

IMPLEMENTASI TEKNIK CRAWLING UNTUK PENGUMPULAN DATA DARI MEDIA SOSIAL TWITTER

PRAMANA YOGA SAPUTRA

Teknologi Informasi, Politeknik Negeri Malang

Email: pramanay@gmail.com

Abstract

Social media is an online communication network with millions of users. Users can interact with the other users, and share the information through the social media. The number of users and the high frequency of data traffic and data exchange, making social media into a data warehouse. Users can obtain valuable information from this data warehouse through the data collecting process, analysis and decision making. One of the techniques that used for collecting data is crawling. Crawling is the utilization of internet bot program to browse the addresses on the World Wide Web, to do indexing and to store important data in accordance with the purpose of the bot is made. In this research, we implementing crawling on Twitter. the reason why using Twitter, because Twitter is one of the biggest social media. There are approximately 700 million tweets per day, with an average of 7500 tweets per second from users around the world. Because of the amount of data stored in Twitter, the researcher chose Twitter, to implement crawling techniques in this study.

Key word: *crawling, social media, twitter*

1. PENDAHULUAN

Media sosial pada saat ini memiliki peran yang penting dalam berbagai aspek kehidupan manusia. Media sosial menjadi sarana untuk berbagi informasi bagi penggunanya. Saat ini tercatat 2.800 miliar pengguna aktif media sosial. Dari jumlah tersebut, dibagi menjadi beberapa kategori pengguna sesuai dengan jenis media sosial yang digunakan. Sesuai dengan kategorinya, media sosial dibagi menjadi lima kategori, yakni:

1. Jaringan sosial, misalnya seperti *Facebook, LinkedIn*, dan lain-lain
2. *Micro blogging*, seperti *Twitter, Tumblr*, dan lain-lain
3. *Photo sharing*, seperti *Instagram, Flickr*, dan lain-lain
4. *Video sharing*, seperti *YouTube, Vimeo* dan lain-lain
5. *Instant messaging*, seperti *WhatsApp, Line*, dan lain-lain.

Pengguna media sosial *Twitter* saat ini mencapai 300 juta pengguna. Kurang lebih terdapat 700 juta *tweet* perhari, dengan rata-rata 7.500 *tweet* per detik. Dengan jumlah tersebut, maka *Twitter*

merupakan gudang data yang menyimpan milyaran data dari seluruh penjuru dunia.

Data yang tersimpan di dalam *Twitter* sangat bermanfaat apabila bisa diolah menjadi informasi. Sebelum diolah menjadi informasi, data-data yang ada di dalam *Twitter* harus dikumpulkan terlebih dahulu, untuk selanjutnya dilakukan analisis dan dilakukan ekstraksi informasi. Untuk melakukan pengumpulan informasi, dapat digunakan sebuah teknik pengumpulan data dari *web* yang disebut dengan teknik *crawling*.

Pada penelitian ini, akan dilakukan implementasi teknik *crawling* untuk mengumpulkan data yang berasal dari *Twitter*. Dimana hasil dari penelitian ini dapat dimanfaatkan untuk analisis serta ekstraksi informasi. *Bot* untuk melakukan teknik *crawling* tersebut akan dikembangkan menggunakan dua bahasa pemrograman yang berbeda yakni Java dan R. Hal ini dilakukan untuk memberikan alternatif bagi pengguna dalam pemanfaatan *bot crawling*.

Penelitian ini berfokus pada implementasi teknik *crawling* pada media sosial *Twitter*, teknik *preprocessing* tidak

dibahas dalam penelitian ini. Pada akhir penelitian akan dilakukan pengujian dengan cara melakukan *crawling* terhadap beberapa kata kunci. Selanjutnya akan dibandingkan teknik *crawling* yang dikembangkan menggunakan Java dengan teknik *crawling* yang dikembangkan menggunakan R.

