

PENERAPAN ALGORITMA BOYER MOORE PADA POSTING TWITTER TMC POLDA METRO JAYA UNTUK MELAPORKAN KONDISI LALULINTAS DAN RUTE JALAN KOTA JAKARTA

Rudi Setiawan

Program Studi Sistem Informasi,
Fakultas Telematika, Universitas Trilogi, Jakarta Selatan
+62-8192454119, Email: rudi@universitas-trilogi.ac.id

Abstrak

Setiap harinya jutaan tweets di share oleh berbagai pengguna Twitter didunia, salah satunya tweets yang terdapat pada account Twitter TMC Polda Metro Jaya, yang mana informasinya didapat dari pengguna Twitter yang peduli terhadap kondisi lalulintas kota Jakarta. Dengan menggunakan algoritma Boyer-Moore yang merupakan algoritma pencarian string, pada data tweets TMC Polda Metro Jaya dilakukan pencarian nama jalan atau lokasi beserta kondisi lalulintasnya, kemudian dilakukan visualisasi menggunakan peta Google Map, sehingga ketika user memilih lokasi tujuan perjalanan dari suatu tempat ke tempat lainnya di kota Jakarta maka akan ditampilkan rute jalan beserta kondisi lalulintasnya berdasarkan data tweets yang telah dilaporkan pada hari tersebut. Berdasarkan hasil penelitian, pada sampel 1 hari percobaan ditanggal 17 November 2014 dengan jumlah data 232 tweets, didapat data tweets yang teridentifikasi kondisi lalulintasnya sebanyak 196 tweets artinya sebanyak 84,48% tweets dapat teridentifikasi dan yang tidak teridentifikasi kondisi lalulintasnya sebanyak 15,52%. Penyebab utama dari tidak teridentifikasinya kondisi lalulintas pada data tweets dikarenakan pada data tweets tersebut tidak melaporkan kondisi lalulintas melainkan melaporkan informasi lain seperti kecelakaan, kondisi jalan berlubang dan lain sebagainya.

Kata Kunci : Algoritma Boyer-Moore, TMC Polda Metro Jaya, Twitter

1. PENDAHULUAN

Fenomena jejaring sosial dan situs *microblogging* mulai dari *Facebook*, *Twitter*, *LinkedIn*, *Path*, *Google+* dan lain sebagainya, kini semakin marak digunakan oleh masyarakat sebagai sarana berbagi informasi. Dengan didukung perangkat mobile dan teknologi jaringan nirkabel yang memadai, kini masyarakat dapat berbagi informasi kapan dan dimanapun mereka berada.

Mengenai maraknya penggunaan situs *microblogging* *Twitter*, pada penelitian yang dilakukan Rodiansyah (2012), dinyatakan bahwa pada pertengahan tahun 2010 *Twitter* memiliki lebih dari 106 juta pengguna diseluruh dunia dan terus meningkat setiap harinya sebanyak 300.000 pengguna, dan *Twitter* setiap harinya mendapatkan lebih dari 3 juta *request*. Dari angka tersebut Indonesia menjadi negara yang menduduki peringkat 8 dalam mengakses situs *Twitter*. *Twitter* menerima *tweets* dari pengguna sebanyak 55 juta pesan setiap harinya.

Dengan adanya *Big Data* yang dimiliki oleh *Twitter*, penulis mencoba memanfaatkan data *tweets* untuk kemudian diolah menggunakan algoritma *Boyer Moore*. Algoritma *Boyer Moore* digunakan untuk mengidentifikasi data *tweets* yang

melaporkan tentang kondisi lalu lintas jalan raya di kota Jakarta, untuk kemudian dengan bantuan *Google Maps* kondisi lalu lintas jalan raya akan divisualisasikan kedalam peta lokasi perjalanan.

