

Analisa Keterhubungan *Ontology* Pada Web Semantik Menggunakan
Terminological-Based Ontology Matching

Fithratul Sauwa¹, Arie Ardiyanti Suryani², Toto Suharto³

Fakultas Informatika Universitas Telkom, Bandung

¹fithratulsauwa@gmail.com, ²rie006@yahoo.com, ³tsuharto@gmail.com

Abstrak

Web semantik memungkinkan data tidak hanya dapat dimengerti oleh manusia sebagai pembaca tetapi juga agar bisa diproses dan dimengerti oleh mesin atau komputer. *Ontology* merupakan teknologi pada web semantik yang memungkinkan hal tersebut dapat terjadi. *Ontology* mendeskripsikan data pada web dan keterhubungan antar data pada web. Heterogenitas merupakan masalah yang paling umum terjadi pada *ontology* di web semantik, misalnya terdapat dua *ontology* dengan nama yang berbeda, *ontology* tersebut memiliki struktur yang berbeda atau didefinisikan dengan cara yang berbeda padahal kedua *ontology* tersebut mendeskripsikan domain pengetahuan yang sama. *Ontology matching* bertujuan untuk mengurangi masalah heterogenitas pada *ontology*. *Ontology matching* merupakan proses untuk membandingkan dua *ontology* dan menemukan keterhubungan diantara kedua *ontology* tersebut. Salah satu teknik yang digunakan pada *ontology matching* untuk menyelesaikan masalah heterogenitas adalah *Terminological-based ontology matching* (TBOM). Teknik *terminological-based ontology matching* ini menggunakan data leksikal dari konsep yang terdapat pada *ontology* untuk mencocokkan konsep dengan cara membandingkan string (*string comparison*) sehingga proses *ontology matching*. Teknik *Terminological-based ontology matching* yang digunakan dalam tugas akhir ini dipengaruhi oleh parameter *Similarity Threshold* yang berperan pada proses *terminological-based ontology matching*. Parameter *similarity threshold* ini memberikan pengaruh yang besar terhadap performansi proses *ontology matching*. Semakin besar masukan nilai *similarity threshold* maka semakin baik nilai performansi yang dihasilkan atau dapat dikatakan nilai performansi mendekati satu.

Kata Kunci : web semantik, *ontology*, *ontology matching*, heterogenitas, *Terminological-based ontology matching*

Abstract

The semantic web not only allows data to be understood by humans as a reader but also in order to be processed and understood by machines or computers. Ontology is a semantic web technology that allows it to happen. Ontology describes data on the web and the connectivity between the data on the web. Heterogeneity is a problem that is commonly happened in semantic web ontology, for example there are two ontologies with different names, different structure or defined in different ways although both of them describe the same knowledge. . Ontology matching aims to reduce the problem of heterogeneity in ontology. Ontology matching is a process to compare two ontologies and find the connectivity between two ontology. One of techniques that is used in ontology matching to solve the problem of heterogeneity is by Terminological-based ontology matching (TBOM). Technique of terminological-based ontology matching uses the data from the lexical concepts which is included in the ontology to match the concept by comparing the string (string comparison) so that the ontology matching process produces accurate results in determining the connectivity of two ontology. The TBOM techniques used in this final project is influenced by Similarity Threshold parameter which contributes in terminological-based ontology matching. The similarity threshold parameter gives a great influence on the performance of ontology matching process. The greater the similarity threshold input value, the better the performance of the resulting value or the value of performance approaching one.

Key words : semantic web, *ontology*, *ontology matching*, heterogeneity, *Terminological-based ontology matching*

1. Pendahuluan

Perkembangan web dari tahun ke tahun menunjukkan peningkatan yang sangat signifikan. Pada saat ini aplikasi non web sudah banyak ditinggalkan. Web 3.0 atau biasa disebut dengan *Semantic Web* menjadi pembuktian bahwa perkembangan web saat ini sudah sangat pesat. Munculnya teknologi *Semantic Web* ini dilatarbelakangi oleh dua hal, yaitu : integrasi data dan perangkat pendukung yang lebih cerdas untuk user^[1]. *Semantic Web* memungkinkan data tidak hanya ditujukan atau dapat dimengerti oleh manusia sebagai pembaca tetapi juga agar bisa diproses dan dimengerti oleh mesin atau komputer.