2. KAJIAN LITERATUR

2.1. Twitter

Twitter merupakan salah satu layanan jejaring sosial, yang memungkinkan penggunanya untuk saling berkomunikasi. Pengguna dapat mengirim serta membaca pesan yang terdiri dari maksimal 140 karakter, yang disebut dengan *tweet*. *Tweet* dari pengguna lain yang diikuti atau di-*follow* akan muncul pada halaman *home* untuk dibaca. Pengguna bisa melakukan *retweet* atau mengirim ulang pesan yang dikirim oleh pengguna lain. Pada pesan yang dikirim, apabila menyebutkan nama pengguna lain maka pada *tweet* tersebut tertulis tanda @ diikuti nama pengguna. Pengguna dapat menggunakan tanda # (*hashtag*) untuk menulis pesan berdasarkan topik.

Twitter didirikan pada Maret 2006 oleh Jack Dorsey. Pertumbuhan jumlah pengguna *Twitter* naik dengan cepat, sehingga kini terdapat 300 juta orang pengguna *Twitter*. *Twitter* naik ke peringkat kedua sebagai situs jejaring sosial yang paling sering dikunjungi di dunia, dari yang sebelumnya menempati peringkat dua puluh dua. Tingginya popularitas *Twitter* menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, misalnya sebagai sarana protes, kampanye politik, sarana pembelajaran, dan sebagai media komunikasi darurat.

Twitter memiliki beberapa fitur utama yang merupakan identitas dari *Twitter* itu sendiri, yang membedakannya dengan media sosial yang lain. Fitur-fitur tersebut antara lain:

1. *Tweet*

Tweet adalah sebuah *posting* di *Twitter*, yang hanya mengizinkan pesan 140 karakter atau kurang. Secara standar, *tweet* pengguna dapat terlihat oleh umum, namun pengguna dapat membatasi pengiriman *tweet* hanya bagi pengikut mereka (*followers*). Pengguna bisa mengirim *tweet* melalui situs *Twitter*, aplikasi eksternal yang kompatibel (seperti untuk telepon pintar), ataupun melalui layanan pesan singkat (SMS) yang tersedia di negara-negara tertentu. Layanan-layanan tersebut bersifat gratis, kecuali layanan SMS, yang dikenakan biaya oleh penyedia layanan seluler.

Pengguna bisa berlangganan *tweet* pengguna lain dengan cara mengikuti (*follow*) pengguna yang bersangkutan, dan pengguna yang mengikuti tersebut akan menjadi pengikut (*followers*) bagi pengguna yang diikutinya. Sebutan lainnya adalah *tweeps*, singkatan dari *Twitter* dan *peeps*. Pengguna bisa memeriksa orang-orang yang membatalkan pertemanan (*unfollowing*) melalui layanan pihak ketiga. Selain itu, pengguna juga bisa memblokir pengguna lain yang telah mengikuti mereka.

2. Konten

Konten adalah isi dari *tweet* pengguna. Dari web www.quicksprout.com diperoleh bahwa konten dari *Twitter* berisi:

- 93% berisi teks saja, tanpa gambar maupun video
- 65% teks berisi tautan ke halaman web lain
- *Tweet* berisi tautan, 86% lebih sering di-*retweet*
- Antara gambar dan video, gambar lah yang paling banyak ditemui dalam *tweet* pengguna
- 62% gambar berupa humor, sisanya 38% adalah tipe lain
- Teks yang berisi *how-to* atau *list-based article* memiliki

kemungkinan tiga kali lebih besar untuk di-*retweet* dibandingkan teks biasa

3. Format

Pengguna dapat mengelompokkan kicauan menurut topik atau jenis dengan menggunakan tagar (hashtag) – kata atau frasa yang diawali dengan tanda "#". Sedangkan tanda "@", yang diikuti dengan nama pengguna, digunakan untuk mengirim atau membalas *tweet* pada pengguna lain. Untuk mengirim ulang *tweet* pengguna lain dan membaginya pada pengikut sendiri, terdapat fitur *retweet*, yang dilambangkan dengan "RT". Pada akhir 2009, Twitter menambahkan fitur "daftar" (*list*), yang memungkinkan pengguna untuk mengirim dan membalas *tweet* pada daftar pengguna, bukannya pada pengguna secara perorangan.

