

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis**

Proses analisis sistem yang dilakukan dalam merancang dan membangun sistem pencarian rute angkot atau lyn di Kabupaten Gresik dilakukan dengan analisa terhadap proses yang terjadi. Dari analisis tersebut didapatkan belum adanya informasi mengenai pencarian rute angkot atau lyn hingga mencapai jalan yang dituju oleh penumpang. Selama ini informasi yang tersedia hanya berupa jenis angkot atau lyn dan rutenya.

Diperlukannya suatu sistem pencarian rute angkot atau lyn di Kabupaten Gresik yang dapat memudahkan penumpang dalam menyediakan informasi mengenai jenis angkot atau lyn Kabupaten Gresik beserta rute-rute untuk sampai pada jalan yang dituju oleh penumpang. Proses pencarian rute dilakukan dengan mencari rute dari angkot atau lyn di Kabupaten Gresik yang saling berpotongan hingga mencapai jalan yang dituju oleh penumpang. Perpotongan tersebut akan menghasilkan titik temu antar jenis angkot atau lyn yang menjangkaunya, sehingga menginformasikan tujuan penumpang melalui pilihan jenis angkot atau lyn dan perpotongan yang tersedia.

#### **3.2 Hasil Analisis**

Sebagai bagian dari persyaratan sistem dan kegiatan perancangan, terdapat beberapa komponen yang berkaitan, yaitu:

a. **Komponen rancang model ontologi**

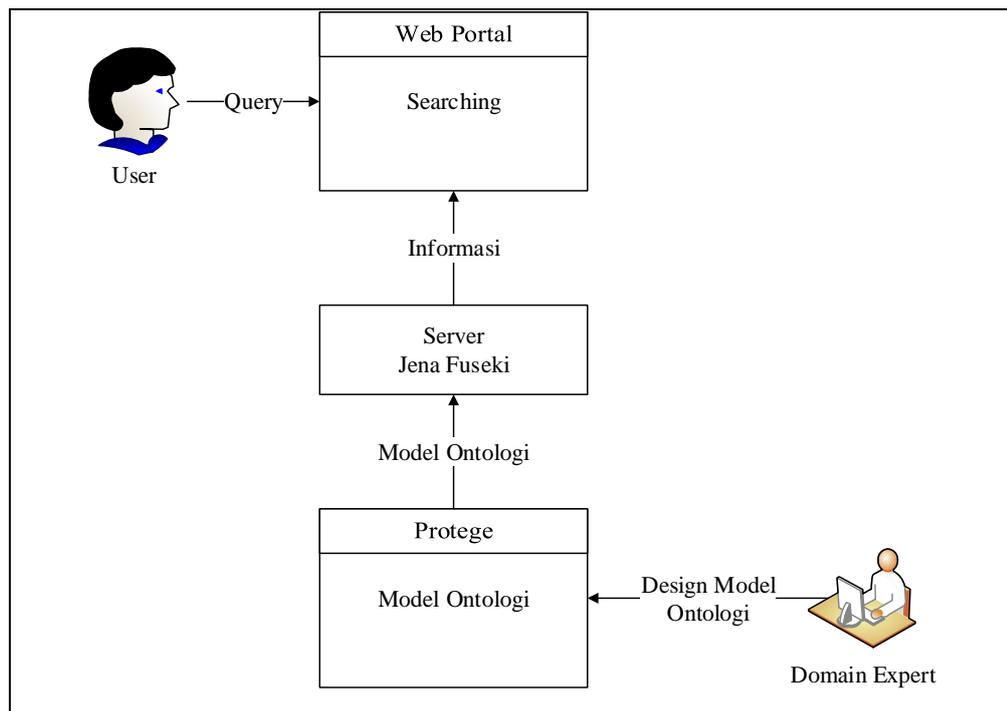
Komponen ini digunakan untuk merancang model ontologi berdasarkan *domain expert*. Dalam pemodelan ontologi memanfaatkan sebuah aplikasi *open-source* Protege. Hasil rancangan berupa model ontologi yang tersusun hirarki taksonomi (*classes - subclasses*). *Domain expert* membuat hirarki taksonomi, *property*, dan *instance* dengan menggunakan Protege dan model ontologi tersebut disimpan dalam *file* berekstensi OWL.

b. Komponen server

Komponen ini digunakan oleh *domain expert* untuk *upload* dan *parsing file* OWL pada server Jena Fuseki untuk kebutuhan *query*.

c. Komponen portal

Komponen ini dibuat dengan tujuan sebagai perantara untuk *user* sebagai wadah informasi yang diinginkan.



Gambar 3.1 Skema Keseluruhan Sistem

Pembangunan sistem dimulai dengan perancangan model ontologi dengan mengimplementasikan *class*, *properties*, dan *instance* melalui Protage oleh *domain expert*. Dilanjutkan dengan *upload* dan *parsing file* OWL pada server Jena Fuseki. Selanjutnya pembuatan portal web semantik sebagai perantara informasi kepada *user*. Portal tersebut memuat *query* yang akan dikirimkan pada server Jena Fuseki. Jena Fuseki memproses *query* yang dikirimkan dan mengembalikan informasi pada portal hasil *searching* yang dilakukan oleh *user*.

### 3.3 Representasi Model

#### 3.3.1 Perancangan Ontologi

Perancangan ontologi memberikan informasi mengenai tahapan-tahapan dalam pembangunan ontologi dan menjelaskan mengenai komponen apa saja yang dibutuhkan dalam penggambaran sebuah informasi.

Tahapan-tahapan dalam pembangunan ontologi adalah sebagai berikut:

- a. Penentuan domain. Domain yang melingkupi ontologi ini adalah Angkot atau lyn Kabupaten Gresik.
- b. Mendefinisikan *class* ontologi dan menyusun *class* tersebut dalam hirarki taksonomi (*subclass-superclass*) dengan menggunakan proses pengembangan top-down dimulai dengan mendefinisikan concept umum dalam domain dilanjutkan dengan concept yang lebih spesifik.
- c. Mendefinisikan *slot* atau *property*.
- d. Mendefinisikan *facets* pada *slot* atau *axiom* pada *properties*. *Properties* memiliki *domain* dan *range* yang spesifik. *Properties* menghubungkan individu pada *domain* dan individu pada *range*.
- e. Membuat *instance*.
- f. Mengisi nilai *slot* pada *instance*.

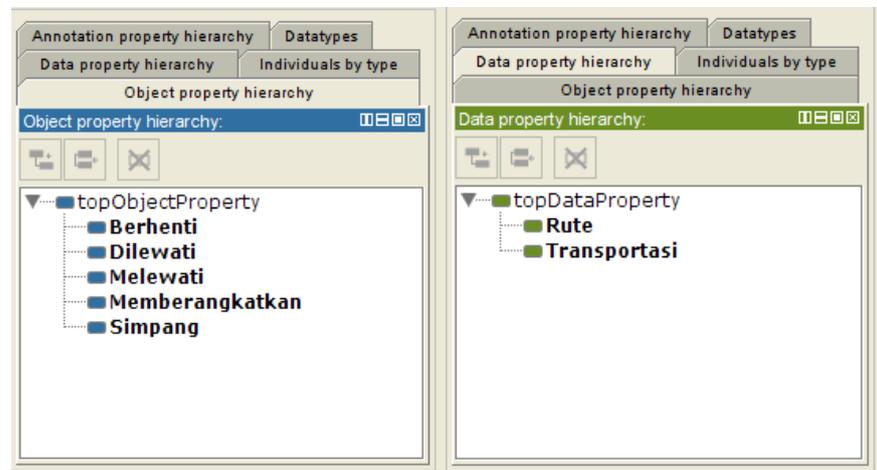
#### 3.3.2 Pembangunan dalam Protege

Perancangan yang telah dilakukan dapat dilanjutkan dengan membentuk model ontologi tersebut ke dalam sebuah *file* OWL dengan menggunakan sebuah aplikasi *open source platform* bernama Protege. Berikut *screenshot* aplikasi Protege dalam membentuk model ontologi yang akan dibangun:



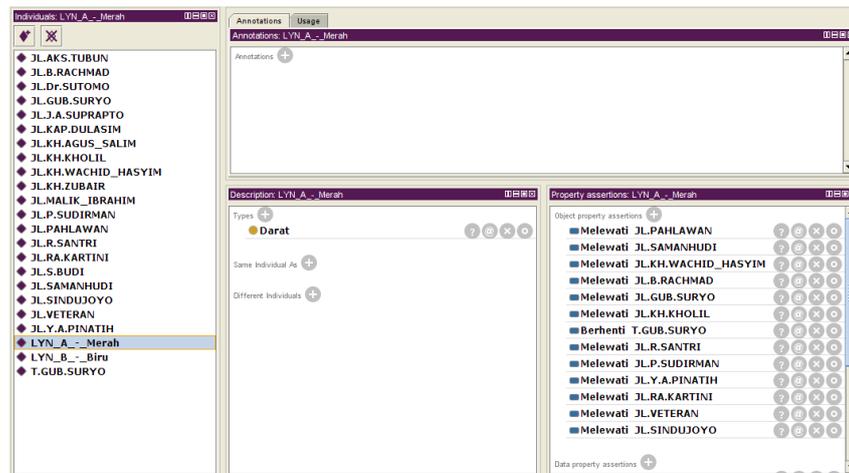
Gambar 3.2 *Class* pada Protege

Pembangunan ontologi untuk menyusun *class* dibentuk dalam hirarki taksonomi (*subclass-superclass*). Penyusunan *class* tersebut dilakukan pada kolom tab Class Hierarchy.



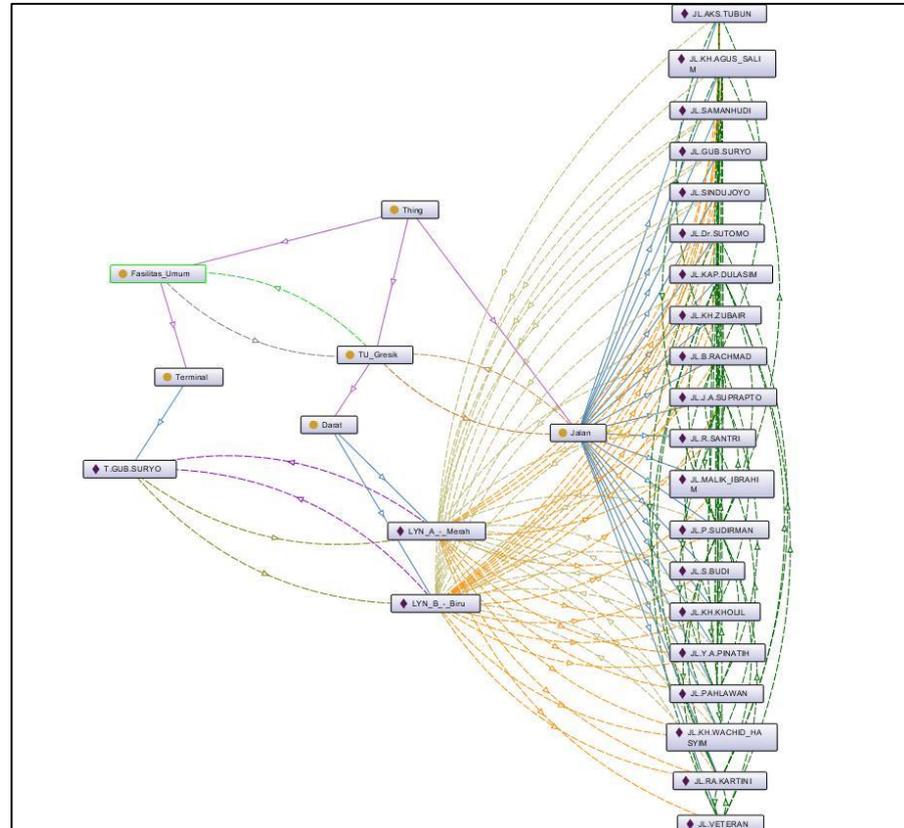
Gambar 3.3 *Object* dan *Data Property* pada Protege

Pembangunan ontologi untuk menyusun *facets* pada *slot* atau *axiom* dapat dilakukan pada kolom tab Object Property Hierarchy, sedangkan untuk menyusun *slot* atau *property* dapat dilakukan pada kolom tab Data Property Hierarchy.



Gambar 3.4 *Individuals* pada Protege

Pembangunan ontologi untuk menyusun *instance* dapat dilakukan pada kolom tab *Individuals*. Pengisian nilai dari *instance* tersebut dimasukkan pada kolom *Description* dan *Property Assertions*.



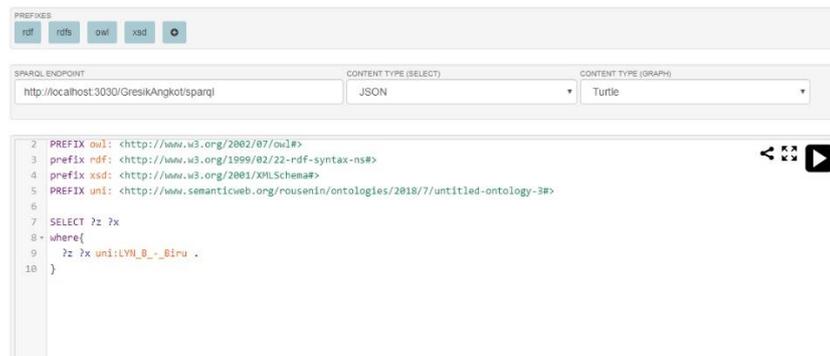
Gambar 3.5 Model Ontologi Angkot atau Lyn di Kabupaten Gresik

Hasil bentuk model perancangan ontologi dapat dilihat pada kolom Tab Ontograf dengan memasukkan *class* yang telah dibangun ke dalam *drawer*. *Class-class* tersebut akan menampilkan keterkaitan antar *class* lainnya dan dapat dijabarkan hingga muncul *instance* lainnya dari *class-class* yang saling berkaitan.

### 3.3.3 Query Jena Fuseki

Salah satu tahapan penting dalam pencarian rute yang akan dibangun adalah tahap *query* di server Jena Fuseki. Jena menyediakan perpustakaan Java yang luas untuk membantu *developer* mengembangkan kode yang menangani RDF, RDFS, RDFa, OWL dan SPARQL sesuai dengan rekomendasi dari W3C yang diterbitkan. Jena juga menyertakan *rule-based inference engine* untuk melakukan penalaran berdasarkan ontologi OWL.

Berikut contoh query dari proses pencarian:



```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX umi: <http://www.semanticweb.org/rousein/ontologies/2018/7/untitled-ontology-3#>

SELECT ?z ?x
WHERE {
  ?z ?x umi:LVM_0_-_Biru .
}

```

Gambar 3.6 Contoh Query Jena Fuseki

*Query* di atas adalah salah satu contoh *query* pencarian rute “angkot atau lyn B” berdasarkan subjek, predikat, dan objek yang diberikan. Berikut contoh hasil dari *query* di atas:

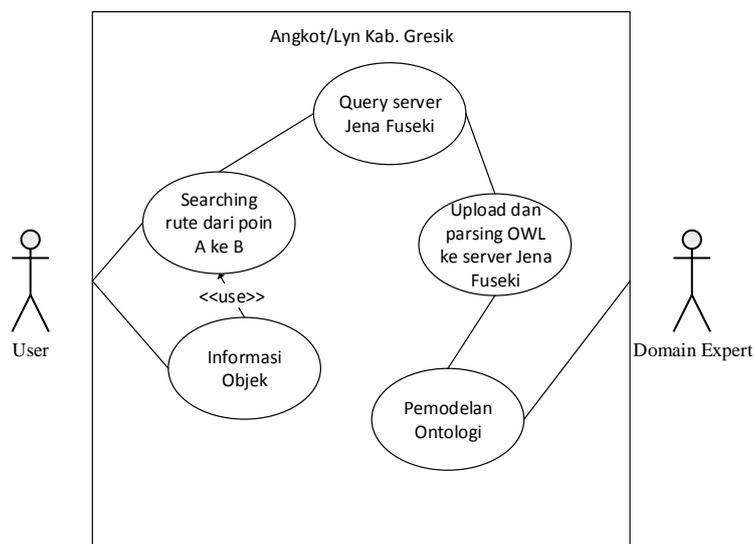
QUERY RESULTS	
<a href="#">Table</a> <a href="#">Raw Response</a>	
Showing 1 to 16 of 16 entries	
z	x
1	uni:JL.KAP.DULASIM uni Dilewati
2	uni:JL.KH.ZUBAIR uni Dilewati
3	uni:JL.JA.SUPRAPTO uni Dilewati
4	uni:JL.RA.KARTINI uni Dilewati
5	uni:JL.SAMANHUDI uni Dilewati
6	uni:JL.YA.PINATHI uni Dilewati
7	uni:JL.KH.KHOLIL uni Dilewati
8	uni:JL.KH.WACHID_HASYIM uni Dilewati
9	uni:JL.SINDUJOYO uni Dilewati
10	uni:JL.S.BUDI uni Dilewati
11	uni:JL.Dr.SUTOMO uni Dilewati
12	uni:JL.KH.AGUS_SALIM uni Dilewati
13	uni:JL.MALIK_IBRAHIM uni Dilewati
14	uni:T.GUB.SURYO uni Memerangkatkan
15	uni:JL.AKS.TUBUN uni Dilewati
16	uni:JL.GUB.SURYO uni Dilewati

Gambar 3.7 Hasil Query Jena Fuseki

Didapatkan hasil dari *query* pencarian rute “angkot atau lyn B” adalah data yang saling berkaitan dengan angkot atau lyn yang dicari dalam ontologi yang dimodelkan, sehingga ditampilkan jalan-jalan yang dilewati oleh angkot atau lyn tersebut serta terminal keberangkatannya sesuai dengan model ontologi yang telah dibangun.

### 3.4 Perancangan Sistem

#### 3.4.1 Use Case Diagram



Gambar 3.8 Use Case Diagram Angkot atau Lyn Kabupaten Gresik

Berikut adalah penjelasan *use case* secara naratif bagaimana sistem akan berjalan.

1. Pencarian Rute

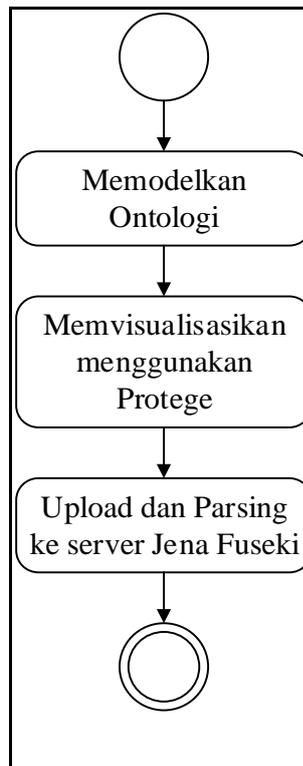
- a. Objective: Pencarian rute angkot atau lyn di Kabupaten Gresik.
- b. Actors: *User*
- c. Pre-condition: *User* menginginkan suatu informasi terhadap jenis angkot atau lyn ataupun jalan maupun rute yang dijangkau angkot atau lyn tersebut.
- d. Normal Flow:
  - 1) *User* membuka halaman web portal.
  - 2) *User* memulai pencarian.
  - 3) *User* mendapatkan informasi mengenai angkot atau lyn, jalan, atau rute yang dicari oleh *user*.
- e. Alternate Flow: *User* tidak mendapatkan informasi yang diinginkan atau informasi tersebut tidak terdapat dalam ontologi model yang sudah dibangun.
- f. Post Condition: *User* memperoleh hasil pencarian.

2. Pemodelan Ontologi

- a. Objective: Membangun desain ontologi dalam domain Angkot atau lyn Kabupaten Gresik ke dalam *class*, *properties* dan *instance* dengan menggunakan tool Protege.
- b. Actor: Domain expert
- c. Normal Flow:
  - 1) *Domain expert* melakukan pemodelan terhadap rancangan ontologi menggunakan Protage dan disimpan dalam bentuk file OWL.
  - 2) File OWL di-*upload* dan di-*parsing* ke server Jena Fuseki.

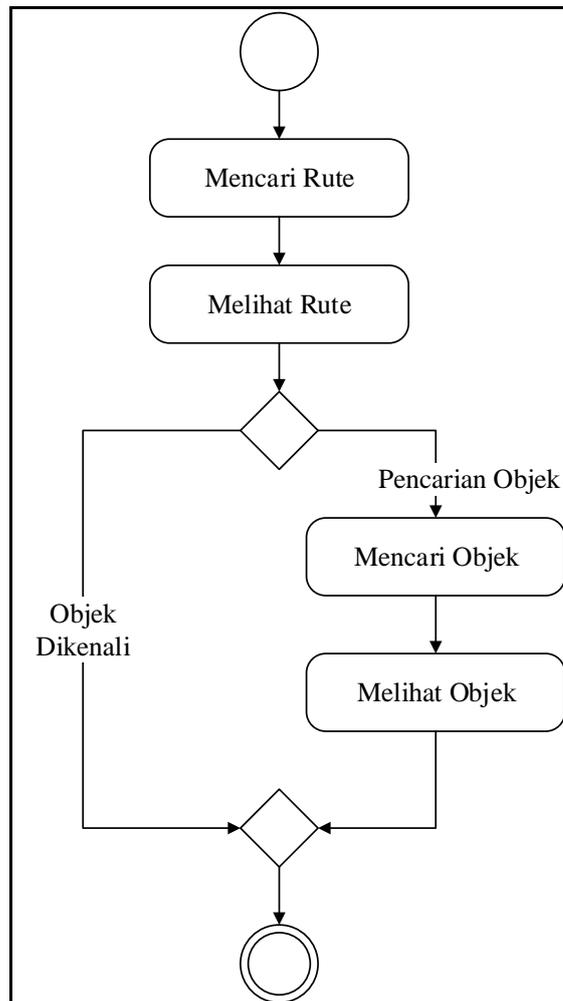
- d. Result: Model ontologi terintegrasi dengan server Jena Fuseki.

#### 3.4.1.1 Activity Diagram



Gambar 3.9 Activity Diagram *Domain Expert* Angkot atau Lyn Kabupaten Gresik

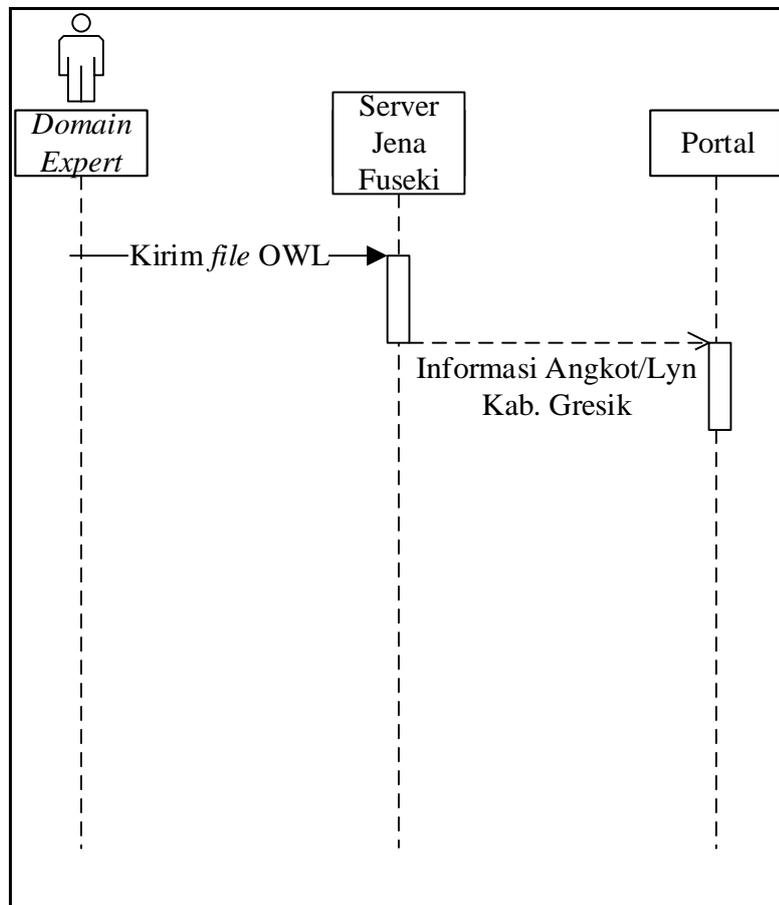
Activity Diagram pada gambar 3.10 menjelaskan awal proses dari seorang *domain expert* dengan memodelkan ontologi. Model ontologi tersebut dibentuk ke dalam sebuah *file* OWL melalui aplikasi *open source platform*, Protege dan kemudian di-*upload* dan *parsing* ke server Jena Fuseki.



