IJCCS, Vol.11, No.2, July 2017, pp. 155~164
ISSN: 1978-1520

Rancang Bangun *Plugin Protégé* Menggunakan Ekspresi *SPARQL-DL* Dengan Masukan Bahasa Alami

Muhammad Fahrurrozi*[1], Azhari SN[2]
[1]Program Studi S2 Ilmu Komputer, FMIPA UGM, Yogyakarta, Indonesia
[2]Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia
e-mail: *[1]**f4.rozi@gmail.com**,[2]arisn@ugm.ac.id

Abstract

Semantik web merupakan teknologi yang memungkinkan kita membangun basis pengetahuan atau ontologi agar informasi dari halaman web bisa dipahami oleh komputer. Salah satu perangkat lunak untuk membangun ontologi berbasis semantic web adalah protege. Protégé menyediakan plugin berupa DL-Query dan SPARQL-Query untuk menampilkan informasi yang melibatkan ekspresi class, property dan individual di dalam ontologi. Permasalahan yang kemudian muncul adalah plugin DL-Query hanya mampu memperoses aturan-aturan yang melibatkan ekspresi class dengan object property saja walaupun sudah dilengkapi dengan fungsi reasoning. sedangkan plugin SPARQL-Query tidak memiliki kemampuan reasoning seperti plugin DL-Query walaupun plugin SPARQL-Query bisa memperoses aturan-aturan yang melibatkan class, property dan individual. Penelitian ini menghasilkan plugin baru menggunakan metode SPARQL-DL dengan masukan bahasa alami karena protege belum menyediakan plugin dengan masukan bahasa alami untuk melihat hasil dari gabungan ekpresi-ekpresi yang terdapat di dalam ontologi sehingga memudahkan developer melihat informasi ontologi dengan bahasa yang lebih mudah dipahami tanpa harus memikirkan struktur query SPARQL yang rumit.

Kata kunci— *Semantic Web, Protégé, Plugin,SPARQL-DL*

Abstract

Semantic web is a technology that allows us to build a knowledge base or ontology for the information of the web page can be understood by computers. One software for building ontology-based semantic web is a protégé. Protege allows developers to develop an ontology with an expression of logic description. Protégé provides a plugin such as DL-Query and SPARQL-Query to display information that involve expression of class, property and individual in the ontology. The problem that then arises is DL-plugin Query only able to process the rules that involve expression of class to any object property, despite being equipped with the function of reasoning. while the SPARQL-Query plugin does not have reasoning abilities such as DL-Query plugin although the SPARQL-Query plugin can query memperoses rules involving class, property and individual. This research resulted in a new plugin using SPARQL-DL with input natural language as a protégé not provide a plugin with input natural language to see results from the combined expression-expression contained in the ontology that allows developers to view information ontology language that is easier to understand without having think of SPARQL query structure is complicated.

Keywords— *Semantic Web, Protégé, Plugin,SPARQL-DL*

Received January 26th,2017; Revised March 21th, 2017; Accepted March 22th, 2017

1. PENDAHULUAN

Informasi di dalam internet telah memberikan kemudahan dalam berbagai bidang, mulai dari bidang kesehatan, politik, kehidupan sosial, sejarah, dan berbagai bidang disiplin ilmu lainnya. Sumber informasi yang banyak dan beragam menimbulkan permasalahan baru dimana mesin pencari yang ada saat ini tidak memberikan informasi secara spesifik mengenai informasi yang kita inginkan, sehingga mesin membutuhkan teknologi yang bisa memahami isi informasi dari halaman web supaya mesin pencari bisa memberikan informasi yang spesifik sesuai dengan permintaan pengguna. Semantik web merupakan teknologi yang memungkinkan kita membangun basis pengetahuan atau ontologi agar informasi dari halaman web bisa dipahami oleh komputer. Semantic web merupakan gabungan teknologi yang memungkinkan mesin untuk memahami makna (semantik) dari informasi di dalam halaman web [1].

