

SKRIPSI

PEMBUATAN *TWITTER BOT* UNTUK Mencari Jalur  
TRANSPORTASI PUBLIK



KEVIN THEODORUS YONATHAN

NPM: 2011730037

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2014



**UNDERGRADUATE THESIS**

**FOLLOWING THE MAJORITY:  
A NEW ALGORITHM FOR COMPUTING  
A MEDIAN TRAJECTORY**



**KEVIN THEODORUS YONATHAN**

**NPM: 2011730037**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2014**



LEMBAR PENGESAHAN

PEMBUATAN *TWITTER BOT* UNTUK MENCARI JALUR  
TRANSPORTASI PUBLIK

KEVIN THEODORUS YONATHAN

NPM: 2011730037

Bandung, 4 Juli 2014

Menyetujui,

Pembimbing Tunggal

Pascal Alfadian, M.Com.

Ketua Tim Penguji

Anggota Tim Penguji

Thomas Anung Basuki, Ph.D.

Dr. rer. nat. Cecilia Esti Nugraheni

Mengetahui,

Ketua Program Studi

Thomas Anung Basuki, Ph.D.



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **PEMBUATAN *TWITTER BOT* UNTUK MENCARI JALUR TRANSPORTASI PUBLIK**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 4 Juli 2014

Meterai

Kevin Theodorus Yonathan  
NPM: 2011730037





## ABSTRAK

...

**Kata-kata kunci:** Lintasan, Homotopy, Fréchet, Lintasan Median, Penyangga



## ABSTRACT

...

**Keywords:** Trajectory, Homotopy, Fréchet, Median Trajectory, Buffer



*Dipersembahkan untuk diri sendiri*



## KATA PENGANTAR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Bandung, Juli 2014

Penulis



## DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xviii</b>
<b>DAFTAR TABEL</b>	<b>xix</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Batasan Masalah . . . . .	3
1.4 Tujuan . . . . .	3
1.5 Metode Penelitian . . . . .	3
<b>2 DASAR TEORI</b>	<b>5</b>
2.1 Twitter . . . . .	5
2.2 Twitter API . . . . .	6
2.2.1 OAuth . . . . .	7
2.2.2 REST API . . . . .	10
2.2.3 Streamming API . . . . .	11
2.3 KIRI API . . . . .	11
<b>DAFTAR REFERENSI</b>	<b>13</b>

## DAFTAR GAMBAR

2.1	flow application-only authentication . . . . .	9
2.2	Ilustrasi sign in . . . . .	10

## DAFTAR TABEL



## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Seiring dengan perkembangan zaman, perkembangan internet di Indonesia sudah semakin maju. Banyak orang sudah menggunakan fasilitas internet untuk berbagai macam kebutuhan. Contoh dari penggunaan internet adalah untuk mencari informasi, email, bermain jejaring sosial online, Internet Banking, online shop, dan lain lain. Menurut Kemkoinfo(8mei2014) *penggunainternetdiIndonesiacapai82Juta*

Twitter adalah salah satu layanan jejaring sosial online yang memungkinkan pengguna memposting pesan berbasis teks hingga 140 karakter. Pengguna Twitter menyebutnya sebagai *tweet*. *Tweet* ini akan meneruskan pesan singkat yang ditujukan ke semua *follower* suatu akun<sup>1</sup>. *Follow* adalah salah satu istilah dalam Twitter yang bertujuan untuk mengikuti aktivitas *tweet* suatu akun. Sedangkan cara seseorang untuk dapat memberi rujukan kepada akun Twitter yang lainnya adalah dengan cara *reply* atau lebih dikenal dengan nama *mention*<sup>2</sup>. Sebagai contoh, diketahui akun bernama @kviniink mem-*follow* @infobdg untuk mengetahui perkembangan apa saja yang terjadi di kota Bandung. Lalu akun @kviniink ingin bertanya tentang info mall yang ramai di Bandung, maka akun @kviniink membuat *mention tweet* yang berisikan "@infobdg Halo saya ingin bertanya apa saja mall yang sedang ramai di Bandung yah?".

