

IF2211 STRATEGI ALGORITMA

TUGAS BESAR 3 - DETEKSI SPAM PADA MEDIA
SOSIAL ATAU *CHAT-MESSENGER* DENGAN
ALGORITMA PENCOCOKAN STRING



Oleh :

Letivany Aldina	13514067
Muhammad Sulthan Adhipradhana	13516035
Ivan Fadillah	13516128

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2017/2018

A. DESKRIPSI MASALAH

Electronic-spam merupakan pesan elektronik yang tidak diinginkan penerimanya, bisa dalam bentuk surat elektronik, SMS, posting atau komentar di media sosial yang muncul di timeline kita, ataupun pesan pada *chatmessenger*. *Spammer* melakukan *spamming* untuk tujuan tertentu, paling banyak untuk menyebarkan iklan. Penentuan spam sangat bersifat subjektif, artinya spam untuk kita, belum tentu spam untuk pengguna lain. Gambar 1-2 merupakan contoh spam pada twitter, facebook, dan line.

Algoritma pencocokan string (*pattern*) Knuth-Morris-Pratt (KMP) dan Algoritma Boyer-Moore merupakan algoritma yang lebih baik daripada *brute force*. Pada Tugas Besar III kali ini Anda diminta membuat aplikasi sederhana deteksi spam pada media sosial dengan kedua algoritma tersebut, plus menggunakan regular expression (regex). Teks yang akan Anda proses adalah posting berbahasa Indonesia. Pengguna aplikasi ini akan memberikan masukan berupa keyword spam, dan menandai daftar posting yang dikategorikan spam menurut berdasarkan tanggal.

Pencocokan string yang anda buat adalah *exact matching* (untuk KMP dan BM) jadi posting yang diproses mengandung string yang tepat sama dengan keyword spam dari pengguna. Sedangkan bila menggunakan regex maka tidak selalu *exact matching*. 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).

Kumpulan posting diambil secara otomatis menggunakan Facebook API atau Twitter API (<https://developer.twitter.com/en/docs/tweets/search/overview>) atau Line API (<https://developers.line.me/en/services/messaging-api/>) atau api dari media sosial lainnya.

B. DASAR TEORI

Pencarian string atau pencocokan string merupakan algoritma pencarian kemunculan string dengan ukuran tertentu (*pattern*) di dalam string dengan ukuran yang lebih panjang (*text*). Persoalan pencarian string dirumuskan sebagai berikut:

- a. Teks (*text*), yaitu (*long*) string yang panjangnya n karakter.
- b. *Pattern*, yaitu string dengan panjang m karakter ($m < n$) yang akan dicari di dalam teks.

Dalam tugas besar 3 ini, digunakan beberapa metode untuk pencarian string tersebut. Terdapat tiga metode yaitu sebagai berikut:

1. Algoritma Knuth-Morris-Pratt (KMP)

Pada algoritma ini, kita menyimpan informasi berupa posisi yang digunakan untuk melakukan jumlah pergeseran yang lebih jauh. Dengan algoritma ini, waktu pencarian dapat dikurangi secara signifikan. Algoritma KMP ini melakukan proses awal terhadap *pattern* dengan menggunakan *border function*, yaitu fungsi yang menghitung pergeseran s terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian *string*. Secara umum, langkah-langkah yang dilakukan dalam algoritma KMP ini adalah sebagai berikut:

- 1) Memproses *border function*.
- 2) Mulai melakukan pencocokan pertama *pattern* di awal teks.
- 3) Dari kiri ke kanan, algoritma melakukan pencocokan karakter per karakter di *pattern* dengan karakter di *text* yang bersesuaian sampai salah satu kondisi berikut terpenuhi:
 - a. Karakter di *pattern* dengan di teks tidak cocok.
 - b. Semua karakter di *pattern* cocok dengan yang di teks, kemudian algoritma akan memberitahu penemuan posisi ini.
- 4) Menggeser *pattern* berdasarkan karakteristik algoritma KMP ini, mengulangi langkah 3 sampai berada di ujung teks.

2. Algoritma Boyer-Moore (BM)

Terdapat dua teknik dalam algoritma BM ini, yaitu

- 1) Teknik *Looking-glass*, yaitu melakukan perbandingan dari bagian belakang *pattern*.
- 2) Teknik *Character-jump*.