Salah satu perkembangan teknologi yang digunakan pada *semantic web* adalah *ontology*. *Ontology* biasanya menyediakan kosakata yang menggambarkan domain tertentu dan spesifikasi istilah makna dari kosakata tersebut^[2]. *Ontology* juga mendeskripsikan data pada web dan keterhubungannya. Dengan semakin berkembangnya teknologi *semantic web*, maka akan semakin banyak pula jumlah *ontology* yang tersedia di internet. Hal ini akan menimbulkan permasalahan heterogenitas. Beberapa permasalahan heterogenitas yang timbul pada *ontology* salah satunya seperti penggunaan istilah yang berbeda untuk suatu hal yang sama pada suatu *ontology*, misalnya *surname* dan *family name*^[6]. Cara yang digunakan dalam mengurangi permasalahan yang timbul karena heterogenitas ini bisa diatasi dengan *Ontology Matching*. *Ontology matching* ini membandingkan dua buah *ontology* dan menemukan keterhubungan diantara kedua *ontology* tersebut. Ada beberapa teknik *ontology matching* yang digunakan dalam menyelesaikan permasalahan heterogenitas diantaranya adalah *terminological-based techniques*, *structure-based techniques*, *instance-based techniques* dan *semantic-based techniques*.

Terminological-based techniques merupakan salah satu teknik *ontology matching* yang akan digunakan dalam menyelesaikan permasalahan tugas akhir ini. Teknik *terminology* ini menggunakan data leksikal dari konsep yang terdapat pada *ontology* untuk mencocokkan konsep dengan cara membandingkan string (*string comparison*). Pada umumnya digunakan metode ini karena kemampuannya untuk masuk kedalam aplikasi dan domain yang berbeda tidak diperlukannya syarat yaitu masukan tambahan (*additional inputs*) dan *instances*^[8].

Perumusan masalah yang akan dibahas pada tugas akhir ini adalah

- Bagaimana menyelesaikan masalah heterogenitas dengan menggunakan metode *terminological-based ontology matching* pada *semantic web*

Dari perumusan diatas dibuat tujuan dari tugas akhir ini :

- Menerapkan metode *terminological-based ontology matching* untuk menyelesaikan masalah heterogenitas yang terjadi pada *semantic web*
- Melakukan analisa keterhubungan *ontology* pada *semantic web* berdasarkan parameter *Similarity Threshold* yang mempengaruhi nilai performansi *recall*, *precision* dan *f-measure* dengan menggunakan metode *terminological-based ontology matching*.

Metode penelitian yang digunakan adalah dengan cara studi literatur, yaitu mencari dan mempelajari materi yang berkaitan dengan permasalahan. Selanjutnya dilakukan perancangan sistem serta analisis. Dibuat juga dataset untuk pengujian sistem. Kemudian mengimplementasi sistem yang telah dirancang dan melakukan pengujian sistem. Lalu dilakukan pengujian untuk skenario pengujian sehingga mendapatkan hasil yang menunjukkan performansi sistem lalu ditarik kesimpulan.

2. Dasar Teori

2.1 Ontology

Salah satu teknologi pada web semantik yang memiliki cara baru untuk mendefinisikan dan menyimpan pengetahuan biasa dikenal dengan *ontology*. Ada beberapa peneliti mendefinisikan *ontology* secara berbeda-beda. Menurut Gruber pada tahun 1993 menyebutkan bahwa *ontology* adalah sebuah spesifikasi secara eksplisit dari konseptualisasi^[4]. Peneliti lain seperti Neches dan rekannya menyebutkan bahwa *ontology* adalah istilah-istilah dasar dan relasi yang terdiri dari kosakata area suatu topic serta aturan untuk menggabungkan istilah dan relasi untuk mendefinisikan perluasan kosakata^[4]. Lalu ada Studer dan rekannya menyebutkan bahwa *ontology* adalah spesifikasi formal eksplisit dari konseptualisasi bersama. Konseptualisasi ini mengacu pada model abstrak dari beberapa fenomena di dunia setelah mengidentifikasi kelas

yang relevan dari fenomena tersebut. Eksplisit berarti bahwa tipe dari kelas yang digunakan dan batasan yang digunakan didefinisikan secara jelas. Formal mengacu pada fakta bahwa *ontology* harus mudah dibaca mesin. Bersama mengacu pada ide bahwa *ontology* menangkap pengetahuan yang berhubungan yakni tidak bersifat pribadi dari individu tertentu akan tetapi dapat diterima oleh kelompok^[4].

