

## PENERAPAN TEKNOLOGI SEMANTIC WEB UNTUK MENENTUKAN PILIHAN JALUR BIS TRANS JOGJA

Indra Aji Setyo Wicaksono<sup>1</sup>, Novrido Charibaldi<sup>2</sup>, Herlina Jayadianti<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknik Informatika UPN "Veteran" Yogyakarta

Jl. Babarsari no 2 Tambakbayan 55281 Yogyakarta Telp (0274)-485323

e-mail: novrido@gmail.com, herlinajayadianti@gmail.com, a\_ztje@yahoo.com

### Abstrak

*Trans Jogja adalah upaya Pemerintah Provinsi Daerah Istimewa Yogyakarta (DIY) untuk meningkatkan pelayanan publik khususnya pada sektor transportasi darat dikawasan perkotaan DIY dengan berbasis bis, mengganti sistem setoran menjadi sistem pembelian pelayanan bis yang berhenti di halte-halte khusus yang mirip dengan busway Trans Jakarta. Namun persoalan muncul jika pencarian mengenai informasi bis Trans Jogja harus sesuai dengan pengetahuan masing-masing pengguna. Pada penelitian ini, pengetahuan-pengetahuan mengenai data jalur bis Trans Jogja disimpan ke dalam model ontologi berbasis Web Ontology Language. Software yang digunakan untuk membangun sistem ini meliputi Protégé, NetBeans, Jena API sebagai library-nya. Bahasa pemrograman menggunakan Java Server Page (JSP) dan bahasa query menggunakan SPARQL.*

**Kata kunci** : *semantic web, ontologi, Jena API, Protégé, jalur bis Trans Jogja*

### 1. PENDAHULUAN

Kini hadir sebuah sarana transportasi baru di Yogyakarta, sarana transportasi ini bernama *busway* atau dikenal Trans Jogja. Trans Jogja adalah upaya Pemerinatah Provinsi Daerah Istimewa Yogyakarta (DIY) untuk meningkatkan pelayanan publik khususnya pada sektor transportasi darat dikawasan perkotaan DIY dengan berbasis bis, mengganti sistem setoran menjadi sistem pembelian pelayanan bis yang berhenti di halte-halte khusus. Sebuah sarana transportasi yang mirip dengan *busway* Trans Jakarta.

Salah satu upaya untuk mengembangkan *web browser* yang dapat mencari data berdasarkan berbagai persepsi pengguna adalah dengan menggunakan pendekatan model ontologi. *Web browser* yang menggunakan penyimpanan pengelolaan pengetahuan dengan konsep model ontologi dapat disebut sebagai *semantic web*. Hal ini dikarenakan *web browser* tersebut menyimpan makna dari sebuah objek dan tidak hanya menyimpan sekedar data. *Semantic web* digambarkan sebagai pengembangan dari *web* saat ini, selain dapat dibaca oleh manusia menggunakan *web browser*, *web* dianotasi menggunakan meta informasi. Meta informasi ini menerangkan tentang *web* tersebut sehingga dapat diproses oleh mesin (Davies, dkk., 2006).

Dalam penelitian ini mencoba menerapkan teknologi baru, yaitu bagaimana membangun dan mengelola pengetahuan dengan berbasis model ontologi. Sehingga diharapkan masyarakat dapat dengan mudah mendapatkan informasi jalur bis Trans Jogja yang akan. Penyimpanan informasi jalur bis Trans Jogja dengan model ontologi mampu menyimpan data secara lebih semantik sehingga nantinya *user* dapat melakukan pencarian *query* atau pemanfaatan kembali data tersimpan sesuai dengan persepsi masing-masing pengguna terhadap informasi bis Trans Jogja.

### 2. TINJAUAN PUSTAKA

Penelitian yang memiliki tema penerapan *semantic web* ini pernah dilakukan oleh Mercy Lambe' Patayang dengan judul Pengelolaan Pengetahuan Data Perumahan Di Yogyakarta Berbasis OWL (*Ontologi Web Language*) dari Universitas Pembangunan Nasional "Veteran" Yogyakarta. Penelitian tersebut dilakukan dengan metodologi *ontological engineering* yaitu *basic topic, design, development, application, dan knowledge sharing and reuse*. Sedangkan metodologi pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah dengan menggunakan metode *waterfall* yaitu analisis, *design*, implementasi serta pengujian.

Teknologi yang digunakan dalam penelitian ini dan pada penelitian yang dilakukan oleh Mercy Lambe' Patayang sama-sama menggunakan teknologi semantik. Namun yang membedakan penelitian ini dengan penelitian sebelumnya yaitu di dalam penelitian ini akan dibangun sebuah aplikasi yang akan menerapkan konsep semantik, sedangkan dalam penelitian sebelumnya sistem yang dibangun hanya sampai pada tahap pembangunan ontologi. Sedangkan ontologi merupakan salah satu tahapan untuk membangun sebuah *semantic web* itu sendiri.

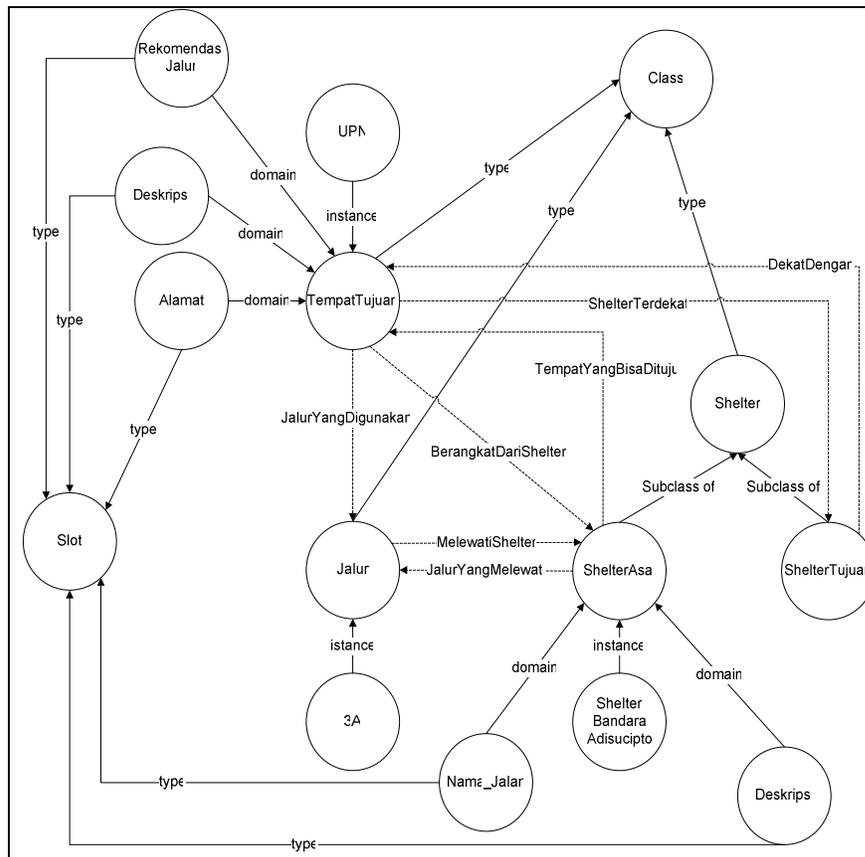
Bahasa pemrograman yang dipakai pada penelitian sebelumnya menggunakan OWL, sedangkan pada penelitian ini selain menggunakan OWL juga menggunakan bahasa pemrograman JSP (*Java Server Pages*) untuk merancang *interface*-nya. *Tools* yang digunakan untuk merancang *interface* menggunakan NetBeans IDE 6.1 dan dengan penambahan Jena API sebagai *library*-nya. Kemudian *tools* yang digunakan dalam perancangan *knowledge base* sama-sama menggunakan Protégé 3.2.

### 3. METODE PENELITIAN

Pada segmen ini perancangan dibagi menjadi dua yaitu perancangan ontologi dan perancangan sistem. Perancangan ontologi untuk merepresentasikan basis pengetahuan, sedangkan rancangan sistem merupakan gambaran atau perancangan dari sebuah sistem pada perangkat lunak yang terdiri dari rancangan proses, perancangan *flowchart* dan rancangan antar muka.

#### 3.1 Perancangan Ontologi

Perancangan model ontologi jalur bis Trans Jogja secara keseluruhan ditunjukkan pada Gambar 1.



Gambar 1. Rancangan Model Ontologi Pencarian Jalur bis Trans Jogja

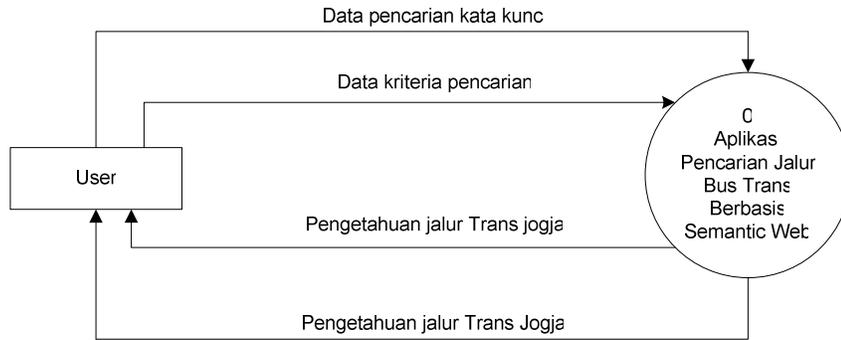
#### 3.2 Perancangan Sistem

Pada perancangan sistem ini dilakukan untuk menggambarkan suatu model aplikasi yang nantinya digunakan dalam pengembangan sistem. Dalam hal ini sistem dirancang agar dapat melakukan proses pencarian jalur trans berdasarkan kata kunci dan kriteria pencarian dimana pada kriteria pencarian, *user* dapat memilih sendiri pengetahuan yang diinginkan.

