

REPRESENTASI PENGETAHUAN PADA WEB SEMANTIK UNTUK MENINGKATKAN NILAI EFEKTIFITAS Pencarian DATA SURAT (STUDI KASUS: PT ANGKASA PURA I CABANG SEPINGGAN BALIKPAPAN)

Tri Setia Yoga Nugroho ⁽¹⁾, Herlina Jayadianti ⁽²⁾, Yuli Fauziah ⁽³⁾.

Jurusan Teknik Informatika, Fakultas Teknik Industri, UPN "Veteran" Yogyakarta,
Jalan Babarsari 2, Kec Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281,
Indonesia.

e-mail: trisetiayoga@gmail.com⁽¹⁾

Abstract

PT. Angkasa Pura I branch Sepinggan, Balikpapan still uses the web with a little interactive (browsing or searching for certain information). The system still has some constraints between, the data searched only in accordance with the origin of the sender of the letter. Information gleaned from web searches will make it hard for users to find other broader content, consequently if the search keyword does not match the text contained in the information, the information will not appear. The method used is semantic ontology method, and use protege as ontology design, then use netbean and library Jena as interface design. The methodology used in this final project is the methodology methodology of Guidelines for Rapid APPLICATION Engineering (GRAPPLE). The GRAPPLE methodology is a simplified framework of the Rapid Application Development (RAD) system development method. GRAPPLE methodology is a system development methodology that each stage consists of several actions, and each action produces a UML diagram. UML is a good modeling language to apply in development. UML has the benefit of providing an overview of systems that are easily understood by developers. UML also makes it easy to translate diagram form into coding. The results obtained in making this application shows that the new application can cover the constraints existing in the previous application, so as to increase the effectiveness of searching a letter. Comparison of test results proved that new applications more minimize the constraints that occur.

Keywords: *Semantic Web, Ontology, Protege, Letter*

Abstrak

PT. Angkasa Pura I cabang Sepinggan, Balikpapan masih menggunakan web dengan sedikit interaktif (*browsing* atau mencari informasi tertentu). Sistem tersebut masih terdapat beberapa kendala yang diantara, data yang dicari hanya sesuai dengan asal pengirim surat. Informasi yang didapat dari pencarian web akan menyulitkan pengguna untuk mengetahui konten lain yang lebih luas, akibatnya apabila kata kunci pencarian tidak sesuai dengan teks yang terdapat pada informasi, maka informasi tidak akan muncul. Metode yang digunakan adalah metode *semantic ontology*, dan menggunakan protege sebagai perancangan ontologi, lalu menggunakan *netbean* dan *library Jena* sebagai perancangan *interface*. Metodologi yang digunakan dalam tugas akhir ini adalah metode metodologi *Guidelines for Rapid APPLICATION Engineering* (GRAPPLE). Metodologi GRAPPLE merupakan penyederhanaan kerangka metode pengembangan sistem *Rapid Application Development* (RAD). Schmuller (2004) mengemukakan bahwa metodologi GRAPPLE merupakan metodologi pengembangan sistem yang setiap tahapannya terdiri dari beberapa tindakan, dan setiap tindakannya menghasilkan produk berupa diagram UML. UML merupakan bahasa pemodelan yang bagus untuk diterapkan dalam pengembangan. UML memiliki manfaat dalam memberikan gambaran terkait sistem yang mudah dimengerti oleh pengembang. UML juga memberi kemudahan dalam penerjemahan bentuk diagram menjadi pengkodean (Sakti et al, 2016). Hasil yang didapat dalam pembuatan aplikasi ini menunjukkan, bahwa aplikasi yang baru bisa menutupi kendala-kendala yang ada pada aplikasi sebelumnya, sehingga mampu meningkatkan efektifitas pencarian sebuah surat. Perbandingan pada hasil pengujian terbukti bahwa aplikasi baru lebih meminimalisir kendala yang terjadi.

