

PENERAPAN TEKNOLOGI *SEMANTIC WEB* PADA ARTIKEL WIKIPEDIA INDONESIA

Irvan Wahyu Nurdian¹, Faisal Rahutomo², Imam Fahrur Rozi³

Program Studi Teknik Informatika, Jurusan Teknologi Informasi
, Politeknik Negeri Malang

¹irvan.wn@gmail.com, ²faisal_rahutomo@polinema.ac.id, ³imam.rozi@gmail.com

Abstrak

Wikipedia adalah sebuah situs yang berisi tentang artikel ensiklopedia dan referensi online tingkat dunia. Artikel yang terdapat pada Wikipedia saat ini total telah mencapai lebih dari 37 juta artikel dari 250 lebih bahasa berbeda. Untuk Wikipedia Indonesia telah mencapai 371.281 artikel. Dengan sekian banyaknya artikel-artikel tersebut, tentunya bukan perkara mudah melakukan pencarian dengan cepat dan tepat sesuai dengan yang diharapkan. Maka dari itu dibutuhkan suatu metode yang dapat memecahkan masalah tersebut, yaitu metode *semantic web*. Penelitian ini dimulai dengan merancang ontologi berdasarkan *class-class* dari hasil *clustering* artikel Wikipedia Indonesia serta dilengkapi dengan pendefinisian *property-property* dari masing-masing *class*. Dilanjutkan dengan perancangan *query* dengan bahasa *SPARQL*. Pencarian artikel pada aplikasi ini diuji dengan menggunakan 3 perhitungan untuk mengukur akurasi sistem, yakni *Precision*, *Recall* dan *F-Score*. Pada perhitungan *Precision* mendapatkan nilai rata-rata 1, *Recall* mendapatkan nilai rata-rata 0.53. Lalu pada perhitungan akhir akurasi sistem didapat nilai pada *F-Score* 0.6935 atau dengan persentase akurasi 69,35%.

Kata kunci : *Semantic Web*, Ontologi, Wikipedia, *SPARQL*

1. Pendahuluan

Perkembangan teknologi internet saat ini, telah merubah cara kita mencari dan memperoleh informasi. Sekarang pencarian informasi sebagian besar beralih ke internet. Mulai dari pencarian hal-hal sepele sampai kepada keperluan studi literatur. Salah satu situs web yang populer menjadi rujukan yaitu Wikipedia.

Artikel yang terdapat pada Wikipedia saat ini total telah mencapai lebih dari 37 juta artikel dari 250 lebih bahasa berbeda. Untuk Wikipedia Indonesia telah mencapai 371.281 artikel. Dengan sekian banyaknya artikel-artikel tersebut, tentunya bukan perkara mudah melakukan pencarian dengan cepat dan tepat sesuai dengan yang diharapkan. Maka dari itu dibutuhkan suatu metode yang dapat memecahkan masalah tersebut, yaitu metode *semantic web*.

Untuk menampilkan informasi yang kaya dan relevan, sebuah mesin terlebih dahulu harus memahami tentang makna atau isi dari sebuah informasi. Hal inilah yang akan ditawarkan *semantic web*. Untuk mewujudkan itu maka diperlukan sebuah *domain knowledge*, yang dalam *semantic web* dikenal dengan istilah ontologi. Dalam ontologi, informasi akan saling terkait membentuk sebuah *knowledge* yang akan dimanfaatkan oleh mesin untuk menyajikan

informasi yang kaya dan sesuai dengan yang pengguna cari.

Dari penjelasan di atas, maka pada penelitian ini penulis mengusulkan untuk merancang aplikasi pencarian berbasis teknologi *semantic web* pada artikel Wikipedia Indonesia menggunakan pendekatan ontologi. Diharapkan dengan penelitian ini akan mampu memudahkan pencarian informasi maupun informasi terkait pada artikel Wikipedia Indonesia.

2. *Semantic Web*

Secara umum *Semantic Web* dapat diartikan sebagai informasi dalam jumlah sangat besar di *World Wide Web* yang terhubung secara global dengan suatu cara tertentu dan dimengerti / dipahami oleh mesin, sehingga dapat diproses secara langsung oleh mesin menjadi *knowledge* untuk ditampilkan kepada *user*. *Semantic Web* pertama kali dicetuskan oleh Tim Berners-Lee, penemu *World Wide Web*. Sekarang, prinsip *Semantic Web* disebut-sebut akan muncul pada Web 3.0, generasi ketiga dari *World Wide Web*.