##### 3.2.1 Rancangan Proses

Pada penelitian ini rancangan proses menggunakan bentuk *Data Flow Diagram* (DFD) sebagai berikut :

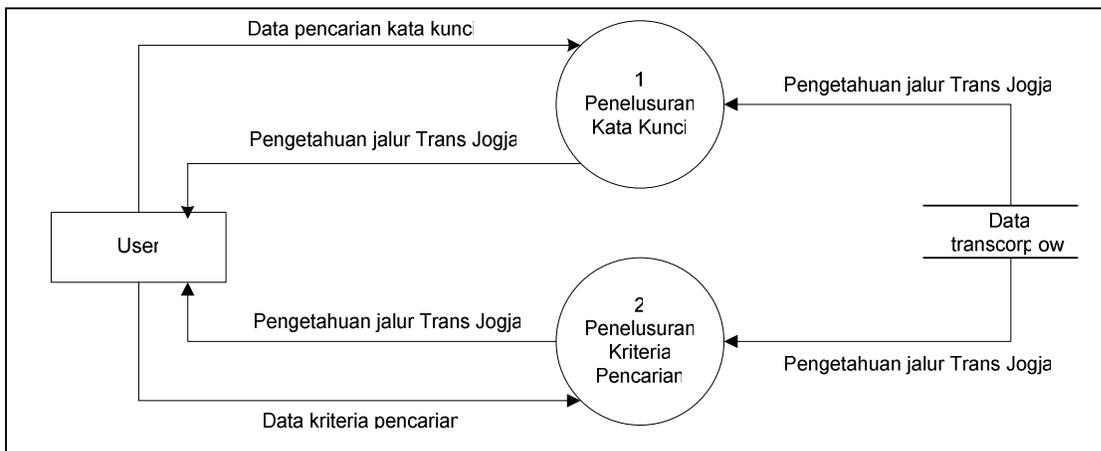
1. DFD Level 0



Gambar 2. DFD Level 0

Pada DFD level 0 ini terdapat satu buah entitas, yaitu entitas *user*. Selain itu juga terdapat satu buah proses yaitu aplikasi pencarian jalur bus Trans Jogja itu sendiri. Pada DFD tersebut terdapat dua aliran data yang masuk (*input*) dan dua aliran data keluar (*output*). Dalam aplikasi, *user* dapat memilih pencarian jalur dengan memasukkan kata kunci atau bisa juga berdasarkan kriteria pencarian. Setelah kata kunci atau kriteria pencarian dimasukkan akan menghasilkan informasi jalur bus Trans Jogja beserta pengetahuan yang terkait.

2. DFD Level 1



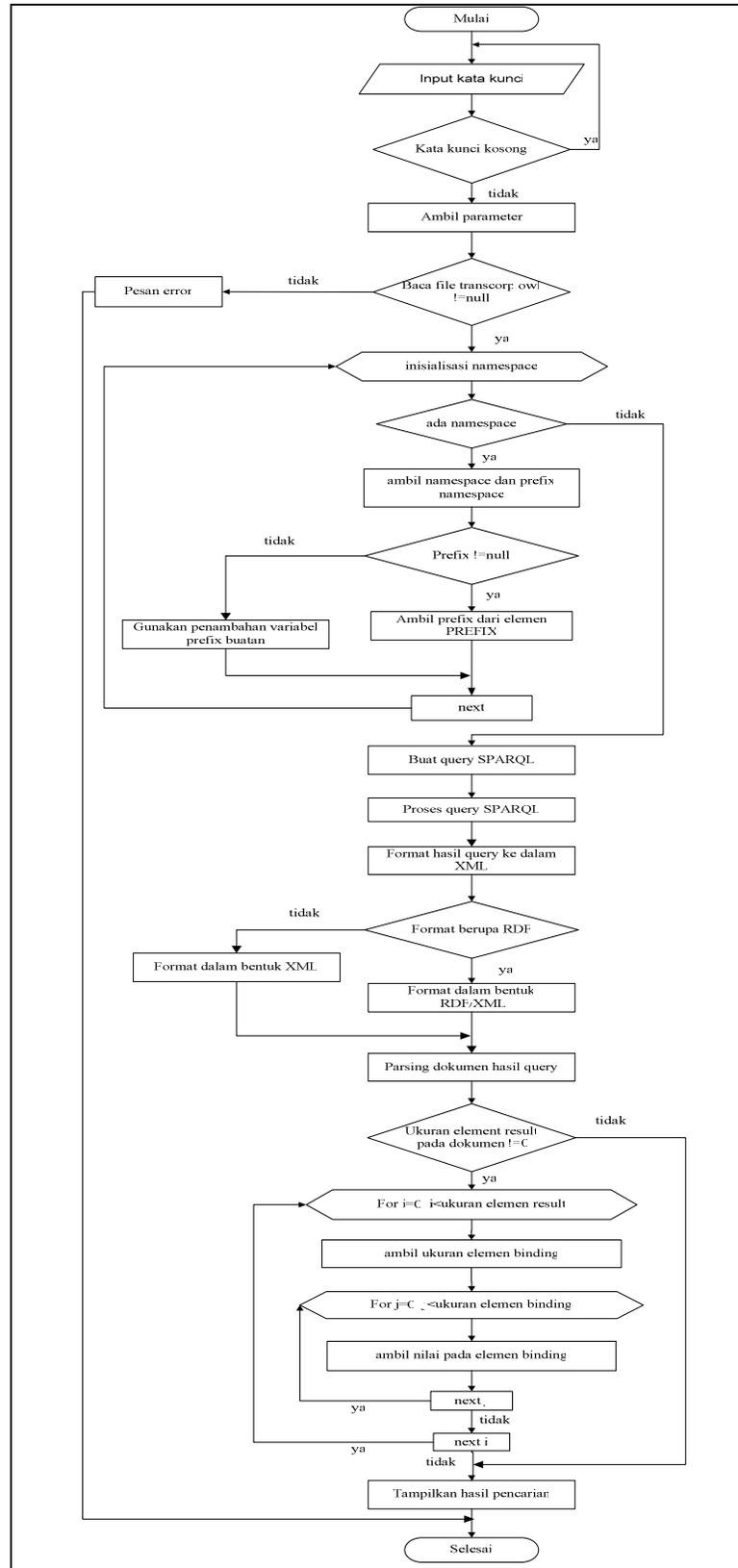
Gambar 3. DFD Level 1

DFD level 1 ini merupakan penjabaran lebih detail dari DFD level 0. Pada DFD level 1 terdapat 2 buah proses yaitu proses penelusuran kata kunci dan proses penelusuran kriteria. Masing-masing proses memiliki satu buah *input* dan satu buah *output*.

3.2.2 Rancangan Flowchart

1. Flowchart Pencarian Kata Kunci

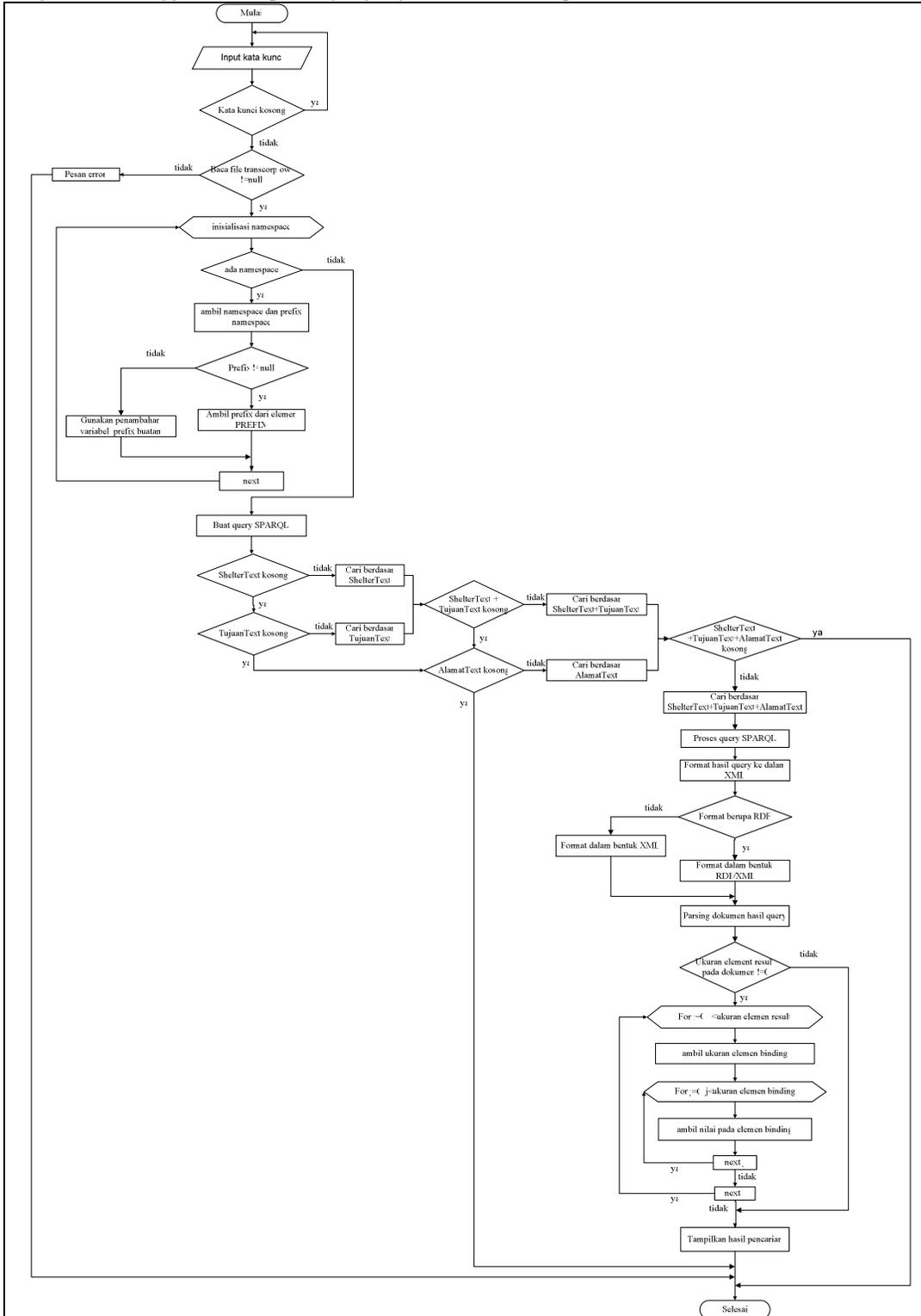
Setelah *user* mentukan ingin mencari informasi jalur Trans Jogja dengan menggunakan pencarian kata kunci, maka *flowchart* pencarian kata kunci ini berfungsi untuk menggambarkan proses yang terjadi dalam pencarian kata kunci dari memasukkan parameter *text* (data *text*) dengan mencocokkan parameter yang ada pada file *transcorp.owl* sampai menemukan hasil pencarian.



Gambar 4. Flowchart Pencarian Kata Kunci

## 2. Flowchart Kriteria Pencarian

Pada *flowchart* ini hampir sama seperti *flowchart* pencarian kata kunci. Perbedaannya hanya pada saat proses parsing dokumen hasil *query*. Hal ini dikarenakan pada metode pencarian dengan menggunakan kriteria pencarian lebih spesifik untuk mencari data dari file *transcorp.owl*. Setelah *user* menentukan ingin mencari informasi jalur Trans Jogja dengan menggunakan kriteria pencarian, maka *flowchart* kriteria pencarian ini berfungsi untuk menggambarkan proses yang terjadi dalam kriteria pencarian.



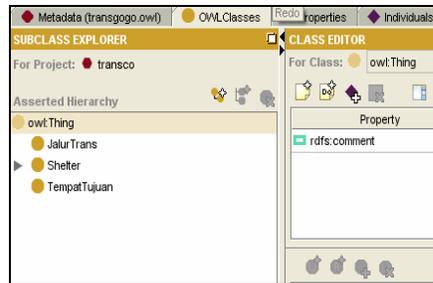
Gambar 5. Flowchart Kriteria Pencarian

## 4. IMPLEMENTASI

### 4.1 Implementasi Ontologi Jalur Bis Trans Jogja

#### 4.1.1 Implementasi Class

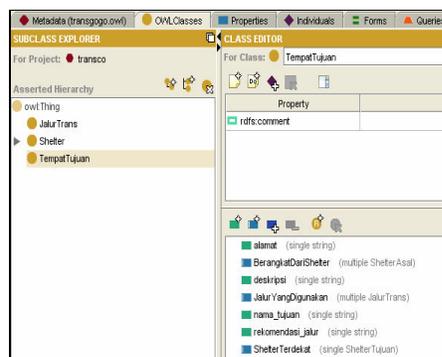
Pembuatan *class* beserta hirarkinya pada ontologi jalur bis Trans Jogja mengacu pada perancangan *class-class* yang ada pada bab sebelumnya. *Class* pada ontologi jalur bis trans Jogja ini terdiri dari tiga *class* utama yaitu JalurTrans, Shelter dan TempatTujuan. *Class* Shelter terbagi menjadi dua *subclass* yaitu *class* ShelterAsal dan *class* ShelterTujuan. Setelah tiga *class* utama tersebut selesai dibuat, maka *class-class* yang tidak memungkinkan memiliki *instance* yang sama dibuat saling *disjoint*. Dalam ontologi jalur bis Trans Jogja, *instance-instance* yang ada saling terhubung maka tidak perlu penambahan *disjoint*. Gambar 6 memperlihatkan implementasi *class-class* yang sudah dibuat ke dalam *tools* Protégé.