Salah satu perangkat lunak yang menangani penerapan dalam membangun ontologi berbasis *semantic web* adalah protege. Protege memungkinkan ontologi *developer* untuk mengembangkan ontologi dengan ekspresi deskripsi *logic* yang sangat kompleks seperti, *class axiom, object property axiom, equivalent* class, *equivalent property* dan lain-lain. Protégé menyediakan *plugin* berupa *DL-Query* dan *SPARQL-Query* untuk menampilkan data yang melibatkan ekspresi *class, property* dan *individual* di dalam ontologi. Permasalahan yang kemudian muncul adalah *plugin DL-Query* hanya mampu memperoses aturan-aturan yang melibatkan ekspresi *class* dengan *object property* saja walaupun sudah dilengkapi dengan fungsi *reasoning* dalam pencarian data. sedangkan *plugin SPARQL-Query* tidak memiliki kemampuan *reasoning* seperti *plugin DL-Query* walaupun *plugin SPARQL-Query* bisa memperoses aturan-aturan yang melibatkan *class, property* dan *individual* yang tidak dimiliki oleh *DL-Query*.

Penelitian tentang penerapan ontologi berbasis ontologi pernah dilakukan diantaranya penerapan metode *Simple Knowledge Organization System* (SKOS) untuk merepresentasikan hubungan semantik dalam basis pengetahuan untuk pencarian katalog perpustakaan menggunakan bahasa alami [2], penerapan ontologi dengan masukan bahasa alami dengan metode *Semantic Web Relational Languange* (*SWRL)* juga dilakukan untuk memproses pencarian pada domain musik dimana aplikasi yang dibangun menunjukkan hasil yang maksimal dan efisien dengan ketentuan kalimat yang diinputkan berbentuk *single value* dan tidak mendukung kalimat berbentuk *multivalue* [3]. Penelitian tentang *Question and Answering* (*QA*) pada domain restoran dengan masukan bahasa alami yang mendukung kalimat berbentuk *multivalue* dengan pendekatan metode menggunakan *SPARQL* [4], selanjutnya pengembangan aplikasi *QA* berbasis ontologi juga digunakan untuk mencari informasi jalur pendakian gunung degan memanfaatkan bahasa alami sebagai masukan dan mengikuti kaidah aturan tata bahasa indonesia. Pendekatan metode yang digunakan adalah denga menggunakan *query SPARQL* [5]. Hasil dari sistem yang dibangun secara kuantitatif menunjukkan bahwa sistem mampu memahami secara acak input yang diambil dari responden sebesar 69 persen. Berbeda dengan penelitian sebelumnya pengembangan aplikasi *QA* berbasis ontologi juga digunakan untuk melakukan pencarian data pemerintahan untuk Kabupaten Nusa Tenggara Barat, dimana ontologi yang dibangun berbasis *Multi-Ontologi* dengan menggunakan metode *SPARQL-DL* [6]. Pemanfaatan ontologi berbasis *SPARQL* juga dilakukan untuk pengembangan jadwal penerbangan pesawat dengan menggunakan pemodelan ontologi, hasil dari pengembangan tersebut memberikan hasil yang signifikan dalam mengatur jadwal penerbangan [7].

2. METODE PENELITIAN

2.1 Protege

Perangkat lunak protege dikembangkan oleh *Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine*. Perangkat lunak protege bersifat *Open Source* dibawah licensi bernama *Mozilla Public License* (*MPL*). Perangkat lunak protege merupakan alat bantu untuk membantu ontologi *developer* untuk memngembangkan sistem yang didasarkan pada basis pengetahuan *Knowledge Base System*. Protege dapat

membuat, mengedit dan menyimpan ontologi dalam format *CLIPS, RDF, XML, UML* dan *Relational Database* [8].

Secara umum protege memudahkan pengguna untuk membuat pemodelan dasar secara lebih sederhana yang dilengkapi dengan visualisasi hubungan *SubClass* dalam *Tree*. Protege juga mendukung berbagai penurunan (*Multiple Inheritance*) dan *root* pada hirarki *Class* yang adalah *Class ``Thing''*.

2.2 Plugin

Plugin merupakan sebuah program komputer yang menambahkan fungsionalitas sebuah program utama. Program utama pada umumnya memberikan *Interface* agar *plugin* dapat berinteraski dengan program utama [9]. Jadi dapat dikatakan *plugin* adalah sebuah program tambahan yang bisa diintegrasikan dengan program utama untuk memberikan fungsi-fungsi lain yang belum tersedia pada instalasi *standart*.