Dengan metode *Semantic Web*, data berbasis HTML dapat diubah menjadi format yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan antara informasi. Web semantik mampu melakukan perubahan ini dengan

bantuan XML (*Extensible Markup Language*) dan data language standards seperti RDF (*Resource Description Framework*) dan OWL (*Ontology Web Language*), dua standarisasi dari W3C (*World Wide Web Consortium*).

3. Ontologi

Istilah ontologi sebenarnya berasal dari istilah filosofi "ontology" yang artinya sesuatu yang sesungguhnya ada dan bagaimana menggambarannya. Dalam dunia komputer ontologi digunakan untuk menspesifikasikan suatu konseptualisasi. Dalam istilah lain ontologi dijelaskan sebagai suatu representasi dari domain pengetahuan tertentu yang berisi istilah-istilah dalam domain tersebut beserta hubungan antara istilah-istilah yang ada.

Terdapat beberapa tahapan yang dilakukan dalam pengembangan ontologi (Noy, 2000) yaitu :

1. Menentukan domain dan batasan ontologi
2. Mempertimbangkan penggunaan ontologi yang sudah ada.
3. Menentukan istilah penting dalam ontologi
4. Mendefinisikan *class* ontologi dan menyusun class dalam hirarki taksonomi (*subclass – superclass*).
5. Mendefinisikan *slot* atau *properties* dan menjabarkan nilai dari *slot* tersebut.
6. Mendefinisikan *facets* pada *slots*.
7. Membuat *instances*.

4. RDF (Resource Description Framework)

RDF (*Resource Description Framework*) merupakan bahasa yang digunakan untuk merepresentasikan metadata. RDF mendukung interoperabilitas antar aplikasi yang mempertukarkan informasi yang bersifat *machine-understandable* di web. *Semantic web* terdiri dari data yang ditulis dalam bahasa yang dapat dimengerti oleh mesin, seperti RDF. RDF menggunakan graf untuk merepresentasikan kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas.

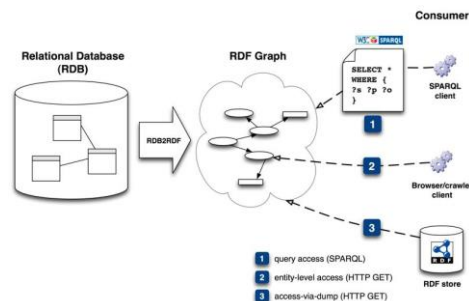
Ada tiga obyek tipe dalam model data RDF (Lasila dan Swick, 1999) :

- a) Sumber Daya (*Resource*)
- b) Properti (*Property*)
- c) Pernyataan (*Statement*)

5. SPARQL

SPARQL merupakan bahasa *query* untuk RDF/OWL. SPARQL menyediakan fasilitas untuk mengekstrak informasi dalam bentuk URI, blank node dan literal, mengekstrak subgraf RDF, dan membangun graf RDF baru berdasar pada informasi dari graf yang di-query (Prud'hommeaux dan Seaborne, 2005). *Query SPARQL* didasarkan pada pencocokan pola graf (Karsanti, 2006). Pola graf

yang paling sederhana adalah triple pattern yang mirip dengan RDF triple, hanya saja pola pada query dimungkinkan pemberian nama diluar terminologi RDF pada posisi subyek, predikat dan obyek.



Gambar 1 : Gambaran umum SPARQL

6. Perancangan Ontologi

Perancangan ontologi untuk artikel Wikipedia Indonesia merupakan langkah awal dalam menentukan konsep dan domain yang akan dibangun. Noy dan McGuinness (2000) telah menjelaskan ada beberapa langkah-langkah yang harus diperhatikan.

1. Penentuan konsep dan domain

Domain dari penelitian ini adalah artikel Wikipedia Indonesia, yang diklasifikasikan menjadi beberapa *class* induk (*superclass*) dan *subclass*.

