

Implementasi Algoritma Pencocokan String dalam Penentuan Tombol Respons Facebook

Raden Fajar Hadria Putra - 13511076
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
fajarghosty@gmail.com

Abstrak—Facebook telah menjadi salah satu bagian hidup dari sebagian besar warga internet di seluruh dunia. Facebook membantu banyak orang untuk berbagi momen-momen penting dalam hidup dalam bentuk status dan berbagi foto dan video. Untuk mendukung pemberian respons dari berbagai teman, Facebook memberikan fitur komentar dan tombol ‘like’. Namun, tombol ‘like’ terkadang tidak dapat merepresentasikan respons teman dengan baik, khususnya jika status atau konten yang ditampilkan merupakan konten negatif atau memiliki suasana sedih. Untuk itu, dibutuhkan tombol lain untuk merepresentasikan respons yang relevan dengan status tersebut.

Kata kunci : Facebook, Tombol Respons, keyword

I. PENDAHULUAN

Facebook merupakan salah satu situs web dan aplikasi jejaring sosial paling populer di seluruh dunia. Hal ini disebabkan oleh mudahnya mencari teman dalam Facebook dan mudahnya berbagi momen, opini, berita, dan curahan hati melalui berbagai bentuk seperti teks, foto, dan video. Facebook pun memberikan fitur yang dapat membantu orang lain untuk memberikan respons terhadap konten yang dibuat oleh pemilik akun. Respons tersebut dapat berbentuk sebuah komentar, yaitu suatu pesan singkat yang akan ditampilkan di bawah konten yang dikomentari. Komentar tersebut dapat berbentuk teks, foto, atau tautan yang dapat menampilkan *preview* dari tautan tersebut. Respons dapat juga berbentuk tombol like, dimana seluruh orang yang memberikan like didaftarkan pada daftar orang yang memberikan ‘like’ pada status tersebut. Fitur ‘like’ pun ditambahkan pada setiap komentar, sehingga pengguna dapat memberikan ‘like’ pada komentar tertentu.



Gambar 1. Contoh komentar berbentuk teks, tautan, dan gambar dan tidak ada orang yang memberikan ‘like’

[1]



Gambar 2. Post Facebook dengan satu buah 'like' pada konten dan satu buah 'like' pada satu komentar [1]

Terdapat kelemahan terhadap fitur 'like', yaitu jika terdapat sebuah status yang memiliki kesan prihatin seperti kabar meninggal atau kecelakaan, akan terkesan tidak etis jika orang lain memiliki pilihan untuk memberikan respons 'like'. Selain itu, jika terdapat suatu status yang meminta jawaban ya atau tidak, respons 'like' kurang menjawab apa yang ditanyakan oleh status itu. Untuk itu, dibutuhkan alternatif tombol yang dapat muncul tergantung dari konteks yang disampaikan oleh status yang bersangkutan.

II. DASAR TEORI

A. Algoritma String Matching

Algoritma string matching atau pencocokan string adalah teknik untuk mencocokkan suatu kata atau frase dengan suatu teks.

Persoalan pencarian string dirumuskan sebagai berikut:

Diberikan:

1. teks (text), yaitu (long) string yang panjangnya n karakter
2. pattern, yaitu string dengan panjang m karakter ($m < n$) yang akan dicari di dalam teks.

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). Berikut adalah beberapa contoh dari perumusan pencarian string (dengan kata tebal pada teks merupakan target pencarian).

Contoh 1:

Pattern: hari

Teks: kami pulang **hari** kamis

Contoh 2:

Pattern: not

Teks: nobody **noticed** him

Contoh 3:

Pattern: apa

Teks: **Siapa** yang menjemput Papa dari kota Balikpapan?

1. Algoritma Brute Force

Salah satu teknik pencarian string adalah dengan memanfaatkan algoritma *brute force*. Dengan asumsi 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.

Contoh 1:

Teks: nobody noticed him

Pattern: not

```
nobody noticed him
s=0 not
s=1 not
s=2 not
s=3 not
s=4 not
s=5 not
```

s=6 not
s=7 not

Contoh 2:

Teks: 10010101001011110101010001

Pattern: 001011

10010101001011110101010001
s=0 001011
s=1 001011
s=2 001011
s=3 001011
s=4 001011
s=5 001011
s=6 001011
s=7 001011
s=8 001011

Kompleksitas dari algoritma *brute force* jika kombinasi teks dan pattern merupakan kasus terbaik adalah $O(n)$. Kasus terbaik terjadi jika yaitu bila karakter pertama pattern P tidak pernah sama dengan karakter teks T yang dicocokkan. Pada kasus ini, jumlah perbandingan yang dilakukan paling banyak n kali, misalnya:

Teks: Ini adalah string panjang yang berakhir dengan zz
Pattern: zz

Kasus terburuk membutuhkan $m(n-m+1)$ perbandingan, dimana kompleksitasnya adalah $O(mn)$, misalnya:

Teks: aab
Pattern: aaaab [2]

III. ANALISIS PERMASALAHAN

Permasalahan pada tombol 'like' yang tertanam di setiap status adalah kemungkinan terjadinya kesalahpahaman pada orang yang memberikan 'like' pada status yang tidak layak diberikan 'like', di mana terdapat kemungkinan orang tersebut ingin memberikan respons yang relevan dengan status tersebut tanpa memberikan komentar. Misalkan terdapat sebuah status sebagai berikut :

“Telah meninggal seorang pahlawan tanpa tanda jasa yang telah mendidik saya selama ini”

Respons yang layak terhadap status tersebut adalah respons turut berduka cita. Namun, pada Facebook, tombol 'like' akan tetap muncul pada status tersebut,

sehingga akan terlihat tidak layak dan ofensif jika terdapat orang yang memberikan 'like' pada status tersebut.



Gambar 3. Contoh ketidaklayakan pemberian tombol 'like' pada status dengan suasana duka cita [1]

Suasana lain yang kurang merepresentasikan respons jika diberikan 'like' adalah status yang memberikan pertanyaan ya atau tidak. Tombol 'like' tidak mampu menjawab apa yang status tersebut pertanyakan. Misalkan terdapat status sebagai berikut :

“Apakah kamu setuju jika harga BBM kembali dinaikkan?”

Orang yang hanya ingin memberikan jawaban ya atau tidak tanpa berkomentar tetap harus membeberikan opininya pada komentar, karena tombol 'like' tidak menjawab apakah orang tersebut setuju harga BBM naik atau tidak. Kalaupun orang tersebut memberikan 'like' karena setuju, tidak ada tombol yang merepresentasikan respons ketidaksetujuan atas kenaikan harga BBM.

Permasalahan tersebut dapat diselesaikan dengan membuat tombol alternatif yang menggantikan tombol 'like'. Misalkan untuk berita duka dan status sedih dibuat tombol 'sympathize' yang merepresentasikan duka cita, dan untuk pertanyaan dibuat tombol 'yes' dan 'no' yang akan memberikan jawaban tanpa menambahkan komentar. Namun, tombol-tombol tersebut harus dapat muncul hanya jika tombol tersebut relevan terhadap status sumber. Penambahan tombol-tombol tersebut pada setiap status tanpa mempertimbangkan konteks dari status tersebut hanya akan menambah kemungkinan respons yang tidak layak. Misalkan jika pada status berkonteks kesenangan terdapat tombol 'sympathize', orang lain dapat memberikan 'condolences' pada kesenangan tersebut yang akan memberikan efek kebalikan dari pemberian 'like' pada berita duka. Pemberian respons 'yes' pada berita duka pun tidak mencerminkan respons yang valid terhadap status tersebut. Oleh karena itu, dibutuhkan proses untuk menentukan konteks dari status secara otomatis, sehingga setiap tombol dapat ditanamkan pada status yang layak menerima respons representasi tombol tersebut.

IV. IMPLEMENTASI

A. Penentuan Keyword Konteks Status

Pada setiap jenis respons, terdapat kata-kata tertentu yang dapat menentukan apakah konteks dari sebuah status. Kata-kata ini akan menjadi *keyword* untuk menentukan tombol respons mana yang akan ditanamkan pada status tertentu. Kategori status dapat dibagi menjadi tiga jenis, yaitu :

1. Status berkonteks kesedihan

Status yang memiliki konteks kesedihan pada umumnya mencerminkan curahan kesedihan pembuat status tersebut dan meminta orang lain untuk bersimpati terhadap status tersebut. Contoh status berkonteks kesedihan adalah status seseorang berduka karena adanya kerabat yang meninggal atau status seseorang kehilangan sesuatu. Status berkonteks kesedihan nantinya hanya akan ditanamkan tombol respons 'sympathize' dan tombol 'like' akan dihilangkan. Beberapa frase *keyword* yang dapat diasosiasikan dengan konteks kesedihan mencakup 'RIP', 'rest in peace', 'telah meninggal dunia', 'Innalillahi', 'saya kehilangan', dan 'saya merasa sedih'.

