

**TUGAS BESAR III**  
**IF2211 STRATEGI ALGORITMA**  
**DETEKSI SPAM PADA MEDIA SOSIAL ATAU CHAT-MESSENGER**  
**DENGAN ALGORITMA PENCOCOKAN STRING**

**LAPORAN**

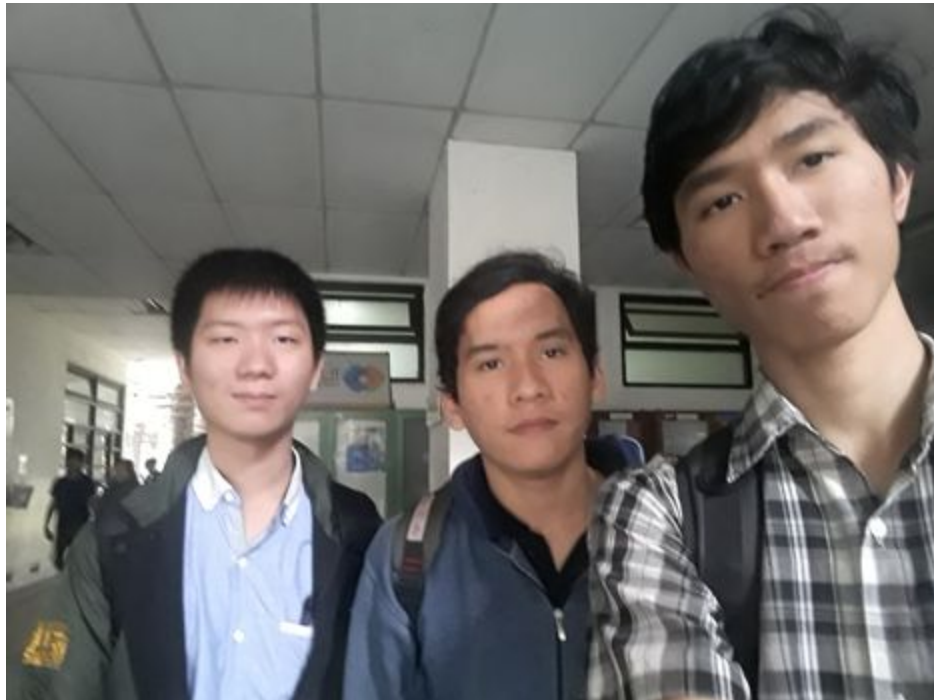
Diajukan untuk memenuhi tugas mata kuliah IF2211 Strategi Algoritma

oleh:

**Gloryanson Ginting 13516060**

**Louis Leslie 13516087**

**Daniel Ryan Levison 13516132**



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**BANDUNG**  
**2018**

## **BAB I**

### **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 chat-messenger. 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.

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.

## BAB II DASAR TEORI

### 2.1. *Spam*

*Spam* adalah usaha memenuhi Internet dengan pesan yang sama dengan tujuan memaksakan pengiriman pesan kepada orang yang tidak ingin dengan kesadarannya memilih untuk menerima pesan tersebut. Kebanyakan *spam* adalah usaha menyiarkan iklan pemasaran, biasanya untuk produk yang diragukan, *scheme* agar cepat kaya, atau jasa palsu. Konsekuensi dari usaha *spam* ini yang diterima pengirim sangat kecil, sedangkan bagi penerima pesan *spam*, konsekuensinya lebih besar.

### 2.2. Algoritma Pencocokan String

Algoritma pencocokan string (*string matching/ pattern matching algorithm*) adalah algoritma yang digunakan untuk mencocokkan sebuah string keyword di dalam suatu string text. Ada tiga jenis algoritma yang digunakan dalam tugas besar ini yaitu algoritma Boyer-Moore, algoritma Knuth-Morris-Pratt, dan regular expression.

#### 2.2.1. Algoritma Boyer-Moore

Algoritma Boyer-Moore adalah salah satu algoritma pencarian string, dipublikasikan oleh Robert S. Boyer, dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencarian string yang ditemukan sebelumnya, algoritma Boyer-Moore mulai mencocokkan karakter dari sebelah kanan pattern. Ide di balik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat.