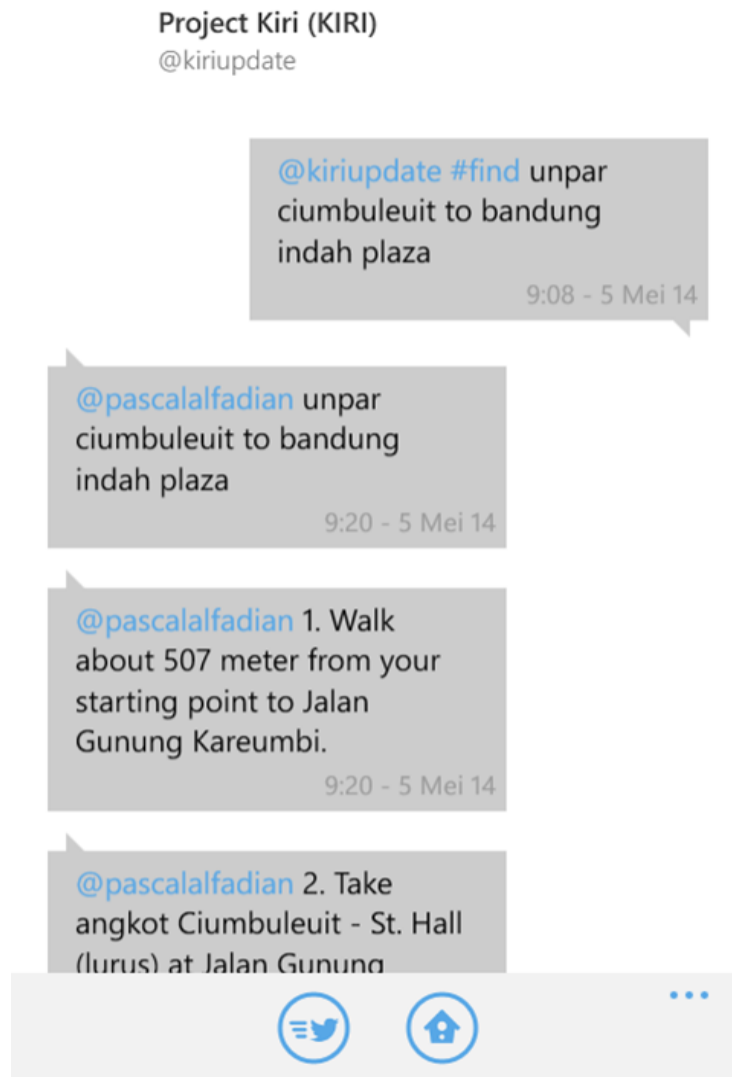
Transportasi publik sudah banyak digunakan oleh kebanyakan orang di dunia, bukan hanya di Indonesia saja transportasi publik ini sudah banyak digunakan di luar negeri. Menurut data, angkutan umum di Kota Bandung pada tahun 2013 sudah lebih dari 12000 unit kendaraan. Keuntungan memakai transportasi publik sudah banyak dirasakan di seluruh dunia yaitu untuk mengatasi kemacetan dan mengurangi pemanasan global. Seiring dengan perkembangan teknologi, menaiki transportasi publik menjadi semakin mudah. Seiring dengan perkembangan teknologi, menaiki transportasi publik menjadi semakin mudah. Dengan adanya KIRI di Indonesia terutama di daerah Bandung, masyarakat dapat naik transportasi publik tanpa harus mengetahui terlebih dahulu kendaraan yang harus dinaikinya. Dengan adanya KIRI, pengguna hanya perlu tahu tempat asal dan tempat tujuan untuk dapat menaiki transportasi publik di Bandung ini.

