

APLIKASI PEMANTAUAN LALU LINTAS YOGYAKARTA MENGUNAKAN GET SOURCE TWITTER DAN GOOGLE MAPS API BERBASIS WEB

Ruli Khairul Anas

Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta Indonesia 55283
Email : ruli.an@students.amikom.ac.id

Abstract

The use of social networks such as Twitter as a means of communication and information sharing is still very popular in Indonesia, especially in Yogyakarta until now. It is certainly can be used by the public to report traffic situation that occurring around such as congestion, accidents, checkpoint, and also flooding. The collected information is expected to be beneficial for all citizens of Yogyakarta in order to avoid a location where exposed to traffic problems. The traffic monitoring system utilizing social networking Twitter that provide Twitter API (Application Programming Language) to retrieve public data tweet about the traffic reports in the vicinity. Furthermore the tweets are processed and displayed on a web map display that utilizes with Google Maps API which shows the location of the user is located..

Keywords - Web Application, Traffic, Twitter API, Google Maps API, Node.js

1. Pendahuluan

1.1. Latar Belakang

Kota Yogyakarta merupakan salah satu kota di Indonesia yang perkembangannya cukup pesat dan juga menjadi destinasi pariwisata yang sangat terkenal di Indonesia. Hal ini tentu saja menimbulkan berbagai masalah klasik perkotaan di bidang lalu lintas. Dengan banyaknya kendaraan bermotor baik itu roda dua maupun roda empat yang setiap hari melintasi jalan-jalan di kota Yogyakarta, tentu saja pada jam-jam dan tempat-tempat tertentu jalanan menjadi macet. Ditambah lagi dengan cuaca yang tidak menentu ketika musim penghujan yang menyebabkan beberapa ruas jalan di Yogyakarta terendam air.

Penggunaan jejaring sosial seperti Twitter sebagai sarana komunikasi dan berbagi informasi masih sangat populer di Indonesia, khususnya Yogyakarta hingga saat ini. Hal ini tentu saja bisa dimanfaatkan oleh warga untuk melaporkan keadaan lalu lintas yang terjadi di sekitarnya seperti kemacetan, kecelakaan, pemeriksaan kendaraan, dan juga banjir. Informasi yang terkumpul ini diharapkan bisa bermanfaat untuk seluruh warga Yogyakarta agar bisa menghindari lokasi yang terkena masalah lalu lintas.

Google Maps dan Twitter telah menyediakan API (*Application Programming Interface*) yang bisa dimanfaatkan untuk diambil data-datanya. Data yang dapat diambil dari Google Maps adalah peta digital yang bisa dimodifikasi dengan memberikan titik koordinat pada lokasi-lokasi tertentu. Sedangkan Twitter menyediakan API *get source* yang berfungsi untuk mengambil data *tweet* dari penggunanya. Dengan menggabungkan keduanya, maka bisa dibuat aplikasi pemantauan lalu lintas. *Tweet* dengan hashtag atau tagar #lalinjogja akan dikumpulkan dan dianalisis berdasarkan geolocation tag-nya. Data ini kemudian langsung ditampilkan dalam wujud visual pada peta digital yang menunjukkan lokasi pengguna berada sesuai dengan posisi yang tercantum pada *tweet* sebagai laporan lalu lintas dari warga.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka permasalahan yang dapat dirumuskan adalah bagaimana cara merancang sebuah Aplikasi Pemantauan Lalu Lintas Yogyakarta Menggunakan Get Source Twitter Dan Google Maps API Berbasis Web?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini dimaksudkan untuk membuat aplikasi yang bermanfaat bagi masyarakat sebagai sarana untuk saling berbagi dan memantau tentang informasi lalu lintas di Yogyakarta

1.4. Tinjauan Pustaka

Wiryawan (2014) melakukan penelitian dengan judul "Incremental Learning Untuk Opinion Mining Pada Tweet Berbahasa Indonesia Menggunakan Data Stream Twitter". Tujuan dari penelitian tersebut adalah membangun sistem yang dapat melakukan opinion mining menggunakan data stream twitter yang menggunakan bahasa Indonesia. Dalam penelitian tersebut menggunakan data stream twitter sebagai sumber data dari sistem yang dibangun.[1]