Kapasitas *tweet* yang ditetapkan dibatasi 140 karakter untuk alasan kompatibilitas dengan pesan SMS, memperkenalkan notasi singkat dan bahasa slang yang umum digunakan dalam SMS. Batas 140 karakter ini telah meningkatkan penggunaan layanan penyingkat URL seperti bit.ly, goo.gl, dan tr.im, serta layanan penampung konten seperti Twitpic, memozu.com, dan NotePub, untuk mengakomodasi konten multimedia dan teks yang panjangnya lebih dari 140 karakter. Sejak Juni 2011, Twitter telah menggunakan domain sendiri, yakni t.co, untuk menyingkat otomatis semua URL yang diposting di situsnya.

4. Trending Topic

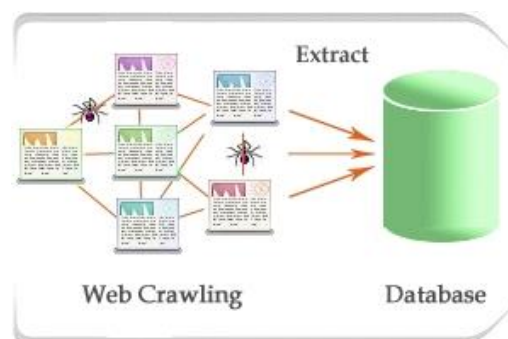
Sebuah kata, frasa, atau topik yang lebih banyak dibicarakan daripada topik lainnya disebut dengan topik hangat/tren (*trending topics*). Suatu topik bisa menjadi tren karena adanya upaya terpadu oleh pengguna, ataupun karena adanya suatu peristiwa yang mendorong orang untuk membicarakan satu topik

tertentu. Topik ini membantu *Twitter* dan penggunanya untuk memahami apa yang sedang terjadi di dunia. *Twitter* sendiri telah mengubah algoritma topik hangat terdahulu untuk mencegah manipulasi tren.

2.2. Crawling

Crawling adalah suatu teknik yang digunakan untuk mengumpulkan informasi yang ada dalam *web*. *Crawling* bekerja secara otomatis, dimana informasi yang dikumpulkan berdasarkan atas kata kunci yang diberikan oleh pengguna.

Alat yang digunakan untuk melakukan *crawling* disebut dengan *crawler*. *Crawler* berbentuk program yang diprogram dengan algoritma tertentu, sehingga dapat melakukan pemindaian ke halaman-halaman *web*, sesuai dengan alamat *web* ataupun kata kunci yang diberikan pengguna. Pada saat melakukan pemindaian, *crawler* akan membaca teks yang ada, *hyperlink* dan berbagai tag yang digunakan di halaman *web* tersebut. Berdasarkan informasi tersebut, *crawler* akan mengindeks informasinya atau pun menyimpan informasi tersebut ke dalam sebuah *file* atau ke dalam *database*. Ilustrasi proses *crawling* ditunjukkan pada gambar 1.



Gambar 1 Web Crawling (<https://cdn-ssl-promptcloud.pressidium.com/wp-content/uploads/2015/02/Web-Crawling.jpg>)

Twitter crawling merupakan penerapan teknik *crawling* pada media sosial *Twitter*. Teknik ini bertujuan untuk mengumpulkan data *tweet* dari media

sosial *Twitter*. Dari data yang telah dikumpulkan, dapat dianalisa dan diambil informasi penting yang bermanfaat bagi pengguna. Untuk menerapkan teknik *twitter crawling* ini, pihak *Twitter* telah memberikan akses bagi pengguna untuk memanfaatkan *Twitter API*. Sehingga dengan memanfaatkan *Twitter API* tersebut, pengguna bisa dengan mudah memperoleh data-data seperti *tweet*, data pengguna dan lain-lain. Untuk selanjutnya dikumpulkan dan disimpan dalam sebuah file atau basis data.

