

Analisis Keterhubungan Ontology Pada Web Semantik Menggunakan *Semantic-Based Ontology Matching*

Ontology Relationships Analysis On Semantic Web Using Semantic-Based Ontology Matching

Andy satria, Anisa Herdiani, S.T., M.T., Veronikha Effendy, S.T, M.T.

Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom andysatria021091@gmail.com,
anisaherdiani@telkomuniversity.ac.id, mhsveffendy@telkomuniversity.ac.id

Abstrak

Web semantik memungkinkan data tidak hanya dapat dimengerti oleh manusia sebagai pembaca tetapi juga agar bisa diproses dan dimengerti oleh mesin atau komputer. Teknologi pada web semantik yang memungkinkan hal tersebut dapat terjadi adalah dengan menggunakan *ontology*. *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* merupakan proses untuk membandingkan dua *ontology* dan menemukan keterhubungan diantara kedua *ontology* tersebut. *Ontology matching* bertujuan untuk mengurangi masalah heterogenitas pada *ontology*. Salah satu teknik yang digunakan pada *ontology matching* untuk menyelesaikan masalah heterogenitas adalah *semantic-based ontology matching*.

Kata kunci: web semantik, *ontology*, *ontology matching*, heterogenitas, *semantic-based ontology matching*.

Abstract

The semantic web enables the data can be understood not only by humans as readers but also to be processed and understood by the machine or komputer. The technology on the semantic web which allows it can happen is by using ontology. Ontology describing data on the web and the connectivity between the data on the web, Heterogeneity is the most common problems occur in the ontology of semantic web, for example, there are two ontology with a different name, the ontology has a different structure or defined in a different way when both ontologies ontology that describes the same domain of knowledge. Ontology matching is a process for comparing two ontologies and find the relations between both of them. Ontology matching aims to reduce the problem of heterogeneity in ontology. One of the technique that usually used in ontology matching to solve the problem of heterogeneity is the semantic-based ontology matching.

Keywords: semantic web, ontology, ontology matching, heterogeneity, ontology-based semantic matching.

1. Pendahuluan

1.1 Latar Belakang

Web semantik merupakan evolusi dan perluasan dari *www (World Wide Web)* yang ditampilkan tidak hanya dengan format bahasa manusia yang umum tetapi juga bisa dibaca dan digunakan oleh mesin. Web semantik juga merupakan perluasan web saat ini dimana informasi yang disampaikan memiliki definisi arti yang baik, memungkinkan komputer dan manusia bekerjasama lebih baik [1]. Perubahan dari teknologi web semantik yang sangat signifikan adalah perubahan dari “Web of documents” menjadi “Web of data”. Salah satu perkembangan teknologi yang digunakan pada web semantik adalah *ontology*.

Ontology merupakan suatu teori tentang makna dari suatu objek, properti dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada suatu domain pengetahuan. *Ontology* juga merupakan spesifikasi dari sebuah konseptual, dengan kata lain *ontology* adalah penjelasan dari sebuah konsep dan keterhubungannya dari sebuah ilmu tertentu.

Ontology mendeskripsikan data pada web dan keterhubungan antar data pada web. Dengan semakin berkembangnya teknologi web semantik, maka akan semakin banyak pula jumlah *ontology* yang tersedia di internet namun oleh karena itu maka ketersediaan *ontology* yang sangat banyak ini dapat menimbulkan heterogenitas. Heterogenitas *ontology* memunculkan permasalahan, misalnya terdapat dua *ontology* dengan nama

yang berbeda, *ontology* tersebut memiliki struktur yang berbeda atau didefinisikan dengan cara yang berbeda padahal kedua *ontology* tersebut mengekspresikan pengetahuan yang sama namun dalam bahasa yang berbeda. Cara yang digunakan untuk mengurangi permasalahan heterogenitas *ontology* pada web semantik adalah *ontology matching*.

Ontology matching merupakan proses untuk membandingkan dua *ontology* dan menemukan keterhubungan diantara kedua *ontology* tersebut. Terdapat beberapa teknik yang dapat digunakan untuk menyelesaikan permasalahan heterogenitas dengan menggunakan *ontology matching* diantaranya adalah *terminological techniques (name-based techniques)*, *structure-based techniques*, *instance-based techniques (extensional techniques)* dan *semantic-based techniques*. Dari beberapa teknik yang digunakan dalam *ontology matching* tersebut, salah satu teknik yang digunakan dalam menyelesaikan permasalahan heterogenitas pada tugas akhir ini adalah teknik *Semantic-based Ontology Matching*. Karakteristik dari *Semantic-based Ontology Matching* ini adalah model teori semantik digunakan untuk membenarkan hasilnya, metode yang seperti ini biasa disebut dengan metode deduktif. Metode deduktif ini tidak tampil secara baik untuk sebuah tugas induktif dasar seperti pencocokan *ontology*. Diperlukan fase preprocessing yang menyediakan 'anchors' yaitu entitas yang dideklarasikan, contohnya agar dapat setara maka didasarkan pada entitas namanya atau input *user* untuk *instance*. Jadi *Semantic-based Ontology Matching* ini didasarkan pada penggunaan *resource* yang ada secara formal untuk menginisiasikan suatu keselarasan yang dapat dipertimbangkan lebih lanjut dengan metode deduktif [2].

Penggunaan metode *Semantic-Based Ontology Matching* ini diharapkan dapat memberikan suatu pemetaan kelas antar data *ontology* yang lebih luas, karena pemetaan kata berdasarkan hubungan semantiknya diyakini akan memberikan korespondensi yang lebih banyak sehingga dapat memberikan keluaran (*output ontology alignment*) baru yang mempunyai cakupan lebih luas.

1.2 Perumusan Masalah

Dalam tugas akhir ini akan dirumuskan masalah sebagai berikut:

1. Bagaimana menerapkan metode *Semantic-based ontology matching* untuk menganalisis dan menentukan keterhubungan *ontology* pada web semantik?
2. Apakah metode *Semantic-based ontology matching* dapat menyelesaikan masalah heterogenitas pada web semantik?
3. Bagaimana nilai parameter *Similarity Threshold* mempengaruhi nilai performansi *precision*, *recall*, dan *f-measure* dari metode *Semantic-based ontology matching* dalam menentukan keterhubungan *ontology* pada web semantik?

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini adalah:

1. Mengaplikasikan metode *Semantic-based Ontology Matching* untuk menangani masalah heterogenitas yang ada pada web semantik.
2. Menganalisis keterhubungan *ontology* pada web semantik berdasarkan parameter *Similarity Threshold* yang mempengaruhi nilai performansi *presicion*, *f-measure*, *recall* pada metode *Semantic-based ontology matching*.

1.4 Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Data input yang digunakan adalah data yang berekstensi *.owl dan *.rdf.
2. Pengimplementasian program menggunakan Java.
3. Proses pencarian kemiripan *semantic (instance similarity measure)* menggunakan Lucene API.
4. Data yang digunakan dalam pencocokan *ontology* adalah data yang berbahasa Inggris.

2. Tinjauan Pustaka

2.1 Web Semantik

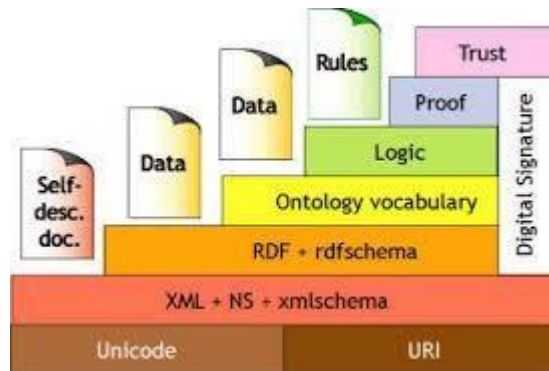
Semantic web atau web semantik merupakan salah satu perkembangan pada aplikasi web. Menurut bahasa, web semantik mempunyai arti yaitu web yang memiliki makna. Dengan kata lain, web semantik merupakan suatu aplikasi web yang mempunyai *knowledge base* tertentu sehingga bisa dikatakan web semantik mempunyai sifat lebih pintar dari web sebelumnya. Web semantik adalah sebuah evolusi dan perluasan dari web yang telah ada, yang memungkinkan komputer untuk memanipulasi data dan informasi [3].

Web semantik dikemukakan pertama kali oleh Tim Berners-Lee [4]. Web semantik sering disebut sebagai web versi 3.0. Web semantik memungkinkan data tidak hanya ditujukan atau dapat dimengerti oleh manusia sebagai pembaca tetapi juga agar bisa diproses dan dimengerti oleh mesin atau komputer.

Sebenarnya Web Semantik terdiri atas 2 buah kata yang masing-masing memiliki pengertian yang cukup berbeda.

1. Web: yang dimaksud web di sini adalah jaringan komputer yang luas yaitu WWW (*World Wide Web*)
2. Semantik: dapat diartikan sebagai ilmu tentang makna atau tentang arti, yaitu salah satu dari tiga tataran analisis bahasa. Fonologi, gramatika dan semantik.

Jadi Web semantik adalah kemampuan aplikasi komputer yang berfungsi untuk lebih memahami bahasa manusia, bukan hanya bahasa yang baku dari para penggunanya tetapi juga bahasa yang lebih kompleks, seperti dalam bahasa percakapan sehingga memudahkan penggunaannya untuk berkomunikasi dengan mesin. Web semantik dapat mengolah bahasa dan mengenali homonim, sinonim, atau atribut yang berbeda pada suatu *database*.



Gambar 1.1: Arsitektur web semantik [5]

Gambar 2.1 merupakan gambaran dari arsitektur web semantik yang dapat dijelaskan sebagai berikut [5]:

1. **URI dan Unicode**: digunakan untuk mengidentifikasi dan menempatkan *resource*. URI dianggap sebagai dasar dari web yang digunakan untuk memberikan nama unik untuk setiap *resources*. *Unicode* merupakan standar representasi karakter komputer.
2. **Extensible Markup Language (XML)** merupakan bahasa *markup* yang dapat dimengerti oleh mesin dan mempunyai format penulisan tersendiri. XML memiliki format teks yang fleksibel dan didesain untuk menggambarkan data serta berperan penting dalam pertukaran jenis data yang berbeda pada web. Setiap dokumen XML diawali dengan pendeklarasian *namespace* menggunakan XML *namespace*.
3. **Resource Description Framework (RDF)** merupakan lapisan pertama dari web semantik. RDF merupakan kerangka dalam menggunakan dan merepresentasikan metadata dan menggambarkan informasi semantik mengenai *resource* web yang mudah diakses mesin atau computer. RDF menggunakan URI untuk mengidentifikasi *resource* dengan menggunakan model grafik.
4. **Ontology Vocabulary** merupakan bahasa yang menyediakan kosa kata umum dan tata bahasa untuk publikasi data serta deskripsi semantik dari data yang digunakan untuk mengamankan dan menjaga *ontology* tersebut agar siap untuk *inference*. *Ontology* menggambarkan semantik data, menyediakan data yang seragam yang memungkinkan komunikasi dapat dimengerti oleh pihak manapun.
5. **Logic dan proof**, pembangunan sistem pada web semantik mengikuti logika pada struktur ontologi. Sebuah *reasoned* dapat digunakan untuk memeriksa dan menyelesaikan permasalahan konsistensi dan redundansi dari terjemahan kelas. Sebuah sistem *reasoning* digunakan untuk membuat *inference* baru.
6. **Trust**, merupakan lapisan terakhir pada web semantic. Komponen ini berkaitan dengan kepercayaan informasi pada web untuk memberikan jaminan kepercayaan terhadap informasi tersebut.

2.2 Ontology

Ontology merupakan salah satu teknologi pada semantik web yang memiliki cara baru untuk mendefinisikan dan menyimpan pengetahuan. *Ontology* menambahkan informasi meta pada konten website, sehingga komputer atau *software agent* bisa mengerti arti dari sebuah informasi. *Ontology* memiliki berbagai macam definisi, banyak peneliti-peneliti yang mendefinisikan pengertian *ontology*. Berikut beberapa definisi mengenai *ontology* yang dipaparkan oleh beberapa peneliti. Neches dan rekannya pada tahun 1991 menjelaskan bahwa *Ontology* mendefinisikan istilah-istilah dasar dan relasi yang terdiri dari kosakata area suatu topik serta aturan untuk menggabungkan istilah dan relasi untuk mendefinisikan perluasan kosakata [6]. Definisi *ontology* yang lain menurut Gruber pada tahun 1993 menyebutkan bahwa *ontology* adalah sebuah spesifikasi secara eksplisit dari konseptualisasi [6]. Definisi yang lain yang dipaparkan oleh Studer dan rekannya pada tahun 1998 menyatakan

bahwa *ontology* adalah spesifikasi formal, eksplisit dari konseptualisasi bersama. Konseptualisasi 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 [6].

Ontology biasanya berbentuk struktur jaringan yang terdiri atas:

1. Kumpulan kelas, biasanya kelas digambarkan sebagai simpul dalam struktur jaringan.
2. Kumpulan relasi yang menghubungkan kelas-kelas, relasi dalam struktur jaringan biasanya digambarkan sebagai garis berarah.
3. Kumpulan *instances* yang terdapat pada kelas-kelas tertentu.

2.3 Ontology Matching

Ontology matching merupakan solusi untuk permasalahan heterogenitas *ontology* pada web semantik. *Ontology matching* memiliki empat teknik dasar yang digunakan dalam proses *matching* dua buah *ontology*, yakni [9] :

- *Terminological techniques* (atau *name-based techniques*)
- *Struture-based techniques*
- *Instance-based techniques* (atau *extensional technique*)
- *Semantic-based techniques*

Teknik-teknik *matching* tersebut dapat digunakan untuk menyelesaikan permasalahan heterogenitas. Terdapat beberapa tipe permasalahan heterogenitas pada *ontology*, antara lain [8]:

1. *Syntactic heterogenity*, heterogenitas ini disebabkan oleh perbedaan format representasi *ontology* misalnya penggunaan bahasa pendefinisian yang berbeda seperti RDF dan OWL. Solusi untuk permasalahan ini adalah dengan mencari kesamaan antara bahasa yang berbeda.
2. *Terminological heterogenity*, heterogenitas ini menunjukkan penggunaan istilah yang berbeda untuk suatu hal yang sama pada suatu *ontology*, misalnya *surname* dan *family name*. Heterogenitas ini sulit untuk diatasi karena bahasa alami berkembang dan setiap orang memiliki kosakata yang berbeda.
3. *Conceptual heterogenity (semantic heterogenity)*, heterogenitas ini bisa dibagi menjadi 3 sub-masalah yakni:
 - ✓ *Coverage difference*, masalah ini terjadi jika *ontology* ditulis dari sudut pandang yang sama misalnya dalam konteks yang sama dan kosakata yang bisa dibandingkan akan tetapi bagian dari domain yang dijelaskan ada yang berbeda.
 - ✓ *Difference in granulariry*, 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
4. *Semiotic heterogenity*, heterogenitas ini disebabkan oleh penafsiran subjektif terhadap istilah yang digunakan oleh manusia. Padahal istilah-istilah tersebut memiliki makna yang sama namun direpresentasikan dalam cara yang berbeda.

2.3.1 Semantic-based Ontology Matching

Karakteristik dari *Semantic-based Ontology Matching* ini adalah model teori semantik digunakan untuk membenarkan hasilnya, metode yang seperti ini biasa disebut dengan metode deduktif. Metode deduktif ini tidak tampil secara baik untuk sebuah tugas induktif dasar seperti pencocokan *ontology*. Diperlukan fase preprocessing yang menyediakan 'anchors' yaitu entitas yang dideklarasikan, contohnya agardapat setara maka didasarkan pada entitas namanya atau input *user* untuk *instance*. Jadi *Semantic-based Ontology Matching* ini didasarkan pada penggunaan *resource* yang ada secara formal untuk menginisiasikan suatu keselarasan yang dapat dipertimbangkan lebih lanjut dengan metode deduktif [2].

2.3.2 Terminological-Based Ontology Matching

Dalam proses *matching* dua buah ontologi dengan menggunakan teknik *Terminological-based Ontology Matching*, teknik 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].

Terminological-based Ontology Matching ini juga dikenal dengan *Schema-based* yang mempunyai kemampuan untuk masuk kedalam domain yang berbeda tidak diperlukannya masukan tambahan (*additional inputs*) 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 (misal: nama, label) [10].

2.3.3 Instance-Based Ontology Matching

Instance-based ontology matching memanfaatkan perluasan dari kelas berupa *instance* yang merupakan contoh dari suatu kelas yang digunakan dalam proses *matching* dua buah *ontology* [11]. Hal ini juga berarti *Instance-based ontology matching* mencocokkan dua buah *ontology* dengan menggunakan ekstensi dari kelas yaitu *instance*. *Instance* merupakan kumpulan-kumpulan objek yang terkait dengan kelas tersebut. Prinsipnya adalah ketika sepasang kelas dari *ontology* memiliki kumpulan *instance* yang sama, maka kelas tersebut dikatakan memiliki keterhubungan atau kemiripan.

2.3.4 Structure-Based Ontology Matching

Structure-Based Ontology Matching memanfaatkan perbandingan struktur entitas dalam *ontology* daripada membandingkan nama atau *id* nya. Perbandingan ini dapat dibagi lagi kedalam perbandingan struktur internal suatu entitas yaitu definisi entitas tanpa mengacu pada entitas lainnya dan perbandingan struktur relasional yaitu himpunan dari relasi suatu entitas dengan entitas yang lain [2].

2.4 Ontology Alignment

Ontology alignment merupakan hasil dari proses *ontology matching* yang terdiri dari kumpulan keterhubungan. *Alignment* mengespresikan hubungan yang terdapat diantara dua buah *ontology*. Keterhubungan pada *alignment* tersebut dideskripsikan dalam 5 *tuple*: $\langle id, e, e', r, n \rangle$ dengan penjelasan sebagai berikut:

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

Ontology alignment ini adalah sebuah proses untuk membangun *alignment* dengan format yang berstandar pada *Alignment API* kemudian membuat perhitungan evaluasi nilai performansi dengan parameter *recall*, *precision*, dan *f-measure* dari *ontology alignment* yang sudah di *generate* oleh sistem dengan bersandar pada *ontology alignment references*. Pada proses *ontology alignment* ini, akan menghasilkan *ontology* baru dalam file ber format *.rdf. Berikut ini adalah rumus yang digunakan dalam menghitung performansi *recall*, *precision*, dan *f-measure*:

$$\text{Recall} = \frac{|\{ \text{Ontology A} \cap \{ \text{Ontology B} \}|}{|\{ \text{Ontology B} \}|} \dots (2 - 1)$$

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

$$\text{Precision} = \frac{|\{ \text{Ontology A} \cap \{ \text{Ontology B} \}|}{|\{ \text{Ontology A} \}|} \dots (2 - 2)$$

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

$$f - \text{measure} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \dots (2 - 3)$$

F-measure adalah nilai harmoni dari nilai *precision* dan *recall*

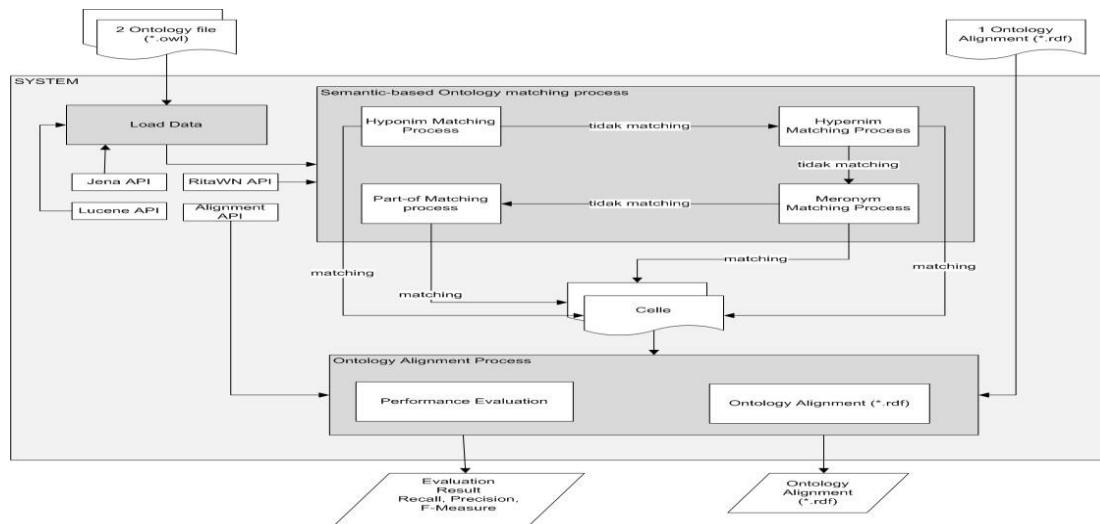
3. Perancangan sistem

3.1 Gambaran Umum Sistem

Sistem yang akan dibangun pada tugas akhir ini, adalah sebuah aplikasi berbasis desktop dengan gambaran umum menggunakan bahasa pemrograman java. Pada gambaran umum sistem yang akan dibangun, tahap pertama yang harus dilakukan adalah *user* menginputkan dua buah *ontology* yang akan dicari

keterhubungannya pada proses *ontology matching* melalui *GUI application* yang dibuat. File *ontology* yang diinputkan dalam sistem adalah dua buah file dengan ekstensi *.owl, dan sebuah file *ontology alignment* yang berekstensi *.rdf. Lalu sistem memproses dua *ontology* tersebut dengan menggunakan metode *semantic-based ontology matching* yang telah diimplementasikan pada sistem. Setelah melalui proses *matching* kemudian dibangun struktur yang menghubungkan kedua *ontology* tadi berdasarkan keterhubungan yang dihasilkan. Kemudian pada *GUI Application* akan ditampilkan struktur yang menghubungkan kedua *ontology* tersebut dan hasil evaluasi dari proses *ontology matching* menggunakan metode *semantic-based ontology matching*.

Didalam system yang dibuat ini, terdapat tiga proses inti yang bekerja, yaitu proses *Load Data*, proses *Semantic-based Ontology Matching*, dan proses *Ontology Alignment Proses*. Pada proses pertama yaitu proses *Load Data*, disini akan dimasukkan dua buah data *ontology* yang berekstensi *.owl untuk dimuat dan dibaca data *ontology* tersebut dengan menggunakan bantuan Jena API dan Lucene API, yaitu sistem pendukung yang bertugas menerima, memuat dan membaca data *ontology* kedalam sistem yang akan dibuat.



Gambar 2.1: Gambaran umum sistem

4. Pengujian dan Analisis

4.1 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui nilai performansi dan tingkat kemiripan dari dua buah *ontology* yang ada pada sistem berbasis desktop yang mencari keterhubungan *ontology* pada web semantic dengan menggunakan metode *Semantic-based Ontology Matching*. Pengujian ini dilakukan dengan 3 buah data yang berekstensi *.owl dan dua buah data yang berekstensi *.rdf. Dari semua file *ontologi* tersebut kemudian akan dicari keterhubungannya adalah dua buah data *ontologi* dan satu data *reference*. Dapat dikatakan bahwa dalam pengujian ini dilakukan dua percobaan dengan data yang berbeda, yang nantinya akan dilihat perbedaan pengaruh dari keterhubungan dua pasangan data tersebut. Pada tugas akhir ini ada dua jenis pengujian yang dilakukan

4.2 Skenario Pengujian

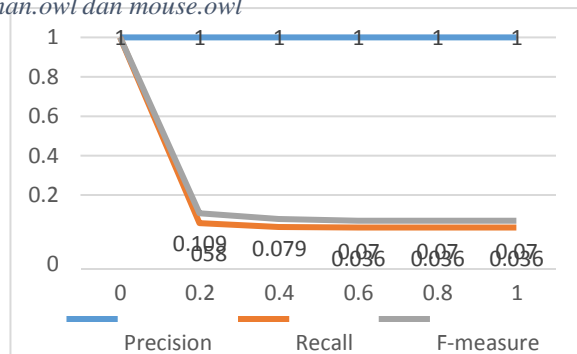
Pengujian untuk mencocokkan dua buah file *ontology* akan dilakukan dengan melakukan dua kali percobaan yang menggunakan sepasang file *ontology* dengan format *.owl yaitu antara "human.owl" dan "mouse.owl", dan satu percobaan berikutnya antara *ontology* "human.owl" dan "plant.owl", akan digunakan untuk mencari korespondensi antar pasangan data *ontology* tersebut. serta menginputkan satu file *ontology* lain dengan format *.rdf pada setiap percobaan, yang bernama "ref.rdf" yang akan dipakai untuk mendapatkan nilai evaluasi *precision*, *recall*, dan *f-measure* serta untuk mendapatkan *ontology alignment* yang baru. Proses pencocokan dua buah *ontology* ini dilakukan dengan diberikan parameter *Similarity Threshold* tertentu pada program yang dibuat.

4.3 Analisa hasil pengujian

Setelah melakukan pengujian nilai *similarity threshold* dengan skenario yang telah ditentukan, maka diperoleh hasil *precision*, *recall*, dan *f-measure* dari hasil proses *matching ontology human.owl* dan *mouse.owl*. Selengkapnya dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4.1: hasil pengujian dengan data ontology *human.owl* dan *mouse.owl*

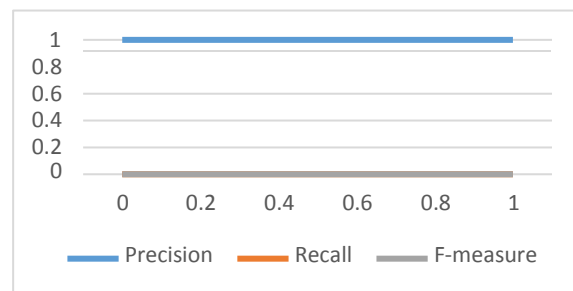
Similarity Threshold	Precision	Recall	F-measure
0	1	1	1
0.2	1	0.058	0.109
0.4	1	0.039	0.079
0.6	1	0.036	0.07
0.8	1	0.036	0.07
1	1	0.036	0.07



Gambar 4.1: Grafik hasil pengujian pengaruh *similarity threshold* terhadap *precision*, *recall*, dan *f-measure* pada pengujian ontology *human.owl* dan *mouse.owl*

Tabel 4.2: hasil pengujian dengan data ontology *human.owl* dan *plant.owl*

Similarity Threshold	Precision	Recall	F-measure
0	1	0	0
0.2	1	0	0
0.4	1	0	0
0.6	1	0	0
0.8	1	0	0
1	1	0	0



Gambar 4.2: Grafik hasil pengujian pengaruh *similarity threshold* terhadap *precision*, *recall*, dan *f-measure* pada pengujian ontology *human.owl* dan *plant.owl*