Algoritma BM ini merupakan variasi lain dari pencocokan string yang memulai perbandingan *pattern* dari bagian kanan (belakang). Algoritma BM juga melakukan proses awal menggunakan *Last Occurrence Function*. Secara umum, langkah-langkah yang dilakukan dalam algoritma BM ini adalah sebagai berikut:

- 1) Algoritma memulai pencocokan *pattern* pada bagian awal teks.
- 2) Dari kanan ke kiri, algoritma BM akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian sampai salah satu dari kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dengan yang di teks tidak cocok.
 - b. Semua karakter *pattern* dengan yang di teks cocok, kemudian algoritma akan memberitahu penemuan posisi ini.
- 3) Algoritma kemudian menggeser *pattern* dengan memaksimalkan nilai penggeseran *good-suffix* dan penggeseran *bad-character* , kemudian mengulangi langkah 2 sampai berada di ujung teks.

3. *Regular Expression*

Regular expression merupakan salah satu teknik pencocokan string yang memanfaatkan teori *formal language* dan automata. Prinsip teknik ini adalah menyimpan informasi posisi karakter yang sama dilanjutkan dengan karakter selanjutnya. Apabila valid, maka pencocokan karakter akan diulang, namun apabila tidak maka pencocokan string akan dimulai lagi dari karakter awal *pattern*.

C. ANALISIS PEMECAHAN MASALAH

Langkah-langkah pemecahan masalah yang dilakukan dalam tugas besar ini adalah sebagai berikut:

1. Algoritma akan menerima masukan pilihan jenis algoritma yang akan memproses, username, dan keyword pencarian.
2. Melalui keyword tersebut, algoritma akan mendapatkan daftar posting sekaligus username yang mengandung keyword pencarian yang didapatkan dari web server melalui JSON dan API yang dipilih dalam tugas ini, yaitu Twitter API.
3. Algoritma juga akan mengecek daftar posting yang *case-sensitive* maupun yang tidak *case-sensitive*.
4. Selanjutnya, melalui pilihan jenis algoritma yang akan memproses keyword pencarian tersebut, pencocokan keyword akan diproses menggunakan algoritma KMP, BM atau menggunakan *regular expression*.
5. Algoritma kemudian akan menampilkan daftar postingan. Apabila terdapat postingan yang mengandung keyword pencarian, maka algoritma akan menandainya sebagai spam.

D. IMPLEMENTASI & PENGUJIAN

1. Spesifikasi teknis program

Pada tugas besar ini, kami menggunakan program berbasis web menggunakan kakas PHP, dengan implementasi teknik pencocokan string menggunakan bahasa Python. File-file yang telah dibuat adalah sebagai berikut:

a. Pada file python (test.py) terdapat beberapa fungsi yang didefinisikan yaitu sebagai berikut:

- 1) `get_timeline(__name__)`, yaitu fungsi untuk mendapatkan sejumlah postingan Twitter melalui Twitter API.
- 2) `spam_filters_insensitive(timeline, keyword, algorithm)`, yaitu fungsi untuk memproses karakter keyword pencarian yang tidak bersifat *case sensitive* untuk masing-masing algoritma.
- 3) `spam_filters_sensitive(timeline, keyword, algorithm)`, yaitu fungsi untuk memproses karakter keyword pencarian yang bersifat *case sensitive* untuk masing-masing algoritma.
- 4) `bm_match(text, pattern)`, yaitu fungsi yang mencocokkan keyword pencarian dengan daftar postingan menggunakan algoritma Boyer-Moore.
- 5) `initialized(text, size)`
- 6) `buildLast(last, pattern, size)`
- 7) `kmp_match(text, pattern)`, yaitu fungsi untuk mencocokkan keyword pencarian dengan daftar postingan menggunakan algoritma Knuth-Morris-Pratt.
- 8) `compute_fail(pattern, M, lps)`
- 9) `main()`

b. Pada file php (home.php) terdapat beberapa fungsi yang didefinisikan yaitu sebagai berikut:

- 1) `highlightTweet($data, $keyword)`, yaitu fungsi yang mencocokkan keyword pencarian dengan data dari web server melalui Twitter API berdasarkan indeks dan panjang keyword pencarian.

2) `createTweet($data, $keyword)`, yaitu fungsi yang menandai sejumlah postingan yang mengandung keyword pencarian dan ditandai sebagai postingan yang bersifat spam.

