

## **Laporan Tugas Besar III Strategi Algoritma IF2211**

### **Aplikasi *String Matching* untuk Disposisi *Tweets* ke Dinas-Dinas dan Instansi di Bawah Pemerintah Kota Bandung**



Anggota Kelompok :

1. Jeremia Jason Lasiman - 13514021
2. Bervianto Leo P - 13514047
3. M. Az-zahid Adhitya Silparensi – 13514095

**Program Studi Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

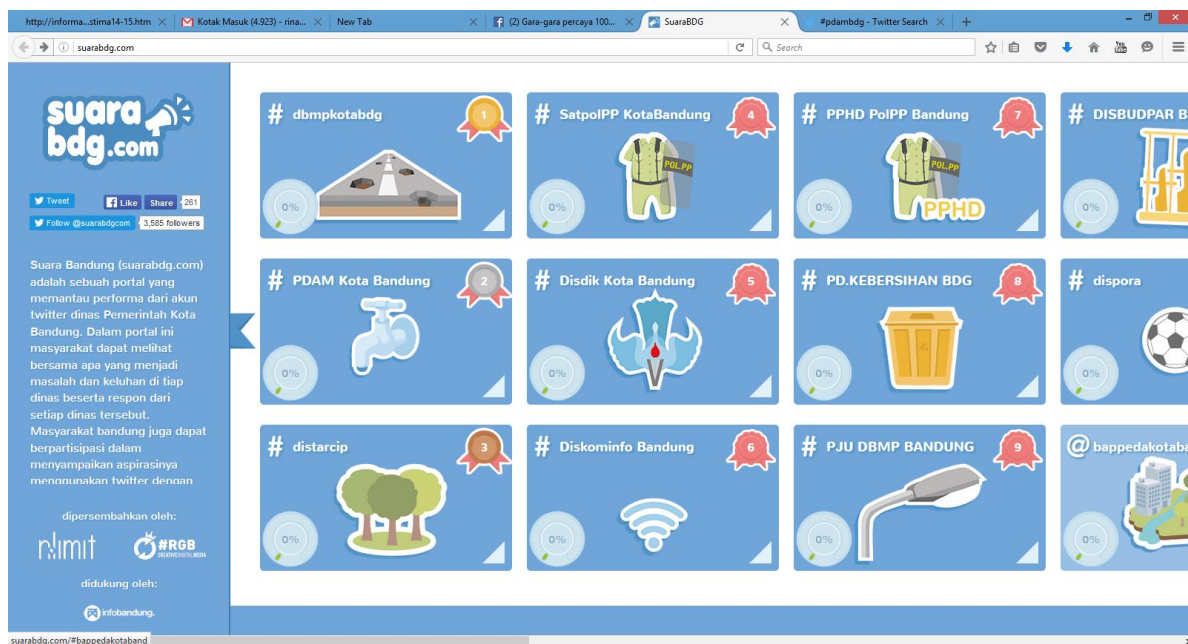
**Semester II Tahun 2015/2016**

# Daftar Isi

<b>Bab 1 Deskripsi Masalah .....</b>	<b>3</b>
<b>Bab 2 Dasar Teori .....</b>	<b>7</b>
<b>2.1 Algoritma Knuth-Morris-Pratt (KMP) .....</b>	<b>7</b>
<b>2.2 Algoritma Boyer-Moore .....</b>	<b>7</b>
<b>Bab 3 Analisis Pemecahan Masalah .....</b>	<b>8</b>
<b>Bab 4 Implementasi dan Pengujian .....</b>	<b>9</b>
<b>4.1 Spesifikasi teknis program .....</b>	<b>9</b>
<b>4.2 Eksperimen/pengujian .....</b>	<b>9</b>
4.2.1 Pengujian 1.....	9
4.2.2 Pengujian 2.....	12
4.2.3 Pengujian 3.....	14
<b>4.3 Analisis hasil pengujian .....</b>	<b>16</b>
<b>Bab 5 Kesimpulan dan Saran.....</b>	<b>17</b>
<b>5.1 Kesimpulan .....</b>	<b>17</b>
<b>5.2 Saran .....</b>	<b>17</b>
<b>Daftar Referensi .....</b>	<b>18</b>