2.3 Semantic Web

Semantic Web merupakan kumpulan teknologi dan aturan-aturan standar yang memungkinkan mesin untuk memahami makna (semantik) informasi di dalam web [1].

2.4 *Resource Description Framework* (*RDF*)

RDF adalah standar yang diterbitkan oleh W3C yang digunakan untuk mendistribusikan informasi atau pengetahuan kepada aplikasi komputer melalui proses dan cara-cara yang terukur [1].

2.5 *Ontologi*

Ontologi secara umum didefinisikan sebagai seperangkat istilah yang digunakan untuk menggambarkan dan mewakili sebuah *domain*. Ontologi mendefinisikan istilah yang digunakan untuk menggambarkan dan mewakili bidang pengetahuan [1].

2.6 SPARQL-DL

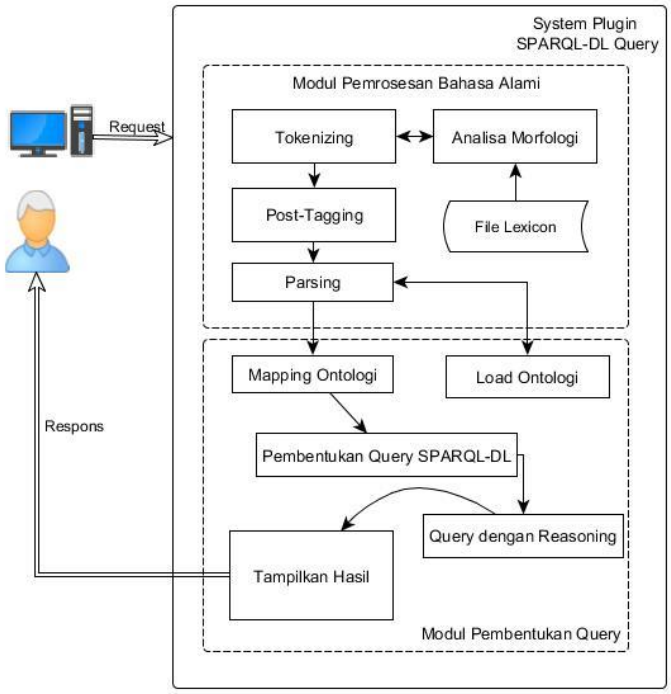
Bahasa query ontologi pada semantik *web* dibagi menjadi dua kelompok yaitu: bahasa query berbasis *RDF* (*Resource Description Framework*) dan bahasa query berbasis *DL* (*Description Logic*). Bahasa query berbasis *RDF* ditujukan untuk melakukan query terhadap *RDF Graph*. Bahasa *query* berbasis *RDF* diantaranya adalah *SPARQL*, *SeRQL* dan *RDQL* [10].

2.7 *Deskripsi Sistem*

Penelitian ini membahas tentang rancang bangun sistem *plugin* untuk protégé yang ditujukan untuk memudahkan pengembang aplikasi membuat sistem berbasis ontologi. *Plugin* yang dibangun merupakan fungsi tambahan untuk melengkapi kekurangan pada sistem aplikasi berbasis ontologi yaitu *Protege*. Sistem *plugin* yang dibangun mempunyai kemampuan untuk melakukan proses *query* terhadap ontologi *OWL-DL* dengan menggunakan metode *SPARQL-DL* dengan cara mentranslasikan bahasa alami menjadi ekspresi *DL*. Aplikasi yang dibangun hanya terdiri dari satu form untuk melakukan input data dan menampilkan hasil pencarian data. Aplikasi yang dibangun ditujukan untuk aplikasi berbasis *desktop* dan hanya berupa sistem tambahan untuk melengkapi kekurangan fitur yang belum ada pada protege.