3) `get_data($url, $data)`

c. `About.php`

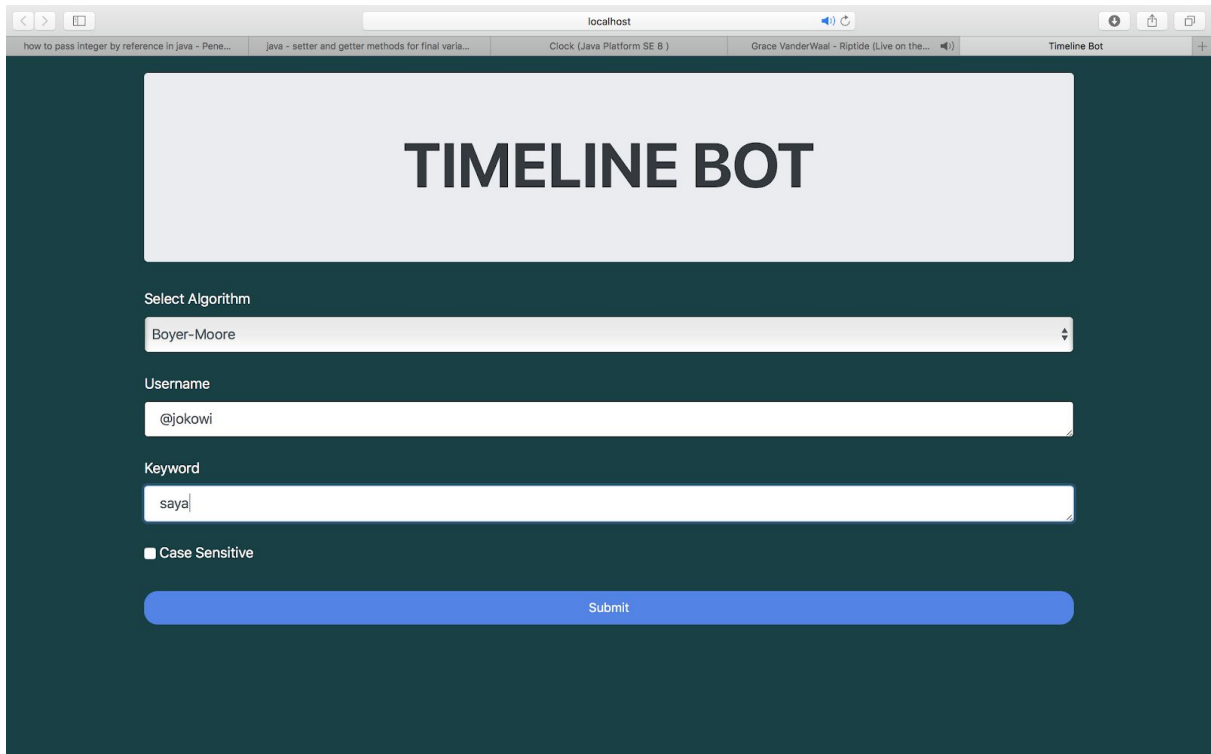
d. `Index.php`

e. `style.css`

2. Pengujian

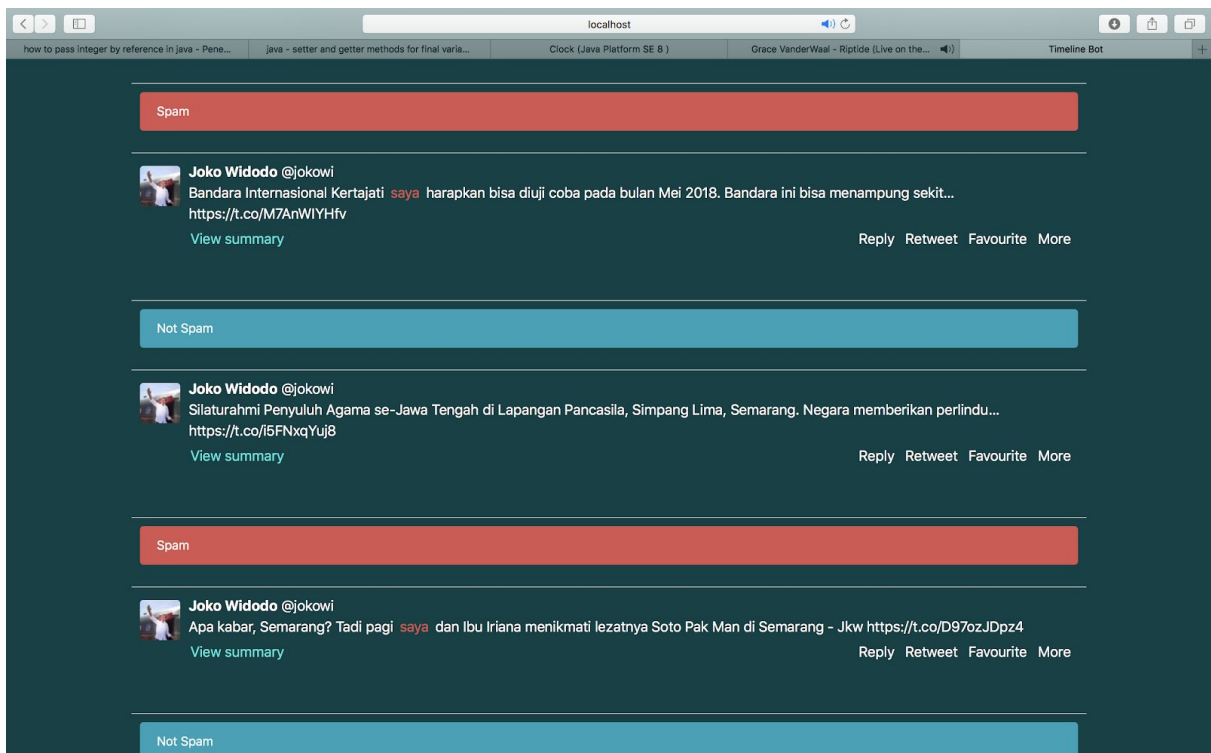
Pada contoh pengujian berikut, kami menggunakan keyword pencarian *saya* pada sejumlah postingan Twitter dengan username *@jokowi*.