KIRI API adalah aplikasi pihak ketiga yang memungkinkan *programmer* mendapatkan data tentang info jalur transportasi publik. Twitter API adalah aplikasi pihak ketiga yang memungkinkan *programmer* melakukan manipulasi dan pengolahan data di Twitter. Dengan memanfaatkan KIRI API dan Twitter API peneliti akan membuat program yang dapat membalas *tweet* untuk mencari jalur transportasi publik. Program yang dibuat akan bersifat *real time* sehingga jika seseorang me-

---

<sup>1</sup>Dusty Reagan, *Twitter Application Development For Dummies*, Wiley, 2010, page 7

<sup>2</sup>Dusty Reagan, *Twitter Application Development For Dummies*, Wiley, 2010, page 9



lakukan mention kepada bot pencari jalur maka bot akan menangkapnya dan membalas mention tersebut berupa jalur yang harus ditempuh. Contoh dari jalannya program adalah ketika akun bernama @kviniink melakukan *mention* kepada @kiriupdate untuk bertanya jalur transportasi publik "@kiriupdate #find bip to ip". Maka Twitter bot @kiriupdate akan mendengarkan mention dari akun @kviniink lalu mention tersebut akan diolah oleh server dan akan di-reply dengan tiga buah tweet "@kviniink istana plaza to bandung indah plaza", "@kviniink Walk about 135 meter from your starting point to Jalan Aceh." , "@kviniink Take angkot Ciroyom - Antapani at Jalan Aceh, and alight at Jalan Pajajaran about 3.6 kilometer later.", "@kviniink Walk about 93 meter from Jalan Pajajaran to your destination.". Karena keterbatasan 140 karakter maka tweet akan dipecah sesuai dengan instruksi yang dikirimkan dari KIRI API.

Oleh karena itu, dalam penelitian ini akan dibangun sebuah perangkat lunak yang dapat memudahkan pengguna dalam mencari jalur transportasi publik. Sebuah aplikasi yang menggabungkan jejaring sosial online Twitter dengan KIRI API. Jadi pengguna bisa melakukan tweet kepada Twitter bot yang dibuat dengan format tertentu yang berisikan tempat asal dan tempat yang akan dituju. Lalu pengguna akan menerima balasan tweet berupa rute jalan yang harus ditempuhnya.

## 1.2 Rumusan Masalah

Mengacu kepada deskripsi yang diberikan, maka rumusan masalah pada penelitian ini adalah:

- Bagaimana membuat *Twitter bot* untuk mencari jalur transportasi publik?
- Bagaimana membuat *Twitter bot* untuk dapat merespon secara real time?
- Bagaimana memformat petunjuk rute perjalanan dalam keterbatasan tweet 140 karakter?

## 1.3 Batasan Masalah

Pada pembuatan perangkat lunak ini, masalah-masalah yang ada akan dibatasi menjadi:

- Input hanya mencakup Kota Bandung saja.
- Input yang diinputkan harus benar, memiliki asal dan tujuan yang jelas di Kota Bandung.
- Hasil yang dikeluarkan berupa tweet jalur transportasi publik.
- Media transportasi publik yang digunakan adalah angkutan umum.
- Pencarian jalur memanfaatkan KIRI API.

## 1.4 Tujuan

Tujuan dari penelitian ini adalah:

- Membuat aplikasi *Twitter bot* untuk mencari jalur transportasi publik.
- Membuat aplikasi Twitter yang bekerja secara *real time*.
- Membuat algoritma untuk memecah instruksi dari KIRI API dan mengubahnya ke dalam bentuk tweet.

## 1.5 Metode Penelitian

Pada perangkat lunak yang dibuat ini digunakan beberapa metode dalam penyelesaian masalah yang menjadi topik pada penelitian ini, antara lain:

1. Melakukan studi literatur, antara lain:
  - KIRI API,
  - REST API Twitter (<https://dev.twitter.com/docs/api/1.1>),
  - Streaming API Twitter (<https://dev.twitter.com/docs/api/streaming>).
2. Mempelajari pembuatan server dalam bahasa Java.
3. Membuat TwitterBot sederhana

4. Melakukan analisis terhadap teori-teori yang sudah dipelajari, guna membangun perangkat lunak yang dimaksud.
  5. Melakukan pengujian terhadap system yang sudah dibangun
- .