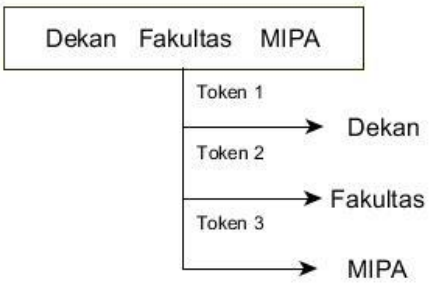
2.8 *Arsitektur Sistem*

Sistem yang dibangun merupakan sistem berbasis desktop dengan menggunakan bahasa Pemrograman JAVA. Aplikasi dibangun dengan struktur *OOP* (*Object Oriented Programing*) yang terdiri dari paket *controller*, pemrosesan bahasa alami, pemrosesan ontologi, pembentukan *query SPARQL-DL*, model serta *helper*. Paket *controller* berisi kelas Main yang berfungsi sebagai *endpoint* yang akan menerima request dari user berupa kalimat tanya. Tahapan proses pencarian informasi dilakukan melalui kelas Main. Arsitektur sistem yang akan dikembangkan diperlihatkan dalam Gambar 1 Arsitektur sistem.



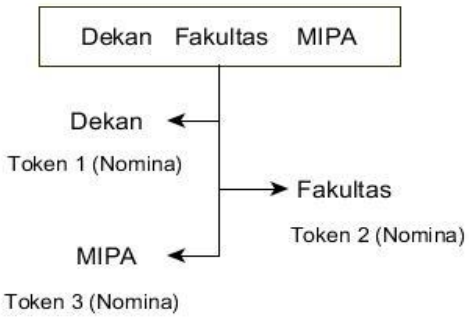
Gambar 1 Arsitektur sistem

Pada Gambar 1 menjelaskan proses perancangan perangkat lunak *plugin* diawali dengan memasukkan bahasa alami berbentuk kalimat tunggal, Contoh ``Dekan Fakultas MIPA". Proses selanjutnya adalah proses *tokenizing* dimana proses tokenizing merupakan proses membagi atau mencacah kalimat tunggal ``Dekan Fakultas MIPA" kedalam token-token atau kata-kata yang dibedakan berdasarkan karakter spasi dalam kalimat seperti yang disajikan pada Gambar 2 Proses Pemisahan Kalimat.



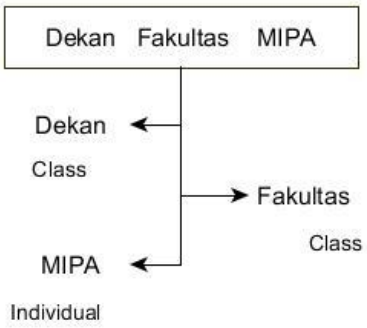
Gambar 2 Proses Pemisahan Kalimat

Gambar 2 merupakan gambaran proses *tokenizing* pada kalimat ``Dekan Fakultas MIPA" kemudian dicacah kedalam beberapa kata berupa ``Dekan", ``Fakultas" dan ``MIPA", Selanjutnya dilakukan proses *Post-Tagging*. *Post-Tagging* merupakan proses pemberian nama entitas seperti pada Gambar 3.



Gambar 3 Proses Pemberian Nama Entitas (*Post-Tagging*)

Pada Gambar 3 kata ``Dekan", ``Fakultas" dan ``MIPA" diverifikasi apakah termasuk dalam kategori *Verb* (Kata kerja), *Nomina (*Kata benda*), Adverbia* dan *adjective (*Kata Sifat*)* yang telah ditetapkan pada data *lexicon*. Jika kata yang terdapat di dalam token tidak ditemukan di dalam data *lexicon* maka akan dilakukan proses analisa morfologi untuk membantu mendefinisikan nama entititas yang tidak terdapat di dalam data lexicon. Setelah proses *Post-Tagging* atau analisa morfologi, selanjutnya dilakukan proses parsing, proses yang dilakukan adalah melakukan pencarian untuk memetakan string kata-kata berupa ``Dekan", ``Fakultas" dan ``MIPA" ke dalam himpunan pola sintaksis yang bermakna dengan mengecek struktur data yang dipetakan sesuai dengan sintaksis atau struktur kalimat berita yang telah dibangun.Setelah proses *parsing* selesai dilanjutkan ke tahap *mapping* Ontologi seperti yang di tunjukkan pada Gambar 4.