Ada 4 komponen utama *ontology* yaitu kelas atau sering juga disebut dengan konsep yang merupakan kelompok abstrak kumpulan atau koleksi dari objek-objek. Selanjutnya ada *instance* atau sering disebut juga dengan istilah *individual* yang merupakan level terbawah dari komponen *ontology* yang menggambarkan objek spesifik dari elemen kelas. Lalu ada relasi digunakan untuk menyatakan hubungan diantara dua buah kelas dalam domain tertentu. Selanjutnya yang terakhir ada aksioma digunakan untuk menentukan batasan atau kondisi nilai dari suatu kelas atau *instance* sehingga aksioma umumnya dinyatakan dengan bahasa logis seperti *first-order logic* yang digunakan untuk memverifikasi kekonsistenan *ontology*^[3]. Kemudian ada 3 bahasa *ontology* yaitu RDF (*Resource Description Framework*), OWL (*Ontology Web Language*) dan OWL FULL.

2.2 Ontology Matching

Ontology Matching merupakan solusi untuk permasalahan heterogenitas *ontology* pada semantik web yang merupakan proses untuk menemukan hubungan diantara *ontology*. Terdapat 4 teknik dasar yang digunakan dalam proses *matching* dalam dua buah *ontology*, yaitu^[2]:

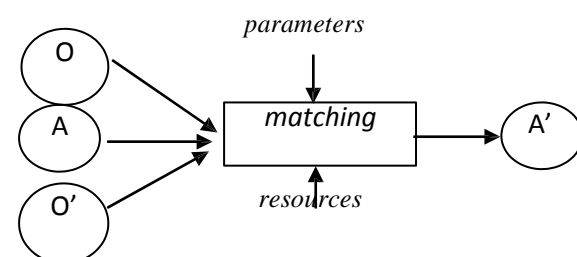
- Terminological techniques* (*name-based techniques*)
- Structure-based techniques*
- Instance-based techniques* (*extensional techniques*)
- Semantic-based techniques*

Teknik ini dapat digunakan untuk menyelesaikan permasalahan heterogenitas. Ada beberapa tipe dari heterogenitas pada *ontology*, diantaranya adalah^[6]:

- Syntactic heterogeneity*, heterogenitas ini disebabkan oleh perbedaan format representasi *ontology* misalnya penggunaan bahasa pendefinisian yang berbeda seperti RDF dan OWL.

- Terminological heterogeneity*, heterogenitas ini menunjukkan penggunaan istilah yang berbeda untuk suatu hal yang sama pada suatu *ontology*, misalnya *surname* dan *family name*.
- Conceptual heterogeneity*, heterogenitas ini dapat dibagi menjadi 3 bagian yaitu :
 - *Coverage difference*, masalah ini terjadi jika *ontology* ditulis dari sudut pandang yang sama.
 - *Difference in granularity*, masalah ini terjadi ketika memiliki bagian yang sama dari domain akan tetapi kedalaman detailnya tidak sama.
 - *Difference in perspective*, masalah ini muncul dilihat dari sudut pandang perancangan *ontology* yang berbeda.
- Semiotic heterogeneity*, heterogenitas ini disebabkan oleh penafsiran subjektif terhadap istilah yang digunakan oleh manusia.

Proses *ontology matching* dapat dilihat sebagai suatu fungsi yang mencocokkan dua buah *ontology* o dan o' , sebuah inputan *alignment* A , sekumpulan parameter p , sekumpulan *resources* r dan menghasilkan *alignment* A' diantara dua buah *ontology* tadi yang disebutkan : $A' = f(o, o', A, p, r)$ atau dapat dilihat dari gambar 2.2



Gambar 2-2 : Proses Ontology Matching
[2]

Gambar diatas mendeskripsikan bahwa proses *ontology matching* ini menentukan *alignment* A' dari sepasang *ontology* o dan o' yang terdiri dari sekumpulan keterhubungan atau *mapping* dari dua buah *ontology* tersebut dan *alignment* A' ini dipengaruhi oleh parameter dan *resource*.

Ada beberapa manfaat dari pengaplikasian *ontology matching* ^[2]:

- Ontology evolution* menggunakan *ontology matching* untuk menemukan perubahan yang terjadi diantara dua buah *ontology*.
- Data integration*, menggabungkan dua data yang berbeda menjadi satu.
- P2P information sharing*, menggunakan *ontology matching* untuk menemukan hubungan diantara *ontology* yang digunakan oleh *peer* yang berbeda.
- Multiagent communication* menggunakan *ontology matching* untuk menemukan hubungan diantara *ontology* yang digunakan oleh dua *agent* dan menterjemahkan pesan yang mereka tukar.
- Query answering* menggunakan *ontology matching* untuk menterjemahkan *query* user di web.
- Semantic web browsing* menggunakan *ontology matching* pada saat melakukan *browsing* halaman web dengan memanfaatkan *ontology* yang mempunyai keterhubungan.

2.3 Terminological-based Ontology Matching

Teknik *terminological-based ontology matching* (TBOM) ini menggunakan data leksikal dari konsep yang terdapat pada *ontology* untuk mencocokkan konsep dengan cara membandingkan string (*string comparison*). Metode ini dapat diterapkan pada nama, label ataupun *comments* dari entitas agar dapat ditemukannya kemiripan dan dapat digunakan untuk membandingkan nama kelas dan/atau URI ^[2].

TBOM ini juga biasa dikenal dengan *Schema-based*, yang pada umumnya digunakan metode ini karena kemampuannya untuk masuk kedalam domain yang berbeda sehingga tidak diperlukannya masukan tambahan dan *instances*. *Schema-based* ini dicapai dengan menggunakan dua pendekatan utama, yaitu pendekatan leksikal dan pendekatan structural. Pendekatan structural dilihat dari unsur-unsur pertukaran *relationships*, *affiliation* dan *position* didalam struktur. Sedangkan pendekatan leksikal sesuai dengan unsur-unsur yang didasarkan dari sifat *string* mereka (misalnya: nama, label) ^[8].

2.4 Ontology Aligment

Ontology alignment merupakan hasil dari proses *ontology matching* yang terdiri dari kumpulan keterhubungan. *Aligment* mengekspresikan keterhubungan yang terdapat diantara dua buah *ontology*. Keterhubungan pada *alignment* tersebut dideskripsikan dalam 5 *tuple*:

<id, e, e', r, n> dengan penjelasan sebagai berikut ^[2]:

- id merupakan identitas unik dari keterhubungan
- e merupakan entitas dari *ontology o*
- e' merupakan entitas dari *ontology o'*
- r menunjukkan relasi dari alignment misalnya = atau <
- n merupakan nilai *confidence* misalnya nilai kemiripan antara dua buah entitas

Ontology alignment juga berupa suatu proses membangun struktur *alignment* dengan standar format yang terdapat pada *Alignment API* dan mengevaluasi nilai performansi *recall*, *precision* dan *f-measure* dari *ontology alignment* yang dihasilkan oleh sistem dengan *ontology alignment references*. *Output file ontology alignment* yang dihasilkan berupa file dalam format *.rdf. Berikut adalah rumus yang digunakan dalam menghitung nilai performansi *recall*, *precision* dan *f-measure* :

$$\frac{\text{Jumlah pemetaan benar}}{\text{Jumlah pemetaan}} \dots (2-3)$$

Precision dihitung dari jumlah pemetaan (*mapping*) benar yang diambil dibagi dengan jumlah pemetaan (*mapping*) yang diambil.

$$\frac{\text{Jumlah pemetaan benar}}{\text{Jumlah pemetaan yang diambil}} \dots (2-4)$$

Recall dihitung dari jumlah pemetaan (*mapping*) benar yang diambil dibagi dengan jumlah pemetaan yang diharapkan.

$$\frac{\text{Jumlah pemetaan benar}}{\text{Jumlah pemetaan yang diharapkan}} \dots (2-5)$$

F-measure merupakan nilai harmoni dari nilai *precision* dan *recall*.