Putra (2014) melakukan penelitian dengan judul "Sistem Multi Agen Untuk Ekstraksi Waktu Dan Tempat Event dari Data Twitter Untuk Web Kalender". Tujuan dari penelitian tersebut adalah menghasilkan aplikasi berbasis multiagen yang mampu memperoleh informasi

tempat, waktu, dan kategori dari data Twitter mengenai event.[2]

Prahasto (2015) melakukan penelitian dengan judul “Pengembangan Aplikasi Layanan Informasi Lowongan Kerja Menggunakan Data Stream Twitter Pada Perangkat Bergerak Dengan Sistem Operasi Android”. Tujuan dari penelitian tersebut adalah menghasilkan aplikasi yang dapat melakukan streaming data pada twitter untuk mendapatkan informasi lowongan kerja dan menyajikan informasi tersebut kepada pengguna dan memberitahu pengguna ketika ada informasi lowongan pekerjaan baru melalui perangkat bergeraknya yang berbasis android.[3]

1.5. Metode Penelitian

Metode pelaksanaan penelitian yang digunakan selama membuat skripsi ini meliputi:

- 1) Metode Pengumpulan Data
Dalam mengumpulkan data, metode yang digunakan adalah metode observasi, yaitu mengamati perilaku pengguna Twitter di Yogyakarta melalui user timeline, hastag atau tagar #lalinjogja, dan akun resmi yang biasa membagikan informasi tentang lalu lintas melalui tweet-nya.
- 2) Metode Analisis
Melakukan analisis terhadap permasalahan yang ada mengenai bagaimana data tweet diambil dari Twitter dengan hashtag tertentu lengkap dengan posisi lokasi geografisnya, yang kemudian diproses dan ditampilkan pada sebuah web.
- 3) Metode Perancangan
Proses perancangan disesuaikan dengan kebutuhan di lapangan, yaitu kemudahan dalam menggunakan aplikasi ini. Sehingga maksud dan tujuan dari pembuatan aplikasi ini dapat terpenuhi.
- 4) Metode Pengembangan
Aplikasi akan dibangun setelah proses perancangan selesai. Aplikasi ini dibangun pada platform komputer dengan spesifikasi yang diperlukan.
- 5) Metode Testing
Aplikasi yang sudah dibangun selanjutnya akan diuji, apakah aplikasi ini dapat bekerja maksimal sesuai dengan kebutuhan pengguna. Jika belum maka akan dilakukan perbaikan.
- 6) Metode Implementasi
Aplikasi yang telah menjalani proses uji coba dan berhasil, maka akan diimplementasikan ke dalam web agar bisa dioperasikan dan diakses pada lingkungan sebenarnya.

1.6. Landasan teori

1.6.1. API, Twitter, Google Maps, dan Node.js

1.6.1.1. API (Application Programming Interface):

API (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh *programmer* saat

membangun perangkat lunak. Dalam API terdapat fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*.[4]

1.6.1.2. Twitter:

Twitter adalah layanan jejaring sosial dan mikroblog daring yang memungkinkan pengguna untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (*tweet*). Twitter didirikan pada bulan Maret 2006 oleh Jack Dorsey. Platform Twitter menyediakan akses untuk data Twitter melalui API. Setiap API merepresentasikan aspek dari Twitter dan memungkinkan developer untuk membangun dan memperluas aplikasi mereka dengan cara baru berdasarkan kreativitas. Developer Twitter terus melakukan perkembangan terhadap API Twitter sehingga API Twitter dapat mengalami perubahan setiap saat.

1.6.1.3. Google Maps:

Google Maps adalah layanan pemetaan *online* yang disediakan oleh Google yang dapat diakses melalui situs <http://maps.google.com>. Situs tersebut menyediakan informasi geografis pada hampir seluruh wilayah di bumi. Google Maps API merupakan suatu fitur aplikasi yang dikeluarkan oleh Google untuk memberikan fasilitas kepada developer yang ingin mengintegrasikan Google Maps ke dalam website-nya dengan menampilkan data point milik sendiri. Dengan menggunakan Google Maps API, Google Maps dapat di-embed pada website eksternal. Agar aplikasi Google Maps dapat muncul pada website tertentu, maka diperlukan adanya API key yang merupakan kode unik yang diberikan oleh Google untuk suatu website tertentu agar server Google Maps dapat mengenali.

1.6.1.4. Node.js:

Node.js adalah *platform* yang dibangun atas JavaScript *runtime* Chrome untuk membuat aplikasi jaringan dengan performa tinggi dan skalabel. Node.js menggunakan I/O model non-blocking serta event-driven, yang membuatnya ringan serta efisien, sehingga tepat untuk aplikasi data-intensive real-time yang berjalan di perangkat terdistribusi. [5] NodeJS membuat web server sendiri, sehingga tidak perlu web-server seperti Apache dan jboss untuk menjalankan. Node.js tidak hanya untuk pemrograman sisi server, front end atau sisi client pun juga tercakup. Sekarang ini sudah banyak sekali framework terkenal menggunakan teknologi ini, salah satunya adalah Express.js. Node.js memiliki Package Management bernama NPM, dan hal ini

mempermudah pencarian plugin atau module untuk pengembangan.

1.6.1.5. *UML (Unified Modeling Language):*

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (Object Oriented). [6]

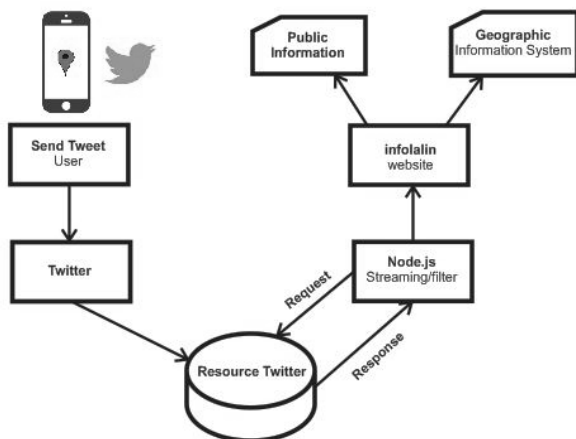
1.6.1.6. *SDLC (System Development Life Cycle):*

System Development Life Cycle (SDLC) adalah suatu model konsep yang digunakan di dalam manajemen proyek untuk menguraikan langkah-langkah yang terlibat dalam suatu proyek. Berbagai metode SDLC telah dikembangkan untuk memandu pengembangan sistem termasuk model waterfall, V-Model, Agile dan sebagainya.[7]

2. Pembahasan

2.1. *Gambaran Umum Sistem*

Pengguna Twitter mengirim *tweet* berupa teks yang berisi informasi lalu lintas di sekitarnya menggunakan perangkat bergerak. Dalam melakukan *tweet*, pengguna Twitter mengaktifkan fitur geolokasi dan menyertakan *hashtag* #lalinjogja. *Tweet* tersebut secara langsung akan terkirim ke Twitter. Sementara itu, sistem melakukan *streaming* terhadap Twitter menggunakan Node.js dan melakukan filter sesuai kriteria. Server yang berada pada Node.js melakukan *request* dan di-*response* oleh Twitter. Selanjutnya, data *tweet* yang sudah berhasil didapatkan ditampilkan pada website melalui peta dan widget (lihat gambar 1).



Gambar 1. *Gambaran Umum Sistem*

2.2. *Analisis Kebutuhan Fungsional*

Analisis kebutuhan fungsional adalah paparan mengenai fitur-fitur yang akan dimasukkan ke dalam aplikasi yang akan dibuat. Fitur-fitur tersebut antara lain sebagai berikut:

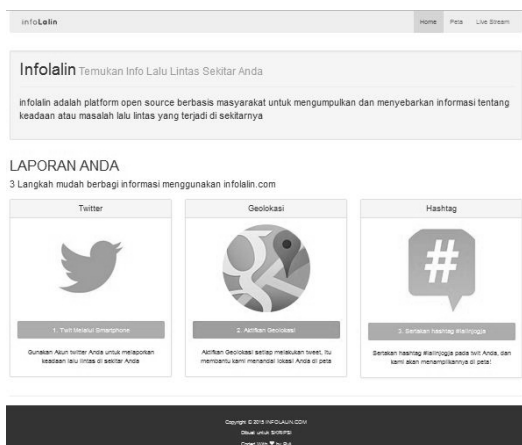
- 1) User:
 - a. User dapat mengakses aplikasi melalui web browser dan mengetahui informasi di dalamnya.
 - b. User dapat memantau lalu lintas melalui Maps pada aplikasi.
 - c. User dapat melakukan live streaming melalui widget Twitter pada aplikasi.
- 2) Aplikasi
 - a. Aplikasi dapat menampilkan peta wilayah Yogyakarta pada web browser menggunakan Google Maps API.
 - b. Aplikasi dapat menampilkan tweet dengan hashtag #lalinjogja pada peta.
 - c. Tweet yang tampil pada peta menyesuaikan dengan lokasi geografis yang tercantum dalam data tweet.

2.3. *Analisis Kebutuhan Non-Fungsional*

- 1) Spesifikasi perangkat keras yang digunakan untuk membangun aplikasi web infolalin adalah :
 - a. Acer Aspire One D270
 - b. Processor : Intel® Atom™ N2600 (1.60GHz, 1 MB L2 cache)
 - c. RAM : 2 GB DDR 3
 - d. Harddisk : 320 GB
 - e. VGA : Intel® Graphics Media Accelerator 3600 Series
- 2) Adapun spesifikasi minimal perangkat lunak yang dibutuhkan dalam analisis kebutuhan perangkat lunak sebagai berikut :
 - a. Perangkat lunak yang digunakan dalam pembuatan aplikasi web infolalin adalah:
 - a) Sistem Operasi Windows 7 32 bit
 - b) Sublime Text 2
 - c) Node.js
 - d) Web Browser Mozilla Firefox versi 41
 - b. Perangkat lunak yang dibutuhkan untuk menjalankan aplikasi web infolalin adalah semua jenis web browser pada perangkat desktop maupun mobile.

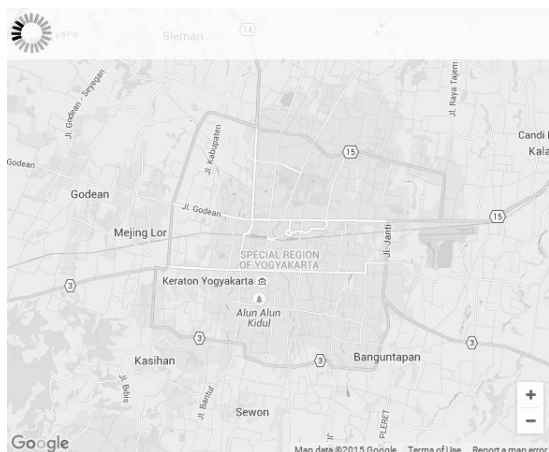
2.4. *Interface*

Halaman utama website ditampilkan setelah user mengakses infolalin melalui *web browser*. Tampilan halaman utama berisi informasi tentang deskripsi *website* dan cara menggunakan aplikasi infolalin. Halaman ini dibuat menggunakan *framework bootstrap* yang terdiri dari menu navigasi utama, *container* untuk konten, dan *footer* (lihat gambar 2).



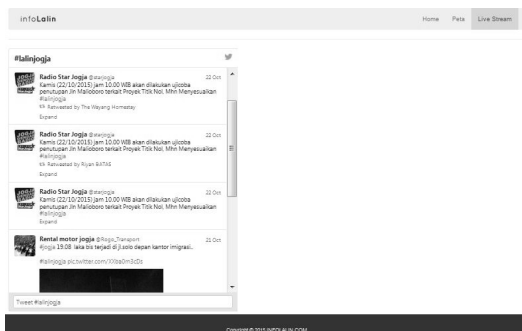
Gambar 2. Tampilan Halaman Utama

Halaman *Maps* adalah peta Yogyakarta yang ditampilkan pada web browser menggunakan Google Maps API. Pada bagian atas terdapat *log bar* sebagai tempat menampilkan data tweet dan *loading image* yang berfungsi sebagai tanda bahwa sistem belum mendapatkan data stream dari pengguna Twitter (lihat gambar 3).



Gambar 3. Tampilan Halaman Maps

Halaman *Live Stream* adalah halaman yang menampilkan *widget live stream tweet* dari Twitter, yang menampilkan *live tweet* dengan hashtag #lalinjogja (lihat gambar 4). Kode widget diperoleh dengan cara membuat *Twitter Embedded Timelines* melalui twitter.com/settings/widget/new/search



Gambar 4. Tampilan Halaman Live Stream

2.5. Pembahasan Kode Program

Program ini dibuat menggunakan bahasa pemrograman HTML dan *javascript* berbasis *client-server* serta *software* program editor Sublime Text 2 dan Node Js untuk memegang kendali *javascript* di sisi *server*.

