

Membuat Bot Untuk Mengkategorikan Informasi Dari Tweet dengan Algoritma KMP dan memberikan notifikasi

Jon Felix Germinian / 13518025
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518025@std.stei.itb.ac.id

Abstract—Di abad ke-20 ini, social media telah berkembang jauh. Banyak dari kita mendapatkan berita terbaru dari berbagai social media, salah satunya adalah twitter. Salah satu kelemahan twitter adalah sifatnya yang pasif Di dalam makalah ini akan didiskusikan cara mengimplementasikan cara membuat bot notifikasi dari twitter seseorang menggunakan algoritma Knuth-Morris-Pratt.

Keywords— Knuth-Morris-Pratt; Algorithm; String Matching;

I. INTRODUCTION

Di dunia post-modern, hampir semua orang yang kita temui memiliki akses ke internet. Dengan akses ke internet yang cukup mudah, social media telah menjadi media berita yang sangat cepat berkembang. Mulai dari berita yang disebarkan lewat whatsapp tentang teman dekat hingga tweet Presiden Amerika Serikat Donald Trump. Meskipun banyak riset yang menunjukkan pemakaian social media berlebihan dapat menimbulkan dampak buruk, penggunaan social media untuk mencari informasi adalah salah satu pemakaian yang ditoleransi secara umum. Dengan menggunakan social media, informasi dapat dengan gampang diedarkan dengan cepat dan mudah.

Diikuti dengan perkembangan teknologi manusia di 20 tahun terakhir, platform social media sudah berkembang. Pada tahun 1990-an, social media yang populer di internet hanyalah relay chat, tetapi masuk ke tahun 2010-an, hampir semua orang di negara maju dan berkempang kenal dengan apa yang dinamakan Twitter, Whatsapp, Facebook, dll. Sebagai contoh, salah satu kegunaan Twitter untuk berbagai orang adalah untuk mendapatkan berita seperti pengumuman anime baru, berita event di dalam sebuah game, dan hal-hal lainnya.

Tetapi mencari berita di dalam social media berita merupakan kegiatan aktif dimana pengguna harus mencari berita tersebut di antara post-post lainnya yang membanjiri social media tersebut. Menerima berita dengan metode ini

kurang efektif. Salah satu solusi untuk menyelesaikan masalah ini adalah dengan notifikasi, tetapi notifikasi tidak dapat membedakan informasi mana yang dapat dikatakan relevan dan informasi mana yang tidak relevan bagi pengguna. Mencari informasi secara aktif dengan *browsing* pada social media juga dapat membuat pengguna kecanduan menggunakan sosial media.

Dengan begitu saya (penulis) melihat kebutuhan untuk metode mendapatkan informasi terkini secara pasif dengan notifikasi untuk para orang yang takut ketinggalan informasi. Di dalam makalah ini, saya akan membahas cara membuat bot yang dapat mengkategorisasikan informasi secara sederhana dengan Algoritma KMP yang diimplementasikan dengan menggunakan Node.js dan IFTTT.

II. LANDASAN TEORI

A. String Matching

String matching adalah algoritma untuk mengecek apakah suatu string tertentu (biasa dinamakan pattern) ditemukan di dalam string yang lebih besar (biasa dinamakan text). Algoritma paling simple untuk mengimplementasikan string matching adalah algoritma brute force atau kadang dipanggil dengan metode naif. Pseudocode untuk metode brute force adalah sebagai berikut :

```
Algorithm BruteForceStringMatch( $T[0\dots n-1]$ ,  $P[0\dots m-1]$ )  
  for  $i \leftarrow 0$  to  $n-m$  do  
     $j \leftarrow 0$   
    while  $j < m$  and  $P[j] = T[i+j]$  do  
       $j++$   
    if  $j = m$  then return  $i$   
  return -1
```

Gambar 1 (Sumber : <http://www.csl.mtu.edu/cs4321/www/Lectures/Lecture%205%20-%20Brute%20Force%20Sorting%20and%20String%20Matching.htm>)

Di luar dari metode brute force, terdapat beberapa algoritma lain untuk string matching diantaranya adalah algoritma Boyer-Moore, algoritma Knuth-Morris-Pratt, algoritma Rabin-Karp, algoritma Finite state automata. Perbedaan kompleksitas dari algoritma-algoritma tersebut adalah sebagai berikut:

String matching algorithm	Time complexity	
	preprocessing	matching
Brute force	$O(n^2)$	$O(mn)$
Rabin-Karp	$\Theta(m)$	Avg $\Theta(n+m)$ worst $\Theta(n \cdot m)$
Finite state automata	$\Theta(m \Sigma)$	$\Theta(n)$
Boyer-Moore	$\Theta(m+ \Sigma)$	$\Omega(n/m), O(n)$
Knuth-Morris-Pratt	$\Theta(m)$	$\Theta(n)$

Tabel 1 : Komparasi kompleksitas berbagai algoritma string matching

B. Aplikasi dari String Matching

String matching telah banyak diaplikasikan untuk berbagai masalah di dunia nyata. Beberapa contoh aplikasi dari string matching adalah sebagai berikut :

1. **Text Editor, Perpustakaan Digital dan Search Engine:** Digunakan untuk pencarian pattern di dalam text
2. **Spam Detection :** Mengidentifikasi apakah sebuah pesan adalah spam (sesuai filter yang diketahui)
3. **DNA sequencing :** Mencari pattern dari asam amino yang ada di dalam data DNA.
4. **Pendeteksi Plagiarisme :** Mengecek 2 text dan menilai kesamaan dari 2 text tersebut

C. Algoritma Knuth-Morris-Pratt

Algoritma pencocokan string Knuth-Morris-Pratt mencari keberadaan sebuah pattern di dalam sebuah text dengan mengimplementasikan beberapa teknik. Ide dasar dari algoritma ini adalah untuk mencari pattern baik dari kiri ke kanan atau sebaliknya tetapi saat sebuah ketidaksamaan

ditemukan algoritma akan melakukan shift yang lebih *pintar* dari algoritma Brute Force.

Pseudocode algoritma Knuth-Morris-Pratt berdasarkan referensi[2] adalah sebagai berikut :

Procedure HitungPinggiran(input m : integer, P : array [1..m] of char, output b : array[1..m] of integer) {menghitung nilai b[1..m] untuk pattern P[1..m]}

Kamus

k,q : integer

Algoritma

```

b[1] ← 0 ;
q ← 2 ;
k ← 0;
for q ← 2 to m do
    while ((k>0) and (P[q]≠P[k+1])) do
        k ← b[k]
    endwhile
    if P[q]=P[k+1] then
        k ← k + 1
    endif
    b[q] = k
endfor

```

Function KMPsearch(input m,n : integer, input P : array [1..m] of char, input T : array [1..n] of char, output idx : integer) : boolean

{mencari kecocokan pattern P di dalam teks T dengan algoritma KMP. Jika ditemukan P di dalam T, maka lokasi awal kecocokan disimpan di dalam peubah idx}

Kamus

i,j : integer
found : boolean
b : array[1..m] of integer

procedure HitungPinggiran(input m : integer, P : array [1..m] of char, output b : array[1..m] of integer)

Algoritma

```

HitungPinggiran(m, P, b)
j ← 0 ;
i ← 1 ;

```

```

ketemu ← false;
while (i ≤ n and not ketemu) do
  while ((j > 0) and (P[j+1] ≠ T[i])) do
    j ← b[j]
  endwhile
  if P[j+1] = T[i] then
    j ← j + 1
  endif
  if j = m then
    return true
  else
    i ← i + 1
  endif
endwhile
return false

```

D. IFTTT

IFTTT (If This Then That) adalah sebuah freeware berbasis web yang dapat membuat rantai perintah jika kondisi tertentu ter-*trigger* yang bernama applet. Sebagai contoh, sebuah applet menunggu tweet seseorang dengan hashtag tertentu, applet tersebut dapat mengaktifkan berbagai perintah seperti mengirimkan text tweet tersebut ke facebook.

E. Chatbot

Chatbot adalah program komputer yang memproses percakapan pengguna lalu memberikan balasan kepada pengguna untuk mensimulasikan percakapan. Kompleksitas chatbot beragam dari chatbot simpel yang hanya memberikan satu balasan hingga chatbot yang dapat melakukan manajemen server di dalam discord.

F. Nodejs

Node.js adalah runtime environment open-source dan cross-platform yang mengeksekusi kode JavaScript di luar web browser. Node.js adalah representasi dari paradigma "JavaScript everywhere", menyatukan proses rekayasa web-application dengan satu bahasa pemrograman.

G. API

API (Application Programming Interface) adalah interface pemrograman yang mendefinisikan interaksi antara berbagai perangkat lunak. API sebuah aplikasi mendefinisikan berbagai perintah, fungsi dan protokol yang dapat digunakan developer untuk membangun aplikasi lain yang akan berinteraksi dengan aplikasi tersebut.

H. Discord API

Discord API adalah API yang dibuat oleh Discord untuk developer lain yang dapat digunakan untuk membuat banyak hal, mulai dari chatbot untuk game, keperluan komunitas, dan aplikasi-aplikasi lainnya yang akan berinteraksi dengan discord. Cara kerja Discord API adalah dengan memberikan data dari discord ke aplikasi dan memberikan akses kepada bot untuk mengirim pesan ke Discord.

I. Social Media

Social media adalah teknologi interaktif yang dimediasi oleh komputer untuk memfasilitasi manusia dalam pembuatan dan penyebaran informasi, ide, dan berbagai ekspresi lainnya melewati komunitas virtual.

Fitur-fitur umum yang disediakan oleh social media adalah kemampuan untuk menyebarkan foto, opini, berita dan sebagainya.

III. IMPLEMENTASI

Yang perlu dilakukan pertama adalah untuk mencari keyword tertentu yang sering muncul dan menganalisis apakah keyword tersebut dapat digunakan. Dengan keyword yang dimasukan ke dalam setiap kategori, saat bot mendapatkan message, bot akan mencari keyword yang telah dimasukan ada di dalam message yang didapat oleh bot. Untuk proses string matching keyword ke dalam text akan digunakan algoritma Knuth-Morris-Pratt.

Implementasi algoritma KMP dalam bot akan dibahas di dalam bab ini.

A. Implementasi fungsi string matching KMP

```

function knuthMorrisPratt(text, pattern) {
  if (pattern.length === 0) {
    return false;
  }

  let textIdx = 0;
  let patternIdx = 0;

  const patternTable = buildPatternTable(pattern);

```

```

while (textIdx < text.length) {
  if (text[textIdx] === pattern[patternIdx]) {
    if (patternIdx === pattern.length - 1) {
      return true; //pattern ditemukan
    }

    patternIdx += 1;
    textIdx += 1;
  } else if (patternIdx > 0) {
    patternIdx = patternTable[patternIdx - 1];
  } else {
    patternIdx = 0;
    textIdx += 1;
  }
}

return false;
}

function buildPatternTable(pattern) {
  const patternTable = [0];
  let prefixIdx = 0;
  let suffixIdx = 1;

  while (suffixIdx < pattern.length) {
    if (pattern[prefixIdx] === pattern[suffixIdx]) {
      patternTable[suffixIdx] = prefixIdx + 1;
      suffixIdx += 1;
      prefixIdx += 1;
    } else if (prefixIdx === 0) {
      patternTable[suffixIdx] = 0;
      suffixIdx += 1;
    } else {
      prefixIdx = patternTable[prefixIdx - 1];
    }
  }

  return patternTable;
}

```

B. Penkategorisasian

Dengan keyword-keyword yang telah dimasukan dibuat sebuah array. Dengan array tersebut dilakukan string matching

untuk setiap elemen array tersebut. Sebagai contoh, pengkategorian dicocokkan untuk twitter [@ArknightsEN](#)

Implementasinya adalah sebagai berikut:

```

let news = ["live", "maintenance"];
let gacha = ["higher rate"];
let pv = ["PV", "Trailer"];

function categorize(text) {
  for (let i = 0; i < news.length; i++) {
    if (knuthMorrisPratt(text, news[i])) {
      return "news";
    }
  }

  for (let i = 0; i < gacha.length; i++) {
    if (knuthMorrisPratt(text, gacha[i])) {
      return "gacha";
    }
  }

  for (let i = 0; i < pv.length; i++) {
    if (knuthMorrisPratt(text, pv[i])) {
      return "pv";
    }
  }
}

```

C. Implementasi chatbot di dalam discord

Menggunakan api discord.js yang telah disediakan oleh tim discord, bot akan login menggunakan token aplikasi. Chatbot pertama akan menginisialisasi dirinya. Dengan trigger sebuah message bot akan menarik sebuah pesan untuk dilakukan string matching. Implementasinya adalah sebagai berikut :

```

bot.on("ready", () => {
  console.log("Bot is Online");
});

bot.on("message", (msg) => {
  let category = categorize(msg.content);
  switch (category) {
    case "news":
      msg.channel.send("@everyone news for arknights");
      break;
  }
}

```

```

    case "gacha":
        msg.channel.send("@everyone new
gacha info");
        break;
    case "pv":
        msg.channel.send("@everyone new PV
is out");
        break;
    }
});

bot.login(token);

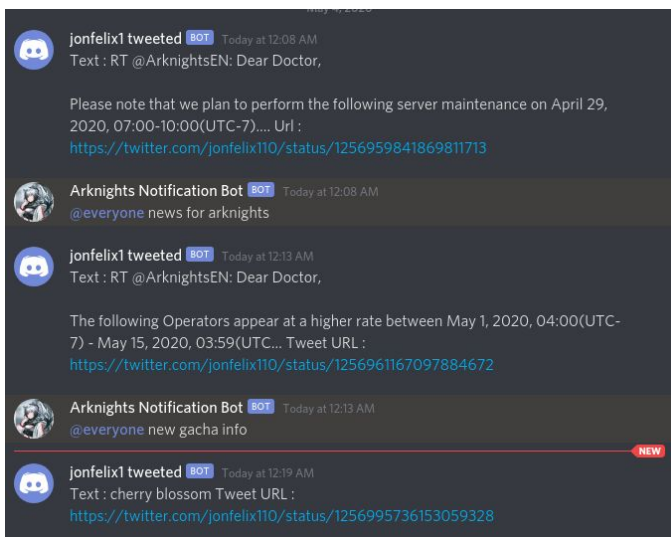
```

D. Implementasi webhook dengan IFTTT

Menggunakan IFTTT, sebuah applet dibuat yang menerima tweet dari seseorang lalu membuat post request dengan json : {"username" : "@{{UserName}} tweeted", "content" : "Text : {{Text}} URL : {{LinkToTweet}}"}"

IV. EKSPERIMEN

Beberapa eksperimen telah dilakukan untuk mengetes keberhasilan chatbot untuk mengkategorisasikan informasi yang diambil dari twitter.



Dari eksperimen tersebut dapat dilihat bahwa ada 1 pesan yang dikategorikan sebagai pesan berita, 1 pesan yang dikategorikan sebagai pesan gacha dan 1 pesan yang tidak dikategorikan. Penjelasan dari eksperimen tersebut adalah sebagai berikut :

- Pesan “ Dear Doctor,
Please note that we plan to perform the following server maintenance on April 29, 2020,

07:00-10:00(UTC-7)...” mengandung kata “maintenance” yang dimasukkan ke dalam keyword berita di dalam program.