Tabel untuk penggeseran bad-character dan good-suffix dapat dihitung dengan kompleksitas waktu dan ruang sebesar  $O(n + \sigma)$  dengan  $\sigma$  adalah besar ruang alfabet. Sedangkan pada fase pencarian, algoritma ini membutuhkan waktu sebesar  $O(mn)$ , pada kasus terburuk, algoritma ini akan melakukan  $3n$  pencocokkan karakter, namun pada performa terbaiknya algoritma ini hanya akan melakukan  $O(m/n)$  pencocokkan.

#### 2.2.2. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

Jika kita melihat algoritma brute force lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

Tabel untuk penggeseran bad-character dan good-suffix dapat dihitung dengan kompleksitas waktu dan ruang sebesar  $O(n + \sigma)$  dengan  $\sigma$  adalah besar ruang alfabet. Sedangkan pada fase pencarian, algoritma ini membutuhkan waktu sebesar  $O(mn)$ , pada kasus terburuk, algoritma ini akan melakukan  $3n$  pencocokkan karakter, namun pada performa terbaiknya algoritma ini hanya akan melakukan  $O(m/n)$  pencocokkan.

### 2.2.2. Algoritma Pencocokan String Regular Expression

Regular expression merupakan notasi yang digunakan untuk mendeskripsikan himpunan karakter string. Sebagai contoh, himpunan yang terdiri dari string “Handel”, “Händel”, dan “Haendel” dapat dideskripsikan dengan “H(ä|ae?)ndel” (atau alternatif lainnya, notasi ini menunjukkan bahwa notasi tersebut sesuai dengan setiap string yang diberikan).

#### Definisi :

Regular expression didefinisikan berdasarkan aturan teori bahasa formal. Regular expression terdiri dari konstanta dan operator yang menunjukkan himpunan-himpunan string dan operasi antar himpunan string tersebut secara berurutan. Konstanta yang telah didefinisikan adalah:

- Himpunan kosong, diberi notasi  $\emptyset$ .
- String kosong, diberi notasi  $\epsilon$ .
- Karakter, diberi notasi sesuai dengan karakter bahasa yang digunakan.

Operator yang telah didefinisikan adalah:

- Konkatenasi, misal  $\{ "ab", "c" \} \{ "d", "ef" \} = \{ "abd", "abef", "cd", "cef" \}$ .
- Alternasi, misal  $\{ "ab", "c" \} \mid \{ "ab", "d", "ef" \} = \{ "ab", "c", "d", "ef" \}$ .
- Kleene star, menunjukkan semua himpunan yang dapat dibuat dengan melakukan konkatenasi 0 atau lebih banyak string dari string yang dilakukan. Misal  $\{ "ab", "c" \}^* = \{ \epsilon, "ab", "c", "abab", "abc", "cab", "cc", "ababab", "abcb", \dots \}$ .

Untuk mengurangi jumlah tanda kurung, diasumsikan Kleene star memiliki prioritas terbesar dilanjutkan dengan operasi konkatenasi baru operasi alternasi.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1. KMP**

Secara sistematis, langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string:

1. Algoritma Knuth-Morris-Pratt mulai mencocokkan pattern pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  - a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
  - b. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser pattern berdasarkan tabel next, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

#### **3.2. Boyer Moore**

Secara sistematis, langkah-langkah yang dilakukan algoritma Boyer-Moore pada saat mencocokkan string adalah:

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  - a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
  - b. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser pattern dengan memaksimalkan nilai penggeseran good-suffix dan penggeseran bad-character, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1. Spesifikasi Teknis Program

##### 1. Spam Engine

Terdiri dari tiga engine terpisah berupa executable file yang dibuat dengan bahasa Go.

###### a. Engine Boyer-Moore

- `func normalize(text string) string;`
- `func analyzeLastOccurence(keywords string) [CHAR_NUMBER]int;`
- `func solve(keywords, text string) int;`

###### b. Engine Knuth-Morris-Pratt

- `func computeLongestSequence(pattern string, j int) int;`
- `func getSequenceTable(pattern string) []int;`
- `func MatchString(pattern, text string) int;`

###### c. Engine Regex

- `func main();`

##### 2. Web Server

Terdiri dari beberapa file terpisah yang memiliki tanggung jawab yang berbeda yang dibuat dengan bahasa PHP.

###### a. oauth.php

- `function init();`
- `function makeOAuthHeader($httpMethod, $baseUrl, $param);`

###### b. request.php

- `function sendRequest($baseUrl, $param, $header=NULL);`
- `function getQuery($arrKeys);`
- `function getTimeLine($param=ALL_TWIT);`

###### c. spam-controller.php

- `function searchKeywords($input, $algorithm);`

###### d. utils.php

- `function encodeMap($map, $type);`
- `function mapToString($map, $mapper="");`
- `function bundle($arr, $separator);`
- `function singleWrap($string, $wrapper);`
- `function wrap($arr, $wrapper);`
- `function encodeUrl($url, $query);`
- `function split($string, $splitter);`
- `function notNull($arr, $keys);`
- `function getVar($var, $wrap=false);`
- `function export($var, $value);`

### 3. Antarmuka Pengguna

Terdiri dari beberapa file terpisah yang mengatur tampilan antarmuka halaman web yang dibuat dengan HTML, CSS, PHP, dan Javascript.

a. `index.php`

Pengatur utama bagi halaman web yang ditampilkan. Memilih file apa yang akan ditampilkan, menerima query dari masukan pengguna, menginisiasi permintaan *timeline* ke Twitter, serta menginisiasi pemilahan spam.

b. `main.html`

Halaman utama yang ditampilkan ke pengguna atau disebut *Home*.

c. `about.html`

Halaman mengenai informasi dari kreator aplikasi.

d. `controller.js`

Pengatur penyisipan data dari server ke halaman yang dilihat pengguna, serta pengatur interaksi pengguna dengan halaman web.

### 4.2. Eksperimen/ Pengujian

Aplikasi ini membaca post (*tweets*) dari timeline Twitter dengan Twitter API dan kemudian melakukan pencocokan *keywords spam message* yang dimasukkan pengguna terhadap semua post yang diambil tersebut sesuai algoritma yang dipilih. Jika pada sebuah *tweet* terdapat *keywords* yang bersesuaian maka *tweet* tersebut ditandai dengan simbol X (Gambar 1.a.). Sedangkan jika pada sebuah *tweet* tidak dijumpai *keywords* yang bersesuaian, maka *tweet* tersebut bebas spam dan ditandai dengan simbol ✓ (Gambar 1.b.).



1.a.



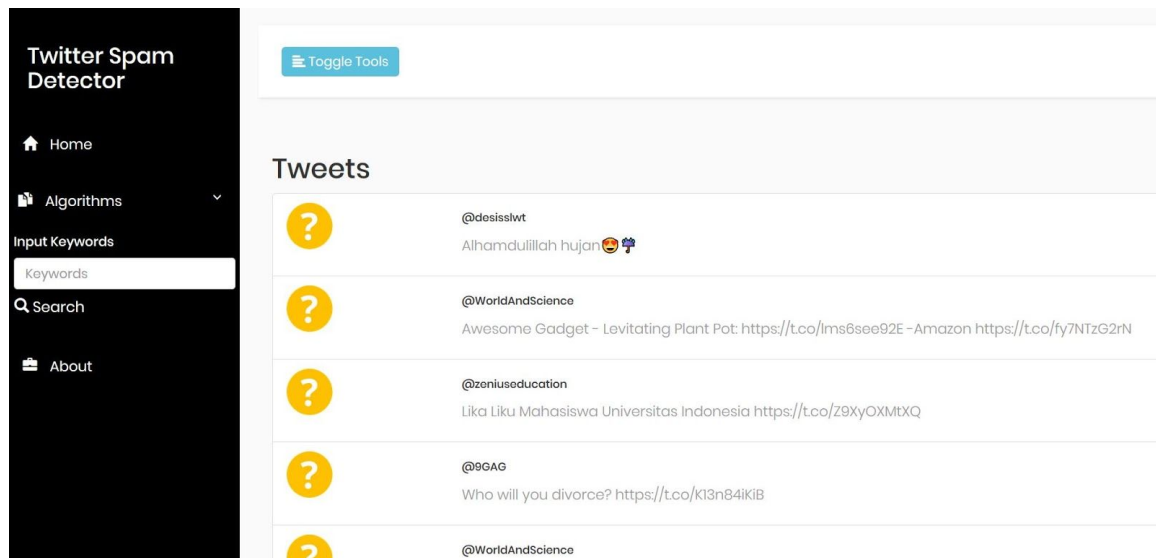
1.b.

Gambar 1.