Pada pembahasan kode program ini akan ditampilkan potongan kode program yang ada dalam aplikasi berdasarkan fungsi dari setiap masing-masing bagian. Potongan kode program pada pembahasan ini merupakan potongan kode program untuk membangun aplikasi infolalin. Kode program aplikasi ini memiliki dua bagian utama, yaitu: kode sisi server yang berjalan pada Node.js, dan kode sisi klien yang berjalan di browser. Server meminta Tweets dari Twitter, mem-*parsing*-nya dan kemudian melakukan *streaming* melalui WebSockets pada halaman web yang terbuka.

2.5.1. Akses Twitter

Sebelum menggunakan API Twitter, maka harus memiliki akses token yang berfungsi untuk melakukan API *request* kepada Twitter. Kode yang dibutuhkan ada 4 macam, yaitu:

1. API key > `consumer_key`
2. API secret > `consumer_secret`
3. Access token > `access_token_key`
4. Access token secret > `access_token_secret`

Untuk mendapatkan kode akses token API, yang pertama dilakukan adalah membuat sebuah aplikasi di Twitter melalui apps.twitter.com/new. Setelah mengisi data aplikasi yang dibutuhkan, kemudian pilih langkah selanjutnya, dan aplikasi yang dibuat sudah jadi. Setelah mendapatkan keempat kode akses API Twitter, maka implementasikan pada kode program `config.js` (lihat gambar 5).

```

1  var keys = {
2    consumer_key: '31eskKLZaMWSgQQX3sWrx4aI',
3    consumer_secret: 'GSu40kFzP9pCZk2oqL36GCqTmA9yNA2CFsq7Ld9inIHQZsQj',
4    access_token_key: '3187704530-PYxV0oxYkhkPHD7PHaCsoFVpx01oKbxD4r2zIS4',
5    access_token_secret: 'mZmSoycBhtX6T3LdgR8uGwNdzwezchzJBObRF6fxqasVhk'
6  };
7
8  exports.getKeys = function() {
9    return keys;
10 };
11
    
```

Gambar 5. Kode Program config.js

2.5.2. Package.json

Package.json menyimpan berbagai metadata yang terkait dengan proyek pembuatan aplikasi dan daftar dependensi. Untuk pembuatan aplikasi infolalin, dibutuhkan modul `socket.io`, `ntwitter`, dan `express`. File `package.json` terletak pada sisi server (lihat gambar 6).

```

1  {
2    "author": "ruli.an@students.amikom.ac.id",
3    "name": "infolalin",
4    "description": "Aplikasi Pemantauan Lalu Lintas Menggunakan Twitter, Google Maps, & node.js",
5    "version": "1.0.0",
6    "repository": {
7      "url": ""
8    },
9    "dependencies": {
10     "socket.io": "*",
11     "express": "*",
12     "twitter": "*"
13   },
14   "devDependencies": {},
15   "optionalDependencies": {},
16   "engines": {
17     "node": "0.8.x",
18     "npm": "1.1.x"
19   }
20 }
    
```

Gambar 6. Package.json

2.5.3. Kode Program Sisi Server

Kode program server.js (lihat gambar 7) berjalan pada sisi server, yaitu melalui node js. Server.js merupakan kode program yang berisi semua logika yang ada untuk menjalankan aplikasi infolalin. Kode ini pertama kali menjalankan web server menggunakan modul express. Server.js juga melayani halaman web statis, memuat modul web socket yaitu socket.io dan memuat modul Twitter API.