- Pesan “Dear Doctor,
The following Operators appear at a higher rate between May 1, 2020, 04:00(UTC-7) - May 15, 2020, 03:59(UTC...” mengandung kata “maintenance” yang dimasukkan ke dalam keyword gacha di dalam program
- Pesan “cherry blossom” sama sekali tidak mengandung keyword sehingga tidak dikategorikan

V. ANALYSIS

Pada bab sebelumnya telah diperlihatkan bahwa bot telah bekerja dengan benar dalam mengkategorisasikan pesan yang didapat dan menotifikasi orang yang ada di dalam server tersebut. Tetapi ada beberapa kelemahan yang dihadapi bot ini:

- Memodifikasi keyword yang untuk string matching diharuskan mengoprek kembali kode program dan tidak bisa langsung di dalam runtime program tersebut yang tidak praktis untuk deployment di dunia nyata.
- Algoritma bukanlah algoritma terbaik untuk teks panjang tetapi untuk tweet yang panjangnya dibatasi oleh twitter algoritma KMP sudah berjalan dengan baik dan cepat.

VI. IDE UNTUK PENGEMBANGAN LEBIH LANJUT

Ada beberapa ide yang dapat diimplementasikan kedalam bot untuk meningkatkan fungsionalitasnya. Ide-ide tersebut adalah sebagai berikut:

- Membuat array keyword dapat dimodifikasi dengan perintah yang dapat diberikan kepada bot.
- Melakukan integrasi langsung dengan API dari berbagai aplikasi
- Membuat aplikasi lebih spesifik dengan melihat hashtag (membuat kategori dan subkategori)
- Tidak menotifikasi semua orang tetapi hanya menotifikasi orang yang telah memilih role untuk mendapatkan notifikasi

VII. KESIMPULAN

Sosial media dapat adalah wadah bagus untuk menyebarkan informasi tetapi dengan membuat sistem notifikasi seperti pada makalah ini, ketergantungan pengguna biasa pada sosial media dapat dikurangi. Seperti yang dibahas

pada bab IV, bot telah berhasil untuk mengkategorisasikan pesan yang didapat dari twitter, tetapi terdapat beberapa kelemahan seperti yang telah dibahas pada bab V. Karena bot yang dibuat pada makalah ini hanyalah *proof of concept* masih banyak pengembangan yang dapat dilanjutkan seperti yang tertera pada bab VI.

VIDEO LINK AT YOUTUBE

<https://youtu.be/M8RdqcdIBa0>

SOURCE CODE PROGRAM

Source code program dapat dilihat di <https://github.com/jonfelix1/arknights-notif-bot>

ACKNOWLEDGMENT

Pertama-tama saya ingin berterima kasih kepada orang tua saya yang memberikan saya kesempatan untuk berkuliah di Institut Teknologi Bandung. Kedua saya ingin berterima kepada Dosen mata kuliah IF2211 terutama untuk Ibu Masayu Leylia Khodra untuk mengajar saya di kelas dan Pak Rinaldi Munir yang telah menyediakan platform pembelajaran online dengan websitenya.

REFERENSI

- [1] Rinaldi Munir, "Pencocokan String", diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/> pada 1 May 2020
- [2] Gahayu Handari Ekaputri, Yulie Anneria Sinaga, "Aplikasi Algoritma Pencarian String Knuth-Morris-Pratt dalam Permainan Word Search" diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Smik/2005-2006/Makalah2006/MakalahSmik2006-07.pdf>
- [3] Dokumentasi Discord API, diakses dari <https://discordapp.com/developers/docs/intro> pada 1 May 2020

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 May 2020



Jon Felix Germinian/13518025