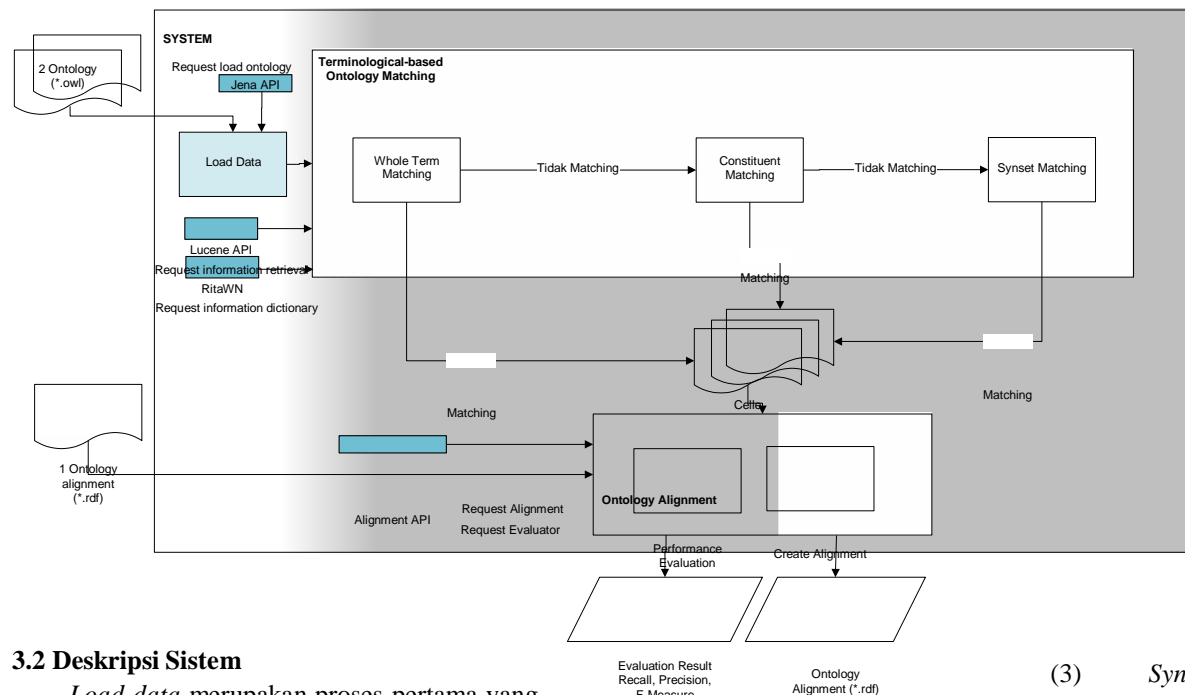
3. Pembangunan sistem

3.1 Gambaran Umum Sistem

Gambar dibawah merupakan gambaran umum dari sistem yang akan dibangun berupa sistem berbasis desktop dengan menggunakan bahasa

pemrograman Java. Ada 3 proses utama yang dilakukan yaitu *load data ontology*, *terminological-based ontology matching* dan proses *ontology alignment*.

skor dari 0 sampai 1. Lalu dari setiap istilah diantara semua pasangan yang cocok hanya ada pasangan yang terbaik (*best-fit*) atau yang memiliki skor tertinggi lah yang dicatat dan disajikan kepada *user*.



3.2 Deskripsi Sistem

Load data merupakan proses pertama yang dilakukan dimana proses ini memuat dan membaca dua buah data file *ontology* O_1 dan O_2 . Proses *load data* ini yang mengimplementasikan Jena API pada sistem.

Selanjutnya proses *terminological-base ontology matching* dimana proses ini merupakan proses mencocokkan dua buah *ontology* O_1 dan O_2 dengan beberapa metode serial yang terdapat didalamnya. Terdapat 3 metode untuk mencocokkan *string* atau label dari dua buah *ontology* yaitu:

(1) *Whole term matching* yang merupakan prosedur yang paling pertama dilakukan dan sederhana untuk dieksekusi. Istilah dari kedua buah *ontology* dikonversi ke huruf kecil (*lowercase*) dan kemudian membandingkan untuk pencocokan nama *string* atau label yang tepat. Pasangan yang cocok diberikan skor 1, jika tidak nilai adalah 0.