2.3. *Twitter API*

Twitter API, merupakan *API* (*Application Programming Interface*) yang disediakan oleh *Twitter* untuk memfasilitasi pengguna agar dapat berinteraksi dengan data-data yang ada pada *Twitter*. Data-data ini misalnya adalah *tweet*, id pengguna, lokasi, waktu pembuatan *tweet* dan lain-lain.

Untuk memanfaatkan *Twitter API*, pengguna harus menggunakan bahasa *server side scripting* seperti php, python, R dan lain-lain. Dengan menggunakan bahasa-bahasa tersebut, pengguna dapat melakukan *request* kepada *Twitter API*, dan respon hasilnya dirupakan dalam format JSON.

Agar komunikasi pengguna dengan *Twitter API* aman, maka *Twitter* menerapkan *OAuth* atau *Open Authorization*. *OAuth* merupakan protokol terbuka yang memungkinkan pengguna untuk berbagi *resource* pribadi seperti foto, video, data pengguna dan lain-lain yang tersimpan di suatu situs web, dengan situs lain tanpa memberikan nama pengguna dan kata sandi pengguna tersebut. *OAuth* mengizinkan pengguna untuk memberikan akses kepada situs pihak ketiga untuk mengakses informasi mereka yang disimpan di penyedia layanan lain tanpa harus membagi izin akses atau keseluruhan data mereka. Proses *OAuth* pada *Twitter* dapat dilihat pada gambar 1.

OAuth bergantung pada tiga set dari token dan secret yang didapat dari server dan client:

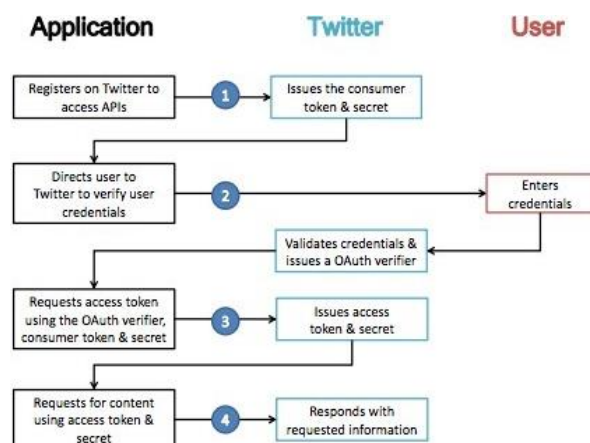
1. *Consumer key* dan *consumer secret*
Consumer key adalah *unique identifier* untuk client yang digunakan client untuk melakukan request untuk mendapatkan request tokens.
2. *Request token* dan *request token secret*
Request token adalah *temporary one-time identifier* yang diberikan oleh server untuk tujuan permintaan kepada user untuk melakukan *grant permission* pada client. *Token secret* digunakan untuk melakukan *sign request* untuk mendapatkan sebuah *access token*.
3. *Access token* dan *access token secret*
Access token adalah *identifier* untuk digunakan oleh client untuk melakukan akses ke *resources* milik user. Sebuah client bisa melakukan akses ke *resources* milik user selama *token* nya masih *valid*. Server bisa melakukan *revoke* kapanpun karena sudah *expire* atau user melakukan *revoke* secara manual. *Secret* digunakan untuk *sign request* ke *resources* yang di proteksi oleh akses user.

Cara untuk mendapatkan consumer key, consumer secret, access token dan access secret adalah sebagai berikut:

1. Client melakukan request ke server menggunakan sebuah consumer key
2. Client menggunakan consumer key untuk mendapatkan sebuah request token dan secret.
3. Client melakukan redirect pada user ke server untuk grant permission untuk client melakukan akses ke resources milik user. Proses ini bisa terjadi jika request token telah diautentikasi.
4. Client melakukan request ke server untuk memberikan access token dan secret . Ianya merepresentasikan sebuah identifier dan shared secret

yang client nya bisa gunakan untuk mengakses resources atas nama user.

