

PENGEMBANGAN APLIKASI KONVERSI OTOMATIS DATA JSON KE OWL ONTOLOGY

Yaya Rizqi Isnanda
Program Studi Teknik Informatika
FTI Universitas Islam Indonesia
Kampus Terpadu UII, Jl.Kaliurang
Km 14.5, Sleman, Yogyakarta
14523131@students.uui.ac.id

Dhomas Hatta Fudholi S.T., M.Eng., Ph.D
Program Studi Teknik Informatika
FTI Universitas Islam Indonesia
Kampus Terpadu UII, Jl.Kaliurang
Km 14.5, Sleman, Yogyakarta
Hatta.fudholi@uui.ac.id

Abstraksi - JSON merupakan standar format dalam pertukaran data. *API (Application programming interface)* merupakan salah satu cara pertukaran data yang menggunakan format penulisan *JSON*. Salah satu keunggulan data *JSON* adalah ringan digunakan, dikarenakan penulisan format *JSON* yang sederhana. Sehingga ukurannya lebih kecil dibanding *XML*. Selain itu, keunggulan dari *JSON* yang lain adalah dapat digunakan dalam berbagai platform, sehingga dalam penggunaannya *JSON* merupakan format pertukaran data yang efisien. Format pertukaran data *JSON* sangat banyak digunakan untuk saat ini. *OWL ontology* dapat digunakan untuk membangun semantic web atau web semantik. Dalam membuat web semantik salah satu komponen terpenting adalah *ontology/basis pengetahuan*. Karena yang membedakan web semantik dan web biasa adalah adanya basis pengetahuan pada web semantik yang dapat membantu web untuk lebih memahami kebutuhan user. Namun untuk saat ini ketersediaan *ontology* masih sangat sedikit. Sehingga developer web samantik akan lebih mengeluarkan effort dalam membuat *ontology* untuk sistemnya. Padahal untuk dapat membuat *ontology* sendiri tidaklah mudah. Perlu serangkaian proses yang tidak setiap developer punya waktu atau bahkan mengerti bagaimana cara membuat *ontology*. Untuk itulah alasan mendasar dari penelitian terhadap pengembangan aplikasi konversi otomatis data *JSON* ke *OWL Ontology*. Dalam penelitian ini penulis *mencoba* membuat aplikasi yang dapat memanfaatkan data *JSON* untuk dapat dikonverikan menjadi *OWL Ontology*. Untuk membuat aplikasi ini terlebih dahulu penulis harus bisa mendapatkan *JSON Schema* dari data *JSON* yang akan dikonversikan. Lalu penulis harus melakukan analisa pada *JSON Schema* tersebut. Sehingga setelah penulis dapat menganalisa struktur *JSON Schema* dan melakukan generate kedalam list yang dibutuhkan untuk membuat *OWL Onology*, penulis dapat membuat aplikasi yang secara otomatis dapat melakukan konversi data *JSON* ke *OWL Ontology*

Kata Kunci – JSON; lits, Ontology; konversi; aplikasi; otomatis; bahasa Indonesia

1. PENDAHULUAN

JSON merupakan standar format dalam pertukaran data. *API (Application programming interface)* merupakan salah satu cara pertukaran data yang menggunakan format penulisan *JSON*. Salah satu keunggulan data *JSON* adalah ringan digunakan, dikarenakan penulisan format *JSON* yang sederhana. Sehingga ukurannya lebih kecil dibanding *XML*. Selain itu, keunggulan dari *JSON* yang lain adalah dapat digunakan dalam berbagai platform, sehingga dalam penggunaannya *JSON* merupakan format pertukaran data yang efisien. Format pertukaran data *JSON* sangat banyak digunakan untuk saat ini. Dikarenakan beberapa alasan yang telah penulis sebutkan diparagraf pertama. Mulai dari perusahaan besar seperti *google* yang menyediakan *API* dengan format *JSON* sampai perusahaan yang baru merintis atau bahkan perorangan/komunitas[1]. Selain *google* yang termasuk salah satu perusahaan besar menyediakan dan memfasilitasi *API* berupa *JSON* data, pastinya user atau pengguna *JSON* data dari *google API* pun tidak sedikit pula. Berikut merupakan gambar penggunaan data *JSON* yang semakin meningkat dan sudah mencapai hitungan yang banyak.