Katakunci : *Semantic Web, Ontologi, Protege, Surat*

1. PENDAHULUAN

Latar Belakang

Web 3.0. atau dapat dikenal dengan sebutan *semantic web* dapat digambarkan sebagai pengembangan dari *web* saat ini. *Semantic web* tidak hanya bisa di baca oleh manusia, namun dapat dibaca oleh mesin dengan memanfaatkan meta informasi (Davies et al., 2003), tidak seperti *web* biasa yang hanya dapat dimengerti oleh manusia saja

Pengolahan surat menyurat PT. Angkasa Pura I cabang Sepinggan, Balikpapan masih menggunakan *web* dengan sedikit interaktif (*browsing* atau mencari informasi tertentu). Sistem tersebut masih terdapat beberapa kendala yang diantara, data yang dicari hanya sesuai dengan asal pengirim surat. Informasi yang didapat dari pencarian *web* akan menyulitkan pengguna untuk mengetahui konten lain yang lebih luas, akibatnya apabila kata kunci pencarian tidak sesuai dengan teks yang terdapat pada informasi, maka informasi tidak akan muncul. Proses pencarian yang memunculkan informasi dengan konten yang lebih luas, dapat dihasilkan apabila informasi memiliki penambahan makna yang dapat dibaca oleh mesin. Pembuatan aplikasi yang mempresentasikan informasi surat menjadi sebuah pengetahuan (*knowledge*) berbasis *semantic-ontology* diperlukan, sehingga diharapkan dapat melakukan proses pencarian informasi surat menjadi lebih luas, dengan menggunakan satu kata, mesin mampu mempresentasikan dalam berbagai makna.

Tujuan

Tujuan dari penelitian ini adalah membuat rancangan data informasi dokumen surat menjadi sebuah data pengetahuan (*knowledge*) berbasis *semantic-ontology* untuk dapat meningkatkan efektifitas pencarian suatu dokumen surat yang berada di Angkasa Pura I cabang Sepinggan Balikpapan.

Batasan Masalah

Berdasarkan latar belakang, maka penelitian ini memiliki batasan masalah antara lain:

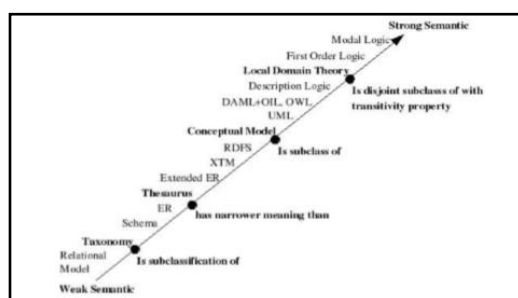
1. Pembuatan *semantic web* dengan menggunakan protégé 3.5 ini menitik beratkan pada pencarian sebuah surat berupa pengetahuan (*knowledge*).
2. Sistem yang akan dibangun difokuskan pada bagaimana mengaplikasikan model *semantic ontology* dengan menggunakan aplikasi protégé 3.5 dan merepresentasikan dalam bentuk pengetahuan.
3. Aplikasi pencarian surat berupa pengetahuan (*knowledge*) dibangun dengan menggunakan bahasa pemrograman *web Java Server Page*, *Jena API*, dan bahasa *query SPARQL*.

2. TINJAUAN PUSTAKA

Ontologi

Menurut Zebua *ontology* istilah ontologi sebenarnya berasal dari istilah filosofi "*ontology*" yang artinya sesuatu yang sesungguhnya ada dan bagaimana menggambarkannya. 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 yang diarahkan dapat memahami makna suatu kata atau kalimat yang diberikan oleh pengguna. (Wijayanto et al., 2013)

Taksonomi (*taxonomy*) adalah sebuah struktur dengan minimal informasi hirarki. *Thesaurus* adalah sebuah pengadaan vokabulari kata, seperti sinonim, antonym, homografi, hirarki, asisiatif, dan sebagainya. *Conceptual* model membawa tingkat *thesaurus* dan taksonomi ke bentuk formal dengan *knowledge* yang semakin kompleks, sedangkan *logical domain theory* dilakukan penambahan fungsi dengan model yang semakin kaya, kompleks, konsisten, dan pengertian yang lebih detail dari *knowledge*.



Gambar 1. Spektrum Ontologi (Daconta et al., 2003)

Semantic Web

Semantic web dicetuskan pertama kali oleh Tim Berners-Lee, James Hendler, dan Ora Lassila, di majalah *Scientific American* pada tahun 2001. Sejak itu, Tim Berners-Lee yang juga merupakan direktur dari W3C membentuk tim kerja yang membidangi *semantic web* serta infrastrukturnya pada tahun 2001.