Pengujian algoritma Boyer-Moore :



The screenshot shows a web browser window with the URL 'localhost'. The page has a dark teal background. At the top, there is a light gray box with the text 'TIMELINE BOT' in large, bold, black letters. Below this, there is a form with the following elements:

- Select Algorithm:** A dropdown menu with 'Boyer-Moore' selected.
- Username:** A text input field containing '@jokowi'.
- Keyword:** A text input field containing 'saya'.
- Case Sensitive:** A checkbox that is currently unchecked.
- Submit:** A large blue button.



The screenshot shows the results of the search on the Timeline Bot application. The page displays a list of tweets from the user 'Joko Widodo @jokowi'. Each tweet is followed by a filter bar with two options: 'Spam' (red) and 'Not Spam' (teal).

- Tweet 1:** "Bandara Internasional Kertajati **saya** harapkan bisa diuji coba pada bulan Mei 2018. Bandara ini bisa menampung sekit...
<https://t.co/M7AnWIYHfv>
View summary Reply Retweet Favourite More
- Tweet 2:** "Silaturahmi Penyuluh Agama se-Jawa Tengah di Lapangan Pancasila, Simpang Lima, Semarang. Negara memberikan perlindu...
<https://t.co/i5FNxqYuj8>
View summary Reply Retweet Favourite More
- Tweet 3:** "Apa kabar, Semarang? Tadi pagi **saya** dan Ibu Iriana menikmati lezatnya Soto Pak Man di Semarang - Jkw <https://t.co/D97ozJDpz4>
View summary Reply Retweet Favourite More

Pengujian algoritma Knuth-Morris-Pratt :

Timeline Bot

Timeline Bot

Select Algorithm

KMP

Username

@jokowi


Keyword

Saya

☒ Case Sensitive

Submit


Not Spam

 **Joko Widodo** @jokowi

Selamat Hari Kartini 2018. Perempuan hebat untuk Indonesia. Mari = Ikut

Timeline Bot


Not Spam

 **Joko Widodo** @jokowi

Bandara Internasional Kertajati saya harapkan bisa diuji coba pada bulan Mei 2018. Bandara ini bisa menampung sekit...
<https://t.co/M7AnWYHfv>

[View summary](#) [Reply](#) [Retweet](#) [Favourite](#) [More](#)


Not Spam

 **Joko Widodo** @jokowi

Silaturahmi Penyuluh Agama se-Jawa Tengah di Lapangan Pancasila, Simpang Lima, Semarang. Negara memberikan perindu...
<https://t.co/l5FNxqYuj8>

[View summary](#) [Reply](#) [Retweet](#) [Favourite](#) [More](#)