(2) *Word constituent matching* merupakan prosedur yang kedua dieksekusi. Setiap istilah dipecah menjadi kata-kata seperti huruf kapital, tanda hubung atau garis bawah. Atau memilah kata-kata yang tidak digunakan dalam proses pencocokan seperti "a", "the", "of", "in" dan lain sebagainya. Kata-kata yang tersisa dari setiap istilah akan diproses secara morfologis dan dibandingkan dalam pencocokan *string* yang tepat dengan kata-kata dari setiap istilah di kedua buah *ontology*. Setiap kata istilah yang cocok memiliki

(3) *Synset matching* merupakan

prosedur yang ketiga dieksekusi. *Synset* ini mengeksplorasi makna semantik dari unsur-unsur kata dengan menggunakan bantuan *Wordnet* dan juga membantu mengidentifikasi sinonim dalam pencocokan. Setiap istilah dari kata-kata yang terdapat pada masing-masing *ontology* dicari kata yang bersinonim dengan bantuan *wordnet*. Hasil pencocokan kata yang bersinonim dari dua buah *ontology* akan diberikan *range score* 0 sampai 1, sama seperti halnya pada prosedur kedua

Selanjutnya proses terakhir yang dilakukan yaitu *Ontology alignment* dimana hasil dari setiap pencocokan dari tahap-tahap TBOM yang telah disalin kedalam *celle* atau keterhubungan kelas-kelas antara dua buah *ontology* tersebut akan diproses selanjutnya kedalam *ontology alignment* untuk dikonversi struktur keterhubungannya menjadi standar *alignment* dalam format **.rdf* dengan mengimplementasi *Alignment API* dan melakukan proses *mapping* serta mengevaluasi nilai performansi *recall*, *precision* serta *f-measure* dari *ontology alignment* yang dihasilkan oleh sistem dengan *ontology alignment references*.

4. Pengujian Sistem

4.1 Tujuan Pengujian

Adapun tujuan dilakukannya proses pengujian terhadap sistem yang dibuat adalah :

1. Menguji tingkat kemiripan dari dua buah data *ontology* yang terdapat pada dataset pertama dan dataset kedua.
2. Menganalisa keterhubungan *ontology* yang dihasilkan dari *terminological-based ontology matching* dengan *ontology reference* yang terdapat pada kedua dataset.
3. Menganalisa pengaruh parameter *similarity threshold* terhadap nilai performansi *precision*, *recall* dan *f-measure* pada dataset pertama dan dataset yang kedua.

4.2 Skenario Pengujian

Pada pengujian sistem tugas akhir ini, user melakukan dua kali percobaan dengan dataset yang berbeda. Pencocokan dua buah *ontology* yaitu *mouse.owl* dengan *human.owl* dan *networkA.owl* dengan *networkB.owl* serta menginputkan *ontology alignment reference.rdf* dan *refnetAB.rdf* pada masing-masing percobaan guna untuk mencari nilai performansi dan dilihat tingkat kemiripan dari kedua buah percobaan yang dilakukan oleh sistem. Data yang telah dipersiapkan digunakan untuk melakukan pengujian terhadap sistem dimana detail skenario pengujian ditunjukkan pada tabel berikut.

No	Nama Skenario	Tujuan	Parameter	Data
1	Skenario 1a	Menguji tingkat kemiripan dari dua buah <i>ontology</i> dengan <i>ontology reference</i> yang tersedia	<i>Similarity threshold</i> = 1	2744 kelas <i>mouse.owl</i> dengan 3304 kelas <i>human.owl</i> dan 1516 keterhubungan kelas <i>reference.rdf</i> (dataset 1)
2	Skenario 1b	Menguji tingkat kemiripan dari dua buah <i>ontology</i> dengan <i>ontology reference</i> yang	<i>Similarity threshold</i> = 1	27 kelas <i>networkA.owl</i> dengan 27 kelas <i>networkB.owl</i> dan 43 keterhubungan kelas <i>refnetAB.rdf</i>

		tersedia		(dataset 2)
3	Skenario 2a	Menganalisa pengaruh parameter terhadap nilai performansi pada dataset 1	<i>Similarity threshold</i> = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0	2744 kelas <i>mouse.owl</i> dengan 3304 kelas <i>human.owl</i> dan 1516 keterhubungan kelas <i>reference.rdf</i>
4	Skenario 2b	Menganalisa pengaruh parameter terhadap nilai performansi pada dataset 2	<i>Similarity threshold</i> = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0	27 kelas <i>networkA.owl</i> dengan 27 kelas <i>networkB.owl</i> dan 43 keterhubungan kelas <i>refnetAB.rdf</i>