Dalam file javascript ini sudah diatur dalam pengaksesannya melalui web browser, yaitu menggunakan port 3000. Adapun direktori index yang diakses berada pada direktori public. Terdapat juga fungsi stream pada Twitter yang kemudian akan melacak tweet yang menggunakan hashtag #lalinjogja. Kemudian tweet tersebut diambil datanya berupa data profil pengguna twitter, data koordinat tempat user melakukan tweet, dan text tweet yang menggunakan hashtag #lalinjogja (lihat gambar 7).

```

1  var express = require('express')
2    , app = express()
3    , server = require('http').createServer(app)
4    , io = require('socket.io').listen(server);
5
6  var port = process.env.PORT || process.env.VMC_APP_PORT || 3000;
7  server.listen(port);
8
9  app.use(express.static(__dirname + '/public'));
10
11 var twitter = require('twitter');
12
13 var tw = new twitter(require('./config').getKeys());
14
15 tw.stream('statuses/filter', {track:'#infolalin'}, function(stream) {
16
17
18   stream.on('data', function (data) {
19     if (data.coordinates) {
20
21       io.sockets.emit('message', {
22         'id': data.id_str,
23         'text': data.text,
24         'lnglat': data.coordinates.coordinates,
25         'sname': data.user.screen_name,
26         'img': data.user.profile_image_url
27       });
28     }
29   });
30   stream.on('error', function (response) {
31     console.log(response);
32     process.exit();
33   });
34   stream.on('end', function (response) {
35     console.log(response);
36     process.exit();
37   });
38   stream.on('destroy', function (response) {
39     console.log(response);
40     process.exit();
41   });
42 });
    
```

Gambar 7. Kode Program server.js

2.5.4. Kode Program Sisi Klien

Kode program client.js (lihat gambar 8), menyiapkan Google Maps untuk ditampilkan pada web browser, yaitu peta Yogyakarta dengan titik koordinat -7.797591, 110.370714 dan tampilan peta bertipe ROADMAP. Kemudian client.js membuat koneksi WebSocket dengan server. Setelah server mengkonfirmasi telah menerima koneksi dan siap mengirimkan tweet, listener yang terkoneksi dipanggil dan klien mengirimkan pesan kembali ke server untuk merespon siap menerimanya melalui fungsi socket.

Data tweet yang ditampilkan adalah berupa hasil *streaming* dari twitter yang telah difilter berdasarkan hashtag #lalinjogja. Secara otomatis, penanda pada Maps akan muncul bersamaan dengan tweet yang ditangkap sesuai dengan letak titik koordinat dari pengguna Twitter.

```

1  $(document).ready(function() {
2    var map = new google.maps.Map(document.getElementById("map_canvas"), {
3      zoom: 8,
4      center: new google.maps.LatLng(-7.797591, 110.370714),
5      mapTypeId: google.maps.MapTypeId.ROADMAP,
6      mapTypeControl: false,
7      panControl: false,
8      streetViewControl: false,
9      zoomControl: true
10   });
11
12   var socket = io.connect();
13   socket.on('message', function(data) {
14     $('#img').attr('src', data.img);
15     $('#id').text(data.id);
16     $('#name').text(data.sname);
17     $('#text').text(data.text);
18
19     var lng = data.lnglat[0];
20     var lat = data.lnglat[1];
21     var marker = new google.maps.Marker({
22       map: map,
23       draggable: false,
24       animation: google.maps.Animation.DROP,
25       position: new google.maps.LatLng(lat, lng)
26     });
27
28     var infowindow = new google.maps.InfoWindow({
29       content: data
30     });
31
32     marker.addListener('click', function() {
33       infowindow.open(map, marker);
34     });
35
36   });
37 });
38
    
```

Gambar 8. Kode Program client.js

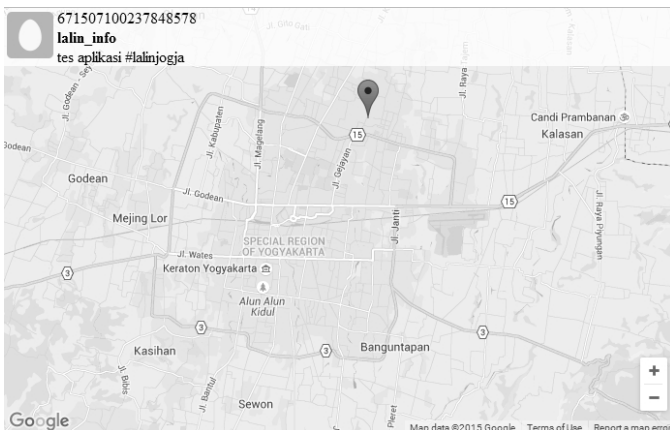
2.6 Black Box Testing

Black box testing merupakan pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Tujuan *black box testing* adalah untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Berikut hasil *test* yang telah dilakukan, yaitu uji coba *scrapping* data *tweet* dengan filter terhadap *hashtag* #lalinjogja dan *geo-tag* yang telah diaktifkan ketika pengguna Twitter melakukan *tweet* menggunakan media yang berbeda (lihat tabel 1).

