

Penerapan Algoritma String Matching pada Rekomendasi Iklan di Media Sosial

Muhammad Fawwaz Naabigh (13519206)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
fnaabigh@gmail.com

Abstract– Iklan merupakan salah satu cara yang dilakukan oleh pemilik produk untuk menggapai konsumen sebanyak mungkin. Namun, seiring berjalannya waktu dan berkembangnya teknologi semakin banyak pula alternatif yang dapat dilakukan dalam melakukan pengiklanan. Salah satu cara adalah melakukan iklan bertarget atau *targeted ad* pada media sosial. Iklan bertarget bertujuan meningkatkan keefektifan dalam proses promosi dengan hanya mengiklankan pada orang-orang yang relevan. Pencarian kerelevanan tersebut, dilakukan dengan melakukan pencocokan dengan algoritma *String Matching*.

Kata kunci– iklan bertarget, *targeted ad*, *string matching*

I. PENDAHULUAN

Iklan atau *advertising* merupakan istilah yang berasal dari bahasa latin, yaitu *ad-yere* yang bermakna mengoperkan pikiran dan gagasan kepada pihak lain. Menurut Kotler (2005) yang dimaksud dengan iklan ialah segala bentuk presentasi non-pribadi dan promosi gagasan, barang, atau jasa oleh sponsor tertentu yang harus dibayar. Berbeda dengan Djaslim Saladin (2002), beliau mendefinisikan periklanan sebagai salah satu alat promosi, biasanya digunakan untuk mengarahkan komunikasi persuasif pada pembeli sasaran dan masyarakat dengan bentuk penyajian iklan ini bersifat non-personal.

Pesan komersial sebenarnya sudah ada sejak bangsa arab kuno. Hal ini dibuktikan dengan bukti sejarah yang menunjukkan bahwa orang-orang mesir kuno menggunakan papirus untuk membuat pengumuman mengenai barang-barang yang dijual serta membuat poster dinding. Bentuk iklan ini kemudian semakin berkembang semenjak mesin cetak ditemukan oleh Gutenberg pada tahun 1450. Pada masa ini, bentuk media periklanan berupa poster, selebaran, serta iklan baris surat kabar.

Semakin berkembangnya zaman secara tidak langsung membuat teknologi semakin berkembang pesat, begitu juga dengan perkembangan iklan. Hal ini kemudian yang menyebabkan iklan tidak hanya terbatas pada poster atau selebaran, melainkan terus berkembang ke ranah media publikasi lainnya, seperti radio dan televisi semenjak tahun 1950. Akan tetapi, perkembangan tersebut tidak berhenti sampai di situ. Pada tahun 1990, kemunculan internet membuka batas baru bagi periklanan. Sejak saat itu, iklan digital lahir dan menjadi opsi baru dalam dunia periklanan.

Internet sebagai media yang memudahkan penggunanya untuk mengakses dan menemukan segala macam informasi sampai ke seluruh dunia menembus batas-batas geografis. Melalui itu, internet sangat sulit dipisahkan dari kehidupan manusia modern dan bahkan menjadi kebutuhan. Keberadaan internet yang sudah menjadi bagian dari kehidupan manusia, menyebabkan berbagai inovasi yang tak terbayangkan sebelumnya, termasuk dari ranah iklan. Iklan tidak lagi terbatas pada media tradisional, tetapi terus merambah ke dunia digital, salah satunya melalui media sosial. Media sosial dengan daya tarik tersendiri membuat beberapa penggunanya rela menghabiskan waktu berjam-jam. Media sosial memudahkan penggunanya untuk berkomunikasi, bertukar pikiran, memberikan informasi dalam bentuk apapun yang dapat disebarluaskan dengan mudah dan cepat, sehingga mempengaruhi acra pandang, gaya hidup, serta budaya suatu bangsa. Dengan demikian, sangatlah efektif untuk menjadikan media sosial sebagai media periklanan digital (Kristiani, N. 2017). Tidak sampai disitu saja, keberadaan teknologi dapat memudahkan iklan yang dulunya mungkin hanya bersifat informasi untuk semua orang, menjadi lebih spesifik dan dapat langsung menuju pada calon konsumen yang relevan. Dengan demikian, kerja iklan menjadi bentuk promosi yang efektif.