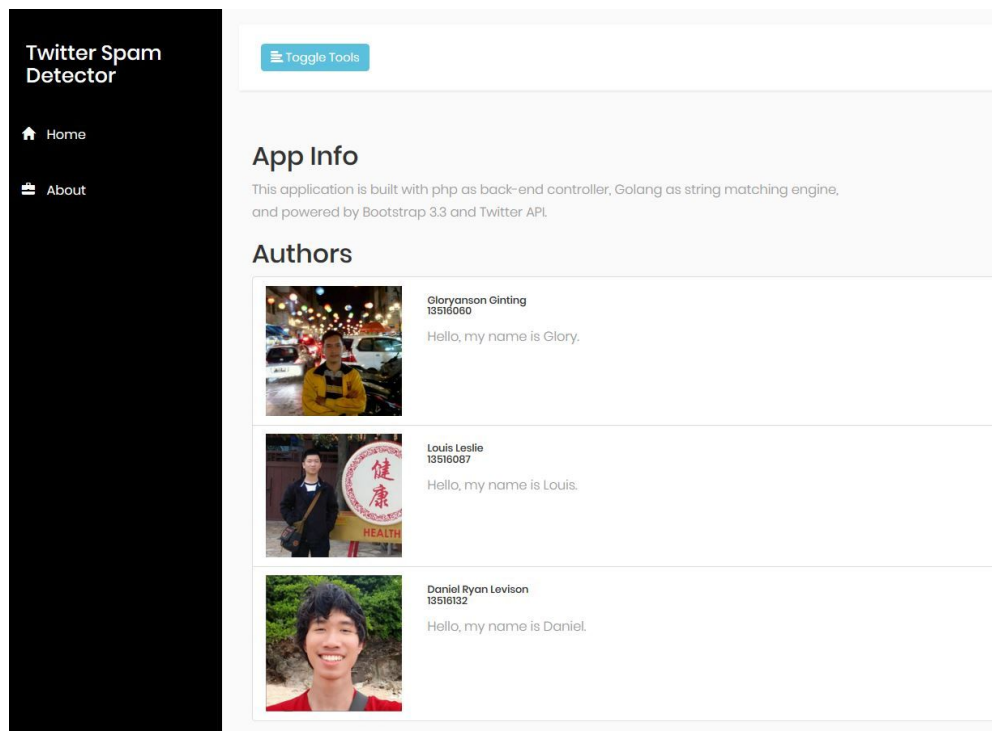
Gambar 1.a. Simbol Spam, Gambar 1.b. Simbol Bebas Spam

### 4.2.1. Tampilan

- Home



- About





## 4.2.2. Boyer-Moore

- Query : The

The screenshot shows the Twitter Spam Detector interface. The left sidebar contains the title 'Twitter Spam Detector' and navigation links: Home, Algorithms, Input Keywords, Search, and About. The 'Input Keywords' field contains the text 'The'. The main content area, titled 'Tweets (Boyer-Moore)', displays a list of tweets with their detection status indicated by a green checkmark (spam) or a red X (not spam).

Status	Username	Tweet Text
Spam (Green Checkmark)	@9GAG	Who will you divorce? <a href="https://t.co/k13n84kIB">https://t.co/k13n84kIB</a>
Not Spam (Red X)	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOzsGt7">https://t.co/njcWOzsGt7</a>
Spam (Green Checkmark)	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 72 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wf8Do5UPPq">https://t.co/wf8Do5UPPq</a> by @fr...
Not Spam (Red X)	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4kw3X0iv4k">https://t.co/4kw3X0iv4k</a>
Not Spam (Red X)	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC8H5bw3">https://t.co/C4vC8H5bw3</a>
Not Spam (Red X)	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/VVBH9M83DQ">https://t.co/VVBH9M83DQ</a>
Not Spam (Red X)	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPQREsWkR">https://t.co/DtPQREsWkR</a>

- Query : who

The screenshot shows the Twitter Spam Detector interface with the 'Input Keywords' field set to 'who'. The 'Tweets (Boyer-Moore)' section displays the following results:

Status	Username	Tweet Text
Not Spam (Red X)	@9GAG	Who will you divorce? <a href="https://t.co/k13n84kIB">https://t.co/k13n84kIB</a>
Spam (Green Checkmark)	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOzsGt7">https://t.co/njcWOzsGt7</a>
Spam (Green Checkmark)	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 72 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wf8Do5UPPq">https://t.co/wf8Do5UPPq</a> by @fr...
Spam (Green Checkmark)	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4kw3X0iv4k">https://t.co/4kw3X0iv4k</a>
Spam (Green Checkmark)	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC8H5bw3">https://t.co/C4vC8H5bw3</a>
Spam (Green Checkmark)	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/VVBH9M83DQ">https://t.co/VVBH9M83DQ</a>
Spam (Green Checkmark)	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPQREsWkR">https://t.co/DtPQREsWkR</a>