2. METODE PENELITIAN

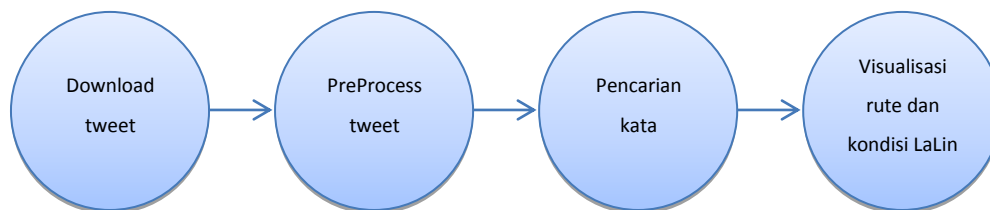
Sistem yang dirancang merupakan sistem yang diharapkan dapat menemukan informasi lalu lintas dari posting *Twitter* yang terdapat pada *account Twitter* TMC Polda Metro Jaya, menggunakan algoritma pencocokan string *boyer moore* sebagai pengolah kata yang terdapat pada *tweets* untuk menemukan kondisi lalu lintas, kemudian menggunakan *GoogleMapsAPI* untuk menampilkan rute perjalanan dari lokasi A ke lokasi B berdasarkan pilihan pengguna.

2.1 PENGUMPULAN DATA

Data yang dikumpulkan merupakan data *tweet* yang terdapat pada *account Twitter* TMP Polda Metro Jaya, yang secara *realtime* didownload dengan memanfaatkan *API search* yang disediakan oleh *Twitter*. Kemudian data yang telah didownload tersebut disimpan kedalam database. Data *tweet* yang dikumpulkan antara lain : isi *tweet*, tanggal dan waktu *tweet*.

2.2 ARSITEKTUR SISTEM

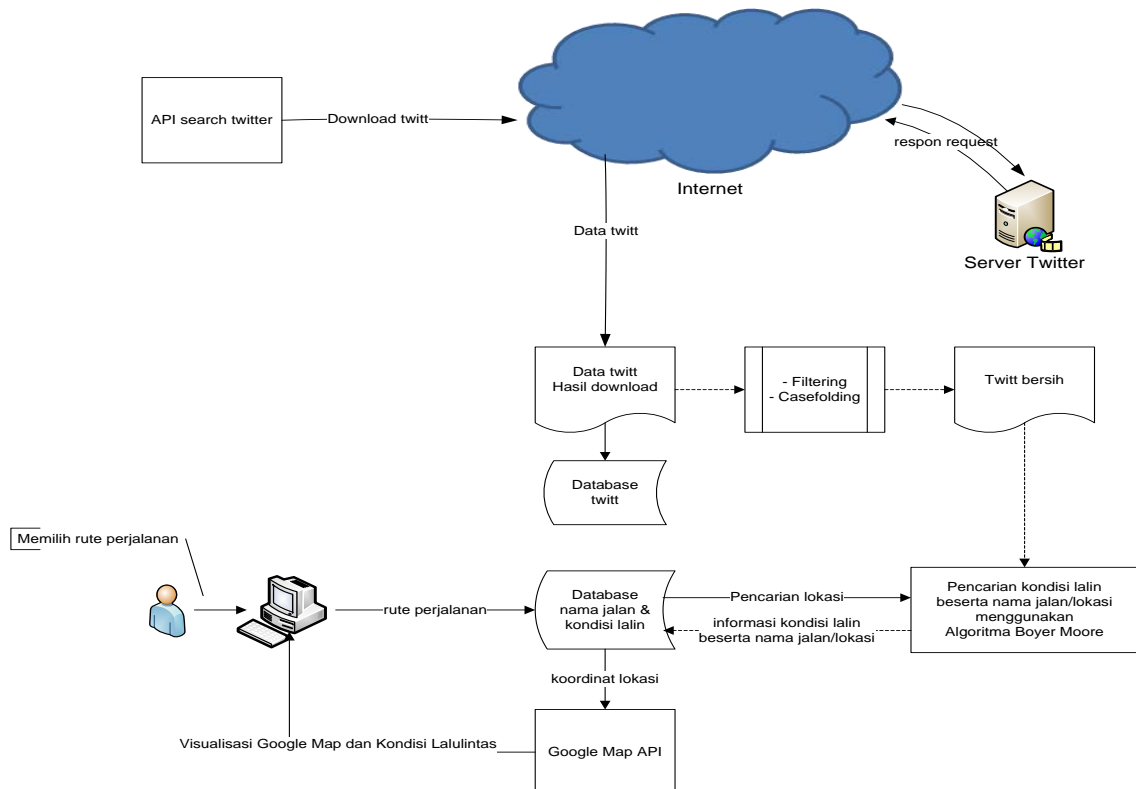
Secara umum sistem yang dirancang terdiri dari empat bagian diantaranya adalah *download tweet* (pengambilan data), *preprocessing tweet*, pencarian kata yang mencerminkan kondisi lalu lintas dan visualisasi rute lalu lintas beserta kondisi lalu lintasnya. Adapun desain arsitektur sistem dapat terlihat pada gambar 1.



