

Email Filtering sederhana pada aplikasi mail klien dengan menggunakan string matching

Akhmad Mukhammad

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if14152@students.if.itb.ac.id¹

Abstrak

Email merupakan senjata/kakas utama mahasiswa pada umumnya dan warga informatika pada khususnya. Saat ini kebanyakan email-email yang sudah masuk pada server biasanya sudah difilter dari keberadaan spam oleh server itu sendiri, jarang sekali aplikasi email klien yang menyediakan fasilitas pemfilteran email, padahal pemfilteran pada sisi klien bisa dilakukan lebih leluasa terutama dalam hal penyaringan isi dari email itu sendiri, dengan catatan bahwa email yang sudah di unduh dari server ke klien berarti sudah terotentifikasi oleh klien yang mengunduh, untuk itu aplikasi klien bisa lebih leluasa untuk mengakses email dan isinya.

Kata kunci : *string matching ,mail filtering , dictionary attack, algoritma Boyer-Moore, spam*

1. Pendahuluan

spam adalah bahasa internet yang menunjuk pada email yang berisi iklan komersial tidak diinginkan. Beberapa orang menamakannya *junk email*. Biasanya email ini berisi iklan komersial, layanan, atau sebuah product.

Sebuah alamat email kemungkinan besar akan menjadi target spam jika alamat email tersebut mudah ditemukan di dunia internet, sederhananya jika suatu email semakin mudah dikenali oleh search engine maka semakin besar kemungkinan alamat email tersebut menjadi target spam.

Saat ini sudah banyak aplikasi email filtering yang sudah kita kenal, spam assassin misalnya, aplikasi yang berjalan di server yang menyaring semua email yang masuk ke server dan menyeleksi email-email yang tergolong sebagai spam, kebanyakan program-program yang berjalan di server ini hanya menyeleksi email berdasarkan subject atau sender, dengan pertimbangan *effort* yang dibutuhkan untuk memproses *subject* suatu email lebih sedikit daripada harus memproses isi dari email tersebut. Padahal jumlah email yang masuk ke suatu mail server tidak bisa dibilang sedikit. Dan sering kali penyaringan email spam berdasarkan *subject* atau *sender* saja kurang efisien, banyak email-email isinya terkategori sebagai spam tidak terdeteksi sebagai spam dan bahkan tidak sedikit juga sebaliknya.

Oleh karena itu, dibutuhkan aplikasi penyaringan dengan memproses *content* dari email yang dijalankan pada sisi klien sehingga email-email berkategori spam bisa terdeteksi lebih jitu dan aplikasi tidak memberatkan server mail itu sendiri.

2. Pemodelan Masalah

spam-filtering pada sisi klien mudah diterapkan dan lebih besar kemungkinan memberikan hasil yang diharapkan.

Dengan bantuan beberapa Algoritma string-matching, mudah untuk menambahkan plugin filtering pada suatu aplikasi mail client seperti horde atau thunderbird, plugin ini menambahkan kemampuan pada aplikasi mail klien untuk melakukan *string-matching* pada body email dan memberikan nilai indeks spam terhadap email tersebut. Semakin banyak kata kunci yang tergolong sebagai spam maka semakin tinggi indeks spam yang akan diberikan pada email tersebut.

3.String Matching

Pencarian string di dalam teks disebut juga pencocokan string (string matching atau pattern matching).

Persoalan pencarian string dirumuskan sebagai berikut:

Diberikan:

1. teks (text), yaitu (long) string yang panjangnya n karakter (dalam kasus ini teks didapat dari body email).
2. pattern, yaitu string dengan panjang m karakter ($m < n$) yang akan dicari di dalam teks. (dalam kasus ini pattern merupakan keyword yang umumnya tergolong sebagai spam)

Carilah (find atau locate) lokasi pertama di dalam teks yang bersesuaian dengan pattern. Aplikasi dari masalah pencocokan string antara lain pencarian suatu kata di dalam dokumen (misalnya menu Find di dalam Microsoft Word).