5. Ketika membuat sebuah request untuk akses ke resource terproteksi , client menyertakan Authorization header yang berisi consumer key, access token , signature method dan sebuah signature , timestamp , sebuah nonce , dan untuk opsionalnya adalah versi dari OAuth yang digunakan.



Gambar 2 Alur Kerja OAuth Twitter

2.4. Twitter4J

Twitter4J merupakan *library* Java untuk menggunakan *Twitter API*. Dengan memanfaatkan *Twitter4J*, pengguna dapat mengintegrasikan layanan *Twitter* dengan aplikasi yang dikembangkan menggunakan bahasa Java. Untuk menggunakan *Twitter4J*, pengguna dapat mengunduh *library* tersebut di <http://twitter4j.org>.

Twitter4J dikembangkan oleh beberapa orang dengan memanfaatkan Github. Pengguna juga dapat ikut memberikan kontribusi, ataupun mencoba contoh pemanfaatan kode programnya. *Twitter4J* terdiri atas beberapa *java package*, antara lain:

- twitter4j
- twitter4j.api
- twitter4j.auth
- twitter4j.conf
- twitter4j.json
- twitter4j.media
- twitter4j.util
- twitter4j.util.function

2.5. twitterR R Package

Bahasa R memiliki *package* yang digunakan oleh user untuk memanfaatkan *Twitter API*. *Package* tersebut diberi nama *twitterR*. *Author* dari *package twitterR* ini adalah Jeff Gentry. Dengan memanfaatkan *twitterR* ini, pengguna dapat dengan mudah mengakses *Twitter API* dan memanfaatkan fitur-fiturnya. Untuk mengetahui fungsi-fungsi dari *package twitterR* dapat dilihat pada tabel 1.

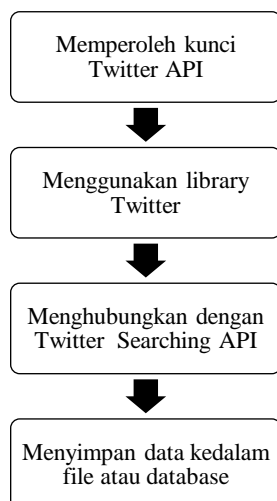
Tabel 1 Daftar Fungsi Pada Package *twitterR*

Nama Fungsi	Keterangan
retweets	Fungsi yang digunakan untuk <i>retweet</i>
taskStatus	Sebuah fungsi untuk mengirim Twitter DM setelah menyelesaikan sebuah tugas
timelines	Fungsi untuk melihat jadwal tweet pada Twitter
get_latest_tweet_id	Sebuah fungsi untuk mengambil ID tweet terbaru dari database
search_twitter_and_store	Fungsi untuk menyimpan tweet yang dicari ke database
getTrends	Berfungsi untuk melihat tren Twitter
register_db_backend	Fungsi untuk menata backend database untuk <i>twitterR</i>
directMessage-class	Kelas "directMessage": Kelas untuk mewakili Pesan Langsung Twitter
getUser	Fungsi untuk mengatur pengguna Twitter
twListToDF	Fungsi untuk mengubah daftar <i>twitterR</i> menjadi data.frames
favorites	Sebuah fungsi untuk mendapatkan tweets favorit
getCurRateLimitInfo	Fungsi untuk mengambil informasi batas tingkat saat ini
setup_twitter_oauth	Mengatur <i>credential</i> OAuth untuk sesi <i>twitterR</i>
friendships	Fungsi untuk merinci hubungan antara diri Anda dan pengguna lainnya

strip_retweets	Sebuah fungsi untuk menghapus retweet
searchTwitter	Cari twitter
import_statuses	Fungsi untuk mengimpor objek twitter dari berbagai sumber
updateStatus	Fungsi memanipulasi status Twitter
load_tweets_db	Fungsi untuk menahan / memuat data twitter ke database
dmGet	Fungsi memanipulasi pesan langsung Twitter
showStatus	Fungsi untuk mengembalikan status
user-class	Objek wadah untuk memodelkan pengguna Twitter
decode_short_url	Fungsi untuk memecahkan kode URL yang dipersingkat
status-class	Kelas berisi status Twitter
registerTwitterOAuth	Mendaftar <i>credential</i> OAuth ke sesi twitter R
use_oauth_token	Menetapkan <i>credential</i> OAuth untuk sesi twitter dari objek Token yang ada