Semantic web merupakan pengembangan dari *world wide web* dimana *content web* yang ditampilkan tidak hanya dalam Bahasa format manusia yang umum (*natural language*) tetapi juga dapat dibaca oleh mesin (*software*). *Semantic web* memiliki informasi yang dimiliki oleh mesin yang memiliki kecerdasan buatan sehingga untuk menemukan dan mengintegrasikan sebuah informasi dapat dilakukan dengan mudah.

Ontologi Web Language (OWL)

Ontology Web Language (OWL) adalah suatu bahasa yang dapat digunakan oleh aplikasi-aplikasi yang bukan sekedar menampilkan informasi tersebut pada manusia, melainkan juga yang perlu memproses isi informasi. Ontologi sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berisi *classes*, *properties*, dan *instances*. Deskripsi ini dapat membantu sistem komputer dalam menggunakan istilah-istilah tersebut dengan cara yang lebih mudah (Bekke, 1992).

SPARQL

Simple Protocol and RDF Query Language (SPARQL) merupakan bahasa yang digunakan untuk *query* terhadap data sebuah *web*. Komponen dasar dari SPARQL merupakan *triple pattern* seperti RDF, tetapi subjek maupun objeknya dapat berupa variabel (Harris, 2013). Pada SPARQL suatu *variable* ditandai dengan tanda tanya ("?) diikuti dengan nama variabelnya. Pada SPARQL terdapat kondisi yang disebut klausa *WHERE* sebagai data yang ingin diproses untuk dikeluarkan, sedangkan *SELECT* untuk data yang ditampilkan.

3. METODE DAN PERANCANGAN SISTEM

Metodologi Pengembangan Sistem

Metodologi yang digunakan dalam tugas akhir ini adalah metode metodologi *Guidelines for Rapid Application Engineering* (GRAPPLE). Metodologi GRAPPLE merupakan penyederhanaan kerangka metode pengembangan sistem *Rapid Application Development* (RAD). Schmuller (2004) mengemukakan bahwa metodologi GRAPPLE merupakan metodologi pengembangan sistem yang setiap tahapannya terdiri dari beberapa tindakan, dan setiap tindakannya menghasilkan produk berupa diagram UML. UML merupakan bahasa pemodelan yang bagus untuk diterapkan dalam pengembangan. UML memiliki manfaat dalam memberikan gambaran terkait sistem yang mudah dimengerti oleh pengembang. UML juga memberi kemudahan dalam penerjemahan bentuk diagram menjadi pengkodean (Sakti et al, 2016). metodologi pengembangan sistem GRAPPLE terdiri dari segmen-segmen seperti berikut (Manongga et al, 2010)

1. *Requirement Gathering*
Pada tahap pertama yang dilakukan dalam pembuatan aplikasi adalah mengambil informasi lengkap, analisis masalah, fungsi, dan kebutuhan sistem termasuk dalam tahap *Requirement Gathering*.
2. *Analysis*
Di tahap *analysis* yang dilakukan adalah menggali lebih dalam hasil yang diperoleh dalam tahap sebelumnya. Tahap ini mengkaji permasalahan pengguna dan menganalisis solusinya.
3. *Design*
Tahap *design* dilakukan untuk merancang solusi yang dihasilkan oleh tahap *analysis* dan *design* dapat berjalan dua arah saling menyesuaikan sampai diperoleh rancangan yang tepat. Termasuk dalam tahap ini antara lain adalah implementasi model dan diagram..
4. *Development*
Tahap ini ditangani oleh pengembang program untuk membangun *code* program dan *user interface*. Pengujian program dan dokumentasi sistem dilakukan pada tahap ini.
5. *Deployment*
Tahap *deployment* adalah tahap pendistribusian produk yang dihasilkan kepada pengguna. Tahap ini mencakup instalasi dan perencanaan *backup* data bila diminta oleh pengguna sesuai dengan perjanjian sebelumnya.

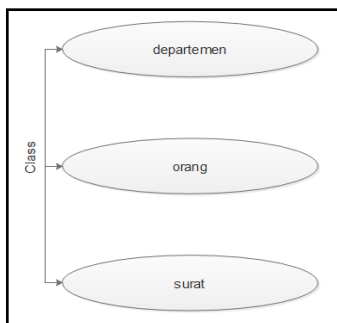
Pada penelitian ini tahapan yang dikerjakan hanya sampai tahap pengembangan program atau *development*.