Historical trends in the usage of structured data formats for websites

This report shows the historical trends in the usage of structured data since August 2017.

	2017	2017	2017	2017	2017	2018	2018	2018	2018	2018	2018	2018	2018
	3-Aug	3-Sep	1-Oct	1-Nov	1-Dec	1-Jan	1-Feb	1-Mar	1-Apr	1-May	1-Jun	1-Jul	2-Aug
None	57.6%	57.0%	56.5%	56.0%	55.6%	55.1%	54.8%	54.1%	53.4%	52.9%	52.5%	52.1%	51.5%
RDFa	36.6%	37.1%	37.6%	38.0%	38.4%	38.9%	39.2%	39.8%	40.4%	40.9%	41.2%	41.6%	42.1%
Meta-tag-based formats	18.0%	18.4%	18.8%	19.2%	19.7%	20.1%	20.5%	21.1%	21.6%	22.1%	22.5%	22.9%	23.4%
JSON-LD	16.2%	16.6%	17.0%	17.3%	17.7%	18.1%	18.4%	18.9%	19.3%	19.7%	20.1%	20.4%	20.8%
Microdata	12.2%	12.4%	12.6%	12.8%	13.0%	13.1%	13.3%	13.6%	13.9%	14.2%	14.5%	14.8%	15.1%
Microformats	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%

Gambar 1. 1 Penggunaan *JSON*

OWL ontology dapat digunakan untuk membangun semantic web atau web semantik. Dalam membuat web semantik salah satu komponen terpenting adalah *ontology/basis pengetahuan*. Karena yang membedakan web semantik dan web biasa adalah adanya basis pengetahuan pada web semantik yang dapat membantu web untuk lebih memahami kebutuhan user. Salah satu bentuk *ontology* atau basis pengetahuan bisa berupa *OWL* yang digunakan oleh sistem bukan hanya menampilkan informasi namun juga memproses informasi data dan ditulis dengan format *RDFS*[2].

Salah satu yang mendasari pembuatan aplikasi konversi ini adalah pentingnya mempunyai basis pengetahuan untuk

membangun sebuah web semantik. Karena dalam proses pembuatan web semantik akan lebih mudah saat sudah mempunyai *owl ontology* (basis pengetahuan). Selain itu berikut adalah beberapa aplikasi yang berbasis web semantik. Services Mashups *The New Generation of Web Applications*[3], Haystack: *A Platform for Authoring End User Semantic Web Applications* [4], Piazza: *Data Management Infrastructure for Semantic Web Applications*[5] dan masih banyak lainnya.

Berikut adalah gambaran proses yang dilakukan dalam merancang sebuah ontology:

- Tentukan domain dan ruang lingkup *ontology*
- Pertimbangkan untuk menggunakan kembali *ontology* yang ada
- Menghitung istilah penting dalam *ontology*
- Tentukan kelas dan hierarki kelas
- Tentukan sifat kelas-slot
- Tentukan sisi slotnya
- Buat *instance* [6]

1

Seperti yang sudah penulis jelaskan diatas untuk membuat *OWL ontology* bukan hal yang mudah. Karena yang membedakan ontology dengan format data lain adalah keunggulan dimana ada relasi atau sebuah pengetahuan yang dapat dipakai oleh sistem dan itu tidak dimiliki format data lain. Beberapa sumber sudah menyediakan ontologi yang bisa langsung digunakan, tapi untuk penggunaan yang spesifik masih sangat kurang jika hanya mengandalkan ontologi yang sudah ada (Miličić, 2014). Oleh karena itu salah satu solusi yang penulis tawarkan adalah sebuah aplikasi konversi format data *JSON* kedalam *OWL ontology*, penulis menggunakan *JSON* karena yang sedang populer saat ini dan dari semua format data *JSON* memiliki keunggulan dibanding format data lain..

2. LANDASAN TEORI

2.1. JSON

JSON merupakan format pertukaran data. Dengan menggunakan *API (Application programming interface)* data dapat dipertukarkan atau digunakan dengan format penulisan *JSON*. *JSON* terbuat dari dua struktur, yaitu:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (hash table), daftar berkunci (*keyed list*), atau associative array.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

2.2.1 Alasan Menggunakan Data JSON

Berikut merupakan alasan penulis mengembangkan aplikasi konversi otomatis ini menggunakan data *JSON*:

- *JSON* merupakan format pertukaran data yang penulisanya sederhana sehingga mudah dibaca dan ditulis oleh manusia
- Karena penulisanya yang sederhana *JSON* juga memiliki keunggulan ringan dalam pertukaran data
- Support untuk *JSON* sendiri sangat besar karena hampir kebanyakan pertukaran data sekarang menggunakan format *JSON* mulai dari perusahaan besar seperti *google* atau bahkan perusahaan kecil dan perorangan.
- *JSON* sendiri mendukung berbagai platform sehingga dari segi penggunaan sangat efisien.

2.1.2 JSON Schema

Dalam penulisan data *JSON* membutuhkan sebuah schema penulisan dimana akan mengatur data *JSON* tersebut agar sesuai dengan standar dari *JSON schema*. Selain itu dalam penelitian ini penulis menggunakan *JSON schema* untuk melakukan analisis data *JSON* yang dimasukkan. Sehingga penulis akan mendapatkan properties dari *JSON schema* yang sudah ada, kemudian dari hasil parsing akan dilakukan pemetakan untuk mendapatkan list yang dibutuhkan dalam membentuk file *owl ontology*.

2.2 OWL Ontology

OWL ontology untuk membangun semantic web atau web semantik. Dalam membuat web semantik salah satu komponen terpenting adalah ontology/basis pengetahuan. Karena yang membedakan web semantik dan web biasa adalah adanya basis pengetahuan pada web semantik yang dapat membantu web untuk lebih memahami kebutuhan user.

Salah satu yang mendasari pembuatan aplikasi konversi ini adalah pentingnya mempunyai basis pengetahuan untuk membangun sebuah web semantik. Karena dalam proses pembuatan web semantik akan lebih mudah saat sudah mempunyai *owl ontology* (basis pengetahuan).

Berikut adalah gambaran proses yang dilakukan dalam merancang sebuah ontology:

- Tentukan domain dan ruang lingkup ontology
- Pertimbangkan untuk menggunakan kembali ontologi yang ada
- Menghitung istilah penting dalam ontology
- Tentukan kelas dan hierarki kelas
- Tentukan sifat kelas-slot
- Tentukan sisi slotnya
- Buat instance

2.3 Penelitian Terdahulu

Sistem konversi otomatis data *JSON* ke *OWL Ontology* setelah penulis dan pembimbing cari belum menemukan sistem yang serupa. Dimana memanfaatkan data *JSON* yang begitu banyak untuk mendapatkan data *OWL Ontology* yang secara struktur lebih kaya dari pada data *JSON* itu sendiri. Untuk itu juga salah satu alasan penulis dan pembimbing sepakat dalam mengembangkan sistem ini.

Untuk penelitian yang hampir serupa penulis menemukan sistem konversi data XML ke OWL Ontology. Dan juga pada disertasi pembimbing tentang penelitian Data JSON ke OWL Ontology. Pada kedua referensi penelitian terdahulu membahas tentang metode baru dalam membuat data OWL Ontology dengan cara yang lebih mudah. Karena menggunakan data dalam bentuk lain yang nantinya melakukan akan dilakukan konversi untuk mendapat data baru berupa OWL Ontology.

Selain itu kenapa penulis mengembangkan sistem konversi ini, karena penulis belum menemukan sebuah sistem atau implementasi dari penelitian terdahulu sampai berbentuk sistem yang dapat digunakan oleh pengguna. Kebanyakan dari penelitian terdahulu baru membahas sampai mana konversi dan mapping datanya belum masuk kedalam implementasi.

Tabel 2. 1 Penelitian serupa terdahulu

Pengarang	Judul	Pembahasan
Dhomas Hatta Fudholi (2016)	<i>Data - Driven Dynamic Common Ontology</i>	Analisis beberapa model data (XML, JSON, RDB, etc) dengan OWL Ontology
Nora Yahia(1), Sahar A. Mokhtar(2) and AbdelWahab Ahmed(3)	<i>Automatic Generation of OWL Ontology from XML Data Source</i>	Membahas bagaimana memperoleh data OWL Ontology dengan data XML yang ada
Hannes Bohring* and Soren Auer+	<i>Mapping XML to OWL Ontologies</i>	Pemetakan data XML kedalam OWL Ontology

3. ANALISIS

3.1. Analisis Masalah

Dalam mengembangkan web semantik *developer* menggunakan *ontology* sebagai basis pengetahuannya. Yang paling membedakan perancangan web semantik adalah adanya *ontology* dalam pengembangannya dari pada web yang tidak menggunakan *ontology*. Seperti yang sudah penulis jelaskan di bab sebelumnya untuk membuat OWL ontology bukan hal yang mudah. Karena yang membedakan *ontology* dengan format data lain adalah keunggulan dimana ada relasi atau sebuah pengetahuan yang dapat dipakai oleh sistem dan itu tidak dimiliki format data lain. Sampai saat ini ada beberapa kendala dalam membuat ontology, yaitu:

- Sulitnya membuat *ontology* dari nol, karena ontology adalah basis pengetahuan dan membutuhkan *effort* besar untuk membuat basis pengetahuan dari nol.
- Salah satu cara mengatasi untuk tidak membuat ontology adalah menggunakan ulang atau me re-use sebuah ontology yang sudah ada. Namun pada kenyataannya ontology yang sudah ada masih sangat terbatas dan kebanyakan domain dari ontology yang sudah ada tersebut sulit untuk dicari kecocokannya dengan ontology

yang di butuhkan, terutama untuk penelitian baru yang memang belum pernah sama sekali ada yang membuat ontology seperti domain yang akan digunakan.

- Begitu banyaknya data JSON dalam penggunaannya juga dapat dijadikan alasan atau masalah kenapa tidak membuat sebuah analisis pemetakan untuk membuat data JSON tersebut menjadi sebuah data yang kaya dan mungkin lebih berguna (dalam lingkup semantic web) .

3.2. Usulan Pemecahan Masalah

Dengan permasalahan yang sedang dihadapi oleh developer web semantik maka pemecahan masalah yaitu dengan merancang dan mengembangkan sebuah aplikasi yang mampu untuk menampung data JSON kemudian mengubahnya menjadi sebuah data owl ontology. Dimana dengan demikian akan lebih mudah bagi *developer* untuk mengembangkan web semantik yang mereka rancang.

Sebuah aplikasi yang mampu membaca data JSON sehingga dapat melakukan analisis pada setiap *properties* data tersebut. Kemudian melakukan mapping / pemetakan dari hasil analisis dan parsing data JSON, sehingga didapatkan sebuah list objek dan *properties* lainnya. Dan kemudian melakukan penyusunan file baru dengan format OWL ontology. Sehingga didapatkan sebuah basis pengetahuan yang dapat dipakai dalam dasar pengembangan web semantik yang akan dibuat.

3.3 Analisis Kebutuhan Informasi

Analisis kebutuhan informasi data dilakukan untuk mengetahui informasi – informasi penting dari data JSON yang akan dikonversi sehingga sistem tahu data mana yang akan dilakukan pemetakan. Setiap informasi data dapat dilihat dari JSON schema yang dilakukan parsing dan pengolahan datanya kedalam bentuk list yang sudah dipisahkan berdasarkan beberapa karakter yang telah penulis tentukan sehingga data bisa langsung digunakan untuk membentuk data owl ontology. Berikut merupakan rincian kebutuhan informasi yang diperlukan oleh sistem

- Data JSOS
Values dari data JSON sangat dibutuhkan dalam pembantuan data owl ontology baru nantinya.
- Data JSON schema
 - Objek dalam JSON
 - Type data dari setiap *properties*
 - Jumlah hirarki kedalaman data JSON
 - Setiap *properties* di semua hirarki
 - Informasi kepemilikan anak disetiap hirarki

3.4 Analisis Kebutuhan Masukan

Analisis kebutuhan masukan dilakukan untuk mengetahui masukan-masukan apa saja yang diperlukan di dalam sistem. Kebutuhan masukan yang digunakan oleh pengguna di dalam sistem yaitu data masukan berupa username dan *password* untuk login, data JSON untuk melakukan konversi ke dalam data owl ontology

3.5 Analisis Kebutuhan Proses

Analisis kebutuhan proses merupakan analisis untuk mengetahui proses-proses apa saja yang diperlukan di dalam sistem. Berikut merupakan perincian kebutuhan proses yang terdapat di dalam sistem:

- Proses *login*
Sistem ini tidak hanya digunakan oleh satu pengguna. Sehingga dibutuhkan proses *login* untuk mengenali siapa yang akan menggunakan sistem dan hal apa saja yang dapat dilakukan pengguna tersebut di dalam sistem.
- Proses input data JSON
Sistem akan mengambil data JSON dari storage sistem dan menampungnya untuk kemudian dilakukan proses selanjutnya.
- Proses konversi data JSON ke owl ontology
Sistem akan melakukan banyak proses, mulai dari parsing, mapping, dan semua proses yang dibutuhkan sedemikian rupa lalu menghasilkan data OWL ontology
- Proses cek history dan bookmark file hasil konversi
Sistem akan menampung hasil konversi dan akan melakukan cek ulang untuk melihat kembali data yang sudah dilakukan konversi.

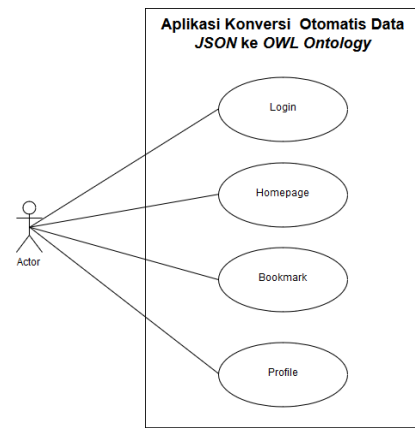
3.6 Analisis Kebutuhan Keluaran

Analisis kebutuhan keluaran dilakukan untuk mengetahui keluaran apa saja yang diperlukan di dalam sistem. Berikut merupakan perincian kebutuhan keluaran dari sistem untuk konversi data JSON ke dalam data OWL ontology:

- Data OWL ontology hasil dari konversi data JSON.
- Detail history dari penggunaan sistem, data yang sudah terkonsversi dan tersimpan sesuai akun login.

3.7 Use Case Diagram

Use case diagram merupakan suatu model untuk memberikan gambaran sistem secara keseluruhan. Diagram ini menggambarkan beberapa atau semua actor dan interaksi-interaksi yang terjadi dengan system untuk mengenalkan system yang dibuat. Adanya use case ini dapat memberikan informasi mengenai fungsi-fungsi apa saja yang terdapat pada sistem yang dibuat. Use case diagram yang terlihat pada gambar 3.1 merupakan rancangan untuk sistem yang akan dibuat.



Gambar 3. 1 Diagram activity sistem

Dalam use case tersebut terlihat bahwa hanya terdapat satu actor dalam sistem ini yaitu pengguna, yang berperan sebagai developer. Pengguna sistem dapat melakukan akses homepage, bookmark, dan profile. Untuk dapat menggunakan fungsi bookmark dan profile pengguna harus melakukan proses logi terlebih dahulu. Berbeda dengan halaman homepage dimana pengguna tidak harus melakukan login terlebih dulu karena pada halaman ini pengguna dapat melakukan konversi data JSON ke OWL Ontology.

3.8 Memetakan JSON ke OWL Ontology

Dalam melakukan mapping / pemetaan membutuhkan sebuah aturan. Disini menggunakan aturan "kesamaan karakter" dan "kesetaraan". Contoh:

Di dalam ontology ada Class, dan class itu mengumpulkan class lain di bawah nya atau bisa juga instance dari class tersebut. sedangkan didalam JSON memiliki object untuk membungkus object lain yang memiliki karakteristik serupa juga.

Dari aturan yang telah ditentukan untuk pemetaan data JSON ke OWL Ontology didapatkan table konversi sebagai berikut:

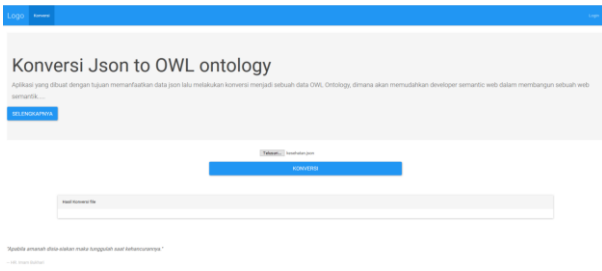
No	Data JSON	OWL Ontology
1	Object yang memiliki karakteristik Umum	owl:Class
2	Relasi Object di antara object di atasnya	rdfs: subClassOf
3	Relasi asosiasi antar atribut di kelas tersebut	owl:objectProperty
4	Atribut yang mempunyai tipe data dan literal	owl:DatatypeProperty
5	Tipe data atribut	rdf:datatype