3. METODE PENELITIAN

Penelitian yang dilakukan untuk melakukan implementasi teknik *crawling* pada media sosial *Twitter* ini menggunakan metode yang ditunjukkan pada gambar 3.



Gambar 3 Proses Implementasi Teknik Crawling

Pada gambar 2 tersebut dijelaskan secara bertahap tentang proses yang dilakukan oleh peneliti dalam mengimplementasikan teknik *crawling*. Diawali dengan cara memperoleh kunci Twitter API, dilanjutkan dengan pemanfaatan *library Twitter*, untuk Java maupun R. Kemudian dilanjutkan lagi dengan menghubungkan kode program dengan *Twitter Searching API*. Dan tahap terakhir adalah mencari kemudian menyimpan data yang sudah ditemukan kedalam file atau database.

Berikut ini adalah penjelasan dari gambar 3:

a) Memperoleh kunci Twitter API

Hal pertama yang harus dilakukan oleh pengguna adalah membuat akun Twitter. Tujuannya untuk memperoleh *API Key*, *API Secret*, *Access Token* dan *Access Token Secret*. Langkah-langkah memperoleh key tersebut adalah sebagai berikut:

- Membuat akun *Twitter*, apabila pengguna belum memilikinya
- Mengakses <https://apps.twitter.com/> kemudian *log in* menggunakan akun yang telah dibuat sebelumnya.
- Membuat aplikasi baru, dengan menekan “*Create New App*”
- Mengisi form, menyetujui *term-term* yang ada, dan klik “*Create Your Twitter Application*”
- Pada halaman berikutnya, klik tab “*Keys and Access Tokens*”, pengguna dapat melihat *API Key*, *API Secret*, *Access Token* dan *Access Token Secret*, dan dapat memanfaatkannya pada aplikasinya.

b) Menggunakan library Twitter

Pada aplikasi yang dikembangkan, dilakukan *import library Twitter*. Pada pengembangan menggunakan Java, maka digunakan *Twitter4J*. Sedangkan untuk R, digunakan *library twitterR*.

c) Menghubungkan dengan Twitter Searching API

Setelah dilakukan *import*, selanjutnya adalah menghubungkan aplikasi yang sedang dikembangkan dengan *Twitter*, caranya adalah memasukkan *API Key*, *API Secret*, *Access Token* dan *Access Token Secret* pada bagian program, dari aplikasi yang sedang dikembangkan.

d) Menyimpan data kedalam file atau database

Setelah aplikasi dijalankan, maka hasil dari *crawling* disimpan dalam file, berupa JSON, XML, CSV, XLS atau ke dalam *database*.

4. HASIL DAN PEMBAHASAN

Dari penelitian yang dilakukan, dihasilkan aplikasi yang digunakan untuk melakukan *crawling tweet* pada *Twitter*. Aplikasi ini dikembangkan dengan menggunakan Java dan R. Keduanya sama-sama memanfaatkan *Twitter API* untuk melakukan *crawling*. Sehingga pada saat implementasi, perlu memasukkan *API Key*, *API Secret*, *Access Token* dan *Access Token Secret*.

Listing kode program untuk aplikasi yang dikembangkan menggunakan bahasa pemrograman Java dijelaskan pada tabel 2.