Perancangan Ontologi

Perancangan Ontologi berlandaskan dari beberapa aturan dasar yang nantinya membentuk sebuah satu kesatuan berupa hirarki, dan dalam proses pembuatannya terdapat beberapa tahapan yaitu:

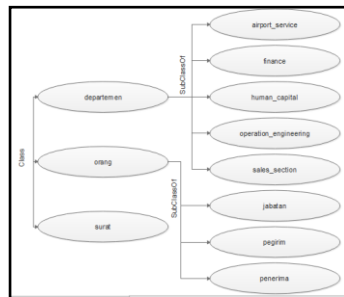
Perancangan Class

Pada pembuatan sebuah *class*, hal yang harus dilakukan pertama adalah membuat sebuah *class* baru, dalam Gambar 2. menunjukkan rancangan *class* pada ontologi yang sebelumnya telah dibuat.



Gambar 2. Rancangan *Class* pada Ontologi

Kemudian pada gambar selanjutnya yaitu Gambar 3. memperlihatkan rancangan *subclass* yang terdapat pada ontologi yaitu terletak pada *class* departemen dan *class* orang. Pada *class* departemen memiliki *subclass* yang diantaranya *airport_service*, *finance*, *human_capital*, *peration_engineering*, dan *sales_section*. Pada *class* orang memiliki *subclass* yang diantaranya jabatan, pengirim dan penerima.



Gambar 3. Rancangan *Subclass* pada Ontologi

Kemudian masuk ke dalam pendefinisian dari masing-masing *class* dan *subclass* dalam ontologi yang diantaranya:

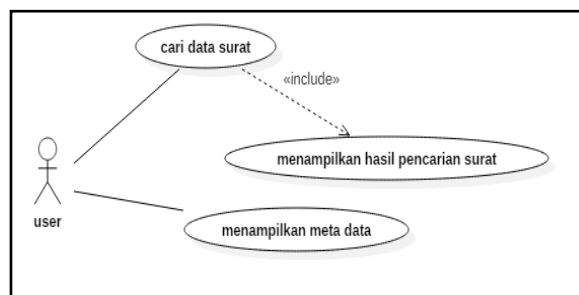
1. *Class* Departemen, merupakan *class* yang mempresentasikan pengetahuan tentang departemen-departemen apa saja yang nantinya akan menerima data surat yang masuk ke perusahaan.
2. *Class* orang, merupakan *class* yang mempresentasikan pengetahuan tentang siapa saja yang bersangkutan dalam ontologi dan memiliki berbagai peran yang diantaranya sebagai pengirim dan penerima.
3. *Class* Surat, merupakan *class* yang mempresentasikan pengetahuan tentang data dari surat baik itu berupa no_surat, perihal, tanggal_agenda, maupun alamat_surat.
4. *SubClass* Penerima, merupakan *subclass* yang mempresentasikan pengetahuan tentang penerima, penerima disini bertugas menerima surat yang masuk ke dalam perusahaan, penerima dapat didefinisikan sebagai sekretaris dari departemen yang bersangkutan.
5. *SubClass* Pengirim, merupakan *subclass* yang mempresentasikan pengetahuan tentang data-data pengirim yang masuk ke dalam perusahaan, yang nantinya akan dimasukkan ke dalam protégé.
6. *SubClass* gelar, merupakan *subclass* yang mempresentasikan pengetahuan tentang jabatan dari *class* orang.

Unified Modeling Language (UML)

Perancangan yang dibangun menggunakan UML dikarenakan pemodelan aplikasi yang berorientasi objek, rancangan yang dibangun terdiri dari tiga diagram, yang diantaranya adalah *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

Use Case Diagram

Use Case Diagram yang dibangun menggambarkan dua kegiatan yang dilakukan oleh pengguna yaitu mencari sebuah dokumen surat, dan menampilkan meta data yang terdapat dalam protege ke *web*, ditambah dengan satu kegiatan pada saat mencari sebuah dokumen surat yaitu menampilkan hasil pencarian surat.