II. LANDASAN TEORI

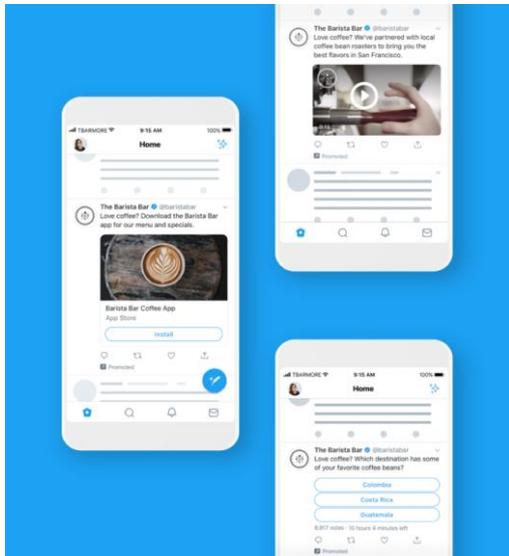
A. *Targeted Advertising*

Iklan di media sosial seperti Instagram bekerja secara lebih kompleks ketimbang iklan tradisional yang ada pada tv atau selebaran dengan tingkat keefektifan yang lebih tinggi. Iklan tradisional bertujuan menarget audiens sebanyak mungkin dengan harapan meningkatkan kesadaran audiens mengenai sebuah produk. Sedangkan, pada iklan yang ada pada media sosial, iklan tidak lagi mengejar jumlah audiens yang banyak, tetapi ditujukan untuk mengejar audiens yang relevan. Sehingga, dengan begitu diharapkan tingkat ketertarikan calon pembeli akan meningkat sebab iklan tersebut relevan dengan audiens.

Iklan dengan konsep demikian disebut dengan *targeted ad* atau iklan bertarget. Sesuai namanya, iklan tersebut menarget audiens tertentu. Tujuan utama dari *targeted ad* adalah meningkatkan calon pembeli ketimbang meningkatkan

awareness sebanyak mungkin sebagaimana iklan tradisional pada radio dan tv.

Cara kerja *targeted ad* mengetahui siapa pengguna yang relevan adalah dengan mengumpulkan data dari pengguna. Data tersebut dapat berupa riwayat pencarian, ketertarikan pengguna berdasar pengguna lain yang dia ikuti, atau melalui unggahan yang disukai oleh pengguna. Tetapi, terdapat kontroversi mengenai metode ini, sebab pengumpulan data semacam itu dianggap dapat menyalahi privasi pengguna.



Gambar 1. Targeted ad pada Twitter.com ditandai dengan tanda “Promoted” pada bagian bawah
Sumber: business.twitter.com

B. Algoritma String Matching

Algoritma *String Matching* merupakan algoritma yang digunakan untuk mencari sebuah pola P, pada teks T. Sebab itu, algoritma *String Matching* juga disebut *Pattern Matching*. Pola P yang akan dicari harus memiliki panjang yang kurang dari atau sama dengan panjang yang dimiliki oleh teks T.

Tabel 2.1. Contoh *string matching*

T:	kasur	ini	rusak
P:	rusak		

Penggunaan *String Matching* dalam kehidupan sehari-hari dapat dijumpai pada berabagai aplikasi. Misalnya, pada saat membaca buku elektronik, seringkali pembaca ingin mencari kata kunci tertentu. Pencarian kata kunci tersebut dapat menggunakan algoritma *String Matching* yang tentu saja perlu dilakukan penyesuaian agar pencarian dapat berjalan dengan cepat.

Contoh lain yang cukup menarik adalah pencocokan Rantai Asam Amino pada DNA. Pada DNA, terdapat pilinan ganda yang berbentuk seperti tangga dan tersusun atas molekul-molekul asam nukleat. Struktur tersebut disebut dengan pilinan

ganda, karena tersusun atas sepasang rantai yang terpilin. Rantai tersebut direpresentasikan dengan 4 huruf yang dapat disusun menjadi 64 kata dan 21 makna. Setiap kata hanya dapat berpasangan dengan kata khusus. Proses *String Matching* pada DNA, terjadi pada saat proses translasi mRNA menjadi tRNA yang kemudian menjadi rantai protein. *String Matching* memastikan pada saat translasi, tidak ada kata yang berpasangan dengan kata yang bukan pasangannya.

C. