

# IMPLEMENTASI *NATURAL LANGUAGE PROCESSING* (NLP) UNTUK APLIKASI PENCARIAN LOKASI

Irkham Huda<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, UGM, Indonesia

Email: <sup>1</sup>irkham@ugm.ac.id

## ABSTRAK

*Location search is one of the today's society needs, it is shown by numerous mapping service provider available. To find a location with reference to certain spatial relationships, users describe it in natural language. Therefore, in order to create a location search system that can understand natural language, the implementation of Natural Language Processing (NLP) is required. The research related to the implementation of NLP for the location search application is necessary, primarily due to the lack of study implementation which support Indonesian Language, existing research is still in English with limited coverage. This study developed NLP System for Location Search application known as NaLaMap. This study utilized location database from Open Street Map (OSM) and a web application client for case studies. In order to transform the input sentence into a spatial query, NLP system is built through five main stages, namely Tokenization, POS Tagging, Tagging NER, Normalization Entities, and Building Query. Then the successfully compiled query executed on the OSM-based location database to obtain a results that will be displayed through the map on the client application. The overall tryout of the system used 45 input sentences obtained from respondents, the result showed adequate value precision 0,97 and recall 0,91.*

**Keywords:** NLP, POS, NER, location search, Indonesian Language

## PENDAHULUAN

Pencarian lokasi menjadi salah satu kebutuhan masyarakat dewasa ini terbukti dengan banyaknya penyedia layanan pemetaan seperti *Google Map*, *Bing Map*, *Open Street Map*, dan lain-lain. Sistem Informasi Geografis (SIG) diperlukan untuk mengelola, menganalisis, dan menampilkan data-data yang berhubungan dengan posisi-posisi di permukaan bumi. Temu balik informasi dalam konteks Sistem Informasi Geografis (SIG) memerlukan penggunaan *Structural Query Language (SQL) Spatial* dan pemahaman dasar tentang struktur data pada basis data SIG tersebut (Huffman, 2013). *Query* spasial sering memerlukan penggunaan unit ukuran ketika melakukan pencarian fitur yang memiliki luas atau panjang tertentu atau beberapa fitur yang memiliki jarak tertentu antara satu dengan yang lain.

Ketika melakukan pencarian lokasi tertentu, kebanyakan pengguna menggunakan kata kunci umum, misalnya “Bandara Adisucipto” maka aplikasi dapat dengan mudah melakukan pencarian lokasi berdasarkan kata kunci tersebut dari basis data. Berbeda jika pengguna menggunakan suatu kalimat tertentu yang mengandung kata yang berhubungan dengan relasi spasial, seperti “Pompa bensin di Yogyakarta” dan “KFC dekat UGM” pada aplikasi

pencarian lokasi yang tidak berbasis *Natural Language Processing* (NLP), maka ada beberapa kemungkinan yaitu 1) aplikasi tidak memberikan keluaran apapun karena *keyword* tidak diketahui padahal data lokasi yang dimaksud mungkin tersedia dalam basis data. 2) aplikasi memberikan hasil tetapi tidak sesuai, misalnya untuk *keyword* “KFC di Yogyakarta” aplikasi memberikan hasil keluaran berupa lokasi kota Yogyakarta, padahal bukan itu yang dimaksud oleh pengguna. Karena itu penggunaan NLP sangat penting agar sistem memahami dengan baik maksud dari kalimat pencarian lokasi yang dimasukkan oleh pengguna.

NLP adalah disiplin ilmu komputer yang bertujuan untuk memahami konsep dan maksud dari bahasa manusia. Sementara manusia cukup mahir memahami sintaks linguistik dan tata bahasa serta hubungan spasial tersirat, komputer memiliki kesulitan besar pengolahan *query* bahasa alami (Allen, 1995). Contoh sistem ini dan penggunaan API adalah *Natural Language Toolkit* (NLTK) dan *CoreNLP* dari *Stanford University*.

*Open Street Map* (OSM) adalah layanan peta yang prinsip kerjanya mirip dengan Wikipedia tetapi khusus untuk informasi spasial. OSM ini disusun terutama berasal dari kontribusi pengguna dan beberapa data yang didonasikan perusahaan, sehingga OSM memiliki dan menyediakan data (Sistem Informasi Geografis) SIG yang ruang lingkungannya hampir seluruh dunia. Terdapat komunitas OSM yang cukup kuat bertugas melakukan verifikasi dan koreksi pada data OSM untuk memastikan keakuratannya. OSM mempunyai ukuran basis data cukup besar yaitu lebih dari 21 *gigabyte* data XML terkompresi dan seluruh data tersebut bebas digunakan oleh siapa saja di bawah lisensi *Open Database License* (ODbL) (Open Street Map, 2013).

Berdasarkan penelitian *Mobile Life Study* (Ries, 2012) hampir 19% pengguna perangkat *mobile* telah memanfaatkan layanan berbasis lokasi. Sekitar 26% dari pengguna layanan tersebut memanfaatkannya untuk menemukan lokasi objek tertentu seperti restoran dan tempat hiburan. Sayangnya, pemanfaatan *Natural Language Processing* (NLP) untuk aplikasi pencarian lokasi melalui berbagai perangkat baik *mobile* maupun web belum banyak diimplementasikan, salah satu aplikasi pencarian lokasi yang sudah memanfaatkan NLP adalah aplikasi *Google Map* dimana data spasial yang digunakan berasal dari Layanan *Google Map* sendiri. Selain itu terdapat beberapa penelitian terkait implementasi NLP untuk aplikasi pencarian lokasi seperti:

- 1) Pengembangan Sistem TEGUS untuk menggali data jalur geospasial dari deskripsi bahasa natural (Blaylock dkk, 2009)

- 2) Mengembangkan antarmuka *query* bahasa natural pada basis data spasial untuk menjawab *query* spasial yang terkait kriminalitas. Sistem ini mendukung sejumlah besar predikat spasial seperti “*within 3 miles/dalam 3 mil*” dan “*close to/dekat*” (Zhang dkk, 2009).
- 3) Mengembangkan sistem tanya jawab yang didesain untuk menjawab pertanyaan terkait posisi geografi yang cukup mendalam dan banyak diminta (Chen dkk, 2013)
- 4) Mengembangkan sebuah *framework* untuk aplikasi ArcGIS berbasis desktop yang menggunakan *tool Natural Language Processing* dan library *full-text indexing* untuk pencarian relasi topologi dasar dalam data OSM (Huffman, 2013).

Beberapa penelitian tersebut di atas mempunyai cakupan yang terbatas seperti hanya bisa menerima kalimat masukan dengan dua nama lokasi atau hanya satu predikat spasial saja dan bahasa yang digunakan adalah Bahasa Inggris. Maka, pengembangan penelitian terkait implementasi NLP untuk aplikasi pencarian lokasi masih dirasa perlu terutama karena belum adanya implementasi penelitian tersebut dalam Bahasa Indonesia.

## METODE PENELITIAN

### 1. Analisis Kebutuhan

Sistem yang dikembangkan pada penelitian ini adalah sistem berbasis NLP untuk aplikasi pencarian lokasi yang memanfaatkan OSM sebagai basis datanya. Aplikasi yang dibuat menggunakan *platform* web. Untuk selanjutnya sistem ini akan disebut sebagai NaLaMap (*Natural Language Processing Based Map*). NaLaMap membantu para pencari lokasi tempat tertentu yang ingin melakukan pencarian lokasi suatu tempat dengan mendeskripsikan kalimat pencarian dalam Bahasa Indonesia dengan batasan tertentu.

#### a. Kebutuhan fungsional

Berdasarkan tujuan pengembangannya NaLaMap, beberapa kebutuhan fungsional yang harus dipenuhi oleh NaLaMap adalah:

- 1) Sistem dapat mengakses data lokasi yang menggunakan basis data *Open Street Map*.  
Basis data *Open Street Map* yang dapat diperoleh secara bebas dari situs *openstreetmap.org*, sudah diproses khusus agar dapat terintegrasi dengan sistem.
- 2) Memberikan keluaran berupa penanda lokasi tertentu pada map sesuai dengan kalimat pencarian yang diajukan oleh pengguna dengan menggunakan Bahasa Indonesia.

Ini merupakan fungsi utama dari NaLaMap. Kalimat pencarian yang diajukan berupa kalimat baku dalam Bahasa Indonesia dengan cakupan:

- Pencarian lokasi suatu tempat dengan nama tertentu.
- Pencarian lokasi suatu tempat yang ada di daerah tertentu.
- Pencarian lokasi suatu tempat yang dekat dengan tempat lain.
- Pencarian lokasi suatu tempat yang dekat dengan tempat lain di daerah tertentu.
- Pencarian lokasi suatu tempat dengan maksimal jarak tertentu dari tempat lain.

#### b. Spesifikasi data

Berdasarkan rincian kebutuhan fungsional di atas, maka spesifikasi data yang digunakan dalam NaLaMap antara lain:

- Input.  
Data input berupa kalimat pencarian dalam Bahasa Indonesia, bisa dalam bentuk kalimat tanya (dengan kata tanya yang digunakan adalah “dimana”) atau kalimat perintah pencarian dengan mendeskripsikan lokasi tempat yang dicari *relative* terhadap tempat lain.
- Data lokasi.  
Data lokasi menggunakan basis data *Open Street Map* yang dapat diperoleh secara bebas dari situs *openstreetmap.org* sebagai sumber data spasial.
- Korpus *Part-of-Speech* Tag Bahasa Indonesia.  
Merupakan korpus yang berisi dokumen teks kalimat dalam bahasa Indonesia yang telah dianotasi nilai *part-of-speech* tag secara manual oleh manusia diperoleh dari [bahasa.cs.ui.ac.id/postag/corpus](http://bahasa.cs.ui.ac.id/postag/corpus) (Dinakaramani dkk, 2014) sebagai data referensi untuk library *POS Tagger Stanford CoreNLP*.
- Data artikel Wikipedia versi Bahasa Indonesia.  
Data artikel ini diperoleh dari DBPedia untuk membuat data referensi pada *library NER Tag Stanford CoreNLP* seperti yang dikerjakan oleh Luthfi, dkk (2014).

## 2. Rancangan Pemrosesan Data

Untuk mengembangkan NaLaMap ini diperlukan beberapa data seperti yang dijelaskan pada Subbab 2.1.2. Data-data tersebut memerlukan pemrosesan, terutama untuk menyesuaikan format agar dapat diintegrasikan ke dalam sistem. Terdapat dua

pemrosesan data yang diperlukan, yaitu pemrosesan Data Spasial OSM dan pemrosesan Data Referensi untuk NLP.

a. Data Spasial OSM

NaLaMap menggunakan basis data Postgree SQL maka diperlukan proses *import* menggunakan perangkat lunak OSMtoPGSQL untuk memasukkan basis data berformat OSM. PBF tersebut menjadi Postgree SQL beserta beberapa proses penyesuaian.

b. Data Referensi NLP

Sistem NLP yang digunakan pada sistem NaLaMap menggunakan basis *tool Stanford Core NLP*. Terdapat dua *tool Core NLP* yang digunakan dalam sistem NaLaMap, yaitu *Part of Speech (POS) Tagger* dan *Named Entity Recognition (NER) Tagger*. Berdasarkan dokumentasi Core NLP (2004), agar masing-masing *tool* dapat mendukung proses identifikasi dalam bahasa selain Bahasa Inggris, diperlukan data referensi dengan format khusus yang dikonversi dari data POS maupun data NER yang berisi daftar kata-kata dalam Bahasa Indonesia beserta *POS Tag* atau *NER Tag*-nya masing-masing.

3. Rancangan Sistem

a. Arsitektur Sistem

Sistem NaLaMap direncanakan menggunakan arsitektur *client/server* terdiri dari bagian *client* yang berupa aplikasi web dan di bagian server berupa Server NLP dan Server Pusat Data Lokasi. Rancangan arsitektur secara umum dapat dilihat pada Gambar 1. Bagian antarmuka *client* akan menerima masukan kalimat dari pengguna, Server NLP akan mengubah *string* pencarian dari Bahasa natural (Bahasa Indonesia) menjadi suatu *query* yang dapat dipahami/digunakan untuk melakukan *query* pada Pusat Data Lokasi. Pada Server NLP juga dilakukan pemeriksaan apabila masukan kalimat pencarian dari pengguna menggunakan *grammer* yang tidak sesuai maka sistem akan menampilkan pesan kesalahan.

b. Pusat Data Lokasi

Pusat Data Lokasi merupakan server yang berisi basis data lokasi yang diperoleh dari *Open Street Map (OSM)*. Pusat Data Lokasi dibangun dengan menggunakan Postgree SQL sebagai DBMS (*Database Management System*), yang dilengkapi ekstensi PostGIS sehingga dapat mengelola dan memproses data-data spasial. Akan terdapat tiga Tabel utama dari hasil *import* yang digunakan oleh Sistem NaLaMap

yaitu Tabel *planet\_osm\_point* untuk menyimpan data spasial bertipe titik, Tabel *planet\_osm\_line* untuk menyimpan data spasial bertipe garis, dan Tabel *planet\_osm\_polygon* untuk menyimpan data spasial bertipe poligon.

c. Server NLP

Di dalam Server NLP terdapat layanan untuk melakukan pemrosesan *string* masukan berupa kalimat pencarian dalam bahasa natural menjadi formulasi *query* untuk memperoleh data lokasi yang diinginkan dari Pusat Data Lokasi. Layanan pada Server NLP ini dikembangkan dengan bahasa pemrograman Java. Alur pemrosesan dari *string* masukan menjadi formulasi *query* dapat dilihat pada Gambar 1. Terdapat 5 proses pada rancangan alur pemrosesan *query* dari *string* masukan bahasa natural, yaitu Tokenisasi, *POS Tagging*, *NER Tagging*, Normalisasi Entitas, dan Penyusunan *Query*.

1) Tokenisasi

Bagian Tokenisasi ini akan menghapus beberapa tanda baca dan kata yang kurang berpengaruh signifikan dalam proses pembuatan formulasi *query*. Beberapa kata yang dihapus antara lain kata tanya, kata perintah, kata bermakna sama dengan lokasi, dan kata paling yang akan digantikan imbuhan ter-. Selanjutnya dilakukan proses identifikasi individu kata-kata dari serangkaian karakter atau proses tokenisasi. Kemudian masing-masing kata yang teridentifikasi akan disimpan untuk kemudian untuk dilakukan proses lebih lanjut.

Contoh: Dimana KFC di Yogyakarta ? → KFC | di | Yogyakarta

2) POS Tagging

Bagian POS Tagging mempunyai fungsi utama yaitu identifikasi *Part of Speech* (POS) untuk setiap kata dalam *string* pencarian. Proses ini dilakukan dengan menggunakan *library Stanford University Part of Speech tagger* (Toutanova dkk, 2003). Sedangkan, data referensi yang digunakan adalah Korpus *Part-of-Speech Tag Bahasa Indonesia* (Dinakaramani dkk, 2014). *POS tagger* akan mengidentifikasi kata benda, kata kerja, kata sifat, kata keterangan, kata ganti, preposisi/kata depan, dan *part of speech* dari *string* masukan. Keberadaan kata depan/preposisi akan sangat menentukan formula *query* yang dihasilkan.

Contoh: KFC | di | Yogyakarta → KFC [NNP] | di [IN] | Yogyakarta [NNP]

3) NER Tagging

Bagian NER Tagging ini untuk mengidentifikasi entitas lokasi dan organisasi yang ada dalam *string* pencarian. Proses ini menggunakan *library Stanford University Named Entity Recognizer*. Data referensi yang digunakan berasal dari sejumlah artikel Wikipedia versi Bahasa Indonesia yang telah dianotasi (Luthfi, 2014). NER dapat mengidentifikasi suatu kata benda termasuk personal, lokasi, atau organisasi. Karena sistem yang dibangun untuk pencarian lokasi, maka jika yang teridentifikasi adalah nama personal maka akan diabaikan sehingga hanya nama lokasi dan organisasi yang diidentifikasi dan digunakan untuk pembentukan *query*.

Contoh: KFC [NNP] | di [IN] | Yogyakarta [NNP] → KFC [NNP/Organisasi] | di [IN/O] | Yogyakarta [NNP/Lokasi]

#### 4) Normalisasi Entitas

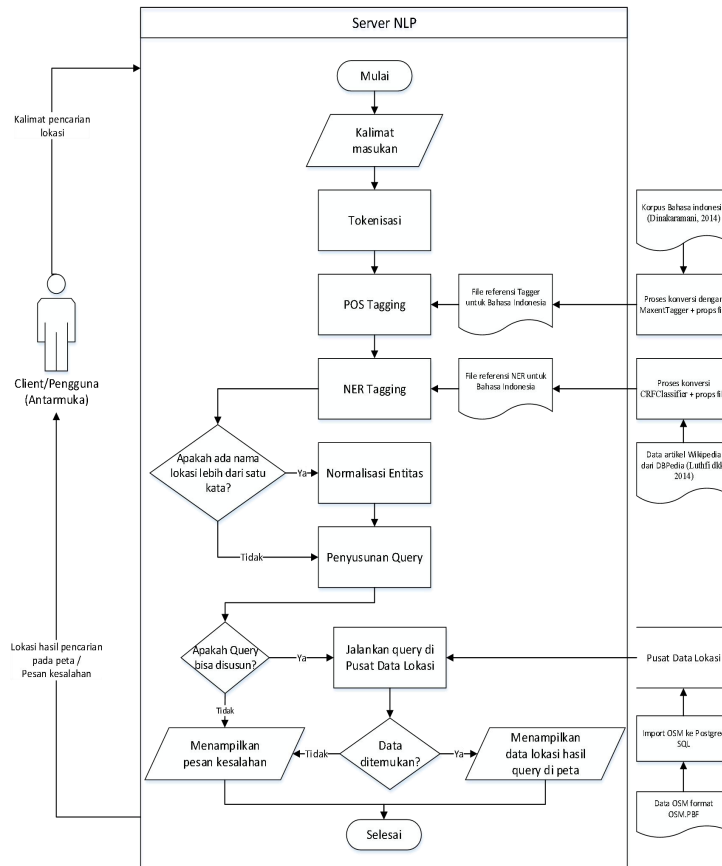
Bagian ini adalah proses tambahan dari NER Tag karena salah satu kelemahannya adalah jika ditemukan nama lokasi yang terdiri dari lebih dari satu kata NER Tag tidak dapat melakukan kombinasi. Maka, modul ini digunakan untuk melakukan penggabungan tag jika ditemukan nama lokasi atau organisasi yang terdiri dari lebih dari satu kata.

Contoh: Jawa [NNP/Lokasi] |Barat [NNP/Lokasi] → Jawa Barat [NNP/Lokasi]

#### 5) Penyusunan *Query*

Pada bagian ini dilakukan penyusunan *query* berdasarkan hasil pemrosesan pada Tokenisasi, POS Tagging, NER Tagging, dan Normalisasi. Untuk mempermudah dan menghindari kesalahan dalam penyusunan *query*, maka selain identifikasi *POS Tag* dan *NER Tag* dari kalimat yang akan disimpan dalam *array* juga diperlukan identifikasi frasa target dan frasa referensi (Zhang dkk,2009). Frasa target adalah suatu frasa yang menunjukkan lokasi yang pengguna ingin temukan, sedangkan frasa referensi yaitu frasa yang menunjukkan suatu lokasi yang digunakan sebagai petunjuk untuk menemukan frasa target. Dalam frasa referensi juga terdapat predikat spasial untuk menunjukkan hubungan spasial antara objek target dan objek referensi. Proses identifikasi frasa target dan frasa referensi ini menggunakan *array* kata, *array* POS Tag, dan *array* NER Tag yang diperoleh dari hasil proses-proses sebelumnya. Tahap selanjutnya, akan dilakukan penyusunan *query* berdasarkan

pola kalimat sesuai dengan frasa target, frasa referensi, dan predikat spasial yang ada pada frasa referensi.



Gambar 1. Rancangan Arsitektur dan Kerangka Kerja Sistem NaLaMap

d. Gramatika Kalimat

Kemungkinan pembentukan kalimat masukan oleh pengguna bisa tidak terbatas, sehingga perlu ada batasan pola gramatika kalimat yang akan dapat dikenali atau dapat diproses oleh Server NLP. Berikut akan dijelaskan daftar pola gramatika kalimat terkait pencarian lokasi yang nantinya dapat diproses oleh sistem NaLaMap. Kemudian pola kalimat secara keseluruhan dibagi ke dalam 3 tipe pola berdasarkan jumlah nama lokasi yang disebutkan dalam kalimat masukan dapat dilihat pada Tabel 1, 2, dan 3. Pola yang dibuat dengan mengabaikan kata sebelum frasa target karena dalam proses akan dihilangkan.

Tabel 1. Pola Kalimat dengan Satu Nama Lokasi

<b>Pola umum</b>	<kalimat> : (a) <lokasi/target> (b) <lokasi/target> ter-<kata sifat> (c) <lokasi/target> paling <kata sifat>
<b>Aturan</b>	(a) S → NNP



<i>Contoh</i>	(b) S → NNP JJ (c) S → NNP RB JJ (a) Yogyakarta (b) KFC terdekat (c) KFC paling dekat
---------------	---

**Tabel 2. Pola Kalimat dengan Dua Nama Lokasi**

<i>Pola umum</i>	<kalimat> : (a) <lokasi/target> <kata sifat> (dari) (regional) <lokasi/referensi> (b) <lokasi/target> <kata depan> (regional) <lokasi/referensi> (c) <lokasi/target> <kata sifat> <kata depan> (regional) <lokasi/referensi> (d) <lokasi/target> <numerik> <satuan> <kata depan> (regional) <lokasi/referensi>
<i>Aturan</i>	(a) S → NNP JJ NNP (b) S → NNP IN NNP (c) S → NNP JJ IN NNP (d) S → NNP CD NND IN NNP
<i>Contoh</i>	(a) KFC dekat UGM (b) KFC di Yogyakarta (c) KFC terdekat dari UGM (d) KFC 5 kilometer dari UGM

**Tabel 3. Pola Kalimat dengan Tiga Nama Lokasi**

<i>Pola umum</i>	<kalimat> : (a) <lokasi/target> [<kata sifat> (dari) <lokasi/referensi>] [<kata depan> <lokasi/referensi>] (b) <lokasi/target> [<numerik> <satuan> (dari) <lokasi/referensi>] [<kata depan> <lokasi/referensi>] (c) <lokasi/target> [<numerik> <satuan> (dari) <lokasi/referensi>] [<kata sifat> <lokasi/referensi>] (d) <lokasi/target> [<kata depan> <lokasi/referensi>] [<kata depan> <lokasi/referensi>] (e) <lokasi/target> [<kata sifat> <lokasi/referensi>] [<kata sifat> <lokasi/referensi>] (f) <lokasi/target> [<numerik> <satuan> (dari) <lokasi/referensi>] [<numerik> <satuan> (dari) <lokasi/referensi>] (g) <lokasi/target> <kata sifat> [<lokasi/referensi>] <kata sambung> [<lokasi/referensi>] (h) <lokasi/target> <numerik> <satuan> (dari) [<lokasi/referensi>] <kata sambung> [<lokasi/referensi>]
<i>Aturan</i>	(a) S → NNP JJ NNP IN NNP S → NNP IN NNP JJ NNP (b) S → NNP CD NND IN NNP IN NNP S → NNP IN NNP CD NND IN NNP (c) S → NNP CD NND IN NNP JJ NNP S → NNP JJ NNP CD NND NNP (d) S → NNP IN NNP IN NNP (e) S → NNP JJ NNP JJ NNP (f) S → NNP CD NND IN NNP CD NND IN NNP (g) S → NNP JJ NNP CC NNP (h) S → NNP CD NND IN NNP CC NNP
<i>Contoh</i>	(a) KFC dekat UGM di Yogyakarta KFC di Yogyakarta dekat UGM (b) KFC 5 kilometer dari UGM di Yogyakarta KFC di Yogyakarta 5 kilometer dari UGM (c) Hotel 2 kilometer dari UGM dekat UNY Hotel dekat UNY 2 kilometer dari UGM (d) Hotel di Jalan Kaliurang di Sleman

	(e) Hotel dekat UGM dekat UNY (f) Hotel 1 kilometer dari UGM 2 kilometer dari UNY (g) Hotel dekat UGM dan UNY (h) Hotel 2 kilometer dari UGM dan UNY
--	---

#### 4. Implementasi

Implementasi Sistem NaLaMap akan terdiri dari beberapa tahap sebagai berikut:

- 1) Mengimport basis data OSM ke Postgree SQL untuk bagian Pusat Data Lokasi ditambahkan optimasi *full text search* untuk mempercepat pemrosesan *query*.
- 2) Pemrosesan data referensi untuk POS Tagger dan NER Tagger.
- 3) Implementasi Server NLP dengan menggunakan Bahasa Pemrograman Java untuk mengubah *string* pencarian lokasi menjadi *query*.
- 4) Implementasi template *query* sesuai pola kalimat yang sudah ditetapkan.
- 5) Implementasi model JSON untuk mengirimkan hasil *query* yang diperoleh dari Pusat Data Lokasi ke anta muka aplikasi web.
- 6) Implementasi antar muka aplikasi client dikembangkan berbasis web menggunakan HTML dan Javascript dengan bagian tampilan utama berupa textbox untuk masukan kalimat pencarian lokasi dan peta untuk menampilkan lokasi hasil pencarian.

#### 5. Pengujian

Pengujian dilakukan untuk mengetahui realibilitas sistem dan untuk memastikan sistem bekerja sesuai dengan rancangan. Pengujian yang akan dilakukan pada sistem NaLaMap ini melalui tiga tahap pengujian utama dengan model strategi integrasi *bottom-up* (Pressman, 2009).

##### 1. Pengujian bagian komponen sistem (*unit testing*).

Pengujian tahap pertama yang dilakukan adalah *unit testing*, merupakan pengujian pada bagian komponen-komponen sistem. Dalam sistem NaLaMap delapan komponen utama yang perlu dilakukan proses *unit testing*, yaitu Tokenisasi, POS Tagging, NER Tagging, Normalisasi Entitas, dan Penyusunan *Query*. Prosedur pengujian untuk setiap komponen dilakukan dengan memberikan berbagai kemungkinan masukan yang sesuai untuk setiap bagian komponen.

##### 2. Pengujian integrasi sistem (*integration testing*).

Pengujian tahap selanjutnya adalah *integration testing*, merupakan pengujian kelompok komponen yang terintegrasi untuk membentuk sub-sistem maupun sistem. Dalam sistem NaLaMap subsistem yang di uji coba adalah Sistem Server NLP.

Subsistem akan dilakukan prosedur pengujian dengan beberapa kemungkinan masukan untuk menguji integrasi sistem.

3. Pengujian sistem dengan berbagai masukan pengguna (*system testing*) .

Pengujian tahap terakhir adalah *system testing*, merupakan pengujian untuk memperoleh prosentase keberhasilan sistem dalam memberikan hasil dari berbagai masukan kalimat pencarian lokasi dari pengguna. Prosedur yang dilakukan pada tahap ini 4 dan 5 kalimat masukan yang sesuai kriteria, diperoleh dari 12 orang responden yang dibagi kedalam 3 tipe pola kalimat sesuai dengan rancangan, kemudian menggunakan berbagai kalimat pencarian lokasi tersebut pada sistem NaLaMap, dan kemudian mencatat bagaimana hasil keluarannya. Untuk mengukur kualitas hasil keluaran sistem, digunakan pengukuran *precision*, *recall*, dan *willingness* (Mollevik, 2013) dengan Rumusan 1, 2,dan 3.

$$willingness = \frac{\#correct + \#incorrect}{\#total} \tag{1}$$

$$precision = \frac{\#correct}{\#correct + \#incorrect} \tag{2}$$

$$recall = \frac{\#correct}{\#total} = \frac{\#correct}{\#correct + \#incorrect} * \frac{\#correct}{\#correct + \#incorrect} = willingness * precision \tag{3}$$

## HASIL DAN PEMBAHASAN

Berdasarkan hasil pengujian dengan menggunakan 4 dan 5 kalimat masukan dari responden dan melalui tiga tahapan pengujian diperoleh hasil seperti yang terlihat pada Tabel 4.

Tabel 4. Hasil Pengujian Sistem

<i>Tipe pengujian</i>	<i>Precision</i>	<i>Recall</i>	<i>Willingness</i>
<i>Pengujian Komponen Sistem</i>			
Modul Tokenisasi	1,0	1,0	1,0
Modul POS Tagger	0,76	0,76	1,0
Modul NER Tagger	0,62	0,62	1,0
Modul Nomalisasi Entitas	1,0	1,0	1,0
Modul Penyusunan <i>Query</i>	0,97	0,93	0,95
<i>Pengujian Integrasi Sistem</i>			
Sistem NLP	0,95	0,91	0,95
<i>Pengujian Kemampuan Sistem</i>			
NaLaMap	0,97	0,91	0,93

Berdasarkan hasil analisis kesalahan dan kegagalan hasil pencarian lokasi pada proses uji coba sistem disebabkan beberapa hal sebagai berikut:

1) Kalimat masukan tidak sesuai dengan batasan masalah.

Jika kalimat masukan tidak sesuai dengan batasan masalah maka Sistem NaLaMap akan gagal memprosesnya atau jika berhasil hasil pencariannya akan salah. Hal ini merupakan ketertasan dari sistem yang dibuat hanya mendukung pemrosesan kalimat masukan sesuai dengan batasan masalah.

2) Kesalahan Identifikasi POS Tag dan NER Tag.

Kesalahan ini terjadi disebabkan karena modul POS Tagger atau *NER Tagger* salah dalam melakukan identifikasi kelas kata dari kalimat masukan. Untuk kata terkait nama lokasi ketika terjadi kesalahan pada bagian POS Tag misalnya teridentifikasi selain NN/NNP, sedangkan pada *NER Tag* kata tersebut telah teridentifikasi sebagai entitas *Place/Organisation* maka akan tetap dianggap sebagai nama lokasi ataupun terjadi kasus sebaliknya, maka kesalahan tidak akan terjadi. Kecuali kalau pada kedua proses baik *POS Tag* maupun *NER Tag* untuk nama lokasi tersebut tidak terjadi kesalahan maka akan tetap. Beberapa kesalahan identifikasi ini menyebabkan Sistem NaLaMap memunculkan pesan kesalahan karena pola kalimat menjadi tidak sesuai.

3) Kesalahan Penyusunan *Query*.