2. Status berkonteks pertanyaan pilihan

Status yang memiliki konteks pertanyaan pilihan pada umumnya mencerminkan pertanyaan dari pembuat status yang berharap mendapatkan jawaban 'ya' atau 'tidak' sebagai respons dari status tersebut. Contoh dari status berkonteks pertanyaan pilihan adalah pertanyaan persetujuan terhadap opini atau pertanyaan penawaran produk. Status berkonteks pertanyaan pilihan nantinya akan ditanamkan tombol respons 'ya' dan 'tidak' dan tombol 'like' akan dihilangkan. Beberapa frase *keyword* yang dapat diasosiasikan dengan konteks kesedihan mencakup 'apakah kamu setuju?', 'apakah kamu ingin?', 'iya atau tidak?', dan 'apakah Anda puas?'.

3. Status berkonteks standar

Status berkonteks standar adalah status yang tidak memiliki konteks kesedihan dan konteks pertanyaan pilihan, sehingga respons 'like' tidak akan diganti karena kemungkinan besar respons 'like' masih relevan dengan status berkonteks standar.

Keyword ini akan digunakan pada algoritma *string matching* sebagai pattern yang akan dicocokkan dengan teks berupa status.

B. Algoritma String Matching pada Penentuan Konteks

Algoritma *string matching* akan digunakan untuk menentukan konteks dari suatu status dan menentukan

tombol respons apa yang akan ditambahkan pada status tersebut. Pada setiap penentuan konteks status, elemen yang dipakai untuk pengecekan adalah :

Text : String yang diambil dari status yang ingin dikategorikan

Pattern : *Keyword* yang diasosiasikan dengan salah satu konteks

Sebagai contoh, terdapat status sebagai berikut :

"Rest in peace, my love"

Misalkan terdapat *keyword* 'rest in peace' di baris pertama pada basis data *keyword* konteks kesedihan. *Keyword* tersebut akan dipisah menjadi satu kata, misalkan 'rest in peace' dibagi menjadi 'rest', 'in', dan 'peace', dan ketiganya akan digunakan sebagai pattern untuk menentukan apakah status mengandung frase 'rest in peace' atau tidak. Setiap kata dari frase tersebut mengandung nilai 'true' atau 'false' yang mencerminkan apakah kata tersebut terdapat pada status atau tidak. Status akan terbilang mengandung frase *keyword* hanya jika seluruh kata yang terbentuk dari *keyword* terdapat pada status. Pada contoh akan digunakan algoritma pencarian string dengan *brute force* :

```
REST IN PEACE, MY LOVE
REST
```

Pada pencarian di atas, terdapat kata 'rest' pada status, sehingga nilai dari 'rest' adalah 'true'

```
REST IN PEACE, MY LOVE
IN
```

```
REST IN PEACE, MY LOVE
IN
```

```
REST IN PEACE, MY LOVE
IN
```

```
REST IN PEACE, MY LOVE
IN
```

```
REST IN PEACE, MY LOVE
IN
```

```
REST IN PEACE, MY LOVE
IN
```

Pada pencarian di atas, terdapat kata 'in' pada status, sehingga nilai dari 'in' adalah 'true'

```
REST IN PEACE, MY LOVE
PEACE
```

```
REST IN PEACE, MY LOVE
PEACE
```

```
REST IN PEACE, MY LOVE
```

PEACE

REST IN PEACE, MY LOVE
PEACE

REST IN PEACE, MY LOVE
PEACE

REST IN PEACE, MY LOVE
PEACE

REST IN PEACE, MY LOVE
PEACE

REST IN PEACE, MY LOVE
PEACE

REST IN PEACE, MY LOVE
PEACE

Terdapat kata 'peace' pada status, sehingga kata 'peace' bernilai 'true'. Karena 'rest', 'in' dan 'peace' seluruhnya bernilai 'true' terhadap status, maka dipastikan bahwa status tersebut berkonteks kesedihan, sehingga akan ditanam tombol 'sympathize' pada status tersebut.

C. Analisis Teknik Pemilihan Keyword dan Pattern

Terdapat dua cara untuk memilih *keyword* yang akan dipakai untuk pencocokan :

1. *Keyword* per frase

Pada contoh di atas, digunakan teknik pemilihan *keyword* per frase, yaitu *keyword* dibolehkan terbentuk oleh lebih dari satu kata, seperti 'rest in peace', 'telah meninggal dunia', 'apakah setuju', dan 'apakah anda ingin'. Ada dua cara pencocokan jika *keyword* menggunakan tipe *keyword* per frase, yaitu :

a. Pencocokan langsung satu frase

Pada teknik pencocokan ini, pattern yang dipakai dalam algoritma pencarian adalah satu frase penuh, seperti pada contoh seperti ini :

T = APAKAH ANDA SETUJU?