### 4.2.3. KMP

- Query : Who

The screenshot shows the Twitter Spam Detector interface. The left sidebar contains the following elements:
 

- Twitter Spam Detector logo
- Home icon
- Algorithms dropdown menu
- Input Keywords section with a search bar containing 'Who'
- Search icon
- About icon

The main content area displays the results for the query 'Who'. At the top, there is a 'Toggle Tools' button. Below it, the section is titled 'Tweets (KMP)'. The results are listed in a table with the following columns: a status icon (red X or green checkmark), the username, and the tweet text. The results are as follows:

Status	Username	Tweet Text
✗	@9GAG	Who will you divorce? <a href="https://t.co/K13n84iKiB">https://t.co/K13n84iKiB</a>
✓	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOZsGi7">https://t.co/njcWOZsGi7</a>
✓	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 72 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wF8Do5UPPq">https://t.co/wF8Do5UPPq</a> by @fr...
✓	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4Kw3X0lv4k">https://t.co/4Kw3X0lv4k</a>
✓	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC6H5bw3">https://t.co/C4vC6H5bw3</a>
✓	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/yVBH9M83DQ">https://t.co/yVBH9M83DQ</a>
✓	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPGREsWkR">https://t.co/DtPGREsWkR</a>

- Query : on

The screenshot shows the Twitter Spam Detector interface. The left sidebar contains the following elements:
 

- Twitter Spam Detector logo
- Home icon
- Algorithms dropdown menu
- Input Keywords section with a search bar containing 'on'
- Search icon
- About icon

The main content area displays the results for the query 'on'. At the top, there is a 'Toggle Tools' button. Below it, the section is titled 'Tweets (KMP)'. The results are listed in a table with the following columns: a status icon (green checkmark or red X), the username, and the tweet text. The results are as follows:

Status	Username	Tweet Text
✓	@9GAG	Who will you divorce? <a href="https://t.co/K13n84iKiB">https://t.co/K13n84iKiB</a>
✗	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOZsGi7">https://t.co/njcWOZsGi7</a>
✗	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 72 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wF8Do5UPPq">https://t.co/wF8Do5UPPq</a> by @fr...
✓	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4Kw3X0lv4k">https://t.co/4Kw3X0lv4k</a>
✓	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC6H5bw3">https://t.co/C4vC6H5bw3</a>
✓	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/yVBH9M83DQ">https://t.co/yVBH9M83DQ</a>
✓	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPGREsWkR">https://t.co/DtPGREsWkR</a>

### 4.2.3. Regular Expression

- Query : `wi*1`

The screenshot shows the Twitter Spam Detector interface. The left sidebar contains a navigation menu with 'Home', 'Algorithms', 'Input Keywords', 'Search', and 'About'. The 'Input Keywords' field contains 'wi\*1'. The main content area is titled 'Tweets (REGEX)' and displays a list of tweets with their detection status. The first tweet is marked as spam (red X), while the others are not (green checkmarks).

Status	Username	Tweet Text
Spam (X)	@9GAG	Who will you divorce? <a href="https://t.co/k13n84kiB">https://t.co/k13n84kiB</a>
Not Spam (✓)	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOzsGi7">https://t.co/njcWOzsGi7</a>
Not Spam (✓)	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 7.2 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wf8Do5UPPq">https://t.co/wf8Do5UPPq</a> by @fr...
Not Spam (✓)	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4kw3X0lv4k">https://t.co/4kw3X0lv4k</a>
Not Spam (✓)	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC6H5bw3">https://t.co/C4vC6H5bw3</a>
Not Spam (✓)	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/YVBH9M83DQ">https://t.co/YVBH9M83DQ</a>
Not Spam (✓)	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPGREsWKR">https://t.co/DtPGREsWKR</a>

- Query : `and?`

The screenshot shows the Twitter Spam Detector interface with the 'Input Keywords' field set to 'and?'. The 'Tweets (REGEX)' list shows that the first two tweets are now marked as spam (red X), while the others remain marked as not spam (green checkmarks).