Gambar 1. Rancangan Arsitektur Sistem

2.3 PERANCANGAN SISTEM

Secara garis besar sistem ini bertujuan melaporkan kondisi lalu lintas berdasarkan data *tweets* dan menggunakan teknik pencocokan string dengan metode pencocokan string menggunakan algoritma *boyer moore*, kemudian hasilnya akan divisualisasikan dengan bantuan *Google Map* berdasarkan rute perjalanan yang telah dipilih oleh pengguna aplikasi. Gambar 2 memperlihatkan gambaran umum sistem.



Gambar 2. Gambaran umum sistem

Cara kerja sistem dimulai dengan mengambil data *tweets* yang tersimpan di server *Twitter* menggunakan bantuan *API Search Twitter*. Pada proses pengambilan data ini juga dilakukan pelabelan data. Kemudian data tersebut diproses untuk tahap persiapan (*preprocessing*) agar data siap digunakan untuk proses identifikasi. Hal ini dilakukan karena tidak semua data *tweets* tersebut dapat digunakan. Pada data *preprocessing* ini dilakukan pembersihan data *tweets* dari kata-kata yang tidak digunakan dan mengganti kata-kata tertentu dengan daftar sinonim yang sudah ada.

3. HASIL DAN PEMBAHASAN

3.1 *Twitter API*

Twitter merupakan sebuah web *micro blogging* dimana penggunanya dapat mengekspresikan *moment* atau ide nya kedalam sebuah teks, foto maupun video. Jutaan *tweets* di *share* secara *realtime* setiap harinya dan *Twitter* menolong anda dalam membuat dan membagi ide dan informasi secara *instant* tanpa ada batasan. Gambar 3 merupakan contoh postingan *Twitter*.



Gambar 3. Postingan Tweets

Untuk dapat menggunakan *API Twitter*, pengguna diwajibkan login di alamat <https://dev.Twitter.com> untuk mendapatkan 4 buah *key* berupa *consumer key*, *consumer secret*, *access token* dan *access token secret* yang akan digunakan sebagai syarat *authentication* untuk dapat mengakses data *Twitter*. Gambar 4 merupakan *source code* untuk melakukan *download tweets*.

```

1 <?php
2 session_start();
3 require_once("twitteroauth.php");
4 $notweets = 5;
5 $twitteruser = "TMC PoldaMetro";
6 $consumer_key = "6I6JldElixZDWR2B3A20ew";
7 $consumer_secret = "C1hCgxI0oqUFYeSuAM9I4oDeiSGCXm16lkmqF14o";
8 $access_token = "461641225-ewmCnrVxZi6hVARnxGgB0ticSKoQGXXnHQ5pWzLP";
9 $access_token_secret = "41aLyepTOKUho2bjmaPRATEH2ngwDR18DLoNe0dUFBvh1";
10
11 function GetConnectionWithAccessToken($cons_key, $cons_secret, $oauth_token, $oauth_token_secret)
12 {
13     $connection = new TwitterOAuth($cons_key, $cons_secret, $oauth_token, $oauth_token_secret);
14     return $connection;
15 }
16 $connection = GetConnectionWithAccessToken($consumer_key,$consumer_secret,$access_token,$access_token_secret);
17 $tweets= $connection->get("https://api.twitter.com/1.1/statuses/user_timeline.json?screen_name=".$twitteruser."&count=".$notweets);
18 include "config.php";
19
20 foreach($tweets as $tweet) {
21     echo '';
22     echo $tweet->created_at.'<br />';
23     echo $tweet->user->name.' ('.$tweet->user->screen_name.') <br />';
24     echo $tweet->text.'<br /> <br />';
25     list($hari, $bulan, $tgl, $wkt, $no1, $thn) = split(' ', $tweet->created_at);
26     $delete = "Insert into twitt (twitt, date) VALUES ('$tweet->text','$tgl $bulan $thn')";
27     $delete_query = mysql_query($delete);
28 }
29 >>