Gambar 6. Class-class pada Ontologi Jalur Bis Trans Jogja

#### 4.1.2 Implementasi Property pada Class

Pembuatan properti-properti pada *class* bisa dilakukan melalui tab *Properties*. Dalam Protégé 3.2 ini telah dibuat subtab dari tab *Properties* yang terdiri dari tab *Object*, tab *Datatype*, tab *Annotation*, dan tab *All*. Tab *Object* digunakan untuk membuat *object properties*, tab *Datatype* digunakan untuk membuat *datatype properties*, tab *Annotation* digunakan untuk membuat *annotation properties*, sedangkan tab *All* digunakan untuk membuat semua jenis properti yang terintegrasi dalam satu jendela. Gambar 7 memperlihatkan implementasi *class-class* yang sudah dibuat ke dalam *tools* Protégé.



Gambar 7. Properti-properti pada Class TempatTujuan

Pada Gambar 7 dapat dilihat bahwa pada *class* TempatTujuan terdapat properti-properti BerangkatdariShelter, JalurYangDigunakan dan ShelterTerdekat bertipe *object type*. Sedangkan properti-properti alamat, deskripsi, nama\_tujuan dan rekomendasi\_jalur bertipe *datatype*.

#### 4.1.3 Implementasi Pembuatan Instance

Setelah desain hirarki dan relasi *class-class* pada ontologi jalur bis Trans Jogja selesai dibuat, maka langkah selanjutnya adalah membuat *instance-instance* untuk masing-masing *class* tersebut. Pembuatan *instance* bisa dilakukan menggunakan Protégé melalui tab *Individuals*. Tab *Individuals* ini terdiri atas tiga jendela utama yaitu jendela *Class Browser* untuk menampilkan *class-class* yang ada pada sebuah ontologi, jendela *Instance Browser* untuk menampilkan *instance-instance* yang dimiliki oleh sebuah *class*, dan jendela *Individual Editor* untuk memasukkan nilai-nilai properti dari sebuah *instance*. Bagaimana suatu *instance* akan ditampilkan, serta *form* yang digunakan untuk mengisi nilai properti dapat diatur pada tab *Forms* yang disediakan Protégé. Contoh *instance*

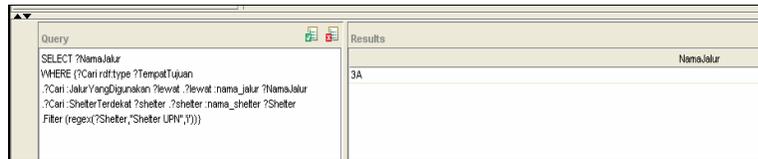
pada ontologi jalur bis Trans Jogja dapat dilihat pada Gambar 8 merupakan salah satu contoh pengisian *instance* pada *class* JalurTrans yaitu jalur 1A.



Gambar 8. Nilai Properti untuk *instance* "1A"

#### 4.1.4 Implementasi Query

Protégé 3.2 telah terintegrasi dengan SPARQL sehingga pada Protégé 3.2 dapat dilakukan peng-*query*-an menggunakan bahasa SPARQL. Pada pengujian ontologi jalur bis Trans Jogja di dalam Protégé, bisa dilakukan melalui *SPARQL Query Panel* dengan memilih menu *OWL - Open SPARQL Query Panel.... SPARQL Query Panel* akan muncul di bagian bawah pada jendela Protégé. Pada Gambar 9 memperlihatkan hasil *query* sederhana dan *code query* SPARQL dapat dilihat pada Modul Program 1.