Status	Username	Tweet Text
Not Spam (✓)	@9GAG	Who will you divorce? <a href="https://t.co/k13n84kiB">https://t.co/k13n84kiB</a>
Spam (X)	@WorldAndScience	The challenges of an alien spaceflight program: escaping super earths and red dwarf stars <a href="https://t.co/njcWOzsGi7">https://t.co/njcWOzsGi7</a>
Spam (X)	@elonmusk	RT @ElectrekCo: Tesla's fleet has driven 7.2 billion miles and its energy products produced 10.3 billion kWh <a href="https://t.co/wf8Do5UPPq">https://t.co/wf8Do5UPPq</a> by @fr...
Not Spam (✓)	@elonmusk	RT @Tesla: The night sky above the Tesla South Australia Powerpack Project. Happy Earth Day <a href="https://t.co/4kw3X0lv4k">https://t.co/4kw3X0lv4k</a>
Not Spam (✓)	@WorldAndScience	The Galton board - the concept of normal distribution <a href="https://t.co/C4vC6H5bw3">https://t.co/C4vC6H5bw3</a>
Not Spam (✓)	@WorldAndScience	String Theory Explained - What is The True Nature of Reality? <a href="https://t.co/ivuHekm7VG">https://t.co/ivuHekm7VG</a> <a href="https://t.co/YVBH9M83DQ">https://t.co/YVBH9M83DQ</a>
Not Spam (✓)	@9GAG	Talented Artist Gives Generic Mass-Produced Dolls Their Own Unique Look <a href="https://t.co/DtPGREsWKR">https://t.co/DtPGREsWKR</a>

### 4.3. Analisis hasil pengujian

Dari 6 (enam) buah query yang diberikan terhadap *tweets* pada *timeline*, berhasil dilakukan pencocokan *keywords* terhadap setiap *tweet* dan berhasil dibedakan *tweets* yang merupakan *spam* dan *tweets* yang bukan merupakan *spam*.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Dibandingkan dengan algoritma Brute Force, pencocokan string yang dilakukan dengan algoritma Knuth-Morris-Pratt dan Boyer-Moore jauh lebih cepat. Meskipun pada kasus terburuk, ketiga algoritma memiliki kompleksitas  $O(nm)$ , dengan  $n$  banyaknya karakter pada teks yang menjadi karakter pertama tiap kali pencocokan berlangsung, dan  $m$  banyaknya karakter pada *string* yang dicari, namun baik algoritma Knuth-Morris-Pratt maupun Boyer-Moore memiliki mekanisme sedemikian sehingga nilai  $n$  diperkecil. Berbeda halnya dengan Brute Force yang sama sekali tidak memiliki mekanisme untuk memperkecil nilai  $n$ .

#### **5.2. Saran**

Pada saat laporan ini dibuat aplikasi ini belum memiliki kamus kata/kalimat yang dapat mengkategorikan suatu teks adalah spam atau bukan. Kata/kalimat hanya baru didapatkan dari pengguna. Untuk pengembangannya dapat dibuat suatu kamus kata/kalimat sehingga aplikasi akan mencari setiap kata/kalimat pada kamus tersebut pada teks dan apabila ditemukan maka teks dikategorikan sebagai spam. Aplikasi ini dapat ditanamkan pada layanan *chatting* atau *email* untuk mengkategorikan pesan sebagai spam atau bukan dan kemudian dapat dilakukan tindakan lanjutan apabila pesan dikategorikan sebagai spam.

#### **5.3. Kesan/ Komentar Tugas**

Tugas ini cukup menarik karena ditugaskan untuk melakukan eksplorasi terhadap berbagai macam API sosial media sehingga didapat pengalaman baru dalam menggunakan dan mengolah data dari berbagai macam API tersebut.

Komentar terhadap tugas ini yaitu bahwa tugas ini terlalu terfokus pada pembuatan web, padahal inti dari tugas ini sebenarnya adalah untuk mengimplementasikan algoritma pencocokan string, sehingga usaha yang dikerahkan untuk tugas ini lebih banyak ditujukan dalam pembuatan web.

**Daftar Pustaka**

Lecroq, Thierry Charras, Christian. 2001. Handbook of Exact String Matching Algorithm.

<http://spam.abuse.net/overview/whatisspam.shtml>

Munir, Rinaldi. 2004. Algoritma Pencarian String (String Matching). Bandung

Thompson, Ken, “Regular Expression Search Algorithm”, Communication of the ACM, Volume 11, Number 6, 1968, hal 419-422.