Kesalahan dalam penyusunan *query* terjadi ketika modul POS Tagger atau NER Tagger melakukan kesalahan identifikasi kelas kata dari kalimat masukan. Proses penyusunan *query* sangat tergantung dari *array* POS dan *array* NER yang diperoleh dari proses sebelumnya. Jadi ketika terjadi kesalahan pada waktu identifikasi POS tag dan NER Tag maka *query* yang disusunpun bisa mengalami kesalahan.

## KESIMPULAN

Dari hasil implementasi dan pengujian Sistem NaLaMap ini dapat diambil kesimpulan sebagai berikut:

- 1) Telah berhasil dibuat sistem untuk mentransformasikan kalimat pencarian lokasi dalam Bahasa Indonesia menjadi *query* spasial Postgrree SQL dengan implementasi NLP, dari hasil pengujian pada bagian Server NLP yang merupakan Pengujian Integrasi Sistem menggunakan 45 kalimat masukan diperoleh nilai precision 0,95 dan recall 0,91. Kesalahan disebabkan karena kalimat masukan dari pengguna tidak sesuai dengan batasan sistem dan kesalahan identifikasi kelas kata pada modul *POS Tagger* atau modul *NER Tagger*.
- 2) Dari hasil pengujian Sistem NaLaMap yang merupakan Pengujian Kemampuan Sistem secara keseluruhan menggunakan 45 kalimat masukan diperoleh nilai precision 0,97 dan recall 0,91. Kesalahan atau kegagalan disebabkan karena kesalahan identifikasi kelas kata pada modul *POS*

*Tagger* atau modul *NER Tagger* sehingga menyebabkan *query* gagal disusun atau *query* yang disusun salah sehingga tidak memberikan hasil yang sesuai ketika dijalankan pada basis data lokasi.

- 3) Aplikasi pencarian lokasi berbasis NLP (NaLaMap) yang telah dibuat dapat menerima masukan dengan pola kalimat paling banyak terdapat tiga nama lokasi atau terdapat dua predikat spasial.

## DAFTAR PUSTAKA

Allen, J. (1995). *Natural Language Understanding 2<sup>nd</sup> Edition*. Benjamin-Cummings Publishing Company.

Blaylock, N., Swain, B., & Allen, J. (2009). *Mining Geospatial Path Data from Natural Language Descriptions*. Institute for Human and Machine Cognition (IHMC). Florida, USA.

Chen, W., Fosler, E., Xiao, N., Raje, S., Ramnath, R., & Sui, D. (2013). *A Synergistic Framework for Geographic Question Answering*. Dept. of Computer Science and Engineering, Dept. of Geography, The Ohio State University.

Dinakaramani, A., Rashel, F., Luthfi, A., & Manurung, R. (2014). *Designing an Indonesian Part of speech Tagset and Manually Tagged Indonesian Corpus*. International Conference on Asian Language Processing (IALP 2014), Kuching.

Finkel, J., Grenager, T., dan Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.

Huffman, G. (2013). *Implementing a Natural Language Processing Framework to Perform Spatial Searches of Open Street Map Features in ArcGIS*. MGIS Capstone Project The Pennsylvania State University.

Location-Based Services: The #1 Most-Wanted Mobile Feature. Here's Why (Research). (2013). <http://therealtime.com/2012/04/27/location-based-services-the-1-most-wanted-mobile-feature-heres-why-research/>.

Luthfi, A., Distiawan, B., & Manurung, R. (2014). *Building an Indonesian Named Entity Recognizer using Wikipedia and DBPedia*. Faculty of Computer Science, Universitas Indonesia.

Mollewik, J. (2013). *Natural Language Interfaces over Spatial Data : Investigations in Scalability, Extensibility and Reliability*. Department of Computing Science, Umea University, Sweden.

OpenStreetMap: Open Database License. (2013). [http://wiki.openstreetmap.org/wiki/Open\\_Database\\_License](http://wiki.openstreetmap.org/wiki/Open_Database_License).

Presman, R. (2009). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York.

Stanford Core NLP, *Stanford Log-linear Part-Of-Speech Tagger*. (2004). <http://nlp.stanford.edu/software/tagger.shtml>.

Stanford Core NLP, *Stanford Named Entity Recognizer (NER)*. (2004). <http://nlp.stanford.edu/software/CRF-NER.shtml>.

Toutanova, K., Manning, C.D. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp. 63-70.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *In Proceedings of HLT-NAACL 2003*, pp. 252-259.

Zhang, C., Huang, Y., Mihalcea, R., & Cuellar, H. (2009). *A Natural Language Interface for Crime-related Spatial Queries*. Department of Computer Science and Engineering, University of North Texas.