjadi:

```
"PREFIX a: <http://www.yani.com/a#>"+
"SELECT ?KodeMK ?Deskripsi ?SKS ?DiampuOleh "+
    "WHERE {"+
    "?lib a:KodeMK ?KodeMK ."+
    "?lib a:Deskripsi ?Deskripsi ."+
    "?lib a:SKS ?SKS ."+
    "?lib a:DiampuOleh ?DiampuOleh ."+
    "?lib a:DiampuOleh ?Dosen }";
```

Gambar 8. Query SPARQL menampilkan data dari file ontologi menggunakan *Prefix*

Table 8. Tabel percobaan pengambilan data ontologi dengan *prefix*

File sumber	Keluaran	Keterangan
*.owl	Tidak ada <i>Error</i>	Data tidak tampil
*.rdf	Tidak ada <i>Error</i>	Data tidak tampil

Percobaan penambahan *prefix* pada query SPARQL dapat menghilangkan pesan kesalahan pada saat *run time*, akan tetapi data masih belum dapat diambil atau ditampilkan dengan baik. Ini dimungkinkan pendefinisian URI (*Universal Resource Identifier*) yang tidak tepat. URI yaitu semacam identitas unik yang tidak harus dapat menghubungkan atau mengakses sumber daya (*resource*).

Untuk mengetahui apakah URI harus sesuai atau sama dengan yang telah didefinisikan pada rancangan ontologi, maka URI pada *prefix* disesuaikan dengan apa yang telah didefinisikan pada file *.owl dan *.rdf. Sehingga modifikasi query SPAQRL adalah sebagai berikut:

```
"PREFIX a: <http://www.semanticweb.org/moon/ontologies/2014/5/untitled-ontology-2#>"
"SELECT ?KodeMK ?Deskripsi ?SKS ?DiampuOleh "+
    "WHERE {"+
    "?lib a:KodeMK ?KodeMK ."+
    "?lib a:Deskripsi ?Deskripsi ."+
    "?lib a:SKS ?SKS ."+
    "?lib a:DiampuOleh ?DiampuOleh ."+
    "?lib a:DiampuOleh ?Dosen }";
```

Gambar 9. Query SPARQL dengan URI *prefix* sesuai dengan file ontologi

Table 9. Tabel percobaan pengambilan data ontologi dengan URI *prefix* disesuaikan

File sumber	Keluaran	Keterangan
*.owl	Tidak ada <i>Error</i>	Data tidak tampil
*.rdf	Tidak ada <i>Error</i>	Data tampil

Pada percobaan ini ternyata file sumber *.rdf dapat menampilkan keluaran Triple yang tersimpan dalam TDBJena sedangkan file sumber *.owl tidak. Berikut adalah keluaran dari *.owl dan *.rdf.

```
run:
log4j:WARN No appenders could be found for logger (org.apache.jena.info).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2faq.html#noconfig for more info.
-----
| KodeMK | Deskripsi | SKS | DiampuOleh |
-----
NOTED SUCCESSFUL (total time: 1 second)
```

Gambar 10. Keluaran Triple dari file *.owl

2.3. Evaluasi

Dari proses perancangan dan pelaksanaan di atas ada dua hal yang perlu dievaluasi untuk memperlancar jalannya uji coba pada kasus sejenis, yaitu sebagai berikut:

- Pembuatan instan tidak terlalu kompleks sehingga kesalahan akibat relasi yang tidak benar akan dapat diminimalisir;
- Hindari penggunaan karakter yang tidak lazim pada data instan untuk menghindari hasil keluaran yang tidak muncul sehingga akan membuat ambigu apakah keluaran tersebut diakibatkan oleh kesalahan program atau kesalahan akibat dukungan karakter pada komputer yang digunakan.

3. Hasil dan Analisis

Hasil yang diperoleh pada uji coba di atas adalah sebagai berikut:

- File sumber yang dapat disimpan dan ditampilkan dengan baik adalah file sumber dengan jenis *.rdf karena dalam file ini Triple didefinisikan secara lengkap dimulai dari kelas, instan, properti data, dan properti obyek;
- Penyesuaian URI *prefix* yang tepat akan menghasilkan keluaran yang tepat pula. Pada percobaan di atas diketahui bahwa URI *prefix* pada query SPARQL harus disesuaikan dengan URI *prefix* yang ada pada file sumber *.rdf

KodeMK	Deskripsi	SKS	Diampu Oleh
"TIKK1054"	"Algoritma dan Pemrograman"	4	<http://.../Moh_Yani_MSc>
"KUPR1012"	"Bahasa Inggris"	2	<http://.../Daddy_Suweadi_MDA>
"TIKB2123"	"Basis Data"	3	<http://.../Eka_TamanLohadi_MFng>
"TTKR2083"	"Jaringan Komputer"	3	<http://.../Willy_Permana_MAng>
"TIKK2072"	"Matematika Diskrit"	2	<http://.../Nadi_Masya_MPMel>
"TTKR2102"	"Melode Numerik"	2	<http://.../Kasyid_MA>
"TIKB1043"	"Sistem Digital"	3	<http://.../Bunarudin_MSc>
"TTKR2083"	"Struktur Data"	3	<http://.../Munengsih_MAng>

Gambar 11. Keluaran Triple dari file *.rdf

4. Kesimpulan

Penelitian ini menghasilkan sebuah pengetahuan tentang cara atau metoda untuk mengimplementasikan proses penyimpanan dan pengambilan ontologi berbasis file menggunakan Java dan Jena. Di mana poin pentingnya adalah bahwa pendefinisian Triple yang lengkap dapat mempengaruhi proses penyimpanan dan pengambilan data. Selain itu penulisan URI *prefix* yang tepat dapat menghasilkan keluaran yang sesuai dengan kebutuhan.

Referensi:

- [1] Bob DuCharme . Learning SPARQL. O'reilly. 2011
- [2] Evis Opari and Marenglen Biba. *A Wiki System for Culture Heritage Data Management with Semantic Content*. Conference of the International Journal of Arts & Sciences. ISSN: 1943-6114. p. 385-400. 2013
- [3] Grigoris Antoniou and Frank van Harmelen . *A Semantic Web Primer*. The MIT Press. 2004
- [4] Mohammad Yani, Eka Ismantohadi. *Implementasi Web Semantik pada Aplikasi Pencarian Data Pengampu Mata Kuliah di Politeknik Indramayu*. Prosiding IRWNS Politeknik Negeri Bandung. 2014
- [5] Saeed M. Sedighi and Reza Javidan . *Semantic Query in Relational Database using a Local Ontology Construction*. *AOSI Open Journals* v108i11/12.2012
- [6] Yinghui Huang, Guanyu Li, and Xingsi Wang. *Semantic Web Roughly Ontology: Definition, Model and Storage Method*. 6th International Conference on Information Management, Innovation Management and Industrial Engineering. 2013
- [7] Yu, L. *A Developer's Guide to the Semantic Web*. Springer. 2011
- [8] Ziyu liu, Xiaoming Zhang, and Xuehui Li . *An Approach of Ontology Construction and Semantic*

Query Expansion for High-speed Railway Domain Knowledge. *Journal of Applied Sciences* 13 (20).
2013

[9] <http://www.w3.org>. Diakses pada 11 Desember 2013 pukul 14.10