Gambar 9 Contoh Query Sederhana

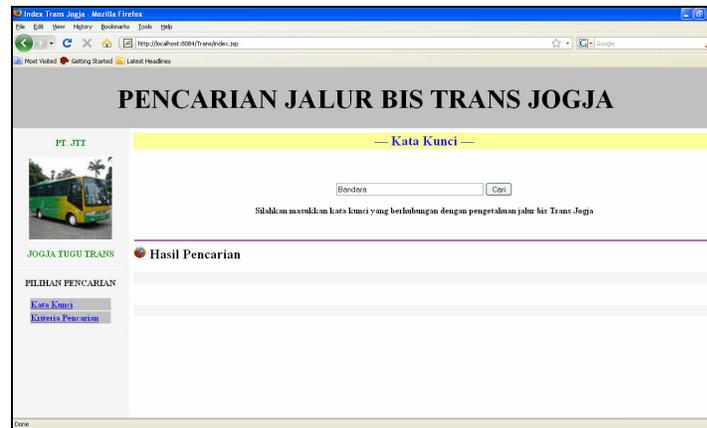
```
SELECT ?NamaJalur
WHERE {?Cari rdf:type ?TempatTujuan
.?Cari :JalurYangDigunakan ?lewat .?lewat :nama_jalur ?NamaJalur
.?Cari :ShelterTerdekat ?shelter .?shelter :nama_shelter ?Shelter
.Filter (regex(?Shelter,"Shelter UPN",'i'))}
```

Modul Program 1 Contoh Query Sederhana

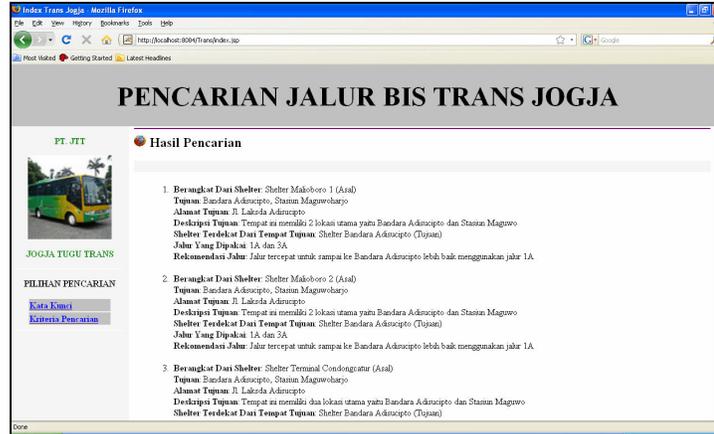
#### 4.2 Tampilan Halaman Utama

Aplikasi pencarian kata kunci akan ditampilkan pada halaman utama saat pertama kali aplikasi dijalankan pada sebuah *web browser*. Teknik pencarian berdasar kata kunci akan meminta *user* untuk memasukkan sebuah kata kunci pada sebuah *text-entry field* yang tersedia. Kata kunci yang dimasukkan akan dicocokkan dengan nilai properti pengetahuan yang dimiliki oleh setiap pengguna bis Trans Jogja.

Sebagai contoh seorang *user* yang ingin mencari jalur Trans Jogja yang berdasarkan tempat tujuan, maka *user* harus memasukkan kata "bandara" pada *text-entry field* bernama *keyword*, seperti terlihat pada Gambar 10. Dan pada Gambar 11 memperlihatkan hasil pencarian dari kata kunci "bandara".



Gambar 10. Aplikasi Pencarian Jalur Bis Trans Jogja dengan Kata Kunci

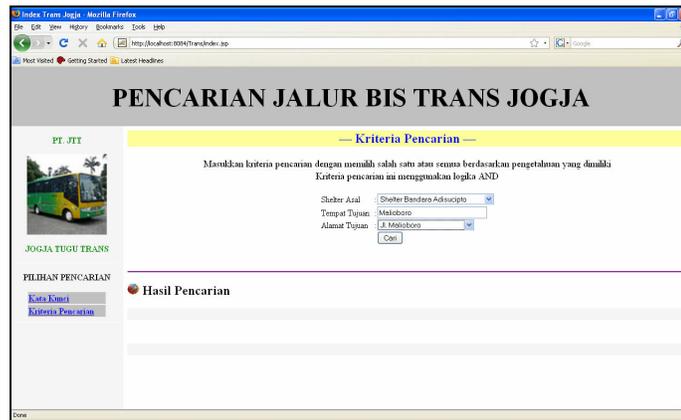


Gambar 11. Hasil Pencarian untuk Kata Kunci "bandara"