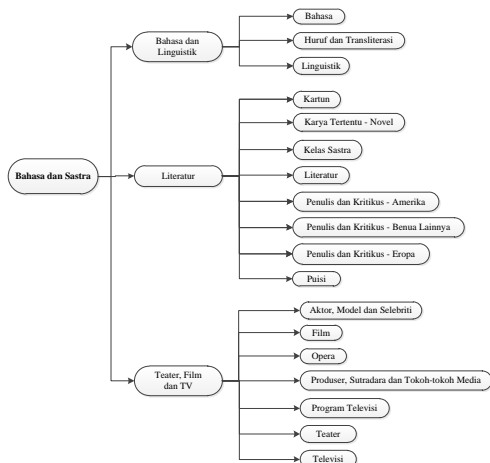
2. Mempertimbangkan ontologi yang sudah ada

Dalam pembuatan aplikasi ini, diasumsikan belum terdapat ontologi yang dapat digunakan untuk pengembangan sehingga dalam hal ini seluruh komponen dari ontologi dikembangkan dari awal.

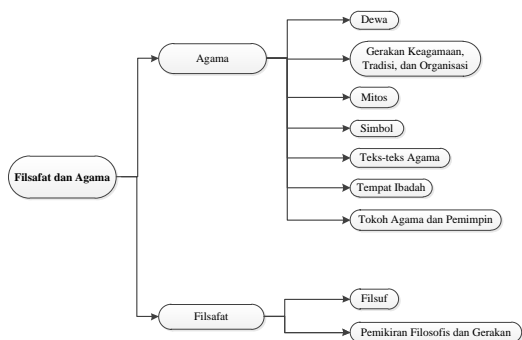
3. Menentukan istilah penting dalam ontologi
Dalam pembuatan ontologi pada aplikasi ini, menentukan istilah penting bertujuan untuk penamaan *class*.

4. Mendefinisikan *class* ontologi dan menyusun *class* dalam hirarki taksonomi (*subclass – superclass*)

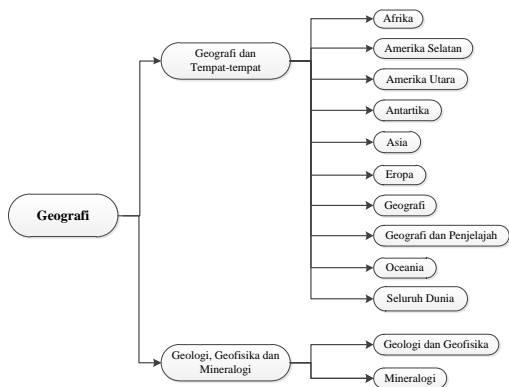
Rancangan *class* artikel Wikipedia Indonesia terdiri dari 10 *class* induk (*superclass*), yaitu sebagai berikut :



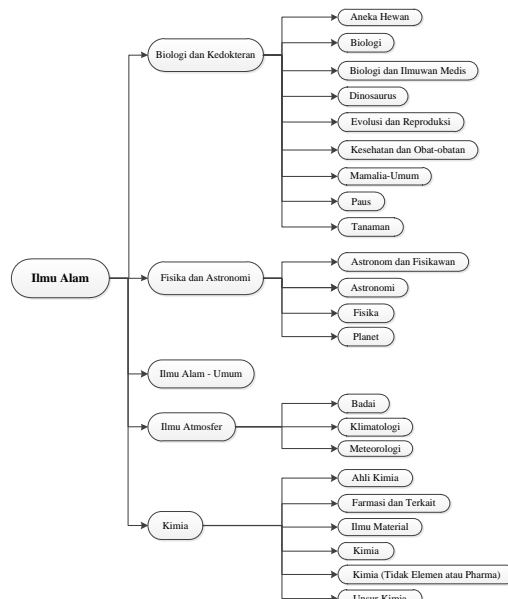
Gambar 2 : rancangan class “Bahasa dan Sastra”