Tabel 1. Uji Coba Scrapping Tweet

Filter	Geo Tag Aktif	Hashtag #lalinjogja	Hasil Scrapping
Desktop Browser	Tidak	Tidak	Salah
	Tidak	Ya	Salah
	Ya	Tidak	Salah
	Ya	Ya	Salah
Mobile Browser	Tidak	Tidak	Salah
	Tidak	Ya	Salah
	Ya	Tidak	Salah
	Ya	Ya	Salah
Mobile App Twitter	Tidak	Tidak	Salah
	Tidak	Ya	Salah
	Ya	Ya	Benar

Setelah data *tweet* berhasil diambil oleh sistem, maka data tersebut langsung ditampilkan pada peta berupa marker yang mempunyai titik koordinat atau geo-lokasi sesuai dengan pengguna Twitter mengirim *tweet*. Kemudian ditampilkan juga foto profil, *user ID*, nama *user*, dan *text tweet* pada *log bar* (lihat gambar 9).



Gambar 9. Hasil Uji Scrapping

3. Kesimpulan

Berdasarkan penjelasan pada bab sebelumnya hingga akhir dari perancangan aplikasi “infolalin” maka dapat disimpulkan :

- 1) Aplikasi berbasis web ini dibangun melalui tahap analisis yaitu dengan menggunakan analisis kelemahan, analisis kebutuhan dan analisis kelayakan, setelah itu tahap perancangan mulai dari rancangan antar muka, dan rancangan sistem.
- 2) Aplikasi ini mampu bekerja menjalankan sistem sehingga dapat melakukan streaming terhadap Twitter dan mengambil data *tweet* sesuai dengan yang dibutuhkan, kemudian menampilkan *tweet*

tersebut pada peta dengan koordinat sesuai letak pengguna ketika mengirimkan *tweet*.

- 3) Aplikasi ini bergantung penuh pada pengguna Twitter di lapangan yang melaporkan keadaan lalu lintas di sekitarnya melalui *tweet*-nya.

Daftar Pustaka

- [1] Wiryawan, Fajri. 2014. *Incremental Learning Untuk Opinion Mining Pada Tweet Berbahasa Indonesia Menggunakan Data Stream Twitter*. Universitas Gajah Mada, Yogyakarta.
- [2] Putra, Fajri P. 2014. *Sistem Multi Agen Untuk Ekstraksi Waktu Dan Tempat Event Dari Data Twitter Untuk Web Kalender*. Universitas Gajah Mada, Yogyakarta.
- [3] Prahasto, Bayu. 2015. *Pengembangan Aplikasi Layanan Informasi Lowongan Kerja Menggunakan Data Stream Twitter Pada Perangkat Bergerak Dengan Sistem Operasi Android*. STMIK AMIKOM YOGYAKARTA, Yogyakarta.
- [4] Setiawan, Bayu. 2013. *Definisi API (Application Programming Interface) Beserta Penerapannya*. <http://bayusetiawan15.blogspot.com/2013/01/definisi-api-application-programming.html>, diakses tanggal 24 Maret 2015 pukul 21.12.
- [5] Kominfo. 2013. *Open Source di Kominfo*. http://kominfo.go.id/index.php/content/detail/3434/Open+Source+di+Kominfo/0/program_prioritas#.VjqCnW7pWA8 diakses 5 November 2015 pukul 06.00AM
- [6] Suhendar, A. dan Gunadi, H. 2002. *Visual Modeling Menggunakan UML dan Rational Rose*. Informatika Bandung, Bandung.
- [7] S. Balaji and Dr. M. S. Murugaiyan. 2012. *Waterfall vs V-Model vs Agile : Comparative Study on SDLC*. International Journal of Information Technology and Business Management. Vol.2 No. 1.

Biodata Penulis

Ruli Khairul Anas, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2015.