Dari dua tabel dan gambar diatas, terlihat bahwa pengujian pengaruh *similarity threshold* terhadap nilai performansi *precision*, *recall*, dan *f-measure* menghasilkan nilai yang berbeda-beda dengan kecenderungan besar nilai parameter *similarity threshold* tidak memberikan banyak pengaruh terhadap nilai performansi *precision*, *recall*, dan *f-measure*. Bahkan semakin besar nilai parameter *similarity threshold* yang diberikan, maka nilai harmoni antara *precision* dan *recall* cenderung semakin kecil. Tetapi, jika dilihat dari hasil keseluruhan proses, metode *Semantic-based Ontology Matching* memiliki performa yang bagus karena dapat memetakan dan mencari korespondensi antar kelas yang lebih baik dari kelas-kelas yang ada pada data *ontology* yang dicari keterhubungannya, dan dapat memberikan keakuratan yang baik.

5. Kesimpulan dan saran

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap sistem yang dibangun dengan menerapkan metode *Semantic-based Ontology Matching*, maka kesimpulan yang dapat diambil adalah:

1. Metode *Semantic-based Ontology Matching* dapat menangani dengan baik masalah heterogenitas, khususnya jenis heterogenitas *Conceptual heterogenity (semantic heterogenity)*. Karena metode ini dapat memetakan dan mencari dengan baik korespondensi antar data *ontology*.
2. Parameter *similarity threshold* tidak memberikan banyak pengaruh terhadap evaluasi nilai *precision*, *recall*, dan *f-measure*. Terlihat dengan semakin besarnya nilai parameter *similarity threshold* yang diberikan, maka berbanding terbalik dengan nilai evaluasi *recall* dan *f-measure*.

5.2 Saran

Setelah menyelesaikan pengerjaan tugas akhir ini, penulis memiliki beberapa saran untuk pengembang tugas akhir dimasa mendatang, penulis menyarankan beberapa hal berikut:

1. Pengujian dicoba dengan menggunakan data inputan yang memiliki jumlah kelas yang lebih banyak.

Menggunakan metode *matching* lainnya seperti metode yang menitik beratkan pada pencarian korespondensi antar kelas berdasarkan strukturnya pada data *ontology*, sehingga dapat dilakukannya perbandingan keakuratan antara beberapa metode *ontology matching*.

Daftar Pustaka

- [1] T. Berners-Lee, J. Hender and O. Lassila, "The Semantic Web," *Scientific American*, 2001.
- [2] E. J and S. P, *Ontology Matching*, Berlin: Springer-Verlag, 2007.
- [3] D. Nandini, *Semantic Web And Ontology*, Bookboon, 2014.
- [4] S. Aghaei, M. Ali N and H. Khorsavi F, "Evolution of The World Wide Web: From Web 1.0 to Web 4.0," vol. 3, no. *International Journal of Web & Semantic Technology (IJWeST)*, 2012.
- [5] M. M. Taye, "Understanding Semantic Web and Ontologies: Theory and Application," *Journal of Computing, Vol. 2, Issue 6*, pp. 182-192, 2010.
- [6] A. G. Perez, F. M. Lopez and O. Corcho, "Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web," Springer-Verlag, London, 2004.
- [7] P. A G and B. V R, "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods," 1999.
- [8] Z. K S, "Instance-Based Ontology Matching and The Evaluation of Matching System," 2010.
- [9] S. B, W. S, I. A and S. S, "Instance-Based Ontology Matching by Instance Enrichment," 2012.
- [10] A. Shareha, M. Rajeswari and D. Ramachandram, "Two-Way Dictionary-based Lexical Ontology Alignment," in *International Conference on Computer and Applications*, Singapore, 2009.
- [11] B. Schopman, S. Wang, A. Isaac and S. Schlobach, *Instance-Based Ontology Matching by Instance Enrichment*, Amsterdam: Springer, 2012.
- [12] N. F. Noy, "Ontology Mapping," 2009.
- [13] M. Ehrig and S. Staab, "QOM - Quick Ontology Mapping," 2004.