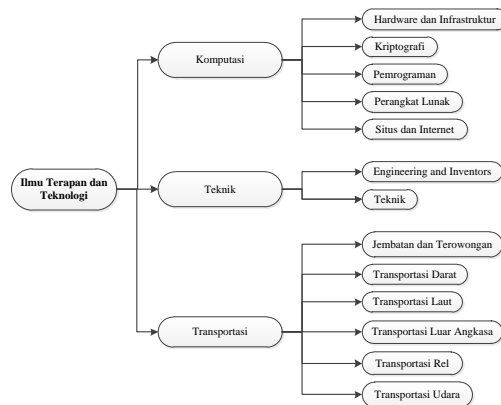
Gambar 3 : rancangan class “Filsafat dan Agama”



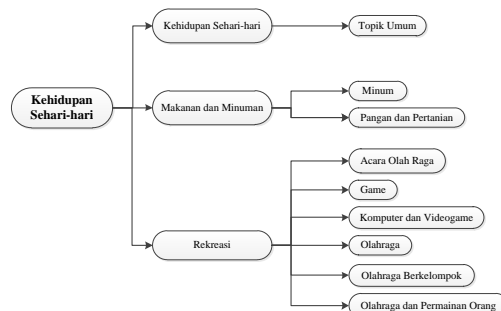
Gambar 4 : rancangan class “Geografi”



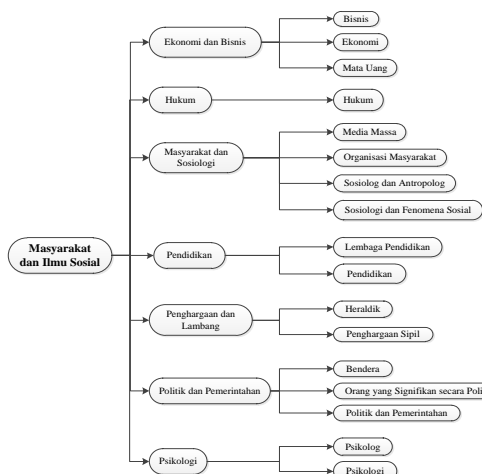
Gambar 5 : rancangan class “Ilmu Alam”



Gambar 6 : rancangan class “Ilmu Terapan dan Teknologi”



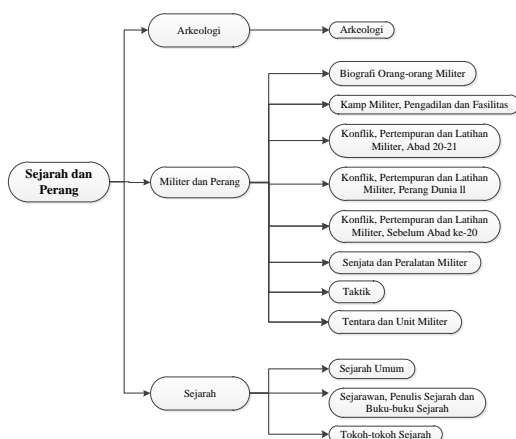
Gambar 7 : rancangan class “Kehidupan Sehari-hari”



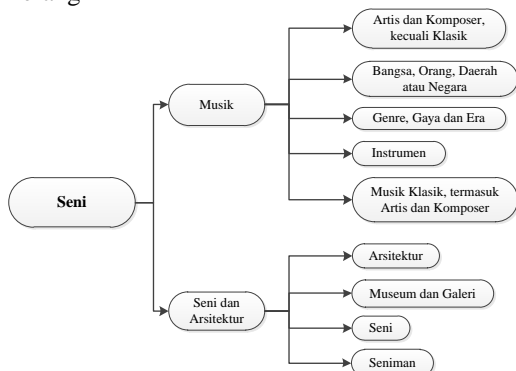
Gambar 8 : rancangan class “Masyarakat dan Ilmu Sosial”



Gambar 9 : rancangan class “Matematika”



Gambar 10 : rancangan class “Sejarah dan Perang”



Gambar 11 : rancangan class “Seni”

5. Mendefinisikan slot atau *properties* pada class

Ada dua tipe *property*, yaitu *Object Properties* berfungsi untuk menghubungkan *instances* antar class dan *Data Properties* yang berfungsi untuk menghubungkan *instances*

dengan *values* dari *property* tiap *instances*. Pada *Object Properties*, didefinisikan dua *property* yaitu *hasMember* dan *isMemberOf*. Sedangkan *Data Properties* didefinisikan tiga *property*, yaitu Isi, Judul dan Nama.