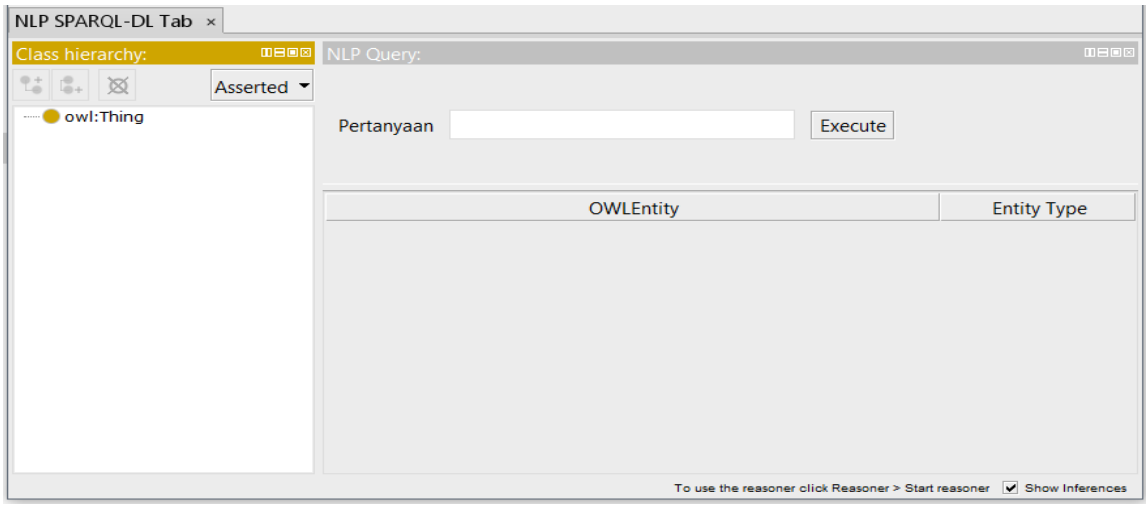


Gambar 4 Proses Mapping Ontologi

Gambar 4 menjelaskan proses *mapping* ontologi yang ditujukan untuk memeriksa dan memastikan apakah kata-kata pada kalimat terdapat di dalam ontologi kemudian dipetakan berdasarkan ontologi yang telah dibangun. Setelah tahapan *mapping* ontologi dilakukan kemudian dilanjutkan ke tahapan pembentukan *query SPARQL-DL* merupakan proses membentuk *statement query SPARQL-DL* yang akan digunakan untuk melakukan *query* terhadap ontologi yang sedang dibangun dalam perangkat lunak *Protege*. Tabel 1 merupakan bentuk dari *query SPARQL-DL* yang digunakan dalam pembuatan *plugin*. Selanjutnya dilakukan proses *query* dan proses *reasoning*. Proses *query* adalah proses eksekusi *query* SPARQL-DL yang telah di bangun yang digabungkan dengan proses *reasoning* untuk mendapatkan informasi yang bersifat *implisit* berdasarkan ekspresi DL yang terdapat dalam objek ontologi baik pada restriksi kelas maupun *object property*. Selanjutnya proses terakhir adalah proses menampilkan hasil dari proses *query* dan *Reasoning*.

2.9 *Impelementasi Sistem*

Implementasi sistem terdiri dari implementasi pembentukan *Graphical User Interface* yang diproses pada *module view*, Proses inisialisasi dan pembentukan fungsi pada sistem yang ditampung pada *module* data model. Pembentukan *module handler* untuk mengeksekusi semua fungsi yang ada dalam *module view* dan *module data model*. Gambar 5 menunjukkan sistem plugin yang telah berhasil dibangun.



Gambar 5 Tampilan *plugin SPARQL-DL*

Pada Gambar 5 diperlihatkan bentuk tampilan *plugin SPARQL-DL* yang berhasil dibangun pada *protégé*. *Plugin* yang dibangun memiliki fasilitas untuk memasukkan pertanyaan, mengeksekusi pertanyaan dengan menekan tombol *Excute* dan fasilitas untuk menampilkan hasil dari pertanyaan yang diberikan kepada sistem.

3. HASIL DAN PEMBAHASAN

3.1 *Pengujian Pemrosesan Bahasa Alami*

Pengujian sistem pemrosesan bahasa alami dilakukan untuk memahami struktur kalimat, diawali dengan tahapan memecah kalimat menjadi beberapa kata, pemberian identitas kata, pengecekan struktur kalimat, pengecekan kata di dalam konstituen ontologi dan memberikan pertanyaan kepada sistem dengan pola yang sudah ditentukan seperti pada Tabel 1.