4. HASIL

4.1. Halaman Utama

Halaman utama atau home merupakan satu satunya halaman yang akan dijumpai oleh pengguna. Dimana pada halaman ini pengguna dapat menginputkan file JSON yang nantinya akan diolah oleh sistem sehingga mendapatkan file

OWL Ontology. Hasil halaman utama dapat dilihat pada gambar 4.1.



Gambar 4. 1 Halaman utama sistem

4.2. Perbandingan File JSON dan OWL Ontology

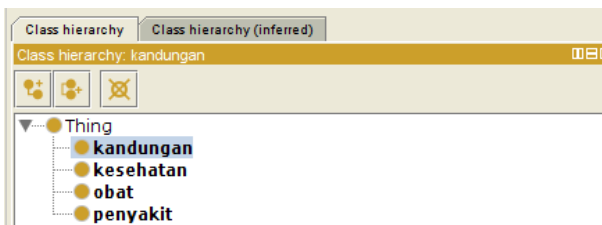
Pada pengujian sistem kali ini. Penulis menggunakan software yang bernama protégé. Dimana software tersebut dapat melakukan operasi dan membaca file OWL Ontology. Kemudian penulis akan membandingkan dan menunjukkan hasil yang ditampilkan di protégé. Apakah list yang dibuat sudah ter-generate menjadi file ontology. Berikut adalah pengujian sistem konversi data JSON kedalam OWL Ontology

```

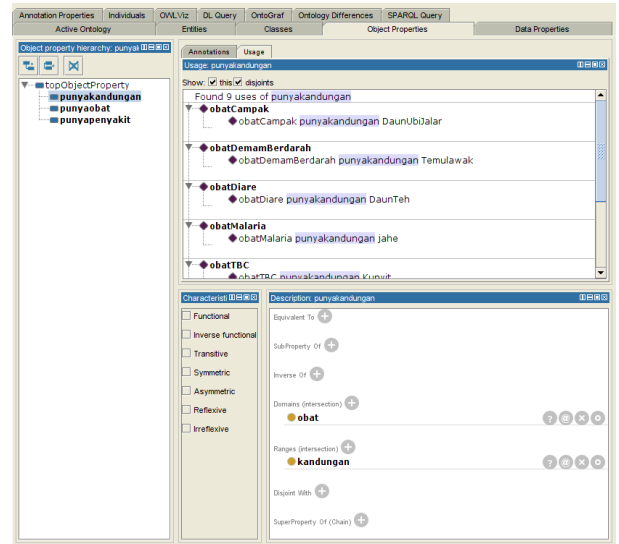
1 {
2   "kesehatan": {
3     "penyakit": [
4       {
5         "idP": 101,
6         "namaP": "Malaria",
7         "obat": {
8           "namaO": "obatMalaria",
9           "kandungan": {
10            "tumbuhan": "jaje",
11            "senyawa": "Hcl"
12          }
13        }
14      },
15      { ... },
16      { ... },
17      { ... },
18      { ... },
19      { ... }
20    ]
21  }
22 }

```

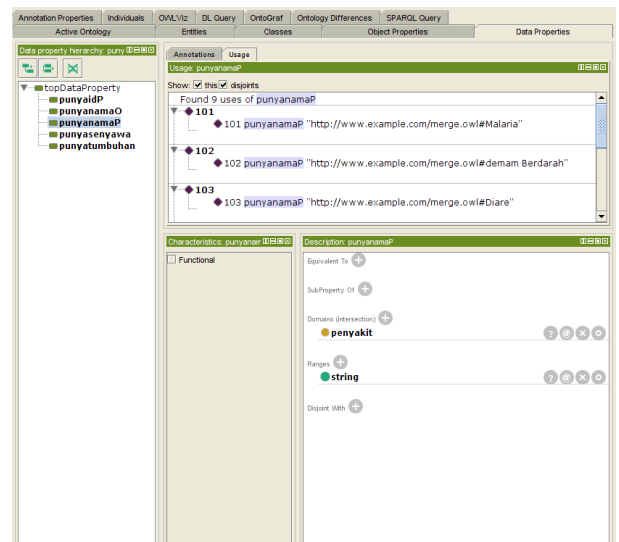
Gambar 4. 2 File JSON yang akan dikonversi



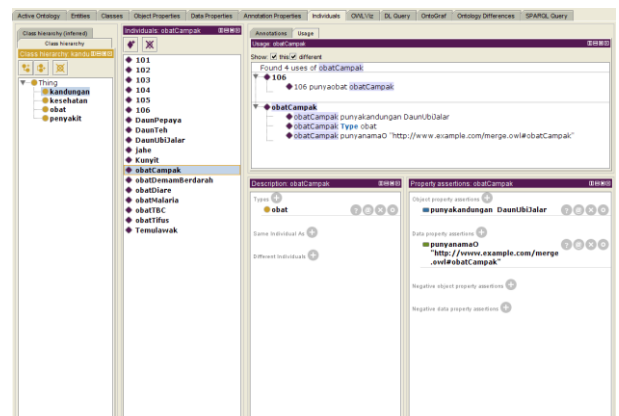
Gambar 4. 3 Protege hasil konfersi kelas



Gambar 4. 4 protege hasil konversi object property



Gambar 4. 5 Protege hasil konversi datatype property



Gambar 4. 6 protege hasil konversi individu

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan perancangan dan juga pengujian yang sudah dilakukan pada penelitian ini maka, didapatkan kesimpulan bahwa penelitian ini yaitu:

- Aplikasi konversi otomatis data JSON ke OWL Ontology dapat melakukan konversi data JSON secara otomatis. Dan juga tanpa perlu mengenerate JSON Schema-nya secara terpisah.
- Aplikasi ini menghasilkan file OWL Ontology sesuai dengan data JSON yang dimasukan. Karna melakukan generate dan menyusun list yang digunakan untuk membentuk OWL Ontology dari file itu sendiri secara otomatis
- Diharapkan dengan adanya pengembangan aplikasi ini developer atau software engineer dapat lebih mudah dalam melakukan pengembangan aplikasi atau software berbasis OWL Ontology

5.2. Saran

Setelah melakukan pengembangan terhadap sistem ini . penulis menyadari bahwa masih banyak kekurangan terhadap sistem yang telah dibuat ini. maka diharapkan pada penelitian selanjutnya dapat menghilangkan atau menutupi kekurangan yang ada pada sistem ini. Bahkan sangat diharapkan penelitian selanjutnya dapat mengembangkan sistem ini lebih luas lagi. Saran untuk pengembangan sistem pada penelitian selanjutnya adalah sebagai berikut:

- Penambahan NLP untuk lebih akuratnya dalam menentukan relasi setiap data hasil generate yang nantinya akan dimasukan kedalam file OWL Ontology
- Support file data selain JSON dengan generate Otomatis.
- Lebih dapat mengenali atau menganalisa list data yang masuk sehingga hasil yang diperoleh lebih relevan lagi

REFERENCES

- [1] The Official Buzz from Blogger at Google, "Blogger JSON API: Getting Started | Blogger | Google Developers," *June 6, 2017.*, 2017. [Online]. Available: https://developers.google.com/blogger/docs/2.0/json/getting_started. [Accessed: 07-Mar-2018].
- [2] I. Horrocks, P. F. Patel-schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, "SWRL : A Semantic Web Rule Language Combining OWL and RuleML," *W3C Memb. Submiss. 21*, no. May 2004, pp. 1–20, 2004.
- [3] D. Benslimane, S. Dustdar, and A. Sheth, "Services Mashups Service Mashups," 2008.
- [4] D. Quan, D. Huynh, and D. R. Karger, "LNCS 2870 - Haystack: A Platform for Authoring End User Semantic Web Applications," *LNCS*, vol. 2870, pp. 738–753, 2003.
- [5] A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov, "Piazza," *Proc. twelfth Int. Conf. World Wide Web - WWW '03*, vol. 32, p. 556, 2003.
- [6] Natalya F. Noy and Deborah L. McGuinness, "What is an ontology and why we need it," 2000. [Online].