## Bab 1 Deskripsi Masalah

*Twitter* (<https://twitter.com/>) merupakan media sosial yang banyak digunakan sebagai sumber informasi instansi yang berasal dari masyarakat. Analisis *twitter* sudah sering menjadi topik riset di Teknik Informatika ITB. Berbagai aplikasi analisis *twitter* yang telah dihasilkan sebagai Tugas Akhir/Tesis, seperti *Blackberry AntiMacTweet* (Hasby angkatan 2009), peringkasan untuk menjelaskan *trending topic* Indonesia (Yosef angkatan 2009), analisis sentimen review film (Novan angkatan 2009), disposisi keluhan (Fauzan angkatan 2010), prediksi Big Five personality (Agnes angkatan 2010), *buzzer matcher* (Afif angkatan 2010; Elisafina & Irfan S2 angkatan 2012), dst. Tidak hanya sebatas riset, beberapa alumni mengembangkan startup analisis tweets seperti *noLimit* (Aqsath angkatan 2005) dengan produk <http://suarabdg.com/>.



Gambar 1. Contoh aplikasi analisis tweets: suarabdg.com

Pemerintah Kota Bandung menggunakan *Twitter* untuk menyerap keluhan dari masyarakat yang berkaitan dengan masalah yang menjadi wewenang dan tanggung jawab dinas-dinas dan instansi di dalam Pemkot (Dinas Kebersihan, Dinas Pendidikan, PDAM, Dinas Sosial, Satpol PP, Damkar, dll). Untuk selengkapnya silakan cari informasi tentang dinas/lembaga di bawah Pemkot Bandung, atau lihat contoh di <http://suarabdg.com>.

Pada Tugas Besar III kali ini Anda diminta membuat aplikasi sederhana analisis *tweet* berbasis kata kunci. *Tweet* yang berhasil diunduh dari *Twitter* dianalisis untuk selanjutnya didisposisikan ke dinas atau instansi yang terkait. Salah satu fungsi dasar yang digunakan dalam sistem analisis teks tersebut adalah pencocokan string. Algoritma pencocokan *string* (*pattern*) yang digunakan adalah *Knuth-Morris-Pratt* (KMP), dan *Boyer-Moore* (BM).

Terdapat banyak dinas maupun instansi di bawah Pemkot Bandung. **Anda pilih lima dinas/lembaga** saja. Pengguna aplikasi analisis *tweet* sederhana ini akan memberikan dua jenis masukan yaitu: (1) *keyword* pencarian *tweet* yang berasal dari tagar (*hashhtag*) atau *mention* #pemkotbdg, @pemkotbdg,

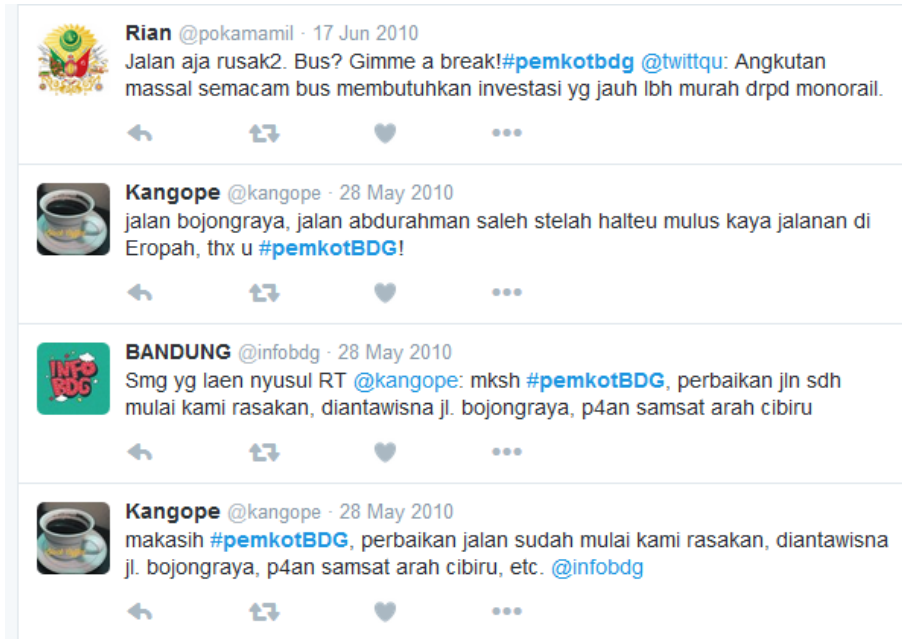
#pemkotbandung, @ridwan kamil, @infobdg, dan yang sejenis dengan itu, dan (2) *keyword*. Tabel berikut menunjukkan contoh masukan dari pengguna.

