



**seminar nasional
informatika 2017**



PROSIDING



**"e-Defense : Menjaga keamanan data
menghadapi cyber warfare untuk memperkuat
kedaulatan Negara Kesatuan Republik Indonesia"**



eDefense
seminar nasional informatika 2017



ISSN 1979-2328

Yogyakarta, 25 November 2017

SUSUNAN PANITIA

Penanggung Jawab : Dekan Fakultas Teknik Industri
Pengarah : 1. Wakil Dekan I FTI
2. Wakil Dekan II FTI
Ketua Umum : Ketua Program Studi Teknik Informatika
Wakil Ketua Umum : Sekretaris Program Studi Teknik Informatika
Ketua Pelaksana : Frans Richard Kodong, S.T., M.Kom.

Reviewer :

Assoc. Prof. Dr. Anton Satria Prabuwo, KSU
Dr. Tech. Ahmad Azhari UGM
Dr. Ir. Lukito Edi Nugroho, MT. UGM
Dr. Ashari SN, UGM
Ir. Balza Ahmad, M.Eng. UGM
Joko Siswantoro, Universitas Surabaya
Dr. Djoko Budianto, Atmajaya Yogyakarta
Dr. Slamet, Universitas Muhammadiyah Malang, Indonesia.
Dr. Abdul Kadir, STMIK Kartika Yani
Nuryono Setyo Widodo, S.T., M.T., Universitas Ahmad Dahlan
Dr. Herlina Jayadianti, S.T., M.T., UPN "Veteran" Yogyakarta
Hafsah, S.T., M.T., UPN "Veteran" Yogyakarta
Hidayatullah Himawan, S.T., M.M., M.Eng., UPN "Veteran" Yogyakarta
Bambang Yuwono, S.T., M.T., UPN "Veteran" Yogyakarta

Komite Pelaksana (Informatika UPN) :

Agus Sasmito Aribowo, S.Kom., M.Cs
Budi Santosa, S.Si., M.T.
Dessyanto Boedi P, S.T., M.T.
Frans Richard Kodong, S.T., M.Kom
Herry Sofyan, S.T., M.Kom.
Heriyanto, A.Md, S.Kom, M.Cs
Heru Cahya Rustamadji, S.Si., M.T.
Juwairiah, S.Si., M.T.
Mangaras Yanu Florestiyanto, S.T., M.Eng
Nur Heri Cahyana, S.T., M.Kom.
Oliver Samuel Simanjuntak, S.Kom, M.Eng
Paryati, S.T., M.Kom.
Rifki Indra Perwira, S.Kom., M.Eng
Simon Pulung Nugroho, S.T.
Wilis Kaswidjanti, S.Si., M.Kom
Yuli Fauziah, S.T., M.T.
Budi Cahyono
Pri Wahyu Eko Setiawan
Rahayu Ari Orbani.
Sugeng Rahmadi
Sukardi
Himpunan Mahasiswa Teknik Informatika (HIMATIF)

DAFTAR ISI

HALAMAN JUDUL		i
KATA PENGANTAR		iii
SUSUNAN PANITIA		iv
DAFTAR ISI		v
1 SISTEM PAKAR BERBASIS WEB MENGGUNAKAN TEOREMA BAYES (STUDI KASUS PENYAKIT SAAT BANJIR DI CIREBON)	<i>Bambang Yuwono, Hidayatulah Himawan, Adi Yusuf</i>	1
2 SISTEM INFORMASI GEOGRAFIS KOMANDO RAYON MILITER (KORAMIL) DAN KECAMATAN BINAAN KORAMIL DI KOTA YOGYAKARTA	<i>Budi Santosa, Sri Rahayu Astari, Wilis Kaswidjanti</i>	13
3 ANALISIS SISTEM MANAJEMEN KEAMANAN INFORMASI ELECTRONIC SECURITY SYSTEM (ESS) MENGGUNAKAN STANDAR ISO 27001 STUDI KASUS KANTOR PERWAKILAN BANK INDONESIA PROVINSI BALI	<i>I Gede Putu Krisna Juliharta, I Made Maha Primananda Budi, I Gusti Agung Lanang Agung Raditya</i>	19
4 IMPLEMENTASI DAN ANALISA BISNIS RENTAL WEB SYSTEM (SEWALOKA.COM) DENGAN PENDEKATAN SOFTWARE ARCHITECTURAL PATTERN MODEL-VIEW-CONTROLLER	<i>I Putu Satwika, I Made Agus Apriliawan</i>	26
5 REKAYASA SISTEM PENERIMA BEASISWA MISKIN DENGAN METODE C4.5 DAN ELECTRE	<i>Made Henny Aryani, Rukmi Sari Hartati , Ni Wayan Sri Ariyani</i>	37
6 APLIKASI SINGLE ACCOUNT BERBASIS WEB SERVICE MENGGUNAKAN AUTHETICATION LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)	<i>Rifki Indra Perwira, Heru Cahya Rustamaji, Hendra Arya Syaputra</i>	42
7 IMPLEMENTASI MAPPING OTOMATIS DARI DATABASE MYSQL 5.6 KE PROTEGE 4.3 DENGAN TURTLE ONTOLOGY, D2RQ, JENA, DAN NETBEANS 7.4	<i>Widiatminingsih, Herlina jayadianti , Heru cahya Rustamaji</i>	53
8 IMPLEMENTASI SISTEM PENGONTROLAN STOK BAHAN BAKU DAN BARANG JADI PADA GUDANG TEH	<i>Wilis Kaswidjanti, Frans Ricard Kodong, Heru Tricahyono</i>	64
9 KOMPARASI METODE DSS UNTUK MENENTUKAN PRIORITAS PROYEK PEMBANGUNAN DAERAH	<i>Maya Marselia, Fathushahib</i>	70
10 SURVEI PADA PENGGUNAAN TEKNIK DATA MINING PADA BIDANG KESEHATAN DI INDONESIA	<i>Siti Khomsah</i>	82
11 ANALISIS KEAMANAN SISTEM INFORMASI AKADEMIK UIN SUNAN KALIJAGA	<i>Aries Firmansyah, Bambang Sugiantoro</i>	91