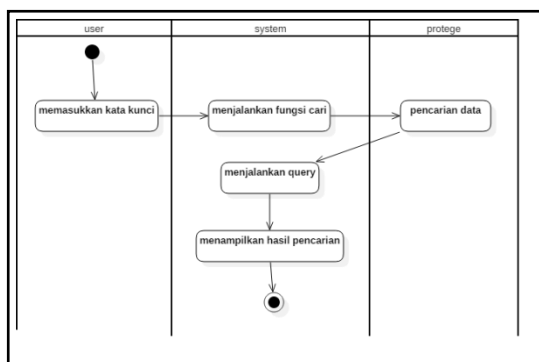
Gambar 4. *Use Case Diagram*

Activity Diagram

Activity Diagram yang dibuat dalam perancangan aplikasi ini menjelaskan tentang aktivitas yang dilakukan dalam sistem, pada tahap ini terdapat dua aktivitas yang dapat dilakukan dan dijelaskan, diantaranya:

1. *Activity Diagram* Pencarian Surat

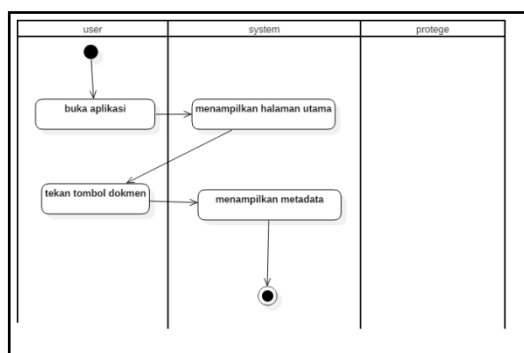
Aktivitas yang terjadi dalam proses ini adalah pengguna memasukkan kata kunci kemudian sistem akan membaca dan menjalankan fungsi cari, setelah itu mengirimkannya ke pencarian data yang terdapat di protege, lalu menjalankan *query* yang telah dibangun dan menampilkan hasilnya pada halaman hasil pencarian.



Gambar 5. Activity Diagram Pencarian Surat

2. *Activity Diagram* menampilkan metadata

Pada aktivitas selanjutnya adalah menampilkan sebuah metada. Proses yang terjadi adalah pengguna membuka aplikasi, sehingga menuju tampilan awal aplikasi, kemudian menekan tombol dokumen, dan menampilkan metadata.



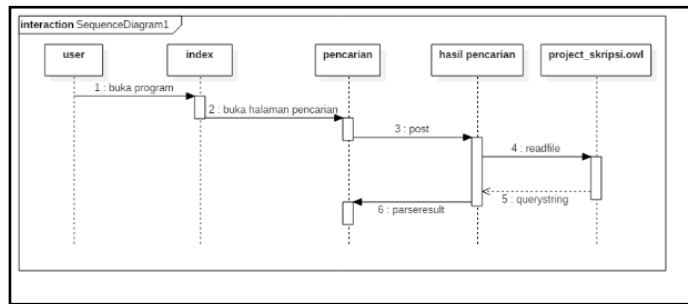
Gambar 6. Activity Diagram menampilkan metadata

Sequence Diagram

Sequence Diagram pada perancangan yang dibangun dalam aplikasi ini menjelaskan interaksi antar objek, terdapat dua interaksi rancangan yang dibangun, diantaranya:

1. *Sequence Diagram* Pencarian Surat

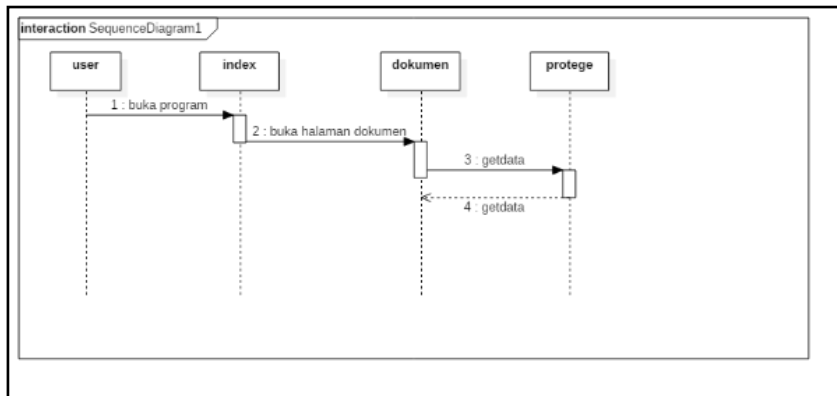
Interaksi objek yang terdapat pada fitur pencarian surat adalah *class* index akan menjalankan perintah dari pengguna untuk membuka program, kemudian pada *class* pencarian, pengguna akan menginputkan kata kunci, lalu menuju proses pengambilan data pada *class* hasil pencarian yang datanya terdapat pada *class* project_skripsi.owl, kemudian hasilnya akan dikembalikan ke halaman hasil pecarian.