```

Gambar 4. Source code download tweets

Pada baris kode ke 26 dan 27 dari gambar 4, data hasil download *tweets* dimasukkan kedalam sebuah *table* yang bernama *tabletweets*, data yang berada ditabel *tweets* tersebutlah yang nantinya akan diproses untuk mendapatkan kondisi lalu lintas dari setiap nama jalan atau lokasi yang telah dilaporkan oleh pengguna *Twitter*. Gambar 5 menunjukkan data *tweets* hasil download.

id	twitt	date
29	09:45 Giat Penertiban/Penegakan Hukum penerobos jalur Trans Jakarta lokasi Kota tua Tamansari Jakbar http://t.co/P2d4CV3yp0 (@budi_30A)	20 Feb 201
30	RT @AreaJakartaRaya: 09:39 Kondisi arus lalin Jl ry Kalimalang arah Jl Halim Baru padat sbiliknya ramai lancar http://t.co/grkKeKZdQj #AJR	20 Feb 201
31	RT @PolsektroTabang: 11:17 Apel pengamanan unjuk rasa di Mall City Walk Jl Kh Mas Mansyur Jakpus http://t.co/X9mdbkP8dx	20 Feb 201
32	RT @fachritanjung: 11:11 Terjadi kecelakaan truk terbalik di KM 59 (pintu tol ciujung kragilan) tol merak. http://t.co/ihntIshXckf	20 Feb 201
33	11:03 Hindari jalur kebon sirih Arus lalin Kebon sirih arah Tugu tani padat imbas vol kendaraan	20 Feb 201
34	11:00 Giat penyampaian pendapat dr Karyawan Depo Pertamina di Ktr PHI Jl MT Haryono Jaksel http://t.co/gPIo2HL62m	20 Feb 201
35	10:53 Arus lalin Serpong raya dari Alam Sutera arah Cikokol ramai lancar	20 Feb 201
36	11:31 Arus lalin Patung Kuda arah Bunderan HI ramai lancar kedua arah	07 Jan 201
37	11:20 Situasi cuaca wil Fatmawati hujan gerimis jalan licin harap berhati-hati	07 Jan 201
38	11:15 Situasi cuaca wilayah Semanggi , Bunderan Pd Indah hujan gerimis jalan licin harap berhati-hati	07 Jan 201
39	11:00 Kegiatan Polisi sahabat anak pengenalan rambu LL di TK Nurul Falah Jakut oleh Dikmas satwil LL Jakut http://t.co/gvDL1e2OB	07 Jan 201
40	10:57 Suasana lalu lintas jl pemuda rawamangun arah pulogadung kedua arah lancar #cuaca cerah http://t.co/c1aTrEvetF @INurfajri	07 Jan 201

Gambar 5. Data tweets hasil download

3.2 Algoritma Boyer Moore

Algoritma *Boyer Moore* dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Pencocokan karakter yang dilakukan menggunakan algoritma *boyer moore* dimulai dari string sebelah kanan *pattern*. Ide di balik algoritma ini ialah dengan memulai pencocokan karakter dari kanan, maka akan lebih banyak informasi yang didapat. Gambar 6, menunjukkan *pseudocode* algoritma *Boyer Moore* pada fase pencarian.

BOYER_MOORE_MATCHER (T, P)

```

procedure BoyerMooreSearch(
    input  $m, n$  : integer,
    input  $P$  : array[0..n-1] of char,
    input  $T$  : array[0..m-1] of char,
    output ketemu : array[0..m-1] of boolean
)

```

Deklarasi:

```

 $i, j, shift, bmBcShift, bmGsShift$ : integer
 $BmBc$  : array[0..255] of integer
 $BmGs$  : array[0..n-1] of integer

```

Algoritma:

```

preBmBc( $n, P, BmBc$ )
preBmGs( $n, P, BmGs$ )
 $i := 0$ 
while ( $i \leq m - n$ ) do
     $j := n - 1$ 
    while ( $j \geq 0$  and  $T[i+j] = P[j]$ ) do
         $j := j - 1$ 
    endwhile
    if ( $j < 0$ ) then
         $ketemu[i] := true$ ;
    endif
     $bmBcShift := BmBc[chartoint(T[i+j])] - n + j + 1$ 
     $bmGsShift := BmGs[j]$ 
     $shift := \max(bmBcShift, bmGsShift)$ 
     $i := i + shift$ 

```

Gambar 6. Pseudocode algoritma Boyer-Moore (Sumber: <http://id.wikipedia.org>)

Algoritma *Boyer Moore* termasuk algoritma *string matching* yang paling efisien dibandingkan algoritma-algoritma *string matching* lainnya (Cormen dkk. 1994). Karena sifatnya yang efisien, banyak dikembangkan algoritma *string matching* dengan bertumpu pada konsep algoritma *Boyer Moore*, beberapa di antaranya adalah algoritma *Turbo BM* dan algoritma *Quick Search*.

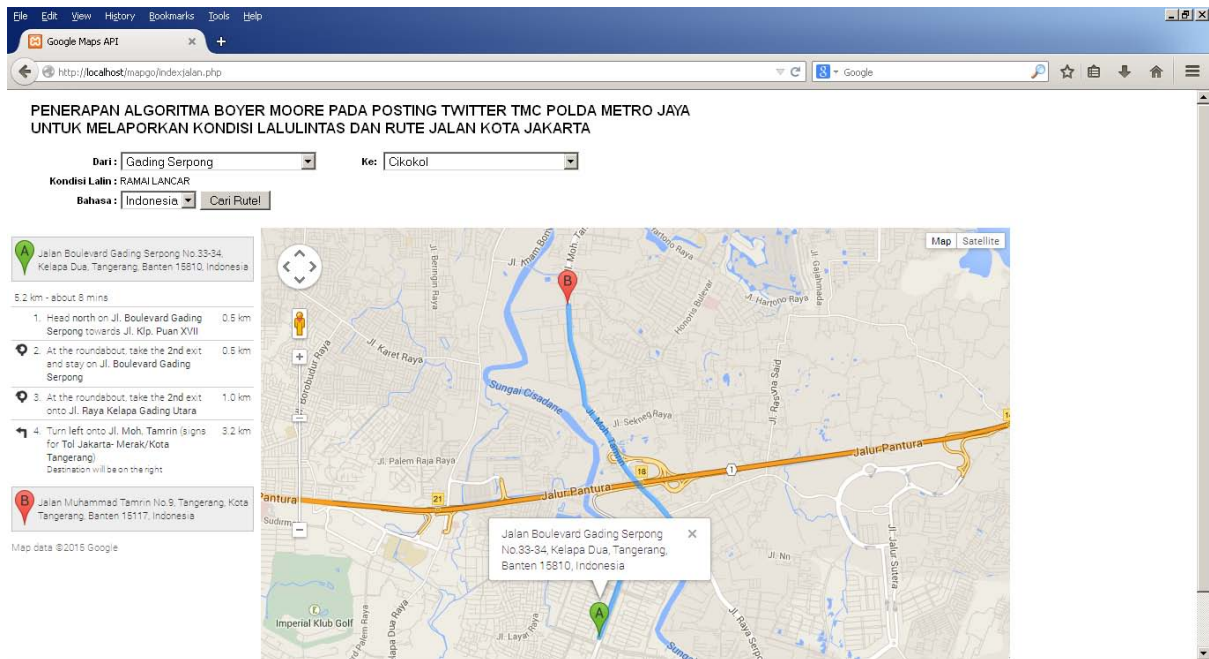
Adapun langkah-langkah pencarian string pada algoritma Boyer Moore sebagai berikut :

1. Pertama diperlukan 2 tabel dengan pendekatan Match Heuristic (MH) dan Occurrence Heuristic (OH) untuk menentukan jumlah pergeseran yang akan dilakukan pada pattern (P) jika ditemukan karakter yang tidak cocok pada proses pencocokan terhadap karakter pada teks (S).
2. Jika pada proses perbandingan terdapat ketidakcocokan karakter antara karakter pada P dan S, maka pergeseran dilakukan dengan melihat kedua tabel dengan nilai pergeseran paling besar yang dipilih.
3. Kemungkinan penyelesaian dalam melakukan pergeseran pada P adalah jika pada pencocokan sebelumnya tidak ada karakter yang cocok maka pergeseran dilakukan dengan melihat nilai pergeseran pada tabel occurrence heuristic. Jika karakter yang sedang dibandingkan tidak terdapat pada P maka pergeseran dilakukan sebanyak jumlah karakter yang terdapat pada P, tetapi jika karakter yang tidak cocok tersebut terdapat pada string P, maka pergeseran dilakukan berdasarkan tabel.
4. Jika karakter pada teks yang sedang dibandingkan cocok dengan karakter pada string P, maka posisi pengecekan karakter pada P dan S masing-masing bergeser ke kiri 1 posisi dari posisi sebelumnya, kemudian lanjutkan dengan pencocokan pada posisi tersebut dan seterusnya, kemudian jika terjadi ketidakcocokan karakter pada P dan S, maka pergeseran dilakukan dengan melihat tabel match heuristic dan occurrence heuristic dimana nilai pergeseran yang terbesar yang akan dipilih dikurangi dengan jumlah karakter yang telah cocok.
5. Jika semua karakter telah cocok, artinya P telah ditemukan didalam S, selanjutnya geser pattern sebesar 1 karakter, lanjutkan sampai akhir pattern S.

3.3 Google Maps API

Google Maps API (Application Programming Interface) merupakan layanan gratis dari google yang memungkinkan peta dunia berbentuk 2 dimensi maupun peta dunia 2 dimensi bercitra satelit dapat ditampilkan kedalam browser. Pada *Google Maps API* terdapat file *library* yang berbentuk *JavaScript*, penggunaan *Google MapsAPI* sangat memungkinkan untuk dilakukan penambahan fitur sesuai kebutuhan pengembang.

Gambar 7 merupakan pemanfaatan *Google Maps API* dalam menampilkan visualisasi peta rute perjalanan yang telah dipilih oleh pengguna. Pada Gambar 7 terdapat kondisi lalulintas dari lokasi perjalanan yang dihasilkan dari data *tweets* yang di *share* oleh pengguna *Twitter* pada saat itu.



Gambar 7. Visualisasi rute dan kondisi lalulintas

4. KESIMPULAN

1. Nama jalan atau lokasi yang terdapat didatabase, yang kemudian dijadikan *pattern* pencarian pada algoritma *Boyer Moore* mampu ditemukan pada data *tweets* yang telah didownload sehingga kondisi lalulintas dari nama jalan atau lokasi tersebut dapat teridentifikasi.
2. Data *tweets* yang tidak teridentifikasi kondisi lalulintasnya, umumnya disebabkan karena pada data tidak mengandung informasi kondisi lalulintas melainkan informasi lainnya seperti kecelakaan, kondisi jalan yang berlubang dan lain sebagainya.
3. Aplikasi yang telah dibuat belum mampu memberikan rute alternative perjalanan sehingga perlu dilakukan pengembangan lebih lanjut.

DAFTAR PUSTAKA

- Rodiyansyah, 2012, *Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification*, Tesis, Universitas Gadjah Mada, Yogyakarta.
- Cormen, Thomas. H., Leiseson, Charles. E., Rivest, Ronald. L., 1994, *Algorithm*. McGraw-Hill Book Company, North America.
- http://id.wikipedia.org/wiki/Algoritma_Boyer-Moore. 12 Agustus 2014.