## BAB 2

# DASAR TEORI

Sebelum bisa membuat Twitter bot untuk mencari jalur transportasi publik, berikut diberikan beberapa definisi yang berkaitan dengan pembuatan Twitter bot. Bab ini akan menjelaskan Twitter, Twitter API, KIRI, KIRI API, dan Twitter4j.

### 2.1 Twitter

Twitter adalah layanan yang memungkinkan pengguna untuk mengirim pesan menggunakan 140 karakter atau kurang. Pesan tersebut dapat diadaptasikan melalui teks, aplikasi *mobile*, atau web. (referensi dari buku Sams teach yourself the twitter api) Berikut ini adalah daftar istilah umum pada Twitter:

Twitter adalah salah satu layanan jejaring sosial online yang memungkinkan pengguna memposting pesan berbasis teks hingga 140 karakter (referensi dari buku The Twitter Book).

- Tweet

Posting pada Twitter disebut sebagai *tweet*. *Tweet* ini akan meneruskan pesan singkat yang ditujukan ke semua *follower* suatu akun<sup>1</sup>. Contohnya adalah seorang akun @kviniink ingin menuliskan bahwa hari ini cuaca cerah, maka @kviniink akan men-tweet 'Hari ini cerah yah..' Tweet juga bisa menyertakan link ke video, foto, atau media lain di internet selain teks biasa. URL link teks termasuk ke dalam 140 batas karakter, namun URL tersebut akan menghabiskan tempat/space dari keterbatasan karakter tweet. Oleh karena itu URL akan dibuat versi singkatnya, contoh saat pengguna memasukkan link 'http://www.chacha.com/gallery/7253/15-movies-that-make-guys-cry', maka akan dibuat menjadi 'bit.ly/1uRi8vV'.

- Follow

Follow adalah satu istilah dalam Twitter yang bertujuan untuk mengikuti aktivitas *tweet* suatu akun. Following adalah ketika sebuah akun mengikuti akun orang lain, dan Follower adalah ketika sebuah akun melakukan aksi follow kepada akun anda.

- Reply

Reply adalah cara seseorang untuk dapat memberi rujukan kepada akun Twitter yang lainnya atau lebih dikenal dengan nama *mention*<sup>2</sup>. Sebagai contoh, diketahui akun bernama @kviniink

---

<sup>1</sup>Dusty Reagan, *Twitter Application Development For Dummies*, Wiley, 2010, page 7

<sup>2</sup>Dusty Reagan, *Twitter Application Development For Dummies*, Wiley, 2010, page 9

mem-*follow* @infobdg untuk mengetahui perkembangan apa saja yang terjadi di kota Bandung. Lalu akun @kviniink ingin bertanya tentang info mall yang ramai di Bandung, maka akun @kviniink membuat *mention tweet* yang berisikan "@infobdg Halo saya ingin bertanya apa saja mall yang sedang ramai di Bandung yah?".

- Retweet

Retweet ini merupakan salah satu yang paling penting dari Twitter(referensi the twitter book halaman 47). Retweet ini berguna ketika pengguna menemukan tweet menarik dan berbagi tweet tersebut dengan follower akun tersebut (*follower*). Retweet ini juga secara tidak langsung mengatakan bahwa "saya menghormati anda dan pesan yang anda buat".

- Hashtag

Sebuah fitur yang diciptakan oleh Twitter untuk membantu pencarian kata kunci dan penandaan suatu diskusi.

- Direct Message(DM)

Direct message digunakan untuk mengirim pesan yang bersifat private antara dua orang. Orang yang mengirim direct message ini hanya bisa untuk orang yang mengikuti akun tersebut.

- Timeline

Timeline adalah sekumpulan tweet-tweet dari semua orang yang anda follow lalu akan ditampilkan di halaman utama.