P = APAKAH SETUJU

Kelemahan dari teknik ini adalah kebutuhan variasi frase yang banyak sebagai *keyword*, karena terdapat kemungkinan sebuah status mengandung frase dalam *keyword*, namun setiap kata dalam frase tersebut terdapat dalam posisi yang tidak sama dengan *keyword*, sehingga untuk mencakup seluruh variasi susunan frase harus ditambahkan banyak frase *keyword* yang mengandung kata-kata yang sama namun memiliki susunan kata yang berbeda.

Pada contoh di atas, jika memakai pencocokan langsung satu frase, "Apakah anda setuju?" tidak akan dikategorikan sebagai status berkonteks pertanyaan pilihan karena meskipun mengandung kata "apakah" dan "setuju", susunannya bukan berupa "apakah setuju".

b. Pemecahan frase berdasarkan kata

Pada teknik pencocokan ini, seperti yang dicontohkan dalam teks "Rest in peace, my love" dan *keyword* "rest in peace", *keyword* yang berupa frase akan dipecah menjadi beberapa kata sesuai dengan jumlah kata yang terdapat pada frase. Misalkan terdapat *keyword* "apakah setuju" sebagai *keyword* status berkonteks pertanyaan pilihan yang akan dicocokkan dengan status "Apakah anda setuju?", pertama *keyword* akan dipecah menjadi "apakah" dan "setuju". Lalu kedua kata tersebut akan dicocokkan dengan status secara terpisah. Kedua pencarian akan menghasilkan nilai 'true', sehingga dapat dipastikan status tersebut merupakan status berkonteks pertanyaan pilihan.

T = APAKAH ANDA SETUJU?

P1 = APAKAH

P2 = SETUJU

Kelemahan dari teknik ini adalah dibutuhkan waktu yang cukup banyak dan kompleksitas yang cukup besar untuk melakukan pemisahan frase, sehingga kompleksitas algoritmanya bertambah.

2. *Keyword* per kata

Pada cara ini, *keyword* yang dipilih hanya mengandung satu kata, seperti "rest" atau "berduka". Teknik pencariannya adalah dengan menambahkan value 'count' untuk setiap kategori khusus pada status, 'count' ini berfungsi sebagai penghitung berapa *keyword* yang cocok pada status. Jika count salah satu kategori telah mencapai angka tertentu, maka status tersebut akan dikategorikan sebagai kategori tersebut. Misalkan pada basis data *keyword* berkonteks kesedihan terdapat kata 'berduka', 'meninggal', 'dunia', dan 'peace' dengan batas minimum 3 kata terdapat pada status agar dikategorikan sebagai status berkonteks kesedihan, maka status "Kami berduka cita, telah meninggal dunia seorang guru" akan dikategorikan sebagai status berkonteks kesedihan.

Namun, terdapat kelemahan juga pada cara ini, yaitu kemungkinan kesalahan pengategorian pada status tertentu. Misalkan terdapat *keyword* 'apakah', 'sejutu', dan 'ingin' untuk kategori status berkonteks pertanyaan pilihan. Status

“Apakah yang membuat saya setuju dengan apa yang dia inginkan?” Akan dikategorikan sebagai status berkonteks pertanyaan pilihan, padahal maksud dari status tersebut adalah meminta opini secara terbuka, tidak hanya ‘ya’ atau ‘tidak’. Selain itu, *keyword* yang dipilih harus benar-benar menekankan konteks setiap kategori, sehingga variasi kosakata yang dapat dipilih lebih terbatas dibandingkan dengan teknik pencocokan langsung satu frase.

V. KESIMPULAN

Kesimpulan dari analisis ini adalah tombol respons ‘like’ yang terdapat pada Facebook dapat digantikan dengan tombol respons lain yang lebih relevan dengan konteks dari status, khususnya pada status berkonteks kesedihan (‘sympathize’) dan status berkonteks pertanyaan pilihan ‘ya’ dan ‘tidak’ (‘yes’ dan ‘no’). Penggantian tersebut dapat dilakukan dengan memanfaatkan algoritma pencarian string dengan berbagai *keyword* yang menekankan konteks kesedihan dan konteks pertanyaan pilihan sebagai pattern untuk dicocokkan dengan status. Pemilihan *keyword* tersebut memiliki beberapa pertimbangan terhadap kompleksitas algoritma, banyaknya daftar *keyword* tiap kategori, dan akurasi pencarian.

REFERENSI

- [1] <http://www.facebook.com>
- [2] Munir, Rinaldi, 2004, IF2251 Strategi Algoritmik – Algoritma Pencarian String (*String Matching*), Bandung : Institut Teknologi Bandung

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, 20 Desember 2013

ttd



Raden Fajar Hadria Putra -13511076