5. Kesimpulan dan saran

Kesimpulan yang dapat diambil dari tugas akhir ini antara lain :

1. Metode *terminological-based ontology matching* dapat mengurangi permasalahan heterogenitas. Hal ini dapat dilihat dari hasil pengujian yang dihasilkan oleh sistem, bahwa pencocokan yang dilakukan proses TBOM menghasilkan kecocokan atau tingkat kemiripan yang cukup akurat antara *ontology O₁* dengan *ontology O₂* dengan tingkat akurasi 80%.
2. Tingkat kemiripan pada proses *ontology matching*, dalam hal ini penerapan metode *terminological-based ontology matching* dipengaruhi oleh banyaknya kelas pada dataset yang diuji dan *ontology alignment reference* yang tersedia. Jika hasil *mapping ontology matching* yang dihasilkan tidak sebanding dengan keterhubungan kelas pada data *ontology reference* yang diberikan, maka tingkat kemiripan dan keterhubungan dari dua buah *ontology* tidak memberikan hasil yang cukup baik begitu juga sebaliknya.
3. Parameter *similarity threshold* memiliki pengaruh besar terhadap kualitas nilai

performansi *precision*, *recall* dan *f-measure*, karena semakin besar masukkan nilai *similarity threshold* maka semakin baik nilai performansi yang diberikan yaitu mendekati Hal ini dipengaruhi oleh banyaknya dataset yang diuji, karena semakin banyak dataset yang diuji maka memberikan hasil *mapping* yang tidak sebanding dengan *ontology reference* yang tersedia.

Saram yang dapat diberikan dan khususnya berkaitan dengan kasus pada tugas akhir ini antara lain:

1. Dalam pengembangan selanjutnya, dapat ditambahkan beberapa proses yang berhubungan dengan data leksikal agar dapat membandingkan nilai performansi yang dihasilkan.
2. Penggunaan metode *ontology matching* yang lainnya seperti *structur-based techniques* dan *semantic-based techniques* sehingga dapat dilakukan perbandingan kualitas performansi *ontology matching* berdasarkan metode atau teknik yang digunakan.

Daftar Pustaka:

- [1] Harmelen, F.V. (2004). *The Semantic Web: What, Why, How, and When*. Berlin: Springer-Verlag.
- [2] Euzenat, J., Shvaiko, P. (2007). *Ontology Matching*. Berlin Heidelberg: Springer-Verlag.
- [3] Mohammad Mustafa Taye. (2010). "Understanding Semantic Web and Ontologies: Theory and Applications," *Journal of Computing*, vol. 2, no. 6.
- [4] Perez, A.G., Lopez, F.M., Corcho, O. (2004). *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. London: Springer-Verlag.
- [5] Perez, A.G., Benjamins, V.R. (1999). *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*. Berlin Heidelberg: Springer-Verlag.
- [6] Zaiss, K.S. (2010). *Instance-Based Ontology Matching and The Evaluation of Matching System*. Germany: der Heinrich-Heine-Universitat Dusseldorf.
- [7] Li, John. (2004). *A Lexicon-based Ontology Mapping Tool*. Palo Alto: Teknowledge Corporation.
- [8] Shareha, Ahmad., Rajeswari, M., Ramachandram, D. (2009). *Two-way Dictionary-based Lexical Ontology Alignment*. Singapore: International Conference on Computer Engineering and Aplication.
- [9] Rahm, E. (2011). *Towards Large-Scale Schema and Ontology Matching*. Germany: Universitas of Leipzig.
- [10] Tim Berners-Lee, James Handler, and Ora Lassila. (2001). *The Semantic Web*. USA: Scientific American.
- [11] Balthasar Schopman, Shenghui Wan, Antoine Isaac, and Stefan Schlobach. (2012). *Instance-Based Ontology Matching by Instance Enrichment*. Amsterdam: Vrije Universiteit Amsterdam.
- [12] W3C. (2014) Introduction to RDF. [Online]. <http://www.w3.org/2001/sw/wiki/RDF>. (diakses pada tanggal 13 Juni 2014)
- [13] W3C. (2014) Introduction to RDF. [Online]. <http://www.w3.org/2001/sw/wiki/OWL>. (diakses pada tanggal 13 Juni 2014)