## 2.2 Twitter API

Twitter API adalah aplikasi pihak ketiga yang memungkinkan *programmer* melakukan manipulasi dan pengolahan data di Twitter. Twitter API tidak seperti API pada umumnya karena Twitter memaparkan hampir semuanya termasuk setup account dan informasi kustumisasi. Ini adalah salah satu bentuk pendekatan dari Twitter yang berfokus pada jaringan dan memungkinkan developer memiliki hak untuk berpikir 'out of the box' untuk membuat aplikasi yang mereka inginkan. Tetapi tetap akan terjadi keterbatasan yang dimiliki Twitter API, yaitu :

- Hanya bisa men-update 1000 per harinya, baik melalui handphone, website, API, dan sebagainya.
- Total pesan hanya bisa sebanyak 250 per harinya, pada setiap dan semua perangkat.
- 150 permintaan API per jam.
- OAuth diijinkan 350 permintaan per jam.

Twitter API sendiri dibagi menjadi dua yaitu REST API dan Streaming API.

### 2.2.1 OAuth

Dengan semakin berkembangnya website, semakin banyak situs yang bergantung pada layanan distribusi dan *cloud computing*. Contohnya adalah menggunakan jejaring sosial dengan menggunakan account media sosial lainnya seperti Google untuk mencari teman-teman yang sudah tersimpan pada kontak Google. Atau bisa juga menggunakan pihak ketiga yang memanfaatkan API dari beberapa layanan.

OAuth menyediakan suatu metode bagi pengguna untuk memberi akses pihak ketiga untuk resources (sumber daya) mereka tanpa berbagi password mereka. Cara ini juga memberikan cara untuk memberikan akses yang terbatas (dalam satu lingkup atau durasi). Sebagai contoh, seorang pengguna web dapat memberikan layanan percetakan (client) untuk mengakses foto pribadinya yang disimpan di layanan berbagi foto (server) tanpa harus memberikan username dan passwordnya. Ia akan mengotentikasi langsung dengan layanan berbagi foto tersebut yang mengeluarkan layanan percetakan.

Dalam model otentikasi client-server tradisional, klien menggunakan kredensial untuk mengakses resources hosted oleh server. Di dalam model OAuth, klien (bukan pemilik resource, tetapi bertindak atas namanya) meminta akses ke resource yang dikenalkan oleh pemilik resource namun diselenggarakan oleh server.

Agar klien dapat mengakses resource, pertama-tama ia harus mendapatkan izin dari si pemilik resource. Izin ini dinyatakan dalam bentuk token dan mencocokkan *shared-secret*. Tujuan dari token ini adalah untuk membuat pemilik resource untuk berbagi kepercayaan kepada klien. Berbeda dengan kepercayaan pemilik resource. Token dapat dikeluarkan dalam ruang lingkup terbatas, durasi yang terbatas, dan akan dicabut secara independen. Referensi (<http://hueniverse.com/oauth/guide/intro/>)

Twitter OAuth yang diberikan memiliki fitur

- secure

Pengguna tidak harus berbagi password mereka dengan aplikasi pihak ketiga untuk meningkatkan keamanan akun.

- Standard

Contoh code yang diberikan sudah kompatibel dengan implementasi Twitter OAuth.

Twitter sekarang memanfaatkan OAuth 1.0A. Otentikasi aplikasi pengguna adalah bentuk paling umum dari otentikasi resource dalam pelaksanaan OAuth 1.0A. Twitter sampai saat ini. Permintaan anda menandatangani baik untuk mengidentifikasi identitas aplikasi anda yang akan menyertakan izin untuk diberikan kepada pengguna. Hal ini bertujuan untuk dapat membuat panggilan API atas nama anda yang diwakili oleh akses token.

#### Application-only authentication

Twitter menawarkan aplikasi yang mampu mengeluarkan permintaan otentifikasi atas nama aplikasi itu sendiri. Dengan menggunakan Application-only authentication anda tidak mempunyai konteks dari otentifikasi pengguna dan ini berarti setiap request API untuk endpoint akan membutuhkan konteks user, seperti memposting tweet tidak akan bekerja. Aplikasi yang akan di dapat adalah:

- Melihat timeline
- Mengakses following dan follower dari suatu akun
- Mencari dalam tweet
- mengambil informasi dari akun manapun

Tetapi application-only authentication tidak bisa melakukan :

- Posting tweet
- Melakukan koneksi dengan Streaming endpoint
- Mencari akun seseorang
- menggunakan geo endpoint manapun
- mengakses DM