6. Mendefinisikan *facets* pada slots

Mendefinisikan batasan (*facet*) *property* atau slot dengan mengisi Nama *Property*, *Range*, *Type* dan Keterangan pada setiap class.

Tabel 1: Rancangan Slot Pada Class “Kat Bahasa dan Sastra”

Property	Range	Type	Keterangan
Nama	Single String	Data Property	Nama Class
hasMember	Multiple Subclass	Object Property	Menghubungkan class Kat_Bahasa_dan_Sastra dengan class Sub1_Bahasa_dan_Linguistik, Sub1_Literatur, Sub1_Teater_Film_dan_TV

7. Membuat *instances*

Langkah pendefinisian sebuah *instanceclass* dimulai dengan memilih class, membuat individu *instance* dari class kemudian yang terakhir mengisi *property* dari individu *instanceclass*. Sebagai contoh, *instance* “Bahasa Indonesia” menerangkan secara khusus anggota dari class “Bahasa”.

Tabel 2 : Rancangan Instance “Bahasa Indonesia” Dari Class “Bahasa”

Property	Value
Judul	Bahasa Indonesia
Isi	res/Wikipedia/Bahasa dan Sastra/Bahasa dan Linguistik/Bahasa/Bahasa Indonesia.xml
isMemberOf	Sub2_Bahasa

7. Implementasi Ontologi

Implementasi class dan struktur hirarkinya pada aplikasi ini dimulai dengan menciptakan class-class dan sub class yang telah ditentukan. Class-class pada ontologi aplikasi ini terdiri dari sepuluh class induk (*super class*), yaitu class “Bahasa dan Sastra”, class “Filsafat dan Agama”, class “Geografi”, class “Ilmu Alam”, class “Ilmu Terapan dan Teknologi”, class “Kehidupan Sehari-hari”, class “Masyarakat dan Ilmu Sosial”, class “Matematika”, class “Sejarah dan Perang” dan class “Seni”.

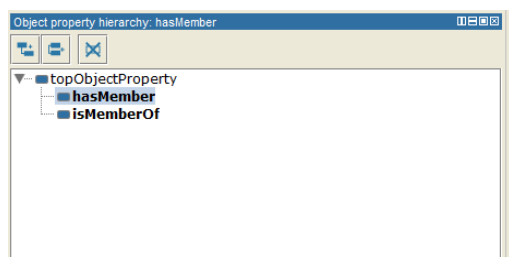


Gambar 12 : Implementasi Struktur dan Hirarki Class Induk (Superclass)

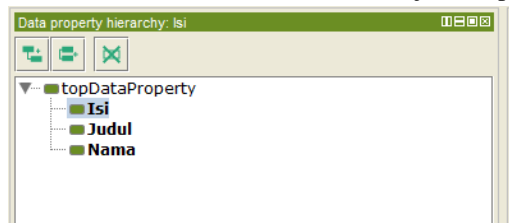


Gambar 13 : Implementasi Struktur dan Hirarki Class “Bahasa dan Sastra”

8. Implementasi Property



Gambar 14 : Pendefinisian Nama Object Property



Gambar 15 : Pendefinisian Nama Data Property

9. Implementasi Instances

Pembuatan *instances* yaitu membuat individual baru dan melakukan pengisian *property* dengan nilai (*values*) pada individual tersebut. Setiap *instances* yang terbentuk akan mewakili *class* yang telah dibuat dan setiap artikel. Pembuatan *instancesclass* dimulai dengan

pemberian nama *instances*, memilih *Types (class)* dan pengisian *property*.



Gambar 16 : Instances “Bahasa_Indonesia”

10. Pengujian

1. Pengujian Fungsional Sistem

Untuk menguji kinerja aplikasi dibutuhkan pengujian fungsionalitas sistem. Pengujian ini dilakukan dengan cara menjalankan setiap fitur dalam aplikasi dan melihat apakah hasilnya sudah sesuai dengan yang diharapkan.