Gambar 7. Sequence Diagram Pencarian Surat

2. *Sequence Diagram* menampilkan metadata

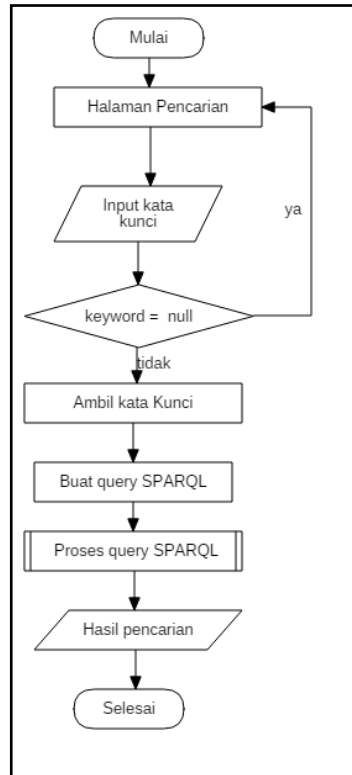
Interaksi objek yang terdapat pada fitur menampilkan metadata adalah *class* index akan menjalankan perintah dari pengguna untuk membuka program, kemudian pada *class* dokumen akan menampilkan metadata yang sebelumnya diambil dari protege.



Gambar 8. Sequence Diagram menampilkan metadata

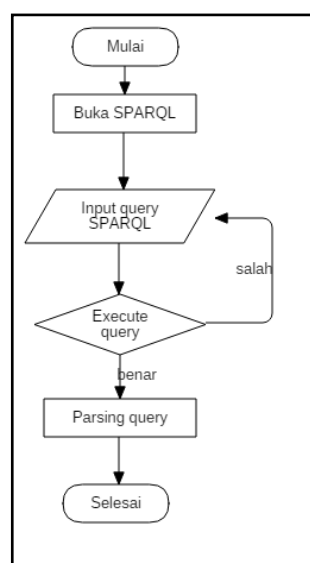
Perancangan Diagram Alir

Pada Perancangan Diagram alir (*Flowchart*) ini akan menjelaskan sebuah alur proses dan prosedur yang terjadi dalam protégé, sehingga dapat digambarkan algoritma yang terjadi.



Gambar 9. Flowchart pencarian dokumen surat

Dari perancangan di atas dapat dijelaskan bahwa mula-mula masuk ke dalam halaman awal dari program, kemudian masukkan kata kunci yang diinginkan, masuk ke dalam kondisi dimana jika kata kunci yang dimasukkan tidak terdaftar, maka akan masuk ke proses memasukkan kata kunci kembali, namun jika kata kunci terdaftar, maka masuk ke proses selanjutnya, dimana program akan mengambil kata kunci. Lalu proses selanjutnya pembuatan *query* SPARQL yang nantinya akan diproses, kemudian dokumen dari *query* tersebut akan *diparsing* (penguraian kalimat) dan yang terakhir akan masuk ke dalam halaman hasil pencarian dari data surat.



Gambar 10. Flowchart proses query SPARQL

Proses *query* dimulai dari membuka halaman SPARQL yang berada di *menu* yang ada di protégé, kemudian *input query* yang telah tersedia di kolom SPARQL, yang di *inputkan* adalah SELECT yang berisi ?subject ?object dan WHERE yang berisi ?subject rdfs:subClassOf ?object, kemudian masuk ke proses *execute* dimana jika *query* benar data akan diuraikan dan menampilkan hasil yang diinginkan.

4. IMPLEMENTASI DAN PENGUJIAN

Implementasi Halaman Awal

Pada halaman awal berisi sebuah *menu* yang terletak di bagian kanan atas, *menu* tersebut diantaranya adalah *menu* utama, bantuan, dan dokumen, dan untuk memulai pencarian, yang perlu dilakukan adalah menekan tombol yang terletak di bawah tulisan pencarian surat, dan implementasinya seperti pada Gambar 11.