3.1 Brute Force

Dengan sumsi bahwa teks berada di dalam array $T[1..n]$ dan *pattern* berada di dalam array $P[1..m]$, maka algoritma *brute force* pencocokan *string* adalah sebagai berikut:

1. Mula-mula *pattern* P dicocokkan pada awal teks T .
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap setiap karakter di dalam *pattern* P dengan karakter yang bersesuaian di dalam teks T sampai:
 - a. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
 - b. dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila *pattern* P belum ditemukan kecocokannya dan teks T belum habis, geser *pattern* P satu karakter ke kanan dan ulangi langkah 2.

3.2 Algoritma Knuth-Morris-Pratt (KMP)

Pada algoritma *brute force*, setiap kali ditemukan ketidakcocokan *pattern* dengan teks, maka *pattern* digeser satu karakter ke kanan.

Sedangkan pada algoritma KMP, kita memelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh, tidak hanya satu karakter seperti pada algoritma *brute force*.

Dengan algoritma KMP ini, waktu pencarian dapat dikurangi secara signifikan. Algoritma KMP dikembangkan oleh D. E. Knuth, bersama-sama dengan J. H. Morris dan V. R. Pratt.

Fungsi Pinggiran (*Border Function*)

Fungsi *pinggiran* $b(j)$ didefinisikan sebagai ukuran awalan terpanjang dari P yang merupakan akhiran dari $P[1..j]$.

j	1	2	3	4	5	6
$P[j]$	a	b	a	b	a	a
$b(j)$	0	0	1	2	3	1

Teks: abcabcabd
 Pattern: abcabd

Mula-mula kita hitung fungsi pinggiran untuk *pattern* tersebut:

j	1	2	3	4	5	6
$P[j]$	a	b	c	a	b	d
$b(j)$	0	0	0	1	2	0

Teks: abcabcabd
 Pattern: abcabd

↑
 $j = 3$

4. Analisis Model Masalah

Pada model masalah yang sudah digambarkan pada bagian sebelumnya, maka dapat dirancang suatu model yang terdiri atas 2 komponen, yaitu :

- Pembangkit kata kunci yang dikenali sebagai spam, kata kunci disini berfungsi sebagai *pattern* yang akan dicari dalam body email.
- Pemberian nilai indeks terhadap body email berdasarkan pencocokan kata kunci dengan body email, semakin banyak kata kunci yang ditemukan dalam body email maka indeks akan semakin tinggi, semakin tinggi indeks maka semakin tinggi kemungkinan email tersebut adalah *spam*.
- Tools sebagai input, memungkinkan user untuk menambahkan kata-kata baru sebagai *pattern* yang digunakan untuk pencocokan string dengan body email..
- Tools yang memungkinkan user memasukkan besar indeks email akan masuk sebagai spam.

Keempat komponen diatas akan berkolaborasi sehingga setiap email yang ada dalam inbox akan memiliki indeks spam masing-masing.

Plugin ini sebenarnya juga bisa memberikan manfaat lain, selain sebagai spam-filter, kita juga bisa memberikan input *pattern-pattern* baru sehingga email akan terindeks berdasarkan *pattern-pattern* tersebut. Misalnya kita memasukkan kata-kata yang berhubungan dengan kesehatan, maka email bisa terindeks berdasarkan indeks kesehatan alih-alih dari indeks spam

5. Kesimpulan dan Saran

Masalah spam-filtering merupakan satu dari masalah yang bisa ditangani dengan menggunakan bantuan algoritma string matching, masih banyak juga hal-hal lain yang bisa diterapkan menggunakan bantuan algoritma string matching ini, plugin untuk mencari email, men-*sortir* email berdasarkan konten edukasi dan beasiswa, dan laen laen..

Algoritma *string-matching*) dalam hal ini dapat dijadikan sebagai salah satu patokan dalam setiap aplikasi yang bersinggungan dengan teks dan string.

6. Daftar Pustaka

1. Munir, Rinaldi. 2005. "*Strategi Algoritmik*". Departemen Teknik Informatika, Institut Teknologi Bandung.