- | | | | |
|-----------|--|--|------------|
| 12 | PERANCANGAN MALWARE LOCAL DAN ANTI-MALWARE MEMANFAATKAN SCRIPT BATCH FILE PADA PLATFORM WINDOWS DENGAN METODE FORWARD CHAIN | <i>Frans Richard, Jefri
Hutama Arbi</i> | 100 |
| 13 | REPRESENTASI BUDAYA YOGYAKARTA PADA DESAIN KAOS MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY BERBASIS ANDROID | <i>OliverSamuel
Simanjunt, Hidayatulah
Himawan¹, Reza
Raditya Setyo Putra</i> | 110 |

IMPLEMENTASI MAPPING OTOMATIS DARI DATABASE MYSQL 5.6 KE PROTEGE 4.3 DENGAN TURTLE ONTOLOGY, D2RQ, JENA, DAN NETBEANS 7.4

Widiatminingsih^a, Herlina Jayadianti^a, Heru Cahya Rustamaji^a

^a Universitas Pembangunan Nasional "Veteran" Yogyakarta, Jln SWK 104 Lingkar Utara, Condongcatur Yogyakarta
Indonesia

Abstract

The problem of changing information into the level of knowledge is still being a big problem. Manually mapping process from MySQL to Protege is still running by user/programmers if they want to build web 3.0 generation. This is a waste of time. By using D2RQ Turtle the transfer process from information into the form of knowledge becomes faster.

1. Pendahuluan

Perkembangan teknologi web terus berkembang dari waktu ke waktu. Permasalahan yang muncul adalah migrasi data. Perpindahan data dari web generasi 2.0 ke web generasi 3.0 tidak mudah. Jumlah data yang cukup besar ini sangat mempengaruhi proses migrasi data dimana diketahui bahwa teknologi web semantik RDF/OWL belum terkoneksi dengan MySQL. Inilah yang akan menjadi fokus penelitian dalam penelitian ini yaitu membuat jembatan *interoperability* dari level informasi ke level *knowledge*. Mengatasi permasalahan perpindahan informasi ke level *knowledge* dalam web semantik, yakni cara memindahkan *database* dengan MySQL kemudian dipindahkan dengan *knowledgebase* dengan format *turtle* (.ttl) yaitu dengan menggunakan *generate mapping* sebagai jembatan *interoperability* yang terdapat pada teknologi D2RQ *turtle*. Jembatan *interoperability* inilah yang menjadi cikal bakal ontologi. Pada proses selanjutnya, data yang sudah diubah ke dalam format *turtle* (.ttl) akan dipindahkan ke dalam *protégé* sehingga format harus diubah menjadi (.owl). Pengubahan format dari (.ttl) ke (.owl) menggunakan *dump RDF/OWL* yang terdapat pada teknologi D2RQ *turtle*. Perpindahan data ke dalam *protégé* dilakukan untuk membuat *object properties*, data *properties*, dan pengecekan *SPARQL*, untuk *class* dan *instance* sudah secara otomatis terbuat. Pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta adalah pusat pelayanan kesehatan masyarakat yang cukup besar di daerah Yogyakarta. Pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta memiliki sistem yang cukup kompleks yang terdiri dari *front office* dan *back office*, dimana didalamnya terdiri atas beberapa modul. *Front office* terdiri dari modul rawat jalan, modul registrasi, modul poli, modul apotek, modul rawat inap, modul rekam medik, modul IGD, dan modul keperawatan. *Back office* terdiri dari modul keuangan, modul eksekutif, modul gudang, dan modul logistik. Besarnya sistem pada pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta maka hanya diambil satu modul untuk proses web semantik yaitu modul rawat jalan pada bagian *front office*. Pengambilan modul rawat jalan dikarenakan setiap pegawai pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota yang melakukan pencarian pada modul rawat jalan pasti memiliki pemahaman *knowledge* yang berbeda sehingga pencarian dibuat berdasarkan *knowledge*.

2. Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah membuat jembatan *interoperability* yang akan memindahkan *database* pada MySQL di web generasi 2.0 ke dalam web generasi 3.0 dengan bantuan D2R server *engine* menggunakan pendekatan *mapping* yang nanti akan dipindahkan dari *database* pada MySQL ke dalam *knowledgebase* dan akan berbasis web yang akan dipakai oleh pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta.

3. Metodologi Penelitian

Pada penelitian ini, metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Pemodelan Sistem, Merupakan tahapan diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk perangkat lunak. Pemodelan sistem yang digunakan dalam penelitian ini adalah *Unified Modelling Language* (UML).

2. Analisis, Merupakan proses menganalisis hal – hal yang dibutuhkan untuk pembuatan atau pengembangan perangkat lunak seperti metodologi riset, arsitektur, rekayasa dan pemodelan sistem, dan analisa kebutuhan perangkat lunak.
3. Perancangan, Merupakan tahapan perancangan *database*, perancangan *knowledgebase* seperti pendefinisian *class* dan hierarki *class*, pendefinisian *properties (slot)*, dan proses *instance* dengan menggunakan *On-To-Knowledge (OTK)* ontologi (Sure, Staab, & Studer, 2004), perancangan sistem dengan *Unified Modelling Language (UML)*, bagan alir (*flowchart*), dan perancangan *user interface*.
4. *Coding*, Merupakan tahapan untuk menerjemahkan algoritma ke dalam bahasa pemrograman yang telah ditentukan. Pada penelitian ini terbagi dua tahapan yakni dengan XML/RDF, *SPARQL* di level *knowledgebase* dan dengan *Netbeans Jena Library*.

5. Fundamental

a. Teknologi Web

Teknologi web adalah teknologi berbasis web yang dikembangkan dengan pemanfaatan bahasa pemrograman web untuk suatu ruang informasi yang diakses menggunakan *browser*.

b. Web Semantik

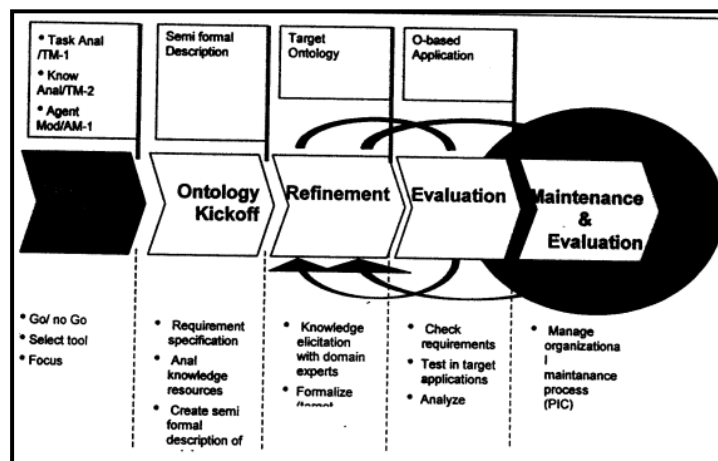
Web semantik merupakan perluasan dari web saat ini, sehingga informasi memiliki arti yang terdefinisi lebih baik, dengan mengupayakan persamaan persepsi antara konsep-konsep yang ada, sehingga memungkinkan manusia dan komputer untuk bekerjasama secara lebih optimal, terutama dalam pengelolaan dan penyajian informasi (Maedche & Staab, 2001).