Gambar 3.10 Activity Diagram *User Angkot atau Lyn Kabupaten Gresik*

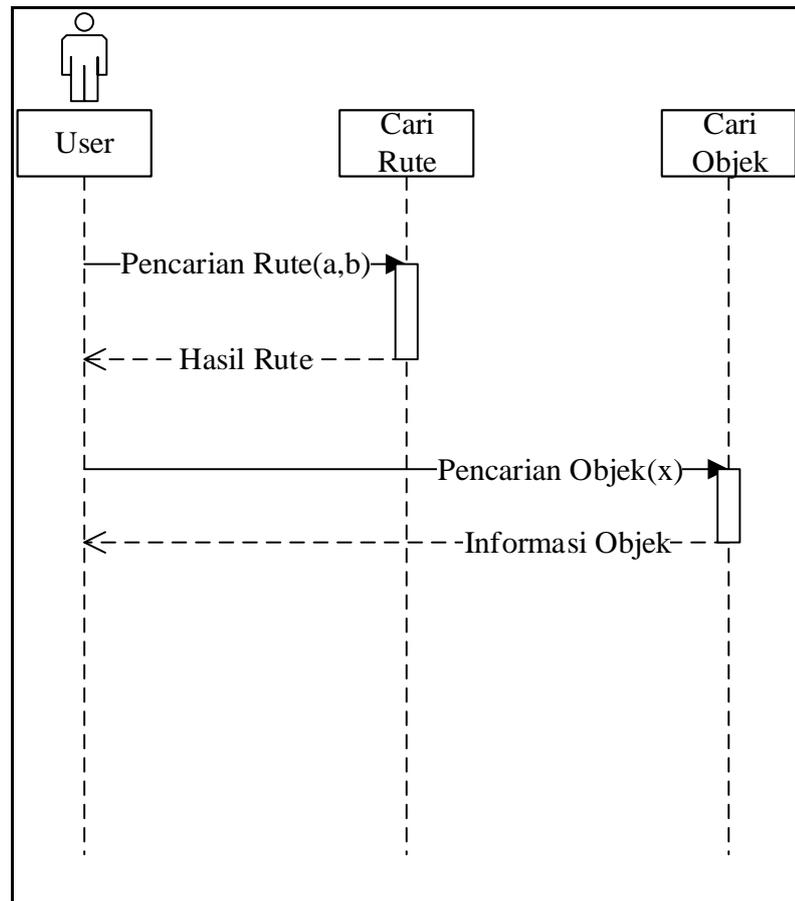
Activity Diagram pada gambar 3.11 menjelaskan awal proses dari *user* dengan melakukan pencarian rute dari poin jalan awal dan poin jalan tujuan penumpang. Rute yang dicari akan ditampilkan kepada *user* melalui objek-objek yang ada. *User* dapat menelusuri lebih lanjut informasi mengenai objek jika diperlukan.

### 3.4.1.2 Interaction Diagram



Gambar 3.11 Interaction Diagram *Domain Expert* Angkot atau Lyn Kabupaten Gresik

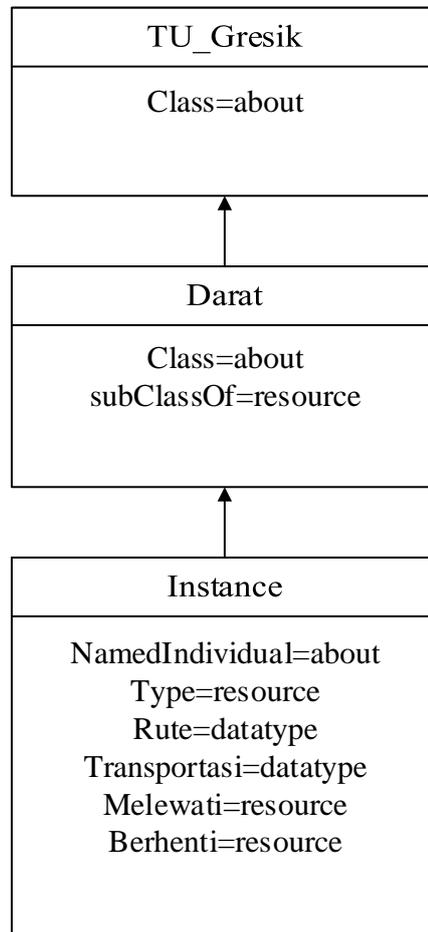
Interaksi Diagram pada Gambar 3.12 menjelaskan interaksi yang dilakukan oleh seorang *Domain Expert* dengan mengirimkan *file* OWL yang telah dibentuk pada Protege ke server Jena Fuseki.