Not Spam

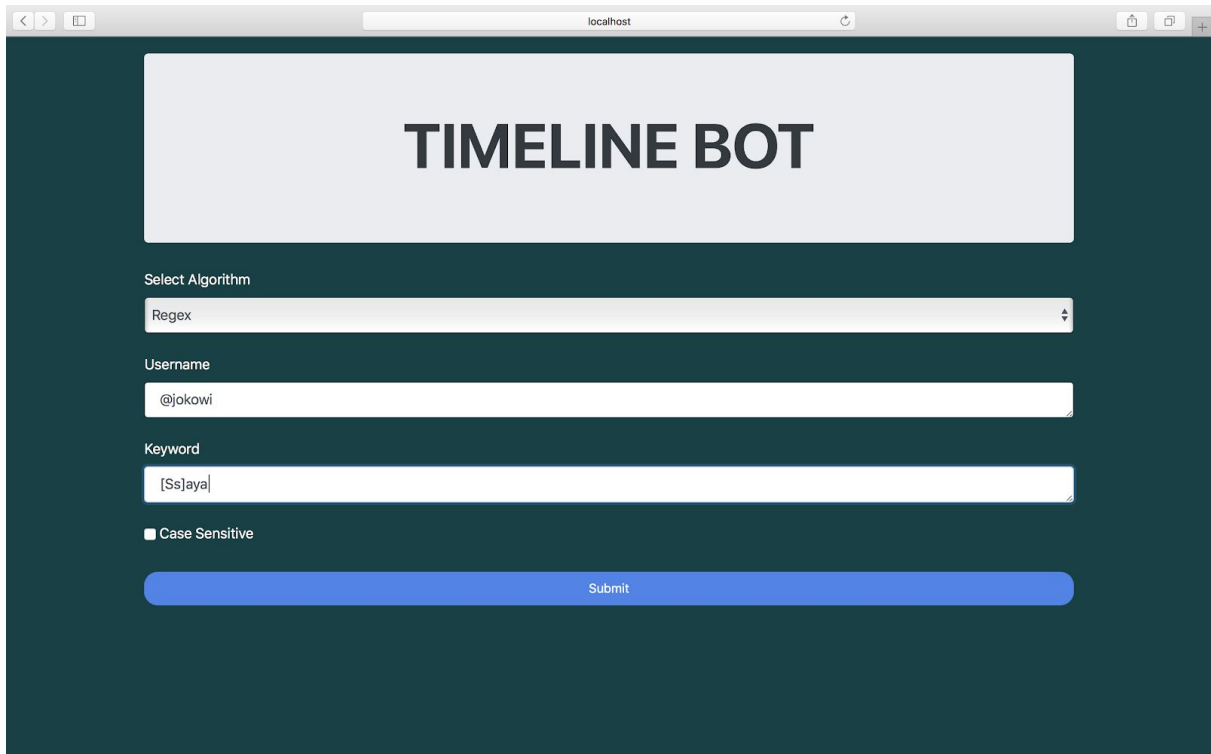
 **Joko Widodo** @jokowi

Apa kabar, Semarang? Tadi pagi saya dan Ibu Iriana menikmati lezatnya Soto Pak Man di Semarang - Jkw <https://t.co/D97ozJDpz4>

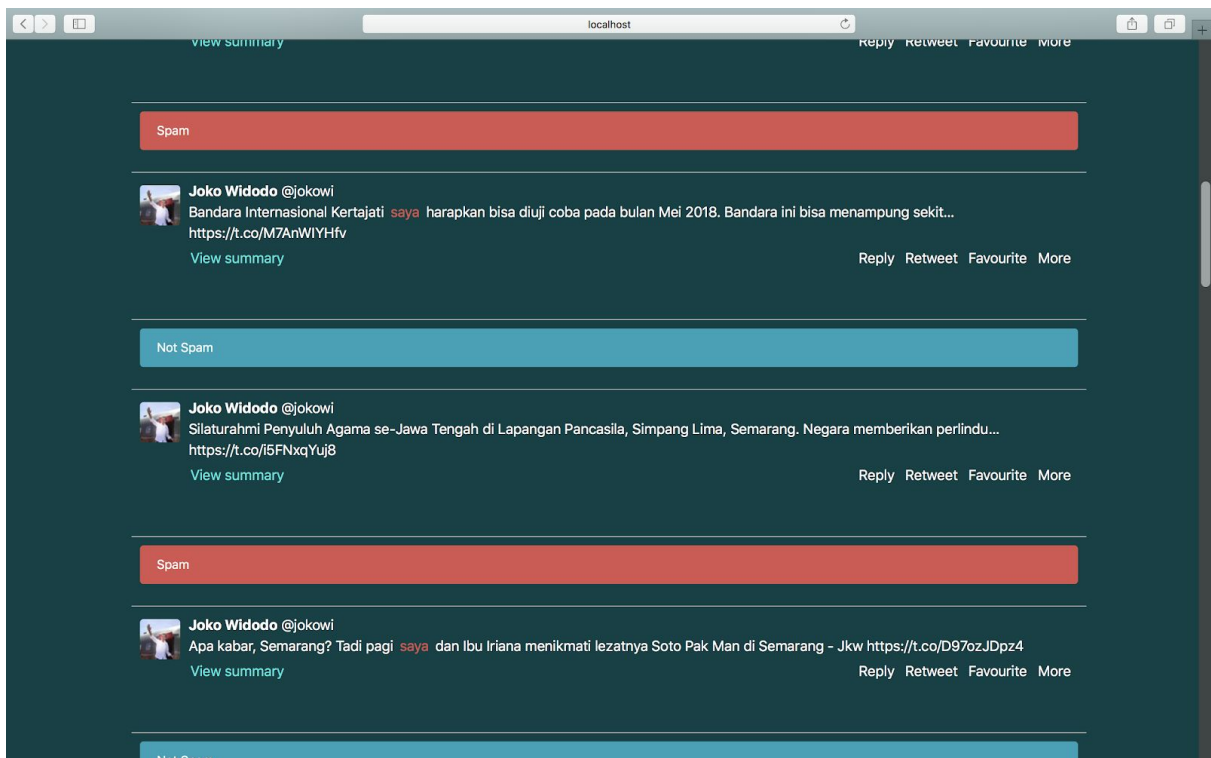
[View summary](#) [Reply](#) [Retweet](#) [Favourite](#) [More](#)