Tabel 1 Pola Pertanyaan

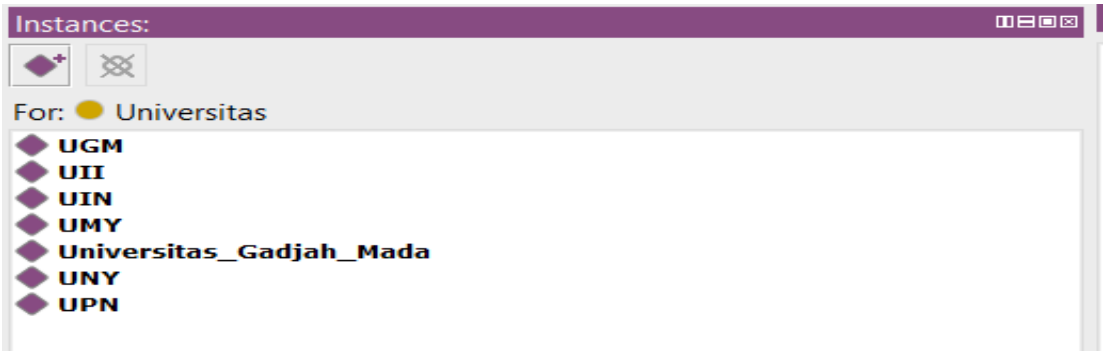
NO	Pertanyaan	Pola Kalimat
r-1	Siapa rektor UGM ?	Predikat-Subjek
r-2	Siapakah rektor Universitas Gadjah Mada ?	Predikat-Subjek
r-3	Siapakah rektor Universitas_Gadjah_Mada ?	Predikat-Subjek
r-4	Rektor UGM Siapakah ?	Subjek- Predikat
r-5	Rektor UGM ?	-

Tabel 1 menjabarkan pertanyaan-pertanyaan yang digunakan untuk pengujian sistem yang dibangun. Pertanyaan yang diberikan bersifat kalimat terbatas dengan pola kalimat predikat subjek, predikat dan predikat, subjek.

3.2 *Pengujian Mapping Ontologi*

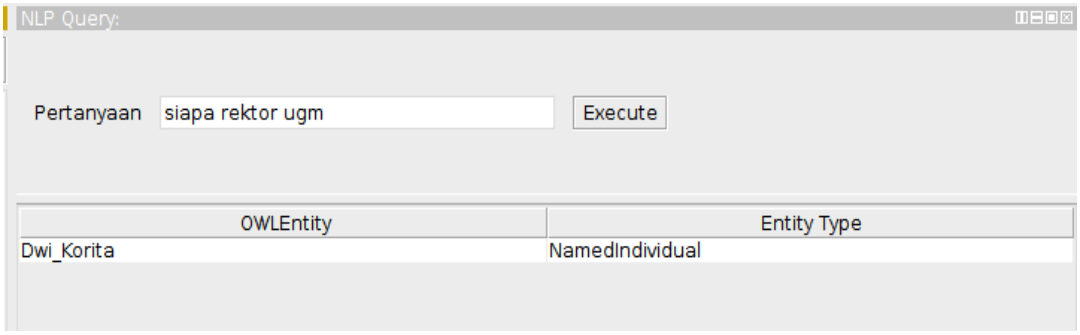
Proses selanjutnya setelah dilakukan pemrosesan bahasa alami dengan menentukan bentuk frasa di dalam kalimat adalah memeriksa konstituen-konstituen kalimat tanya ke dalam

ontologi. proses mapping bertujuan untuk mengetahui kemampuan sistem dalam melakukan pemetaan frasa terhadap objek ontologi yang terdiri dari dua kata atau lebih. Sistem harus mampu mengadaptasi pemetaan frasa dengan menggabungkan konstituen-konstituen tersebut menjadi satu kesatuan konstituen yang bermakna sama seperti pada data ontologi. pada Gambar 6, 7, 8 dan 9 diperlihatkan hasil *mapping* dengan menggunakan bentuk pertanyaan pada Tabel 1.