Auth Flow Langkah-langkah dari application-only auth terdiri dari beberapa langkah yaitu : Sebuah aplikasi dikodekan berdasarkan *consumer key* dan *secret* ke dalam satu set khusus yang dikodekan secara kredensial. Aplikasi membuat *request* ke POST OAuth2/token endpoint untuk merubah kredensial tersebut untuk *token bearer*. Ketika mengakses REST API, aplikasi menggunakan *token bearer* untuk otentifikasi. Karena tidak ada kebutuhan untuk menandatangani request, pendekatan ini lebih sederhana dari model standar OAuth 1.0a

Tentang application-only auth Token adalah *password*. Perlu diingat bahwa *consumer key* dan *secret*, *bearer token credential*, dan *the bearer token* itu sendiri memberikan akses untuk membuat permintaan atas nama aplikasi itu sendiri. Point-point ini harus dianggap sensitif layaknya password dan tidak boleh dibagikan atau didistribusikan kepada pihak yang tidak dipercaya atau tidak berkepentingan

SSL benar-benar dibutuhkan karena ini adalah cara otentifikasi yang aman. Oleh karena itu semua *request* (baik untuk mendapatkan atau menggunakan token) harus menggunakan endpoint HTTPS, yang juga merupakan syarat untuk menggunakan API v1.1.

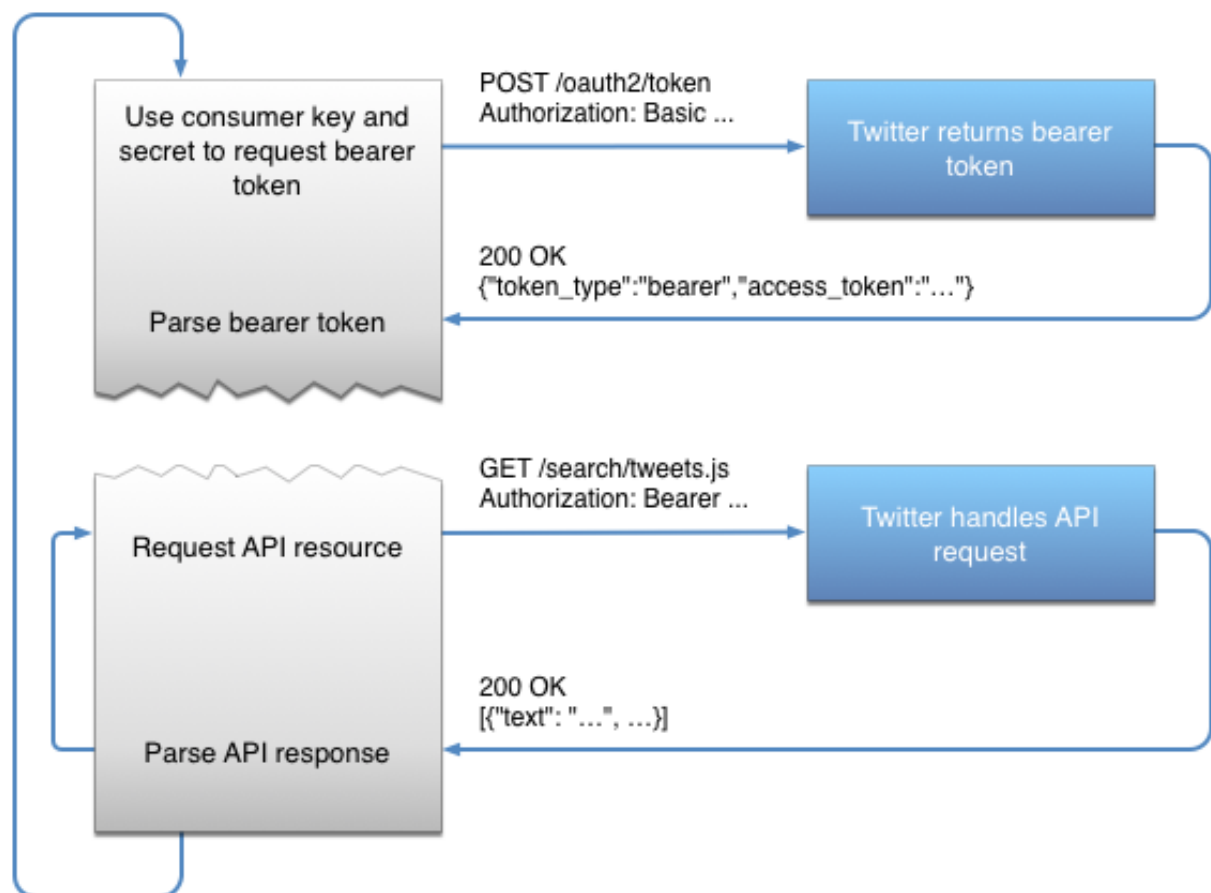
Tidak ada konteks pengguna. Ketika mengeluarkan permintaan menggunakan application-only auth, tidak ada konsep '*current-user*'. Karena itu endpoint seperti POST status / update tidak akan berfungsi dengan application-only auth.