Keyword Pencarian Tweet	Keyword	Output ke Dinas/Lembaga
#pemkotbdg @pemkotbdg @ridwan kamil @infobdg	Pipa bocor; air ga ngalir; pdam mati; ga ngocor ...	PDAM Bandung
	PJU; Penerangan mati; lampu padam; terang; jalan gelap; ...	Dinas Bina Marga
	sampah; bau busuk; ...	Dinas kebersihan

Tabel 1 Contoh Keyword dan Output



Gambar 2. Contoh tweets @pemkotbdg



Gambar 3. Contoh tweets #pemkotbdg

Berdasarkan *keyword* pencarian *tweet* yang dimasukkan pengguna, aplikasi akan menggunakan *Twitter API* untuk mendapatkan minimal 100 *tweets* terbaru (*recent tweets*). Lalu, aplikasi menentukan kategori dinas/instansi dari setiap *tweet* dengan mengecek apakah *tweet* tersebut mengandung *keyword* yang diberikan untuk kategori dinas/lembaga tersebut.

- Jika *tweet* mengandung lebih dari satu *keyword* dari kategori dinas yang berbeda, *tweet* tersebut masuk dalam kategori dengan *keyword* yang muncul lebih dulu. Sebagai contoh, jika terdapat *keyword* “sampah” (kategori “dinas kebersihan”) dan “taman kotor” (kategori “dinas pertamanan”), *tweet* masuk kategori dinas kebersihan jika “sampah” muncul lebih dulu daripada kata “taman”.
- Jika tidak mengandung *keyword* apapun, *tweet* tersebut masuk dalam kategori dinas “unknown”.

Pencocokan *string* yang anda buat adalah *exact matching*, jadi *tweet* yang diproses mengandung *string* yang tepat sama dengan *keyword* yang telah ditentukan oleh pengguna. Di sini Algoritma KMP dan Boyer-Moore dapat digunakan. Untuk *keyword* berupa frase, maka perlu dilakukan penanganan secara khusus untuk menemukan *tweet* yang mengandung kata-kata di dalam frase. Pencarian juga tidak bersifat *case sensitive*, jadi huruf besar dan huruf kecil dianggap sama (hal ini dapat dilakukan dengan menganggap seluruh karakter di dalam *pattern* dan teks sebagai huruf kecil semua atau huruf kapital semua).

Jika *tweet* mengandung nama tempat (dimulai dengan kata depan “di”), maka sebagai **bonus**, Anda dapat menampilkan lokasi dari setiap kategori dengan *GoogleMaps API*.

Tweet	Lokasi Hasil Ekstraksi
#pemkotbdg 09.16 : lampu di jln tamansari jika malam padam	tamansari
@infobdg: 08.23 : Macet (lagi) di jln rancaekek - cicalengka maju dikit2 ga terlalu pic.twitter.com/gyK7PaCTVM   @Latifaa Aulia:	rancaekek - cicalengka

<b>#suaraBDG</b> via <b>@dionmudjenan</b> : Hatihati di jembatan perbatasan komp GBI td ada yg kena begal pelaku mabok bawa golok nyandra yg bawa motor	jembatan perbatasan komp GBI
<b>@ridwan kamil</b> : di terminal leuwipanjang banyak anak jalanan yg ngemis2 sementara ibunya diem di pos polisi sambil bbman X_X <b>#Suarabdg</b> <b>@dinsos_bdg</b>	terminal leuwipanjang

## Bab 2 Dasar Teori

### 2.1 Algoritma Knuth-Morris-Pratt (KMP)

Algoritma KMP pertama menghitung *border function* pada setiap katakter *pattern* yang ingin dicari. *Border function* digunakan untuk menentukan berapa banyak karakter yang perlu dilewati ketika ternyata katakter yang sekarang diproses tidak sama dengan *pattern* yang ingin dicari. *Border function* didefinisikan sebagai ukuran dari *prefix* terpanjang dari  $P[1..k]$  yang juga merupakan *suffix* dari  $P[1..k]$ .

Algoritma KMP ini mencari *pattern* dalam teks dari kiri ke kanan (seperti algoritma brute force). Ketika ternyata karakter yang diproses tidak sama dengan *pattern*, maka pencarian akan dilanjutkan dari indeks awal karakter yang sama ditambah dengan *border function* dari karakter yang terakhir sama. Pencarian berakhir ketika sudah menemukan kumpulan katakter yang cocok dengan *pattern* atau sudah mencapai karakter terakhir.

### 2.2 Algoritma Boyer-Moore

Algoritma Boyer-Moore ini berdasarkan pada dua teknik, yaitu

- *Looking-glass technique*

Pencarian *pattern*(P) dalam suatu teks (T) dengan menggunakan pencarian dari belakang

- *Character-jump technique*

- Ketika ketidakcocokan terjadi pada  $T[i] \neq x$ ,
- karakter dalam *pattern*  $P[i]$  tidak sama dengan  $T[i]$

Terdapat 3 kemungkinan yang dapat terjadi, yaitu :

- a. Jika P mengandung x pada sebelah kiri, maka geser P sehingga x dalam P sejajar dengan  $T[i]$
- b. Jika P mengandung x pada sebelah kanan dan tidak ada di kiri, maka geser P sebanyak  $T[i+1]$
- c. Jika P tidak mengandung x, geser P sehingga  $P[1] = T[i+1]$

Untuk optimasi, sebelum memulai algoritma ini dapat menyimpan indeks suatu karakter terdapat pada *pattern* menggunakan *Last Occurrence Function*.

## Bab 3 Analisis Pemecahan Masalah

Langkah Pemecahan Masalah :

- Ambil data dari halaman web berupa query untuk searching, query untuk klasifikasi, dan cara pencarian.
  - Halaman web menggunakan ASP.Net yang terdiri dari label untuk penunjuk, text box untuk memasukan query, dan button untuk memilih antara algoritma KMP atau algoritma Boyer-Moore.
- Tweet disearch menggunakan API Tweetinvi.
  - Pencarian tweet menggunakan API maka programmer hanya menerima hasil dari API. Tweet yang didapat 100 buah berdasarkan kata kunci pada halaman web.
- Hasil dari API tersebut dimasukkan ke kategori dinas-dinas sesuai dengan query untuk dikategorikan dengan algoritma KMP atau Boyer-Moore.
  - Hasil tersebut dimasukan kedalam sebuah struktur data yang bernama TweetResult yang berisi list QueryCategory. QueryCategory menyimpan data dari tweet sesuai dengan kategorinya.
  - Kategori yang ada adalah Dinas Binawarga, Dinas Kesehatan, Dinas Sosial, Dinas Pendidikan, Dinas Pemuda dan Olahraga, dan No Category.
- Kemudian menampilkan tweet sesuai dengan kategori dinas yang dimasukan.



## Bab 4 Implementasi dan Pengujian

### 4.1 Spesifikasi teknis program

- Controller
  - TweetiFormController.cs  
Mendapatkan nilai query dari view. Mengambil data tweet. Memproses data tweet sekaligus membaginya sesuai kategori.
  - ResultController.cs  
Mendapatkan hasil pengkategorian dan mengirimkannya ke ShowResult.cshtml.
  - StringMatching.cs  
Algoritma pencocokan string.
- View
  - ShowResult.cshtml  
ShowResult.cshtml menangani bagian untuk mengeluarkan hasil dari pencarian.
  - SearchForm.cshtml  
SearchForm.cshtml menangani bagian default page, dan mengoper nilai query ke controller.
- Model
  - HasilTweet.cs  
Kelas untuk menyimpan hasil dari proses tweet.
  - QueryCategory.cs  
Kelas untuk menyimpan 1 kategori dari query beserta hasil proses dari seluruh tweet yang masuk pada kategori tersebut.
  - Tags.cs  
Kelas untuk menyimpan query yang dimasukan user.
  - TweetResult.cs  
Kelas untuk menyimpan seluruh QueryCategory didalam

### 4.2 Eksperimen/pengujian

#### 4.2.1 Pengujian 1

- Halaman awal :

SearchForm

Name

Query Dinas Kesehatan :

Query Dinas Sosial :

Query Dinas Pendidikan :

Query Dinas Binamarga & Pengairan :

Query Dinas Pemuda & Olahraga :

☒ KMP  
☐ Boyer-Moore

© 2016 - Tweety

- Halaman hasil :

Result:

Dinas Kesehatan


Jumlah Tweet : 0;


Dinas Binamarga

Jumlah Tweet : 0;

Dinas Pemuda

Jumlah Tweet : 2;

 Mangga warga Bandung, ngapain di rumah? Ke booth @bobotohfm di @tmpyouthfest\_ ada kang @ridwankamil #bandungbanget <https://t.co/EZA5hy12aN>  
24/04/2016 09:43:40

 @bobotohswiss @mfathur12 @ridwankamil @vikingbanten\_ ngelesnya pake ngedit kaya anak smp. Jadi mana tiket yg disebut dibagiin gratis kang?  
24/04/2016 23:55:33

Dinas Pendidikan

ShowResult - Tweety


localhost:64539/Result/ShowResult

Tweety About

Jumlah Tweet : 0;


### Dinas Pemuda

Jumlah Tweet : 2;



Mangga warga Bandung, ngapain di rumah? Ke booth @bobotohfm di @tmpyouthfest\_ ada kang @ridwankamil #bandungbanget <https://t.co/EZA5hy12aN>

24/04/2016 09:43:40



@bobotohswiss @mfathur12 @ridwankamil @vikingbanten\_ ngelesnya pake ngedit kaya anak smp. Jadi mana tiket yg disebut dibagiin gratis kang?


24/04/2016 23:55:33

### Dinas Pendidikan

Jumlah Tweet : 0;

### Dinas Sosial

Jumlah Tweet : 30;



RT @ridwankamil: Selamat #HariahNU ke 93. Semoga selalu menjadi benteng Islam yang kokoh dan toleran di Nusantara.

25/04/2016 01:11:45


ShowResult - Tweety

localhost:64539/Result/ShowResult

Tweety About


### Dinas Sosial

Jumlah Tweet : 30;




RT @ridwankamil: Selamat #HariahNU ke 93. Semoga selalu menjadi benteng Islam yang kokoh dan toleran di Nusantara.

25/04/2016 01:11:45




RT @ridwankamil: Selamat #Milad18PKS . Semoga selalu terdepan berkhidmat untuk rakyat. hatur nuhun.

25/04/2016 00:58:27




RT @ridwankamil: Selamat #Milad18PKS . Semoga selalu terdepan berkhidmat untuk rakyat. hatur nuhun.

25/04/2016 00:49:12




RT @ridwankamil: perhatian untuk para jejak tukang tikung. <https://t.co/FFiz8vnytW>

25/04/2016 00:40:46



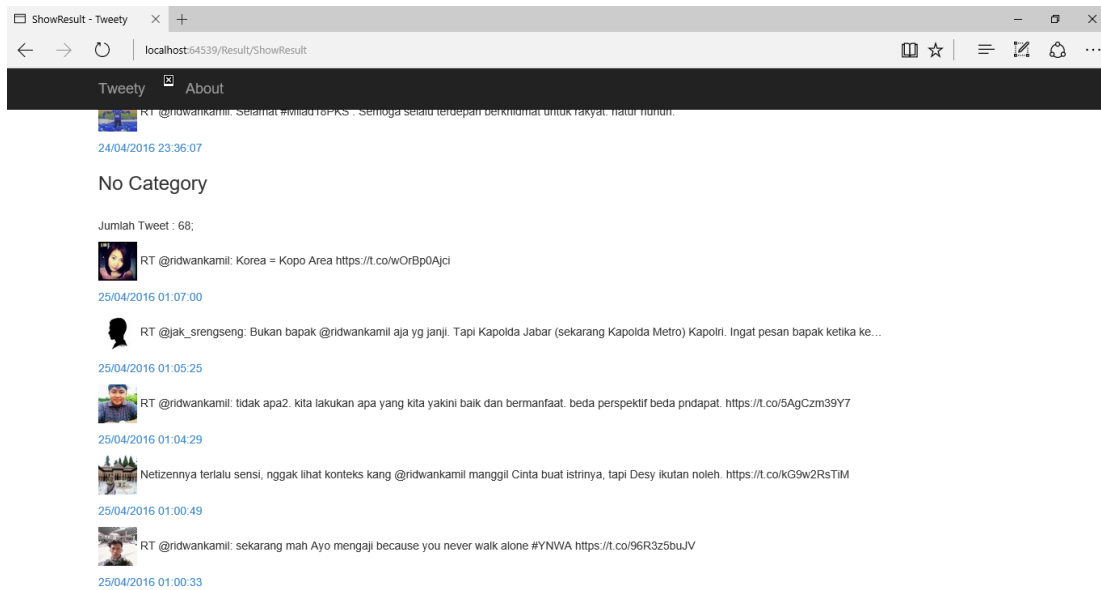
RT @ridwankamil: Selamat #Milad18PKS . Semoga selalu terdepan berkhidmat untuk rakyat. hatur nuhun.

25/04/2016 00:40:44



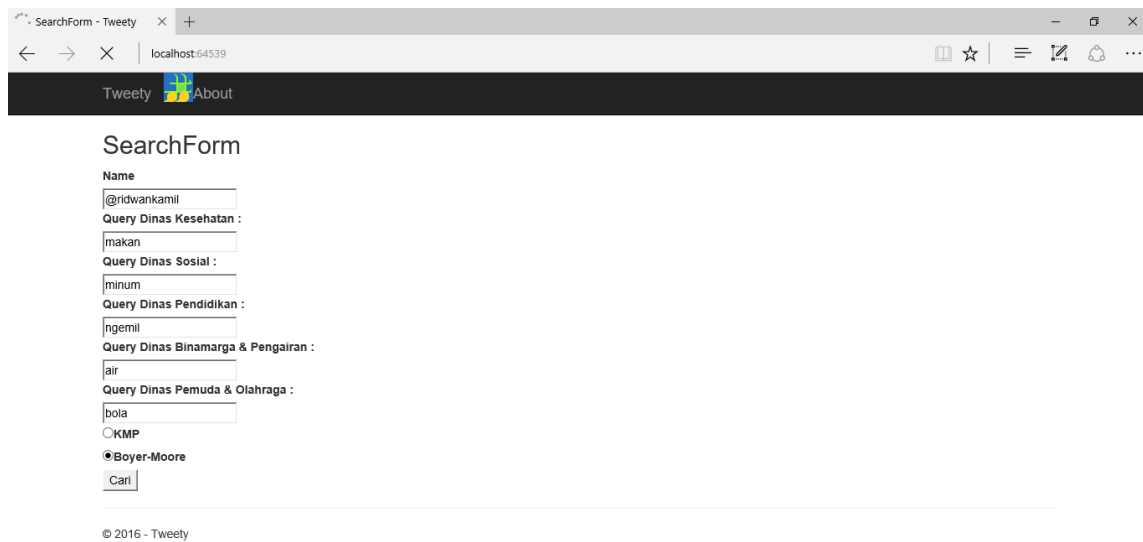
RT @ridwankamil: Selamat #Milad18PKS . Semoga selalu terdepan berkhidmat untuk rakyat. hatur nuhun.

25/04/2016 00:30:45



## 4.2.2 Pengujian 2

- Halaman awal :



- Halaman hasil :



## Result:

### Dinas Kesehatan

Jumlah Tweet : 0;

### Dinas Binamarga

Jumlah Tweet : 0;

### Dinas Pemuda

Jumlah Tweet : 0;

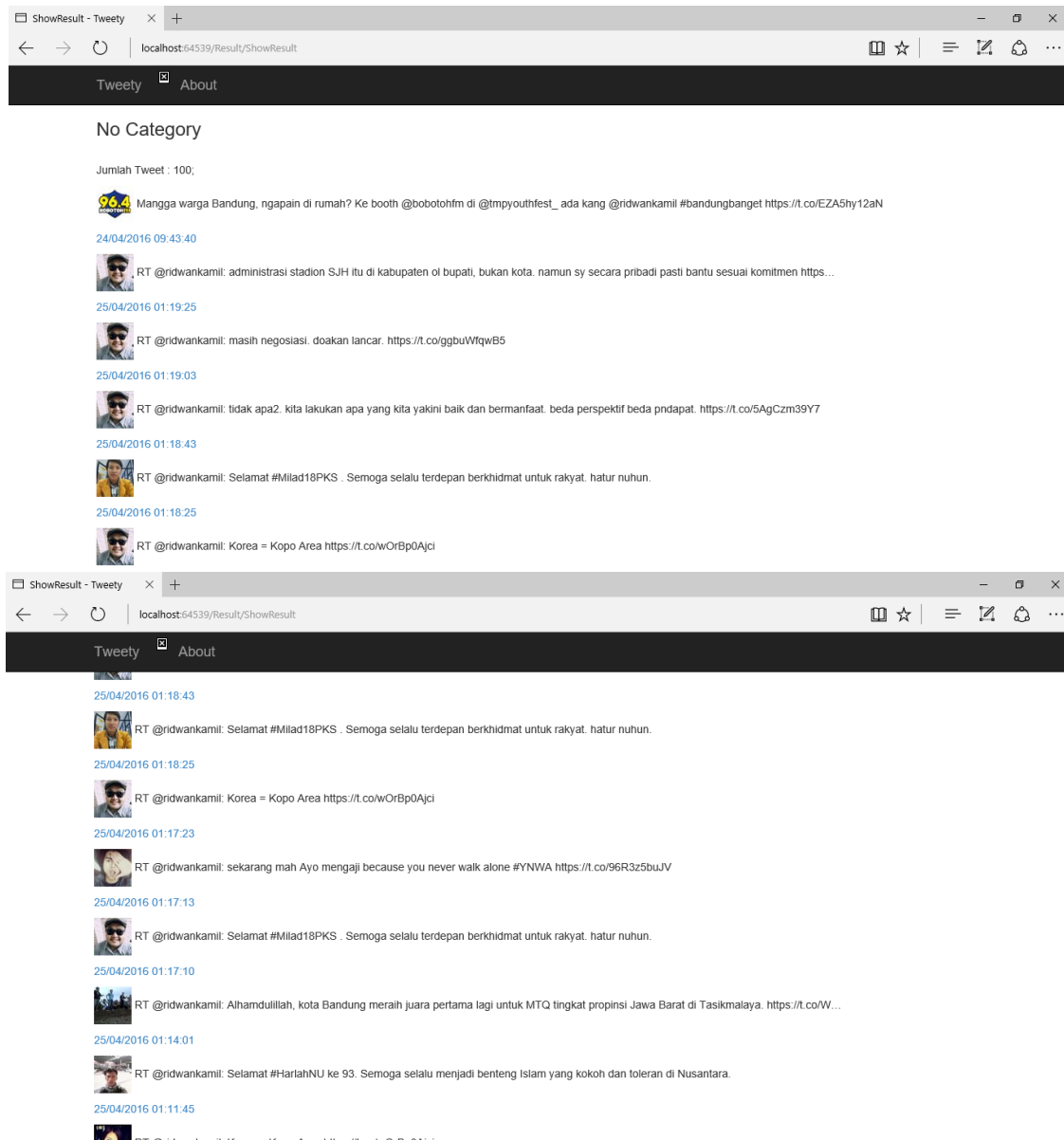
### Dinas Pendidikan

Jumlah Tweet : 0;

### Dinas Sosial

Jumlah Tweet : 0;

### No Category



#### 4.2.3 Pengujian 3

- Halaman awal :

SearchForm - Tweety x +

localhost:64539

Tweety About

## SearchForm

Name

@ridwankamili

Query Dinas Kesehatan :

sehat

Query Dinas Sosial :

hatur nuhun

Query Dinas Pendidikan :

prestasi

Query Dinas Binamarga & Pengairan :

pengairan

Query Dinas Pemuda & Olahraga :

balap

☐ KMP

☒ Boyer-Moore

Cari

© 2016 - Tweety

- Halaman hasil :

ShowResult - Tweety x +

localhost:64539/Result/ShowResult

Tweety About

## Result:

Dinas Kesehatan

Jumlah Tweet : 0;

Dinas Binamarga

Jumlah Tweet : 0;

Dinas Pemuda


Jumlah Tweet : 0;

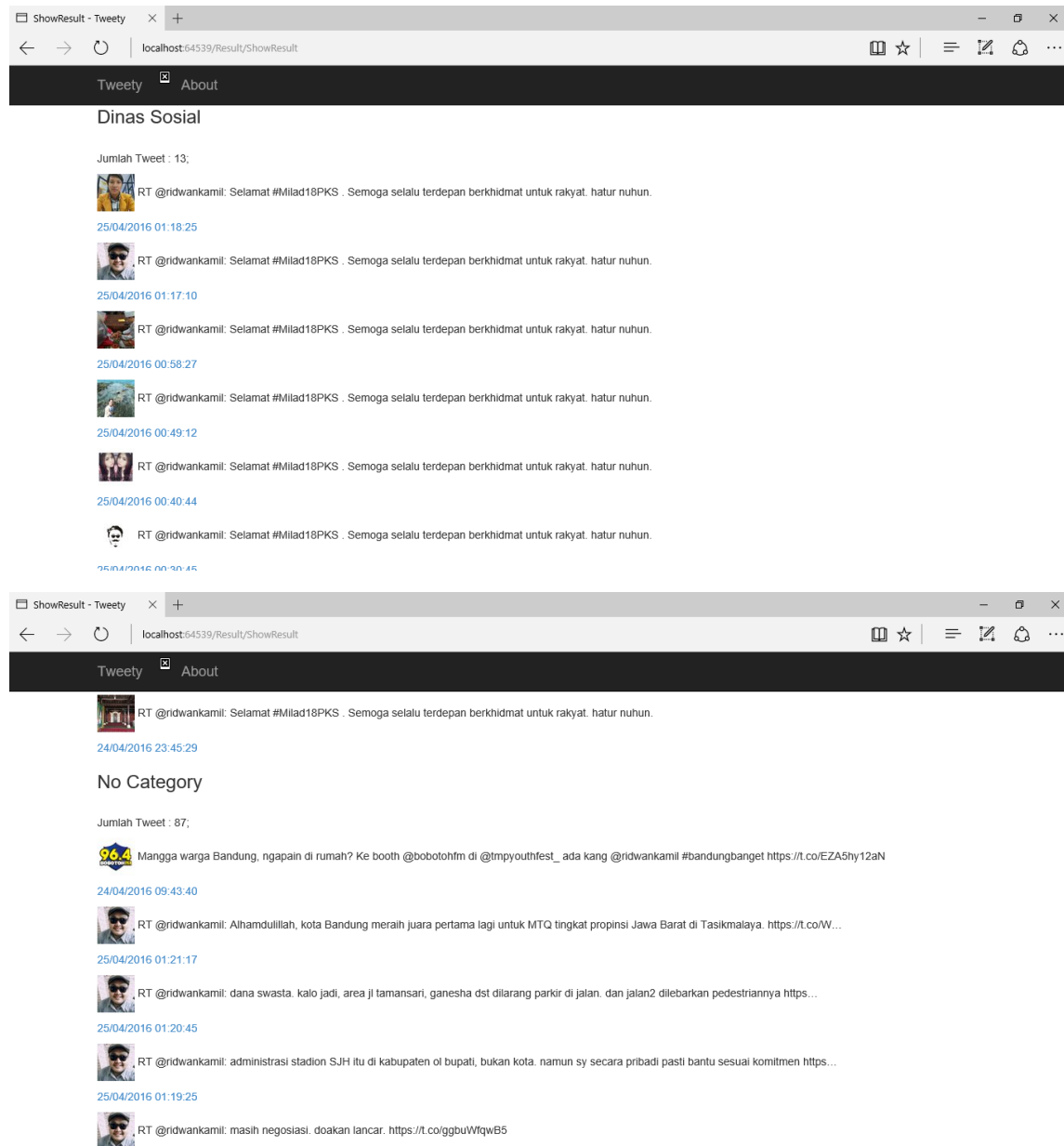
Dinas Pendidikan

Jumlah Tweet : 0;

Dinas Sosial

Jumlah Tweet : 13;

 RT @ridwankamili: Selamat #Milad18PKS. Semoga selalu terdapat herkhidmat untuk rakyat hatur nuhun



### 4.3 Analisis hasil pengujian

Dari ketiga percobaan tersebut metode yang digunakan tidak begitu mempengaruhi hasil yang didapatkan. Dalam persoalan waktu yang digunakan tidak berbeda. Dikarenakan metode Boyer-Moore harus mendefinisikan huruf-huruf yang akan diuji sehingga huruf (elemen dari *string*) diabaikan saja. Metode KMP cukup cocok dikarenakan tidak memiliki masalah terhadap karakter-karakter asing pada html atau sejenisnya. Sedikit sulit mencari *query* yang cocok untuk setiap dinasnya dikarenakan perlu analisis lebih lanjut mengenai gaya (*style*) pengguna twitter untuk mengungkapkan setiap keluhannya.



## Bab 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari tugas besar kali ini kami bisa menyimpulkan beberapa hal sebagai berikut. Pertama Algoritma Boyer-Moore kurang efektif dalam hal space pada program, karena pada saat pembandingan karakter jika string mempunyai sebuah character yang melebihi total array integer untuk menyimpan nilai karakter terakhir pada keyword maka akan terjadi `arrayOutOfBounds` pada array tersebut.

Kedua Algoritma KMP (Knuth-Morris-Pratt) algoritma ini efektif untuk kasus kali ini karena karakter pada tweet tidak mempengaruhi algoritma ini yang membuat hampir tidak mungkin untuk `ArrayOutOfBounds`.

### 5.2 Saran

Pada tugas besar kali ini kelompok kami menyarankan jika menggunakan Algoritma boyer-moore harus berhati-hati dengan `ArrayOutOfBounds`. Salah satu cara untuk mengatasinya yaitu jika lebih dari indeks array maka langsung mengeluarkan nilai `-1` untuk pergerakan elemennya.

Kemudian sebaiknya menggunakan tools yang saling compatible, seperti C# dan ASP.Net. Karena jika tidak compatible seperti PHP dan C# maka untuk menggabungkan kedua kode tersebut akan menjadi sulit.

## Daftar Referensi

- Slide Kuliah *String Matching* : [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Pencocokan%20String%20\(2015\).ppt](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Pencocokan%20String%20(2015).ppt)