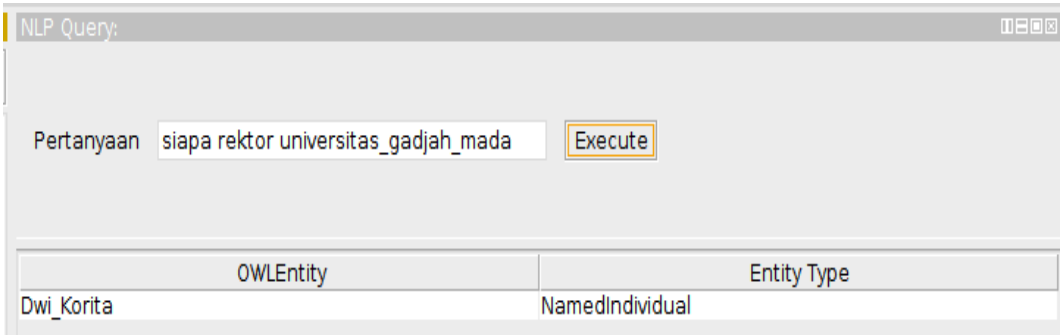


Gambar 6 Cuplikan *Instace Individual* dari kelas Universitas

Gambar 6 merupakan contoh pengujian sistem dengan bentuk pertanyaan pada r-2. Frasa "Universitas Gadjah Mada" memerlukan proses *mapping* untuk mendapatkan bentuk kata tunggal Universitas_Gadjah_Mada seperti yang terdapat pada ontologi dari data *instance class* Universitas.



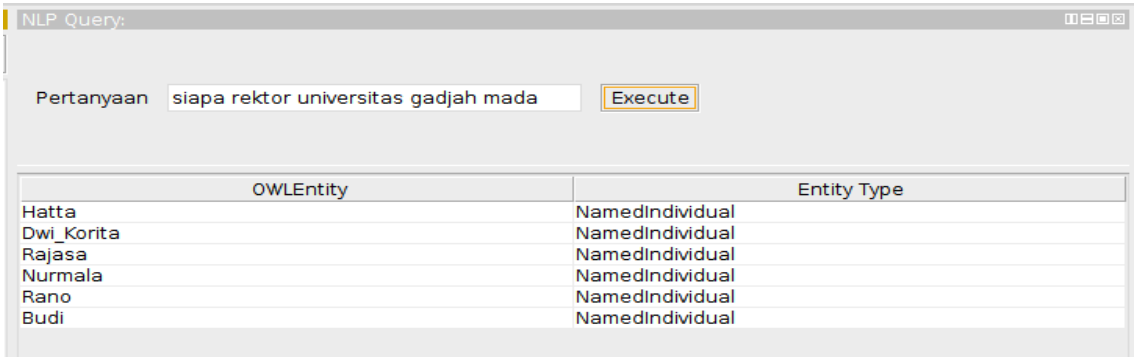
Gambar 7 Cuplikan proses *mapping* dikenali



Gambar 8 Cuplikan proses *mapping* dikenali

Pada Gambar 7 dan 8 menunjukkan proses *mapping* mempunyai pengaruh terhadap hasil pencarian yang diiginnkan, dikarenakan jika frasa "Universitas Gadjah Mada" tidak

bermakna satu kesatuan maka data yang ingin dicari di dalam ontologi tidak akan ditemukan, sehingga frasa "Universitas Gadjah Mada" harus di rubah dalam bentuk satu kesatuan kata yang berbentuk sama dengan data yang ada di dalam data ontologi. Jika proses mapping berhasil maka menghasilkan data yang diinginkan.



Gambar 9 Cuplikan proses *mapping* tidak dikenali

Gambar 9 memberikan contoh hasil jika proses *mapping* tidak berhasil maka akan menampilkan keseluruhan data *instance individual* dari *class* universitas tidak hanya *instance* yang dari frasa "universitas gadjah mada".

3.3 *Pengujian Query dengan Reasoning*

Pengujian *query* dengan *reasoning* bertujuan untuk memperlihatkan kemampuan sistem dalam melakukan pencarian informasi yang terdapat di dalam ontologi baik secara implisit dengan memanfaatkan klasifikasi *Same Individual As, Equivalent To, object property assertion*.