Tabel 3 : Pengujian Sistem

Pengujian	Data Masukkan	Hasil yang diharapkan	Hasil
Pencarian berdasarkan kata kunci	Kata kunci “Bahasa Indonesia”	Menampilkan judul dan isi artikel dari kata kunci beserta artikel terkait dan kategori terkait	Berhasil
	Kata kunci “Indonesia Bahasa”	Menampilkan beberapa kemungkinan tautan dari kata kunci “Indonesia Bahasa”	
	Kata kunci “Bahasa”	Menampilkan beberapa kemungkinan tautan dari kata kunci “Bahasa”	
	Kata kunci “presiden” yang tidak terdapat pada ontologi wiki	Menampilkan pesan bahwa kata kunci tidak ditemukan	
Pencarian berdasarkan tautan pada artikel terkait	Tautan “Bahasa Aceh”	Menampilkan judul dan isi artikel “Bahasa Aceh” beserta artikel terkait	Berhasil
Pencarian berdasarkan tautan pada kategori terkait	Tautan “Huruf dan Transliterasi”	Menampilkan tautan artikel dalam kategori “Huruf dan Transliterasi”	Berhasil
Pencarian berdasarkan tautan pada kategori utama artikel	Tautan “Bahasa dan Sastra”	Menampilkan tautan sub kategori dalam kategori “Bahasa dan Sastra”	Berhasil

2. Pengujian Akurasi Sistem

Pengujian akurasi sistem bertujuan untuk menghitung tingkat relevansi hasil pencarian oleh sistem dengan kata kunci yang dimasukkan

oleh pengguna (*query*). Dalam pengujian ini terdapat 2 nilai yang diukur, yaitu :

1. Jumlah artikel sebenarnya (**N1**), yaitu didapat dari perhitungan jumlah artikel yang sesuai atau terkait dari keseluruhan artikel dengan kata kunci yang diujikan berdasar pengetahuan penguji.
2. Jumlah artikel oleh sistem (**N2**), yaitu didapat dari perhitungan jumlah artikel yang ditampilkan sistem.

Pada pengujian ini juga digunakan tiga perhitungan untuk mengukur akurasi sistem yang menerapkan metode *semantic web*, yaitu :

1. *Precision*, merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Nilai *Precision* tertinggi adalah 1, yang berarti seluruh dokumen yang ditemukan adalah relevan. Pada pengujian ini, *Precision* dirumuskan sebagai berikut.

$$Precision = \frac{N2 \cap N1}{N2}$$

2. *Recall*, merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Nilai *Recall* tertinggi adalah 1, yang berarti seluruh dokumen dalam koleksi berhasil ditemukan. Pada pengujian ini, *Recall* dirumuskan sebagai berikut.

$$Recall = \frac{N2 \cap N1}{N1}$$

3. *F-Score*, merupakan gabungan dari *Precision* dan *Recall*. Pada pengujian ini, *F-Score* dirumuskan sebagai berikut.

$$F-Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)}$$

Tabel 4 : Data Hasil Pengujian Akurasi Sistem

No	Query	N1	N2	Preci-sion	Recall
1	Bahasa Indonesia	9	3	1	0.333
2	Mitologi	6	3	1	0.5
3	Amerika Serikat	6	3	1	0.375
4	Hukum Fisika	5	3	1	0.6
5	Pesawat Terbang	4	3	1	0.75
6	Bola Basket	4	3	1	0.75
7	Daftar Sosiolog Indonesia	8	3	1	0.375
8	Matematika	4	3	1	0.75
9	Soekarno	6	3	1	0.5
10	Galeri Nasional Indonesia	8	3	1	0.375
Rata-rata				1	0.53083
F-Score				0.693522047	
(%)				69.35 %	

Berdasarkan pengujian fungsional sistem yang telah dilakukan, diketahui bahwa sistem pada penelitian ini telah berjalan dengan baik secara

fungsional dan menghasilkan *output* yang diharapkan.

Pada perhitungan N1, didapatkan nilai jumlah artikel yang berhubungan atau relevan dengan *query* uji berdasarkan pengetahuan penguji. Pada perhitungan N2, didapatkan nilai jumlah artikel yang ditampilkan oleh sistem berdasarkan *query* uji. Dari data N1 dan N2, kemudian digunakan untuk menghitung akurasi sistem melalui tiga perhitungan, yaitu *Precision*, *Recall* dan *F-Score*.