Not Spam

Pengujian menggunakan *regular expression* :



The screenshot shows a web browser window with the address bar set to 'localhost'. The page has a dark teal background. At the top, a light gray box contains the text 'TIMELINE BOT' in large, bold, black capital letters. Below this, there is a form with the following elements: a 'Select Algorithm' dropdown menu currently showing 'Regex'; a 'Username' input field containing '@jokowi'; a 'Keyword' input field containing '[Ss]aya'; a checkbox labeled 'Case Sensitive' which is currently unchecked; and a large blue 'Submit' button at the bottom.



The screenshot shows the results of the search on the 'TIMELINE BOT' application. The browser window has the address bar at 'localhost'. The page has a dark teal background. At the top, there are links for 'view summary' and 'Reply Retweet Favourite More'. Below this, there is a list of tweets. Each tweet is preceded by a colored bar indicating its status: a red bar for 'Spam' and a blue bar for 'Not Spam'. The tweets are from 'Joko Widodo @jokowi' and contain text about 'Bandara Internasional Kertajati' and 'Simpang Lima, Semarang'. Each tweet also includes a 'View summary' link and 'Reply Retweet Favourite More' options.

3. Analisis hasil pengujian

Dari hasil pengujian tersebut, implementasi algoritma BM dan KMP sudah sesuai dengan apa yang diekspektasikan pada awalnya. Algoritma dapat menemukan sejumlah posting bersifat spam dengan keyword pencarian *saya*. Pada implementasi *regular expression*, keyword pencarian yang dimasukkan adalah [Ss]aya. Pada tugas kami ini juga terdapat pilihan untuk melakukan pencarian yang bersifat *case sensitive* maupun yang tidak. Untuk ketiga jenis implementasi, semuanya menampilkan daftar posting, baik yang bersifat spam maupun yang non-spam. Pada postingan yang mengandung spam, keyword pencarian akan di-*highlight* dengan warna merah menandakan di dalamnya terdapat keyword pencarian, dan akan terdapat header dengan tulisan Spam yang mempunyai warna *background* merah pula. Pada postingan yang tidak mengandung spam, header postingan akan ditandai dengan warna *background* biru.

E. KESIMPULAN & SARAN

1. Kesimpulan

Dari tugas besar yang telah kami buat dan kembangkan, baik algoritma Boyer-Moore, algoritma Knuth-Morris-Pratt, maupun *regular expression* memberikan hasil yang sama terhadap pencocokan string pada tugas ini, yakni deteksi postingan yang bersifat spam. Masing-masing algoritma mempunyai cara kerja yang berbeda, namun sama-sama bersifat efektif dalam melakukan pencocokan string.

2. Saran

Dari tugas yang telah kami pelajari dan kembangkan ini, masih ada terdapat kekurangan. Sehingga diharapkan ke depannya dapat dilakukan pembelajaran dan pengembangan lebih lanjut agar hasil yang didapatkan juga lebih baik.