Sistem yang di bangun mampu menerapkan klasifikasi *Individual* dengan memanfaatkan *Same Individual As*, dimana *individual* UGM bermakna sama dengan *individual* Universitas_Gadjah_Mada sehingga ketika diberikan pertanyaan seperti pada r-1 maka *reasoner* akan memberikan jawaban yang sama seperti pertanyaan r-3. Pemanfaatkan *Same As Individual* akan menganggap UGM dan Universitas_Gadjah_Mada sebagai data yang sama. Pada Gambar 11 mengilustrasikan penggunnaan *Same Individual As*.

Sistem yang dibangun juga mampu memahami *property Equivalent To* dimana *reasoner* mampu melakukan klasifikasi terhadap data *individual* berdasarkan kriteria *restriction* yang dimiliki oleh sebuah kelas. dimana kelas rektor tidak mempunyai hubungan langsung dengan data *individual*. Sedangkan *property object assertion* digunakan untuk membangun hubungan antara *individual* dengan *individual* dengan memanfaatkan relasi *object property*

4. KESIMPULAN

Dari uraian keseluruhan yang telah dikemukan pada bab-bab sebelumnya, maka dapat diambil kesimpulan yaitu :

1. Sistem *Plugin* SPARQL-DL pada Protégé berhasil dikembangkan
2. Sistem hanya mampu memberikan jawaban dari gabungan ekpresi class, object property dan instance dari data individual.
3. Sistem mampu memberikan jawaban dari gabungan ekspresi *Class*, *Property* untuk mendapatkan *Instance* data *Individual*.
4. Sistem mampu menemukan jawaban dari sumber ontologi universitas yang telah dibangun.

5. SARAN

Saran yang diusulkan untuk peneliti yang berkeinginan mengembangkan plugin untuk protégé yaitu :

1. Kesalahan dalam penulisan kata tanya menyebabkan sistem gagal dalam memahami pertanyaan yang diberikan sehingga diperlukan sebuah sistem yang bisa mengatasi masalah tersebut.
2. Kurangnya data di dalam data lexicon khususnya bahasa Indonesia yang dipakai pada penelitian ini menyebabkan kegagagalan sistem dalam membentuk fungsi sintaksis frasa-frasa. Sehingga diperlukan penambahan untuk data *lexicon*.
3. Sistem plugin yang dibangun belum bisa memberikan hasil dari *datatype property* yang dibutuhkan oleh data individual sehingga perlu ditambahkan metode lain yang bisa mengakses *datatype property.*

DAFTAR PUSTAKA

[1] Yu. L., 2011, *A Developer's Guide to the Semantic Web*, Springer, New York, USA.

[2] Andri, 2011, Rancang Bangun Online Public Acces Catalog (OPAC) Berbasis Web Semantic, *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[3] Riswanto, E., 2012, Rancang Bangun Model Semantic Search Dengan Metode Rule Based Sebagai Aplikasi Web Semantic (Studi Kasus Pada Informasi Musik), *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[4] Haryawan, C., 2014, Pemanfaatan Sparql inferencing notation(SPIN) dalam prototipe pencarian semantik pada data restoran, *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[5] Admojo, F.T., 2015, Sistem pencarian Informasi berbasis ontologi untuk pendakian gunung menggunakan query bahasa alami dengan penyajian peta interaktif, *Tesis*, Program Pascasarjana S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[6] Muttaqin, S., 2016, System Question Answering Data Kabupaten Di Nusa Tenggara Barat Berbasis Multi-Ontologi, *Tesis*, Program Pascasarjana S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[7] Azhari dan Sholichah, M., 2006, Model Ontologi untuk Informasi Jadwal Penerbangan Menggunakan Protege, *Jurnal Informatika*, No.1, Vol.7, Hal.J67-J76.

[8] Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A., 2004, The Prot´eg´e OWL Plugin: An Open Development Environment for Semantic Web Applications, *Stanford Medical Informatics*, Stanford School of Medicine 251 Campus Drive, Stanford, CA 94305-5479.

[9] Kolengsusu, H., 2012, Rancang Bangun Plugin Untuk Sistem Informasi Akademik Dengan Ajax Dan Web Service, *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta

[10] Sirin, E dan Parsia, B., 2007, SPARQL-DL: SPARQL Query for OWL-DL, OWL: Experiences and Directions, Innsbruck, Austria : http://derivo.de/en/resources/sparql-dl-api/.pdf diakses Tanggal 20 Mei 2016.