*Rate limiting.* *Request* yang dibuat atas nama pengguna tidak akan menguras ketersediaan *rate limit* dan *request* tidak akan menguras batas penggunaan **limit** dalam *user-based auth*.

### 3-legged authorization

Cara kerja dari 3-legged authorization adalah dengan memberikan aplikasi yang anda buat untuk mengambil *access token* dengan cara melakukan *redirect* user dengan Twitter dan memberikan mereka sebuah otorisasi dari aplikasi yang anda buat. Cara kerja ini hampir identik dengan cara kerja yang dijelaskan pada implementasi *Sign in* dengan Twitter, hanya saja terdapat dua pengecualian yaitu:

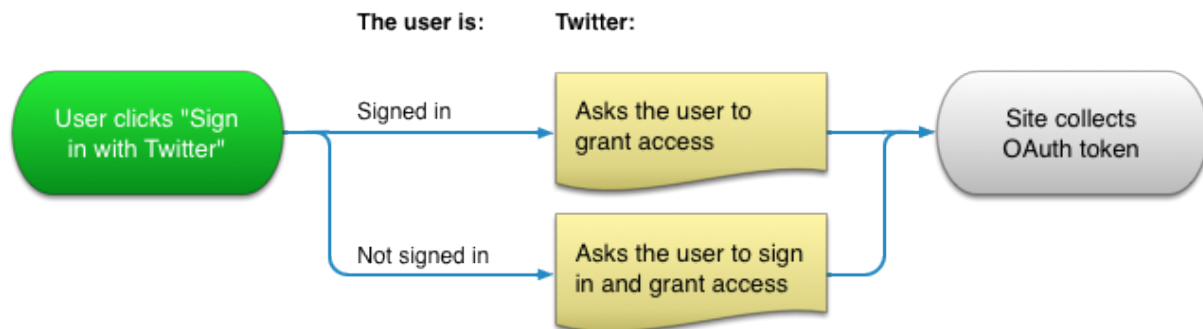
application-only authentication.png



Gambar 2.1: flow application-only authentication

- *GET oauth endpoint* digunakan sebagai pengganti *GET oauth*
- User akan selalu diminta untuk mengotorisasi akses ke aplikasi anda, bahkan jika akses sebelumnya telah diberikan

Beginilah ilustrasi interaksi *sign in* dengan menggunakan *following flowchart*



Gambar 2.2: Ilustrasi sign in

## PIN-based authorization

cara kerja dari PIN-based authorization ini ditujukan untuk aplikasi yang tidak bisa mengakses atau menanamkan *web browser* untuk mengarahkan user kepada *authorization endpoint*. Contohnya adalah aplikasi yang bersifat *command-line*, *embedded systems*, *game konsol*, dan beberapa jenis aplikasi *mobile*.

### Implementasi

Implementasi PIN-based ini memiliki cara kerja yang sama seperti implementasi Sign in dengan Twitter dan *3-legged authorization*, perbedaannya terletak pada nilai dari *oauthcallback* yang harus di set menjadi *oob* saat proses pemanggilan *POST oauth/requesttoken*.