Berdasarkan perhitungan *Precision*, didapatkan nilai 1 atau nilai tertinggi untuk seluruh *query* uji, yang berarti seluruh dokumen yang ditampilkan oleh sistem adalah relevan. Sedangkan pada perhitungan *Recall*, diperoleh nilai rata-rata 0.51.

Pada perhitungan akhir akurasi sistem, pada perhitungan *F-Score* diperoleh nilai persentase 69,35%, yang berarti bahwa sistem mampu menampilkan artikel dengan cukup relevan.

11. Kesimpulan dan Saran

Berdasarkan pembahasan pada makalah ini mengenai penerapan teknologi *semantic web* pada artikel Wikipedia Indonesia maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Ontologi pada aplikasi Semantic Wikipedia Indonesia dimodelkan berdasarkan *class-class* dari hasil *clustering* artikel Wikipedia Indonesia serta dilengkapi dengan pendefinisian *property-property* dari masing-masing *class*.
2. Pencarian pada aplikasi Semantic Wikipedia Indonesia ini mampu menampilkan artikel terkait dan kategori terkait yang relevan dengan artikel hasil pencarian.

Aplikasi Semantic Wikipedia Indonesia ini dapat dikembangkan dengan menambah jumlah artikel Wikipedia Indonesia yang digunakan agar sistem mampu menampilkan lebih banyak artikel terkait.

Daftar Pustaka:

Antoniou, Grigoris and van Harmelen, Frank. 2008. "A *Semantic Web Primer, Second Edition*". Massachusetts Institute of Technology

Gunawan dan Halim, Fandi. 2014. "Penerapan *Web Semantik Untuk Aplikasi Pencarian Pada Repositori Koleksi Penelitian, Studi Kasus: Program Studi Sistem Informasi Stmik Mikroskil Medan*". Skripsi. Medan : STMik Mikroskil

<https://en.wikipedia.org/wiki/Wikipedia> [11 Desember 2015]

<https://id.wikipedia.org/wiki/Istimewa:Statistik> [11 Desember 2015]

Lassila, Ora and Swick, Ralph R. 1999. *Resource Description Framework (RDF) Model and Syntax Specification*. [Online] Tersedia :

- <http://www.w3.org/TR/1999/REC-RDF-syntax-19990222/>. [16 Maret 2016]
- Noy, N. F. dan McGuinness. D. L. 2000. *Ontology Development 101: A Guide to Creating Your First*, Stanford University, Stanford. [Online] Tersedia : http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html [9 Maret 2016]
- Nugroho, Atmoko. 2012. “Membangun Ontologi Jurnal Menggunakan Protégé”. JURNAL TRANSFORMATIKA, Volume 10, No.1, Juli 2012 : 20 – 25. Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang
- Rahman, Muhammad Arief. 2015. “Penerapan Teknologi Semantic Web Pada Ensiklopedia Alam”. Skripsi. Malang : Politeknik Negeri Malang
- Rahutomo, F. dan Sarno, R. (2008), “Semantic Search Wikipedia by applying Agent Matcher Architecture”, *Proceedings International Conference on Information & Communication Technology and System*, FTIF Institut Teknologi Sepuluh Nopember, Surabaya, hal. 646-653.
- Seaborne, A. dan Prud'hommeaux, E. 2008. *SPARQL Query Language for RDF*, [Online] Tersedia: <https://www.w3.org/TR/RDF-sparql-query> [16 Maret 2016]
- Tanaamah, Andeka Rocky dan Wellem, Theophilus. 2009. “Semantik Web sebagai Solusi Pemecahan Masalah Promosi Kepariwisata di Indonesia”. Jurnal Teknologi Informasi - Aiti, Vol.6. No. 2, Agustus 2009:101-200. Fakultas Teknologi Informasi Universitas Kristen Satya Wacana
- Zebua, Javier dan Mustikasari, Metty, S.Kom., MSC. “Aplikasi Pencarian Buku Berbasis Web Semantik Untuk Perpustakaan Smk Yadika 7 Bogor”. Skripsi. Jakarta : Universitas Gunadarma.