Gambar 11. Implementasi Halaman Awal

Implementasi Halaman Pencarian

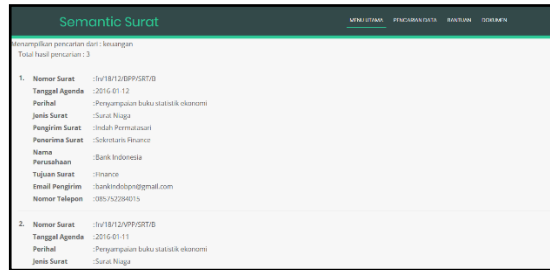
Halaman pencarian merupakan halaman yang digunakan untuk memasukkan sebuah kata kunci untuk mencari sebuah informasi surat yang diinginkan. Untuk memasukkan kata kunci, yang perlu dilakukan adalah mengisi *field* yang telah disediakan pada tampilan, lalu menekan tombol cari yang berwarna hijau, berikut implementasinya pada Gambar 12.



Gambar 12. Implementasi Halaman Pencarian

Implementasi Halaman Hasil Pencarian

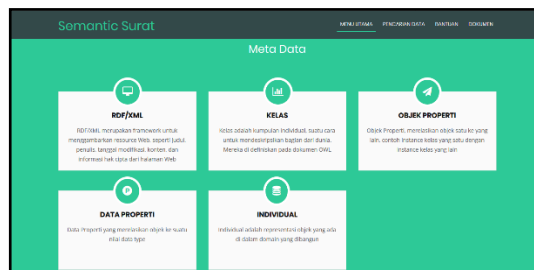
Pada halaman ini akan menampilkan hasil dari pencarian kata kunci yang sebelumnya telah dimasukkan pada halaman pencarian. Hasil dari pencarian berupa informasi sesuai dengan keinginan pengguna, seperti nomor surat, tanggal agenda, perihal, jenis surat, pengirim surat, penerima surat, nama perusahaan, tujuan surat, email pengirim, dan nomor telepon. Sebagai contoh pencarian dengan kata kunci "keuangan", dan hasilnya dapat dilihat pada Gambar 13.



Gambar 13. Implementasi Halaman Pencarian

Implementasi Halaman Dokumen

Pada halaman dokumen berisi metadata yang berfungsi sebagai informasi mengenai *source code* dari ontologi yang dibangun, contoh dari metadata tersebut adalah metadata dari RDF/XML, kelas, objek properti, data properti dan individual. Berikut implementasinya pada Gambar 14.



Gambar 14. Implementasi Halaman Dokumen

Implementasi Halaman Bantuan

Pada halaman bantuan berisi bagaimana cara mencari sebuah informasi, sehingga dengan adanya halaman ini dapat membantu pengguna dalam pencarian. Berikut implementasinya pada Gambar 15.



Gambar 15. Implementasi Halaman Bantuan

Pengujian Sistem

Setelah semua perancangan yang telah dilakukan pada bab-bab sebelumnya, perlu dilakukan sebuah pengujian terhadap sistem yang dibangun. Pengujian dilakukan berdasarkan perbandingan antara web yang sebelumnya dengan web yang dibangun. Berikut Tabel 1 merupakan tabel hasil pengujian sistem.

Tabel 1. Hasil Pengujian Sistem

No	Pengujian	Aplikasi lama	Aplikasi baru
1.	Pencarian dalam konteks yang lebih luas	Tidak menemukan hasil	Menemukan hasil
2.	Hasil pencarian berupa pengetahuan(knowledge)	Hanya menampilkan informasi	Menampilkan pengetahuan(knowledge)
3.	Mesin dapat memahami pengguna	Tidak memahami	Memahami
4.	Menampilkan informasi	Menampilkan	Menampilkan
5.	Mempermudah pengguna	Sedikit mempermudah	Mempermudah
6.	Pencarian kata "Finance"	Menampilkan kata "Finance"	Menampilkan kata "Finance"
7.	Pencarian kata "keuangan"	Tidak menemukan hasil	Menampilkan kata "Finance"
8.	Relasi antar class	Hanya sesuai dengan tabel	Memiliki relasi yang luas
9.	Pemanfaatan protégé	Tidak menggunakan	Menggunakan
10.	Pengembangan aplikasi	Dapat dilakukan	Dapat dilakukan

Dari hasil pengujian yang telah dilakukan, dapat diketahui bahwa aplikasi yang baru dapat memiliki beberapa keunggulan dari aplikasi sebelumnya, sehingga pengembangan aplikasi dapat dikatakan berjalan dengan lancar. Adapun saran dari pengujian ini antara lain dengan memanfaatkan sebuah metode tambahan, aplikasi ini mampu meminimalisir batasan masalah yang sebelumnya telah dibuat, dan penambahan aplikasi yang mampu diakses melalui sistem operasi *Android* maupun *IOS*.