c. Ontologi

Ontologi merupakan suatu teori tentang makna dari suatu obyek, properti suatu obyek serta relasi obyek tersebut yang memungkinkan terjadi pada suatu *domain* pengetahuan (Noy & McGuinness, 2001). Ontologi dapat digunakan untuk menerangkan tentang struktur suatu disiplin ilmu atau *domain* pengetahuan tertentu. Suatu ontologi atas beberapa komponen penting yaitu *individuals (instances)*, *classes (concepts)*, *attributes* dan *relationships*.

d. OTK (*On-To-Knowledge*)

On-To-Knowledge (OTK) adalah metodologi yang memiliki tahapan dan proses umpan balik seperti pada Gambar 1.



Gambar 1 Proses Pengembangan Ontologi dengan OTK (*On-To-Knowledge*)

Tahapan pada OTK adalah studi kelayakan (*feasibility study*), penentuan kelanjutan (*ontology kickoff*), penyempurnaan (*refinement*), dan pemeliharaan dan evolusi (Sure, Staab, & Studer, 2004).

e. Turtle

Turtle adalah format untuk menyatakan data dalam RDF. Model data RDF dengan sintaks mirip dengan *SPARQL*. RDF menampilkan informasi seperti *triple* yang masing-masing terdiri dari subyek, obyek, dan predikat. Item-item tersebut dinyatakan dalam URI web. *Turtle* menyediakan cara untuk menggabungkan tiga URI untuk menjadi sebuah *triple* dan menyediakan cara-cara untuk mempersingkat informasi. *Turtle* dijelaskan oleh Dave Backet sebagai bagian dari bahasa Notation3 (N3) tim Berners-Lee dan Dan Connolly, dan supersets minimal format *N-Triples*. Tidak seperti N3 yang hanya mempunyai kekuatan ekspresif yang melebihi RDF, *turtle* hanya bisa menserialisasi grafik RDF yang valid. *Turtle* adalah alternatif dari RDF/XML yang sebenarnya sintaks unik dan standard untuk menulis RDF. Kebalikan dari RDF/XML, *turtle* tidak bergantung pada XML

dan lebih dapat dibaca dan mudah diedit secara manual dari XML. *MySQL* adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License) dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*).

f. D2R server

D2R adalah *tool* untuk mempublikasikan konten dari *relational database* dalam web semantik. Konten *database* dipetakan dalam RDF dengan *mapping* yang dideklarasikan untuk menspesifikasikan bagaimana *resources* diidentifikasi dan bagaimana properti *value digenerate* dari konten *database*. D2R server mengijinkan RDF dan HTML browser untuk mengarahkan konten dari non-RDF *database* dan membolehkan aplikasi untuk melakukan *query* pada *database* menggunakan SPARQL *query* dengan SPARQL *protocol* (Bizer & Cyganiak, 2006).

D2RQ Mapping Language

D2RQ *language* merupakan bahasa yang dideklarasikan untuk mendeskripsikan relasi antara *relational database schema* dan RDFS *vocabularies* atau OWL *ontologies*. D2RQ merupakan *mapping* dokumen RDF yang disimpan dalam bentuk *turtle*. *Mapping* diekspresikan sesuai dengan term D2RQ *namespace*: <http://www.wiwiiss.fu-berlin.de/suhl/bizer/D2RQ/0,1#>. *Mapping* menjelaskan *virtual RDF Graph* yang terdiri atas informasi dari *database* (Cyganiak, Bizer, Garber, Maresch, & Chritian, 2012).

g. Database Connection

Mendefinisikan koneksi JDBC ke lokal atau *remote database relational*. D2rq: *Database* menggunakan propertinya dalam melakukan koneksi ke *database*. Selain menggunakan properti yang memiliki sendiri, d2rq: *Database* melakukan koneksi ke *database* juga dapat disukung dengan menggunakan *jdbc:namespace* (*namespace* URI: <http://d2rq.org/terms/jdbc/>). Contoh penulisan code untuk koneksi *database* MySQL ditunjukkan dalam Modul Program 1

```
map:database a d2rq:Database;  
  d2rq:jdbcDriver "com.mysql.jdbc.Driver";  
  d2rq:jdbcDSN "com.mysql://localhost/kerusakan";  
  d2rq:username "root";  
  jdbc:autoReconnect "true";  
  jdbc:zeroDateTimeBehavior "convertToNull";
```

Modul Program 1 Koneksi Database MySQL

h. Pembuatan RDF Resources

D2rq: *ClassMap* menyatakan *class* atau kelompok *class* yang serupa dari OWL *ontologies* atau RDFS *schema*. Sebuah *ClassMap* mendefinisikan bagaimana *instances* dari *class* diidentifikasi. D2rq: *ClassMap* terhubung dengan d2rq: *Database* dan memiliki kelompok d2rq: *PropertyBridges* yang melampirkan *property* yang melampirkan *property* ke *instances*. D2rq: *ClassMap* merupakan *instances* dari RDFS atau OWL *class*. D2RQ secara otomatis membuat *rdf:type triples* bila dibutuhkan (Cyganiak et al., 2012).

i. NetBeans

NetBeans merupakan merk dagang yang hak atas kekayaan intelektualnya dimiliki oleh Sun Microsystems, Inc. di Amerika Serikat dan negara-negara lainnya. *NetBeans* dimiliki serta diedarkan di bawah lisensi *Sun Public License*. *NetBeans* adalah versi *open source* dari Forte IDE, sebuah IDE untuk pengembangan aplikasi *java* yang dikembangkan oleh Forte Software, dimana Forte saat ini sudah diakuisisi oleh Sun dan menjadi salah satu produk penguat jajaran divisi *software* sun.

j. Jena RDF API

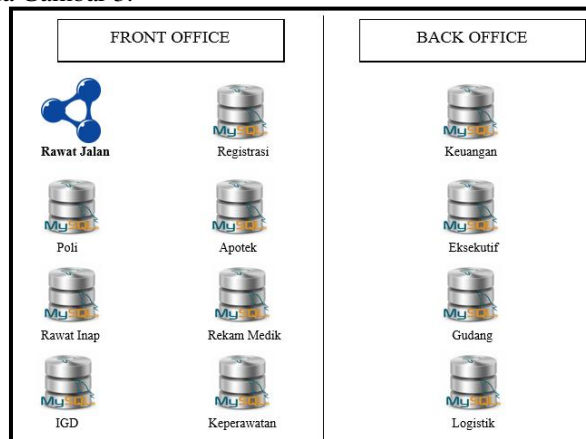
Jena merupakan suatu *Java* API dalam bahasa pemrograman *Java* untuk membuat dan memanipulasi graf RDF (McBride, 2001). *Jena* dikembangkan untuk memenuhi dua tujuan, yang pertama untuk menyediakan para programmer bahwa API lebih mudah digunakan daripada peralatan alternatif lainnya dan yang kedua dapat digunakan untuk menyesuaikan spesifikasi RDF. *Jena* API didesain khusus untuk bahasa pemrograman *Java*. *Jena* mengintegrasikan kedua jenis program menjadi *single* API (API tunggal). Aplikasinya dapat dituliskan menggunakan pendekatan pernyataan *centric*, pendekatan *resource centric* atau campuran keduanya.

k. Java Server Pages

Java Server Pages (JSP) merupakan salah satu *tool* untuk mengembangkan halaman-halaman web yang bersifat dinamis yang berbasis bahasa *Java*. Teknologi ini dikembangkan oleh Sun Microsystems.

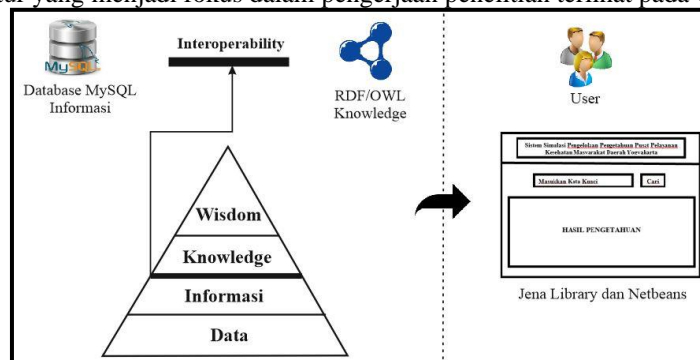
6. Case Study

Pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta mempunyai sistem yang kompleks yang terdiri dari dua bagian, yakni *front office* dan *back office* yang terdiri dari beberapa modul. *Front office* mempunyai modul rawat jalan, modul registrasi, modul poli, modul apotek, modul rawat inap, modul rekam medik, modul IGD, dan modul keperawatan. *Back office* mempunyai modul keuangan, modul eksekutif, modul gudang, dan modul logistik. Setiap modul-modul dari *front office* maupun *back office* tidak bisa langsung dibuat menjadi sistem web semantik secara utuh karena terbatasnya waktu sehingga pada penelitian ini hanya mengambil satu modul yaitu modul rawat jalan pada bagian *front office*. Berikut adalah arsitektur sistem yang akan dibangun yang terlihat pada Gambar 3.



Gambar 3 Arsitektur Sistem yang Akan Dibangun

Dijelaskan pada Gambar bahwa yang akan dikerjakan pada penelitian ini hanya pada modul rawat jalan sehingga agar tercipta sistem web semantik pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta yang utuh maka *database* modul-modul yang berada pada bagian *front office* dan *back office* dibuat sama seperti pada modul rawat jalan. Pada sistem ditampilkan metadata agar dapat digunakan untuk mengintegrasikan antar modul-modul yang berada pada bagian *front office* maupun *back office*. Didapatkan dari modul rawat jalan, arsitektur yang menjadi fokus dalam pengerjaan penelitian terlihat pada Gambar 4.



Gambar 4 Arsitektur Gambaran Umum

Gambar 4 dijelaskan bahwa penelitian ini dibagi menjadi dua tahapan. Pertama membuat jembatan *interoperability* dari level informasi ke dalam level *knowledge* yang bisa terlihat pada segitiga pada Gambar dimana informasi berupa *database* pada *MySQL* yang nantinya akan dipindahkan ke dalam *knowledge* dengan mengubah format menjadi *RDF/OWL*. Kedua membuat *interface* dengan *Netbeans* dan *Jena Library* sehingga bisa digunakan oleh *user*. Arsitektur proses apa saja yang dikerjakan dalam penelitian terlihat pada Gambar 5.



Gambar 5 Arsitektur Proses yang Akan Dikerjakan

Gambar 5 dijelaskan bahwa data Puskesmas Karangwaru, Tegalrejo, Kota Yogyakarta yang disimpan dalam *database* di *MySQL* yang diberi nama *puskesmas_tegalrejo*. Pada *puskesmas_tegalrejo* dibagi menjadi lima tabel, yaitu tabel pasien, tabel penyakit, tabel pembayaran, tabel dokter, dan tabel registrasi. Langkah selanjutnya adalah memindahkan data Puskesmas Karangwaru, Tegalrejo, Kota Yogyakarta yang berada pada *database* *puskesmas_tegalrejo* ke dalam *D2R server engine*. Pemindahan dari *MySQL database* ke dalam *D2R server engine* dimaksudkan untuk mengubah format dari (.sql) menjadi *turtle* (.ttl). Hasil dari pembentukan *mapping file* disimpan ke dalam format .owl agar bisa dimasukkan ke dalam aplikasi *protégé*. *Protégé* melakukan pembentukan *class*, *object properties*, *data properties*, dan pengujian *SPARQL*. Langkah terakhir pembentukan *user interface* dengan *Netbeans* dan *Jena Library*.

7. Rekayasa dan Pemodelan Sistem

Berdasarkan studi pustaka dan observasi yang sudah dilakukan maka dibuatlah rekayasa pemodelan sistem. Pemodelan sistem yang dipakai dalam penelitian ini adalah *Unified Modelling Language* (UML) karena didasarkan pada pendekatan *object oriented* sehingga dapat menganalisa apa saja yang dilakukan oleh aktor dalam menjalankan sistem. Perancangan ontologi yang dipakai dalam penelitian ini menggunakan pendekatan *On-To-Knowledge* (OTK). OTK dipakai dalam penelitian ini karena pada pendekatan ini memiliki proses umpan balik sehingga apabila terjadi kesalahan dalam membangun ontologi bisa kembali ke dalam *protégé* tanpa harus kembali ke *database*. Sistem pada aplikasi ini memiliki kemampuan untuk *user* mendapatkan pengetahuan mengenai data rawat jalan yang melakukan pemeriksaan pada tempat pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta sehingga hasil yang didapat berupa pengetahuan informasi yang ada pada tempat pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta, seperti tanggal periksa, nama pasien, tanggal lahir pasien, alamat pasien, nama penyakit, nama dokter, dan jenis pembayaran.

8. Analisa Kebutuhan Perangkat Lunak

Analisis kebutuhan merupakan langkah awal untuk menentukan perangkat lunak seperti apa yang akan dihasilkan. Langkah yang dilakukan adalah memindahkan web generasi 2.0 dalam *database* pada *MySQL* ke dalam web generasi 3.0 dengan menggunakan *D2R server engine* untuk menghasilkan format *turtle* (.ttl) *file* dan *protégé* (.owl) yang dapat digunakan untuk lokal *Java Application* (*NetBeans*) menggunakan *Jena Library* sehingga menghasilkan pengetahuan ontologi. Proses *instance*, *Instance-instance* dalam ontologi sumber data pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta ini dikembangkan melalui tahap-tahap sebagai berikut.

1. Pemilihan *class*
2. Pembuatan sebuah *instance* (*individual*) dari *class* tersebut
3. Pengisian nilai *properties* (*slot*) dari masing-masing *instance* (*individual*).

Informasi dan pengetahuan dari sistem pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta dari ontologi yang telah dibangun.

9. Implementasi

Langkah yang dilakukan dalam penelitian ini adalah membuat *database* pada *MySQL* dengan nama *puskesmas_tegalrejo*, dimana *database* *puskesmas_tegalrejo* memiliki lima tabel. Berikut ditampilkan data-data dari setiap tabel pasien, tabel penyakit, tabel pembayaran, tabel dokter, dan tabel registrasi.

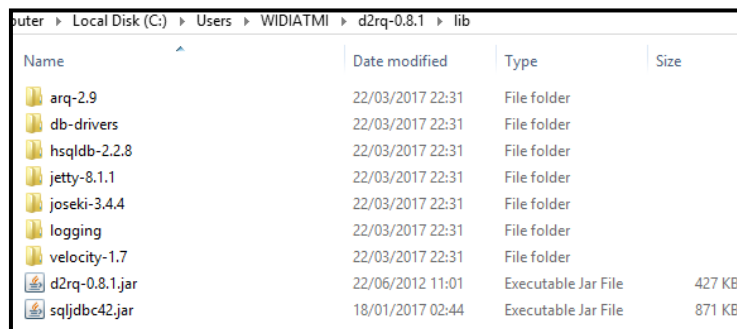
	id_pasien	nama_pasien	alamat	tanggal_lahir
<input type="checkbox"/> Edit Copy Delete	1.3838	Nanda Sari	Jatimulyo 01/01	1996-07-07
<input type="checkbox"/> Edit Copy Delete	1.4209	Eko Haryanto	Jt Mulyo 14/03	1988-08-30
<input type="checkbox"/> Edit Copy Delete	1.4802	Marinah	Jl. Jambon 27B	1942-07-07
<input type="checkbox"/> Edit Copy Delete	1.5526	Patinah	Kricak Kidul 35/08	1954-02-13
<input type="checkbox"/> Edit Copy Delete	1.5979	Tukini	Bangunrejo 51/11	1954-01-23
<input type="checkbox"/> Edit Copy Delete	1.6366	Daniel Kristian	Jatimulyo 02/01	1986-01-02
<input type="checkbox"/> Edit Copy Delete	18.02.0033	Nunuk Setyati	Bangunrejo 35/10	1957-01-13
<input type="checkbox"/> Edit Copy Delete	18.42.04.0032	Ridho Fathurohman	Bener 07/02	2011-01-01
<input type="checkbox"/> Edit Copy Delete	18.42.04.0319	Sutini	Bener 02/01	1983-01-19
<input type="checkbox"/> Edit Copy Delete	18.44.01.0009	Sri Umi	Kricak 45/10	1954-01-30
<input type="checkbox"/> Edit Copy Delete	18.44.01.0011	Bartini	Bangunrejo 40/10	1946-03-19
<input type="checkbox"/> Edit Copy Delete	18.44.01.0110	Suwarto	Jatimulyo 13/04	1961-01-17
<input type="checkbox"/> Edit Copy Delete	18.44.01.0544	Masidi	Jatimulyo 24/05	1936-12-02
<input type="checkbox"/> Edit Copy Delete	18.44.01.0564	Siswanto	Bangunrejo 50/11	1963-02-15
<input type="checkbox"/> Edit Copy Delete	18.44.01.0664	Masri	Bangunrejo Tr. I/1787	1962-12-04
<input type="checkbox"/> Edit Copy Delete	18.44.01.1194	Krisna Murti	Jl. Jambon 27B	2007-08-09
<input type="checkbox"/> Edit Copy Delete	18.44.01.2086	Edi Waskito	Jati Mulyo 16/04	1972-10-30
<input type="checkbox"/> Edit Copy Delete	18.44.01.2738	Sumini	Bangunrejo 45/10	1941-01-15
<input type="checkbox"/> Edit Copy Delete	18.45.02.0032	Rusman	Bangunrejo 35/10	1949-06-29
<input type="checkbox"/> Edit Copy Delete	18.45.02.0082	Maulani	Karangwaru 56/02	2003-08-18

Gambar 6 Tampilan Data Dari Tabel Pasien

Jembatan *Interoperability* Untuk Pindah Ke Web Generasi 3.0 Perpindahan di dalam D2R server engine

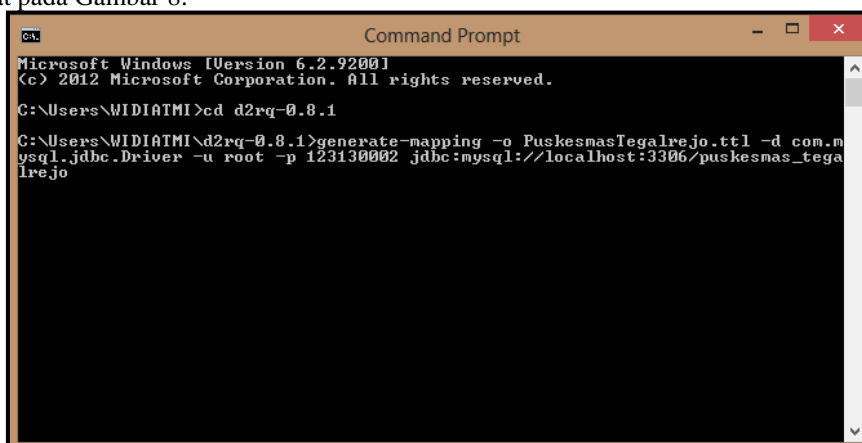
Langkah yang dilakukan dalam perpindahan di dalam D2R server engine adalah *generate mapping* dan *dump* RDF/OWL, tetapi sebelum langkah-langkah tersebut dilakukan terlebih dahulu melakukan penginstalan D2RQ. Adapun langkah-langkah yang harus dilakukan untuk melakukan instal D2RQ sebagai berikut:

1. Mengunduh D2RQ-0.8.1 pada www.d2rq.org.
2. Mengekstrak file D2RQ-0.8.1 dan menempatkan pada local disk C >> users >> widiatmi.
3. Menambahkan sqljdbc42.jar pada folder lib yang terdapat pada file D2RQ-0.8.1 seperti yang terlihat pada Gambar 7.



Gambar 7 sqljdbc42.jar

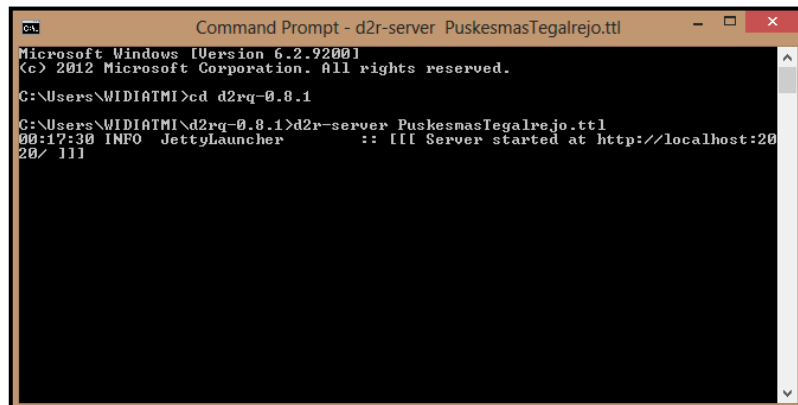
Langkah kedua melakukan *generate mapping* atau *mapping* data menggunakan *command prompt* seperti yang terlihat pada Gambar 8.



Gambar 8 Perintah Mapping

Pada Gambar 8 dijelaskan bahwa langkah pertama masuk pada *directory* D2R server *engine*. Langkah kedua mengetikkan “generate-mapping -o PuskesmasTegalrejo.ttl -d com.mysql.jdbc.Driver -u root -p 123130002 jdbc:mysql://localhost:3306/puskesmas_tegalrejo”. Terdapat perintah “generate-mapping” yaitu perintah *mapping* data, lalu terdapat “-o PuskesmasTegalrejo.ttl” adalah output dari *mapping* menggunakan sintaks *turtle*, “-d com.mysql.jdbc.Driver” adalah *driver database MySQL*, “-u root -p 123130002 jdbc:mysql://localhost:3306/puskesmas_tegalrejo” adalah pengambilan data dari *database* puskesmas_tegalrejo. Langkah berikutnya adalah melakukan pengecekan pada D2R server *engine*. Adapun langkah-langkah yang dilakukan untuk melakukan pengecekan pada D2R server *engine* sebagai berikut:

1. Mengetikkan “d2r-server PuskesmasTegalrejo.ttl” pada *command prompt* seperti yang terlihat pada Gambar 9.



Gambar 9 Perintah Memulai D2R Server *Engine*

2. Membuka <http://localhost:2020> pada *web browser* yang terlihat pada Gambar 10.



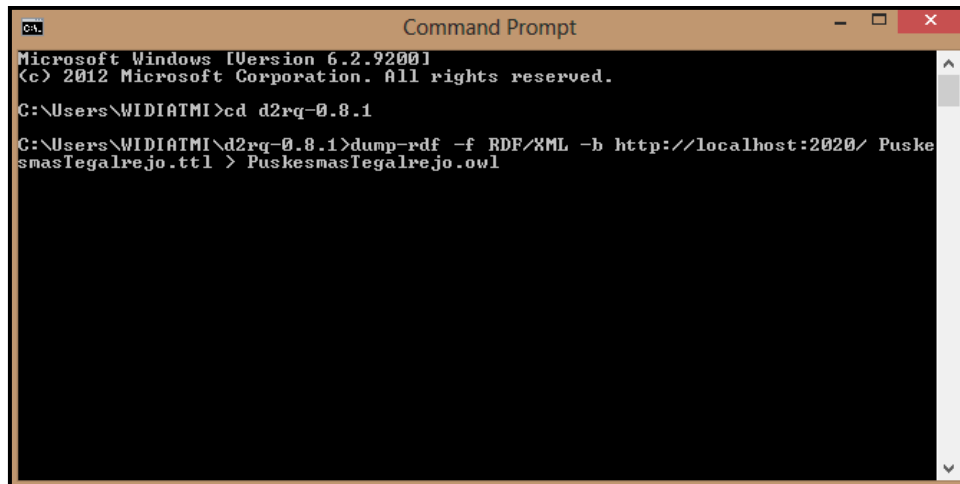
Gambar 10 Tampilan D2R Server *Engine*

Setiap akan mengakses <http://localhost:2020> terlebih dahulu melakukan langkah pertama tanpa menutup *command prompt*.

Perpindahan di dalam *protégé*

Langkah pertama yang dilakukan dalam penelitian ini adalah mentransfer data yang ada pada D2R server *engine* ke dalam *protégé* 4.3. Adapun langkah-langkah yang dilakukan dalam penelitian ini untuk mentransfer data yang ada pada D2R server *engine* ke dalam *protégé* 4.3 sebagai berikut:

1. Membuat RDF ke dalam bentuk format .owl yaitu dengan mengetikkan “dump-rdf -f RDF/XML -b <http://localhost:2020/> PuskesmasTegalrejo.ttl > PuskesmasTegalrejo.owl” pada *command prompt* seperti yang terlihat pada Gambar 12 atau bisa juga untuk melakukan *generate mapping*, pengaksesan server D2R server *engine* dan *dump* RDF dibuat dalam bentuk format (.bat) yang nantinya dimasukkan ke dalam folder d2rq-0.8.1.



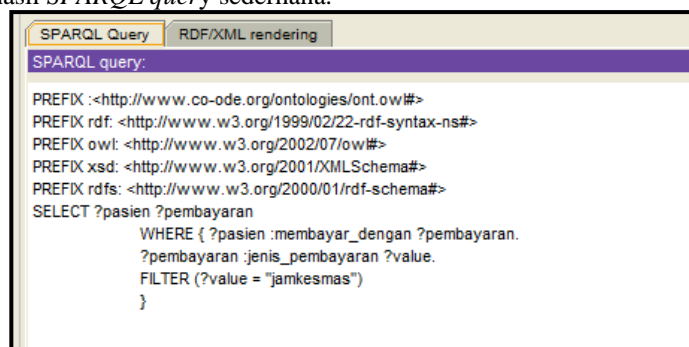
Gambar 11 Perintah Dump RDF

2. Membuka file PuskesmasTegalrejo.owl pada aplikasi *protégé* 4.3, karena bersifat semi otomatis maka data yang berhasil ditransfer ke dalam *protégé* 4.3 hanya *class* dan *individuals* saja maka untuk membuat *object properties* dan data *properties* dilakukan pada *protégé* 4.3.

Langkah kedua yang dilakukan dalam penelitian ini adalah membuat *object properties* dan data *properties* di dalam *protégé* 4.3 pada file PuskesmasTegalrejo.owl yang sudah dibuat di D2R server engine sebelumnya.

Pengujian SPARQL query.

Contoh ke-1 menampilkan pasien yang membayar dengan jamkesmas pada ontologi yang sedang dibangun. Pada Gambar 12 memperlihatkan hasil SPARQL query sederhana.



Gambar 12 Contoh SPARQL Query Menampilkan Pasien yang Membayar dengan Jamkesmas

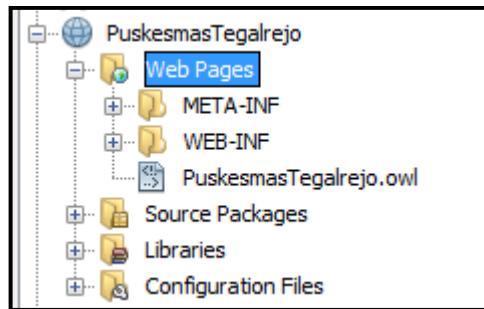
Dijelaskan bahwa “SELECT ?pasien ?pembayaran” adalah *class* yang akan ditampilkan pada hasil pengetahuan. “WHERE {}” menunjukkan syarat atau ketentuan dimana dalam contoh pada Gambar 12 pasien yang membayar dengan pembayaran. Dalam menampilkan pasien yang membayar dengan jamkesmas saja maka menggunakan “?pembayaran :jenis_pembayaran ?value. FILTER (?value = "jamkesmas)”. *Value* adalah variabel yang digunakan untuk mendefinisikan “jamkesmas”, sehingga hasil pengujian SPARQL query yang menampilkan pasien yang membayar dengan jamkesmas terlihat pada Gambar 13.

pasien	pembayaran
'pasien #18.44.01.0110'	'pembayaran #b001'
'pasien #18.44.01.0664'	'pembayaran #b001'
'pasien #1.6366'	'pembayaran #b001'
'pasien #18.45.02.0877-'	'pembayaran #b001'
'pasien #1.3838'	'pembayaran #b001'
'pasien #1.5979'	'pembayaran #b001'
'pasien #18.45.02.1107-'	'pembayaran #b001'
'pasien #18.42.04.0032'	'pembayaran #b001'
'pasien #18.45.02.0082'	'pembayaran #b001'
'pasien #18.42.04.0319'	'pembayaran #b001'
'pasien #18.45.02.1052-'	'pembayaran #b001'
'pasien #1.5526'	'pembayaran #b001'

Gambar 13 Hasil SPARQL Query Menampilkan Pasien yang Membayar dengan Jamkesmas

Hasil yang dimunculkan pada Gambar 13 adalah *primary key* dari setiap *class* pasien dan *class* pembayaran, dimana pembayaran #b001 adalah kode pembayaran dari jamkesmas.

Pembuatan tampilan *interface* pada penelitian ini membutuhkan *NetBeans* 7.4, server tomcat, *Jena* API dan *Protégé-OWL* API. Langkah selanjutnya adalah membuat *project* baru dengan PuskesmasTegalrejo adalah memasukkan file PuskesmasTegalrejo.owl pada *packages Web Pages* yang terlihat pada Gambar 14.



Gambar 14 Proses Memasukkan Ontologi

Langkah selanjutnya adalah menambahkan *Jena* API dan *Protégé-OWL* API yang berekstensi jar ke dalam *packages Libraries* dengan cara mengklik kanan *packages Libraries* >> *Add JAR/Folder* >> memilih file *jena* yang berekstensi jar pada bagian *lib* >> *open*. Langkah yang dilakukan dalam implementasi *query* dimulai dengan file *PuskesmasTegalrejo.owl* sebagai basis pengetahuan dari aplikasi web. Pembacaan file *PuskesmasTegalrejo.owl* tersebut menggunakan *method* *readFile* ("PuskesmasTegalrejo.owl"). Pada Modul Program 1 memperlihatkan *method* untuk pembuatan model ontologi.

```
model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_RDFS_INF);
```

Modul Program 1 *Method* untuk Pembuatan Model Ontologi

Kemudian file OWL tadi akan dibaca sebagai *InputStream* seperti yang diperlihatkan pada Modul Program 2.

```
InputStream in=FileManager.get().open(sc.getRealPath("/"+inputFileName));  
Model.read(new InputStreamReader(in), "")
```

Modul Program 1 Pembacaan File OWL sebagai *InputStream*

Setelah model tersebut membaca file *PuskesmasTegalrejo.owl* maka selanjutnya akan diambil elemen PREFIX dan *namespace* URI dari PREFIX tersebut dengan *method* *model.getNSPrefixMap()* dan *model.getNsURIPrefix(ns)*. *Prefix* hasil *method* *readFile*("PuskesmasTegalrejo.owl") ini disimpan ke dalam variabel global bernama *prefix*.

Pembuatan *query SPARQL* dilakukan dengan melewati berbagai parameter pencarian yang ditangkap oleh halaman antarmuka JSP ke *method* *parseSearchKeys* dan dikembalikan sebagai *string query* setelah digabung dengan *prefix* yang disimpan dalam variabel *prefix*. *String query* tersebut kemudian dieksekusi dengan memanggil *method* *execute(model.query)* yang akan menghasilkan dokumen hasil bertipe *ResultSet*. Cuplikan kode pada *method* *util.execute(model.query)* dapat dilihat pada Modul Program 3 sebagai berikut:

```
Query query = QueryFactory.create(queryString);  
QueryExecution qexec = QueryExecutionFactory.create(query, model);  
return qexec.execSelect ();
```

Modul Program 3 Cuplikan *Method* untuk Mengeksekusi *String Query*

Hasil *query* tersebut perlu diformat ke dalam format RDF/XML jika hasilnya berupa RDF atau ke dalam format XML jika hasilnya bukan berupa RDF. Untuk memformat hasil *query* tersebut digunakan *method* *prepareResult*, yang nantinya akan menghasilkan sebuah dokumen sesuai dengan format hasil *query SPARQL* yaitu berformat XML. Pada *method* *parseResult* dilakukan pemisahan untuk setiap solusi *query* yang dinyatakan pada elemen *result* dan *results* pada dokumen XML tersebut diperlihatkan pada Modul Program 4.

```
List list= d.getRootElement().element("results").elements("result");
```

Modul Program 2 Pemisahan untuk Setiap Solusi *Query*

Halaman awal merupakan halaman yang pertama kali muncul pada sistem simulasi pengelolaan semantik registrasi pasien pada pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta. Teknik pencarian dilakukan berdasar kata kunci akan meminta *user* untuk memasukkan sebuah kata kunci pada sebuah *text-entry field* yang tersedia. Kata kunci yang dimasukkan akan dicocokkan dengan nilai *property*/nama *instances* pengetahuan yang dimiliki oleh sistem. Gambar 15 merupakan hasil implementasi rancangan halaman pencarian kata kunci.



Gambar 15 Halaman Pencarian Terminologi

Saat tombol *Search* diklik maka semua data yang ada pada *form* di halaman *index.jsp* akan dikirim ke halaman *hasil_cari_kata_kunci_1.jsp*. Pada halaman *hasil_cari_kata_kunci_1.jsp* kata kunci dari *text-entry field* akan ditangkap dan disimpan ke dalam variabel *searchText*, dengan *statement* yang diperlihatkan pada Modul Program 5.

```
String searchText = request.getParameter("keyword")
```

Modul Program 5 *Statement* untuk Menangkap Kata Kunci

Variabel *searchText* tersebut kemudian dicocokkan dengan nilai properti. Modul Program 6 berikut menunjukkan potongan *query* yang memperlihatkan proses pencocokkan.

```
"FILTER (REGEX(str(?value)," + searchText + "','i') || REGEX(?nama_pasien," + searchText + "','i') ||  
REGEX(?tanggal_lahir," + searchText + "','i') || REGEX(?alamat," + searchText + "','i') ||  
REGEX(?tanggal_periksa," + searchText + "','i') || REGEX(?nama_dokter," + searchText + "','i') ||  
REGEX(?jenis_pembayaran," + searchText + "','i'))";  
queryString += "} ORDER BY ?nama_pasien";
```

Modul Program 6 Proses Pencocokan *Query*

Setelah proses pencocokkan selesai dan diperoleh hasil yang sesuai, maka hasil *query* tersebut akan ditampilkan. Informasi yang ditampilkan adalah informasi dari pasien seperti tanggal periksa, nama pasien, alamat, tanggal lahir, nama penyakit, nama dokter, dan jenis pembayaran yang dilakukan oleh. Sesuai dengan klausa *SELECT* pada *query* SPARQL yang ditunjukkan pada Modul Program 7.

```
SELECT DISTINCT ?tanggal_periksa ?nama_pasien ?tanggal_lahir ?alamat ?nama_penyakit  
?nama_dokter ?jenis_pembayaran
```

Modul Program 7 Klausa *SELECT* pada *Query* SPARQL

10. Kesimpulan

1. Data yang dipakai dalam penelitian ini adalah pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta pada bagian *front office* khususnya rawat jalan pada tahun 2014.
2. Jembatan yang digunakan untuk melakukan perpindahan *database MySQL 5.6* ke dalam *protégé 4.3* adalah D2RQ.
3. Penggunaan *semantic web search* pada pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta dilakukan dengan menggunakan pendekatan *mapping*.
4. Pembuatan tampilan *interface* pada *semantic web search* pusat pelayanan kesehatan masyarakat Karangwaru, Tegalrejo, Kota Yogyakarta dengan menggunakan *Jena Library* dan *Netbeans 7.4*.

11. Daftar Pustaka

- Maedche, A., & Staab, S. (2001). *Ontology Learning for The Semantic Web*. Retrieved from <http://ceur-ws.org/Vol-40/maedche+staab.pdf/>
- Noy, F. ., & McGuinness, D. . (2001). *Ontology Development 101: A Guide to Creating your First Ontology*. Retrieved from <http://protege.stanford.edu/publications/ontology-development/ontology101-noy-mcguinness.html>
- Sure, Y., Staab, S., & Studer, R. (2004). *On-To-Knowledge Methodology (OTKM)*. In P. D. S. Staab & P. D. R. Studer (Eds.), *Handbook on Ontologies* (pp. 117–132). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24750-0_6
- Bizer, C., & Cyganiak, R. (2006). *D2rq Server-Publishing Relational Database on Semantic Web*. ISWC.

Cygniak, R., Bizer, C., Garber, J., Maresch, O., & Chritian, B. (2012). The D2RQ Mapping Language. Retrieved from <http://d2rq.org>
McBride, B. (2001). Jena: Implementing the RDF Model and Syntax Specification, Hewlett Packard Laboratories Filton Road, Stoke Gifford Bristol, UK. Retrieved from <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-40/mcbride.pdf>

ogyakarta: Universitas Ahmad Dahlan.