Tabel 2 Listing Kode Program Menggunakan Java

```
package twitcrawl;
import twitter4j.*;
import
twitter4j.conf.ConfigurationBuilder;
import java.util.*;
public class TwitCrawl {
public static void main(String[]
args){
    ConfigurationBuilder cb = new
ConfigurationBuilder();
    cb.setDebugEnabled(true);
    cb.setOAuthConsumerKey(fill
using your consumer key);

    cb.setOAuthConsumerSecret(fill using
your consumer secret);
    cb.setOAuthAccessToken (fill
using your access token);
    cb.setOAuthAccessTokenSecret
(fill using your access token secret);
    TwitterFactory tf = new
TwitterFactory(cb.build());
    Twitter twitter =
tf.getInstance();
    Query query = new Query(fill
using your keyword);
    int numberOfTweets = 10000;
    long lastID = Long.MAX_VALUE;
    ArrayList<Status> tweets =
new ArrayList<Status>();
    while(tweets.size() <
numberOfTweets){
        if (numberOfTweets -
tweets.size() > 100){
            query.setCount(100);
```

```
        }
        else{
            query.setCount(numberOfTweets -
tweets.size());
        }
        try{
            QueryResult result =
twitter.search(query);
            tweets.addAll(result.getTweets());

            System.out.println("Gathered
            "+tweets.size()+" tweets");
            for(Status t:
            tweets){
                if(t.getId()<lastID){
                    lastID =
                    t.getId();
                }
            }
        } catch(TwitterException
te){
            System.out.println("Couldn't connect:
            "+te);
        }
        query.setMaxId(lastID-
1);

        for (int i = 0; i <
tweets.size(); i++) {
            Status t = (Status)
tweets.get(i);
            GeoLocation loc =
t.getGeoLocation();
            String user =
t.getUser().getScreenName();
            String msg =
t.getText();
            String date =
t.getCreatedAt().toString();
            String time = "";
            if (loc!=null) {
                Double lat =
t.getGeoLocation().getLatitude();
                Double lon =
t.getGeoLocation().getLongitude();

                System.out.println(i + " USER: " +
user + " wrote: " + msg + " located at
                " + lat + ", " + lon + " - " +date);
            }
            else
                System.out.println(i + " USER: " +
user + " wrote: " + msg + " - " +date);
        }
    }
}
```

Setelah dijelaskan tentang *listing* kode program dalam bahasa pemrograman Java, pada tabel 2. Selanjutnya pada tabel

3, dijelaskan listing program pada bahasa R.

Tabel 3 Listing Kode Program R

```
library(twitteR)
api_key <- "fill using your
consumer key"
api_secret <- "fill using your
consumer secret"
access_token <- "fill using your
access token"
access_token_secret <- "fill
using your access token secret"
setup_twitter_oauth(api_key,api
_secret,access_token,access_tok
en_secret)
# harvest some tweets
some_tweets =
searchTwitter("fill using your
keyword ", n=10000, resultType =
"recent")
tweets.df <-
twListToDF(some_tweets)
# get the text
some_txt = sapply(tweets.df,
function(x) x$getText())
#write to xls
library(xlsx) #load the package
write.xlsx(x = tweets.df, file =
"test.excelfile_R.xlsx",
sheetName =
"TestSheet", row.names = FALSE)
```

Dalam penelitian ini dilakukan pengujian untuk melihat kinerja teknik crawling yang diimplementasikan menggunakan Java maupun R. Pengujian dilakukan dengan memberikan beberapa *term* kepada aplikasi, untuk dicari di dalam *Twitter*. Selain *term*, yang menjadi parameter pencarian adalah banyaknya *tweet* yang akan di-*crawl* oleh aplikasi. Daftar parameter yang dimasukkan kedalam aplikasi, tercantum pada tabel 4.

Tabel 4 Daftar Parameter Pencarian

No	Term	n-cari
1	kebijakan publik	500
2	#pilkadadki2017	1.000
3	BBM Satu Harga	2.000
4	#ahokdjarot	5.000
5	indonesia	10.000

Hasil ujicoba dengan menggunakan parameter yang tertulis

pada tabel 4 menunjukkan bahwa untuk *term* 1, 2, dan 4 menghasilkan jumlah *tweet* yang sama antara aplikasi yang dikembangkan menggunakan Java dan R. Sedangkan untuk *term* 3 dan 5, hasil *tweet* yang diperoleh berbeda. Hasil ujicoba ditunjukkan pada tabel 5.