KESIMPULAN

Dari proses analisis, perancangan dan implementasi yang dilakukan pada bab-bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut:

1. Implementasi dari pencarian surat yang sebelumnya berupa sebuah informasi dapat menjadi pengetahuan yang memiliki makna lebih luas, dengan begitu dapat mempermudah pengguna dalam menggunakan aplikasi ini.
2. Hasil yang didapat dalam pembuatan aplikasi ini menunjukkan, bahwa aplikasi yang baru bisa menutupi kendala-kendala yang ada pada aplikasi sebelumnya, sehingga mampu meningkatkan efektifitas pencarian sebuah surat. Perbandingan pada hasil pengujian terbukti bahwa aplikasi baru lebih meminimalisir kendala yang terjadi.

SARAN

Beberapa saran yang dapat diberikan sebagai hasil tugas akhir ini adalah:

1. Konsep-konsep yang ada pada ontologi pencarian dokumen surat ini dapat dikembangkan dan diperluas lagi agar dapat mencakup semua pengetahuan tentang domain surat menyurat.
2. Perlu adanya pengembangan sistem untuk menciptakan aplikasi yang dapat melakukan penambahan *instance* dan pemberian nilai-nilai *property* pada pencarian dokumen surat sehingga dapat mempermudah pengolahan basis pengetahuan yang ada.

3. Perlu dikembangkan menggunakan sebuah metode yang lain, seperti metode *semantic web rule language*(SWRL), SWRL merupakan perluasan dari OWL dengan menambahkan aturan berbasis *Horn*. Hal ini memungkinkan suatu basis pengetahuan yang dibuat dalam OWL dilengkapi dengan aturan.

DAFTAR PUSTAKA

- A. Y. N. Sakti, Y. Priyandari, and M. Hisjam, "Perancangan Sistem Informasi Manajemen Auditor Teknologi pada Sistem Nasional Audit Teknologi (SNAT) Menggunakan Metode Grapple," 2016.
- H. Wijayanto, W. L. Y. Saptomo, and T. Susyanto, "PENERAPAN WEB SEMANTIK DALAM PENCARIAN KATALOG BUKU DI PERPUSTAKAAN STMIK SINAR NUSANTARA SURAKARTA," *J. Teknol. Inf. Dan Komun. TIKomSiN*, vol. 1, no. 1, 2013.
- D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, *Spinning the semantic web: Bringing the World Wide Web to its full potential*. MIT press, 2008.
- M. C. Daconta and L. Obrst, *The Semantic Web: a guide to the future of XML, Web services, and knowledge management*. Wiley Indianapolis, 2003.
- H. Wijayanto, W. L. YS, and T. Susyanto, "Penerapan Web Semantik Dalam Pencarian Katalog," *TIKomSin*, pp. 60–68, 2013.
- D. Manongga, S. Papilaya, and S. Pandie, "Sistem Informasi Geografis Untuk Perjalanan Wisata di Kota Semarang," *J. Inform.*, vol. 10, no. 1, pp. 1–9, 2010.
- M. Nurkamid, "Aplikasi Bibliografi Perpustakaan Berbasis Teknologi Web Semantik," *Sains Dan Teknol.*, vol. 2, no. 2, 2009.
- A. Andri, "Aplikasi Konversi Basisdata Relasional Mysql Ke Ontologi Sebagai Basis Pengetahuan Aplikasi Berbasis Web Semantik," *J. Ilm. Matrik*, vol. 14, no. 1, pp. 1–10, 2012.
- Z. Eri and M. Khabib, "Crawling Web berdasarkan Ontology." 2009.
- [R. Neches *et al.*, "Enabling technology for knowledge sharing," *AI Mag.*, vol. 12, no. 3, p. 36, 1991.