Gambar 3.12 Interaction Diagram *User* Angkot atau Lyn Kabupaten Gresik

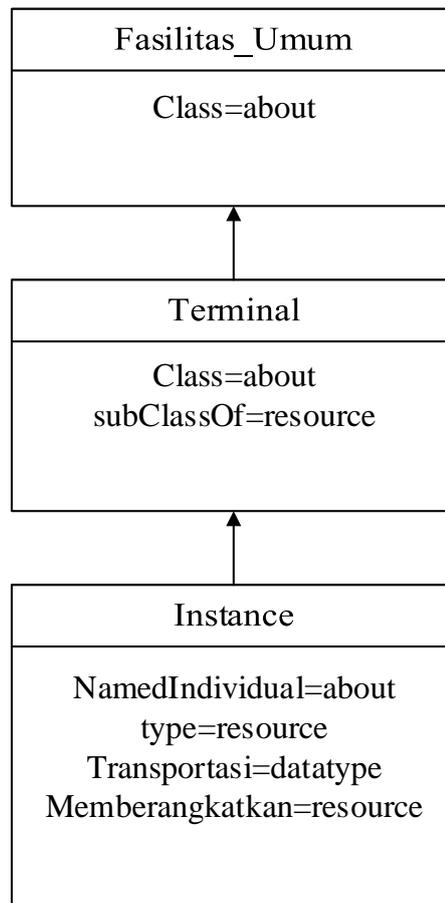
Interaksi Diagram pada Gambar 3.13 menjelaskan interaksi yang dilakukan oleh *user* dengan melakukan pencarian rute melalui poin jalan awal dan poin jalan tujuan penumpang. *User* mendapatkan kembalian informasi berupa angkot atau lyn yang dijangkau, rute, serta perpotongan dari poin jalan awal hingga poin jalan tujuan penumpang. *User* dapat melakukan pencarian objek dari objek-objek yang muncul dari informasi yang tertampil. Pencarian objek tersebut menghasilkan informasi mengenai objek tersebut dan dikembalikan kepada *user* kembali.

### 3.4.1.3 Class Diagram



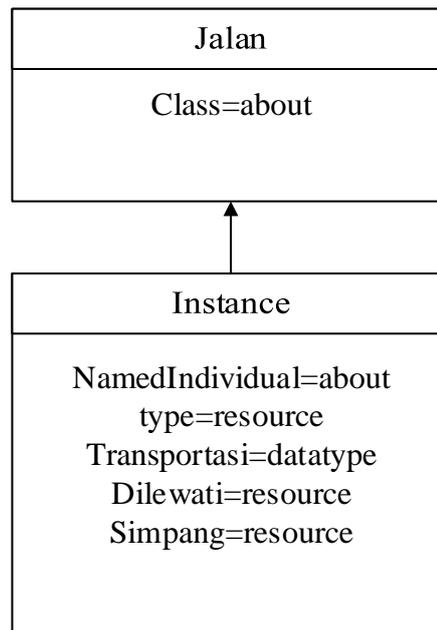
Gambar 3.13 *Class* TU\_Gresik

*Class* TU\_Gresik terbagi ke dalam sebuah subclass Darat untuk mewakili jenis kendaraan terhadap *instance*, yaitu kendaraan angkot atau lyn di Kabupaten Gresik.



Gambar 3.14 *Class* Fasilitas\_Umum

*Class* Fasilitas\_Umum terbagi ke dalam sebuah subclass Terminal untuk mewakili pemberhentian terhadap *instance*, yaitu terminal kendaraan angkot atau lyn di Kabupaten Gresik.

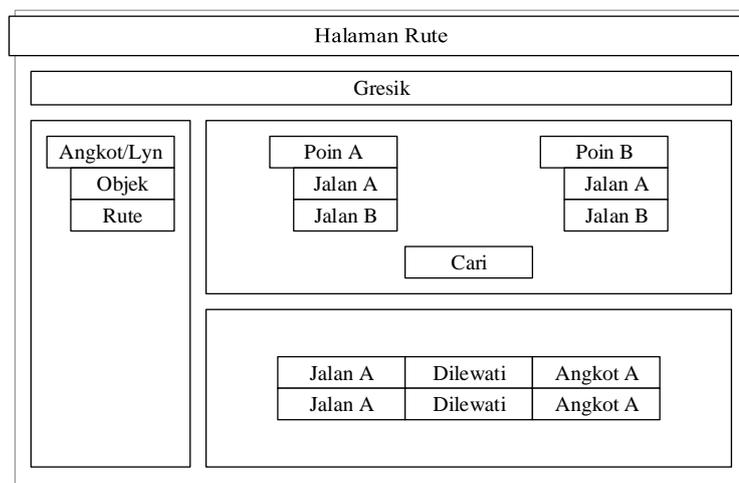


Gambar 3.15 Class Jalan

*Class Jalan* yang terdiri dari *instance* nama-nama jalan yang dijangkau oleh angkot atau lyn di Kabupaten Gresik.

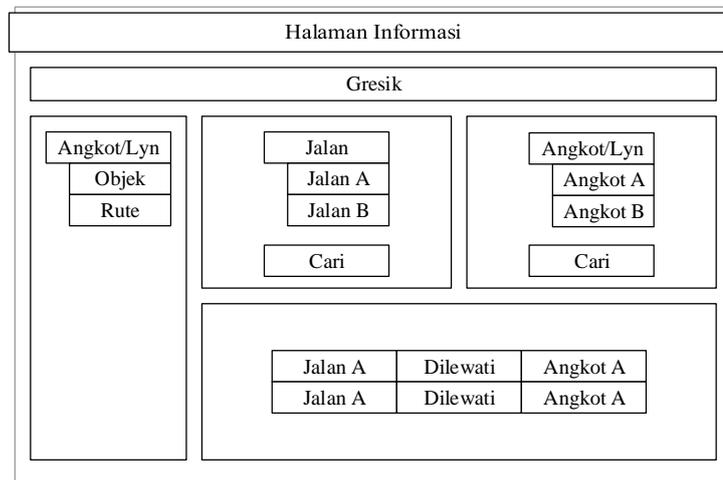
### 3.5 Perancangan Portal

Portal web adalah situs web yang dibangun dengan menyediakan kemampuan tertentu dalam menyajikan informasi yang diinginkan *user*. Berikut adalah rancangan portal yang akan dibangun:



Gambar 3.16 Rancangan Halaman Pencarian Rute

Pada halaman pencarian rute, *user* akan menemukan *list box* dalam bentuk dua kolom serta sebuah tombol cari untuk melakukan pencarian, dan sebuah kolom dibawahnya sebagai hasil pencarian yang dimaksud. Pada kolom pertama sebelah kiri terdapat *list box* sebagai pin awal dan kolom kedua sebelah kanan terdapat *list box* sebagai pin akhir. *List box* tersebut berisikan nama-nama jalan yang dijangkau oleh angkot atau lyn Kabupaten Gresik. Hasil pencarian akan ditampilkan pada kolom bagian bawah. Ketika *user* mengklik hasil objek akan diteruskan pada halaman informasi untuk mengetahui informasi objek lebih lanjut.



Gambar 3.17 Rancangan Halaman Objek

Pada halaman objek, *user* akan menemukan *list box* beserta tombol cari dalam bentuk dua kolom untuk melakukan pencarian, dan sebuah kolom dibawahnya sebagai hasil pencarian yang dimaksud. Pada kolom pertama sebelah kiri terdapat *list box* yang berisikan jenis-jenis angkot atau lyn yang tersedia di Kabupaten Gresik, sedangkan kolom kedua sebelah kanan terdapat *list box* yang berisikan nama-nama jalan yang dijangkau oleh angkot atau lyn Kabupaten Gresik. Pada bagian bawah terdapat sebuah kolom untuk menampilkan informasi mengenai objek yang dicari oleh *user*.

### 3.6 Skenario Pengujian

Pada penelitian ini, untuk mengukur kerja *query* ontologi yang dibangun dilakukan pengujian *precision*, dan *accuracy*.

**Tabel 3.1** Parameter perhitungan *precision* dan *accuracy*.

Keteranga	Relavan	Tidak Relavan
Terambil	True positif (tp)	False positif (fp)
Tidak terambil	False negatif (fn)	True negatif (tn)

Rumus perhitungan *precision*:

$$Precision = \frac{tp}{tp + fp} \times 100\%$$

Rumus perhitungan *accuracy*:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \times 100\%$$

Nilai *precision* dan *accuracy* dinyatakan dalam persen. Semakin tinggi kedua nilai tersebut menunjukkan semakin baiknya kinerja *query* pada model ontologi yang dibangun. Evaluasi yang akan dilakukan dalam penelitian ini adalah menghitung nilai dari *precision* dan *accuracy* berdasarkan *query* data ontologi yang berhasil dihasilkan. Penentuan nilai dari *precision* dan *accuracy* harus didapatkan jumlah data yang relevan terhadap suatu objek yang menghasilkan informasi yang dimaksud.

Menurut Rijsbergen (1979) relevansi merupakan sesuatu yang bersifat subjektif. Setiap orang mempunyai perbedaan dalam mengartikan suatu data yang relevan terhadap sebuah topik informasi. Sehingga dalam pelaksanaan pengujian sistem ini dibutuhkan seorang pakar yang dianggap mampu menilai apakah data yang dihasilkan dikatakan relevan atau tidak relevan. (Waluya, Onny Kartika. 2017)

### 3.7 Spesifikasi Pembangunan Sistem

Kebutuhan perangkat keras serta perangkat lunak dari sistem sebagai berikut:

**3.7.1 Kebutuhan Perangkat Keras**

- a. CPU Intel N4000 setara atau lebih tinggi.
- b. RAM 4Gb setara atau lebih.
- c. HDD 500Gb setara atau lebih.
- d. Mouse, keyboard, monitor standar.

**3.7.2 Kebutuhan Perangkat Lunak**

- a. OS Windows 7/8/8.1/10.
- b. Protege 4.3 build 304.
- c. Server Apache Jena Fuseki 3.9.0.
- d. XAMPP v3.1.0