Tabel 5 Daftar Parameter Pencarian dan Hasil

N o	Term	n-cari	n-hasil	
			Java	R
1	kebijakan publik	500	298	298
2	#pilkadadki2017	1.000	1.000	1.000
3	BBM Satu Harga	2.000	1.093	1.108
4	#ahokdjarot	5.000	3.831	3.831
5	indonesia	10.000	7.850	10.000

Pada *term* ke-1, ke-3 dan ke-5, yakni “kebijakan publik”, “BBM Satu Harga”, dan “#ahokdjarot” hasilnya kurang dari jumlah parameter pencarian. Hal tersebut disebabkan karena jumlah *tweet* yang ada pada *Twitter*, untuk *term* tersebut hanya sejumlah *n-hasil*. Sedangkan untuk *term* ke-2, yakni “#pilkadadki2017”, hasil *tweet* yang diperoleh sama dengan jumlah parameter pencarian. Untuk *term* “indonesia”, hasil yang diperoleh dari aplikasi yang dikembangkan menggunakan Java berbeda dengan aplikasi yang dikembangkan menggunakan R. Aplikasi Java hanya memperoleh 7.850 *tweet*, sedangkan R memperoleh 10.000, sesuai dengan jumlah parameter pencarian. Perbedaan ini disebabkan karena pada aplikasi yang dikembangkan dengan Java mengalami *rate limit exceeded*. Penyebabnya adalah untuk memenuhi parameter jumlah *tweet* 10.000, aplikasi Java menggunakan lebih dari 15 *requests*, yang merupakan batas *request* yang diberikan *Twitter* untuk aplikasi atau pengguna yang memanfaatkan *Twitter API*. Sedangkan untuk aplikasi yang dikembangkan menggunakan R, tidak melampaui batas *request* yang diizinkan *Twitter*.

5. KESIMPULAN

Penelitian telah dilakukan dengan membuat aplikasi *crawling Twitter* dalam dua bahasa pemrograman, yakni Java dan R. Hasil dari kedua aplikasi yang dikembangkan berbeda untuk jumlah data pencarian yang banyak. Untuk pencarian *tweet* sebanyak 10.000 pada aplikasi Java hanya diperoleh hasil sebanyak 7.850. Sedangkan untuk aplikasi R, diperoleh 10.000 *tweet* sesuai dengan jumlah pencarian yang diinginkan. Hal tersebut karena pada aplikasi Java melampaui batas request yang diizinkan *Twitter* untuk memenuhi jumlah *tweet* yang dicari. Dari hasil tersebut dapat disimpulkan bahwa implementasi teknik *crawling* menggunakan R memiliki performa yang lebih baik dibandingkan dengan implementasi teknik *crawling* menggunakan Java.

6. REFERENSI

- Akadol, P.L.J. 2010. *OAuth Untuk Pengamanan Credentials Pada Akses REST API*. Indonesian Security Conference 2010.
- Azam, N., Jahiruddin, Abulaish, M., Halder, N.A. 2015. *Twitter Data Mining for Events Classification and Analysis*. Proceedings of the 2nd International Conference on Soft Computing and Machine Learning Intelligence (ISCMI'15), Hong Kong, IEEE CPS, Nov. 23-24, 2015
- Kumar, S., Morstatter, F., Liu, H. 2013. *Twitter Data Analytics*. Springer
- Sembodo, J.E., Setiawan, E.B., Baizal, Z.K.A. 2016. *Data Crawling Otomatis pada Twitter*. Ind. Symposium on Computing Sept 2016. pp. 11-16
doi:10.21108/indosc.2016.111
- Wanda, R.L. 2011. *Apasih Pengertian OAuth (Open Authorization)?*. Diperoleh 20 Maret 2017, dari <http://wwdq.blogspot.co.id/2011/05/apasih-pengertian-oauth-open.html>