#### 4.3 Kriteria Pencarian (*Advanced Search*)

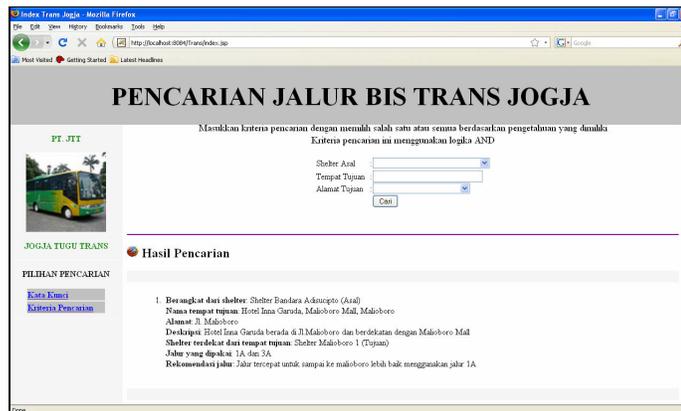
Jika pilihan kriteria pencarian diklik maka akan ditampilkan halaman `indexadvance.jsp`. Halaman ini berisi *form* untuk melakukan pencarian dengan memasukkan kriteria pencarian yang lebih beragam. Teknik kriteria pencarian menyediakan lebih banyak kriteria-kriteria yang bisa diisi oleh *user*. Semakin banyak kriteria pencarian yang diisi oleh *user* maka hasilnya akan semakin akurat.

Contoh penggunaan aplikasi kriteria pencarian sebagai berikut : misalnya mencari jalur Trans Jogja yang berangkat dari Shelter Asal Bandara Adisucipto menuju ke Tempat Tujuan Malioboro dan Alamat Tujuan Jl. Malioboro ditunjukkan pada Gambar 12.



Gambar 12. Antarmuka Kriteria Pencarian

Hasil dari pencarian kriteria pencarian diatas dapat dilihat pada Gambar 13.



Gambar 13. Antarmuka Hasil Kriteria Pencarian

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Dari pembahasan bab-bab sebelumnya mengenai penerapan teknologi *Semantic Web* pada aplikasi pencarian jalur bis Trans Jogja maka dapat disimpulkan sebagai berikut :

1. Pengetahuan mengenai informasi pencarian jalur bis Trans Jogja dapat disimpan dalam model ontologi berbasis OWL dan dibagi ke dalam 3 *class* utama, yaitu *Class JalurTrans*, *Class Shelter*, dan *Class TempatTujuan* serta dilengkapi dengan pendefinisian properti-properti dari masing-masing *class*.
2. Aplikasi pencarian jalur bis Trans Jogja dapat dilakukan dengan dua jenis pilihan teknik pencarian yaitu kata kunci dan kriteria pencarian sehingga *user* dapat memilih teknik yang akan digunakan disesuaikan dengan pola pencarian yang diinginkan dan keakuratan hasil yang diharapkan.
3. Pencarian informasi jalur bis Trans Jogja dengan memanfaatkan ontologi jalur bis Trans Jogja sebagai basis pengetahuan mampu membantu *user* untuk menemukan jalur yang relevan.

### 5.2. Saran

Beberapa saran yang dapat diberikan sebagai hasil penelitian ini adalah :

1. Konsep-konsep yang ada pada ontologi pencarian jalur bis Trans Jogja ini dapat dikembangkan dan diperluas lagi agar dapat mencakup semua pengetahuan pada domain jalur bis Trans Jogja.
2. Perlu dikembangkan penelitian untuk menciptakan metode aplikasi anotasi untuk melakukan penambahan *instance* dan pemberian nilai-nilai properti pada ontologi pencarian jalur bis Trans Jogja sehingga memudahkan dalam memperbarui dan melengkapi basis pengetahuan tersebut.

## 6. DAFTAR PUSTAKA

- Berners-Lee, T., Hendler, dan J., Lasilla, O., 2001., *The Semantic Web, American Scientific*, <http://www.scian.com>
- Brickley, D., dan Guha, R., 2004, *RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation*, <http://www.w3.org/TR/rdf-schema>
- Candrasekaran, B. dan Josehson, J., 1999, *What are Ontologies, and Why Do We Need Them?*, *IEE Intelligent System*, Vol 14(1), hal 20-26.
- Davies, J., Studer, R., dan Warren, P., 2006, *Semantic Web Technologies : Trend and Research in Ontology-based System*, John Wiles and Sons, Ltd, Chichester.
- Geser, G., dan Steemsen, M., 2003, *Semantic Web Terms and Reading, List:A-X*, University of Glasgow, Scotland.
- Horidge, M., Knublauchz, H., Rector, A., dan Stevens, R., Wroe, C., 2004, *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition1*, <http://www.code.org/resource/tutorials/protegeowltutorial.pdf>
- Noy, F.N., dan McGuinness, D.L., 2001, *Ontology Development 101: A Guide to creating Your First Ontology*, <http://protege.stanford.edu/Publications/ontology-development/ontology101-noy-mcguinness.html>
- Passin, T.B., 2004, *Explore's Guide to the Semantic Web*, Manning Publications Co, Greenwich. <http://rpidshare.com/files/1633701/a4.Rar>
- Pramudiono, I., 2006, *Semantic Web Generasi Baru WWW*, <http://www.beritaiptek.com>
- Protege. <http://protege.stanford.edu/>, 2005