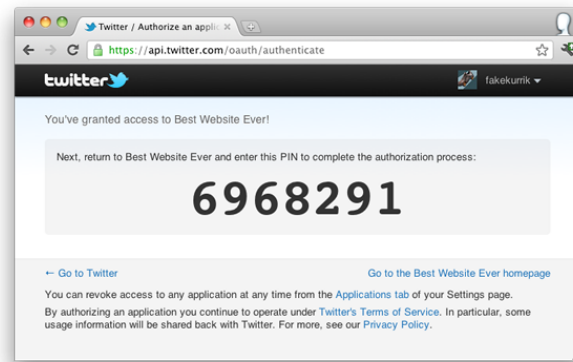
Setelah aplikasi anda telah mendapatkan *GET oauth/authenticate* atau *GET oauth/authorize URL*, tampilkan URL kepada user agar mereka dapat menggunakan web browser untuk mengakses Twitter.

Ketika *callback oob* diminta dan user mengunjungi Twitter, user tidak akan dipindahkan secara otomatis ke aplikasi setelah menyetujui akses. Sebaliknya, mereka akan melihat PIN code, dengan instruksi untuk kembali ke aplikasi dan memasukkan nilai dari PIN code tersebut.

Aplikasi anda harus memungkinkan user untuk memasukkan *PIN code* ini untuk menyelesaikan flow tersebut. Nilai dari *PIN code* harus lolos sebagai *oauthverifier* untuk *POST oauth/accesstoken request*. semua request akan berjalan normal kedepannya.

## 2.2.2 REST API

REST API menyediakan akses program untuk membaca dan menulis data Twitter. Data tersebut seperti menuliskan tweet baru, membaca profile, melihat follower, dan lainnya. REST API mengidentifikasi aplikasi Twitter dan pengguna menggunakan OAuth.



### 2.2.3 Streaming API

Streaming API adalah contoh *real-time* API. API ini ditujukan bagi para pengembang dengan kebutuhan data yang intensif. Contohnya jika mencari cara untuk membangun sebuah data produk data-mining atau tertarik dalam analisis penelitian. Streaming API memungkinkan melacak kata kunci yang ditentukan dalam jumlah besar dan melakukan suatu aksi (seperti tweet) secara langsung atau *real-time*.

Twitter menawarkan beberapa endpoint streaming, disesuaikan dengan kasus yang terjadi.

- Public stream

Streaming data publik yang mengalir melalui Twitter. Dipergunakan untuk mengikuti sebuah akun atau topik tertentu. Selain itu juga public stream digunakan untuk data mining.

- User Stream

Single-user streams, mengandung hampir semua data korespondensi ...

- Site Stream

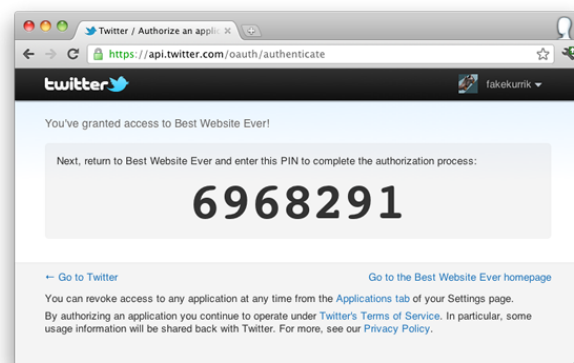
Versi dari multi-user stream. Site stream harus terhubung dengan server yang terkoneksi dengan twitter atas nama banyak pengguna.

Perbedaan antara Streaming API dan REST API yaitu ...

## 2.3 KIRI API

KIRI API adalah aplikasi pihak ketiga yang memungkinkan *programmer* mendapatkan data tentang info jalur transportasi publik. KIRI API dapat diakses dengan beberapa cara. Semua request harus berisikan API key, yang dapat diambil melalui KIRI API Management Dashboard. Berikut adalah spesifikasi dari KIRI API

- Routing Web Service
- Search Place Web Service
- Nearest Transports Web Service





## DAFTAR REFERENSI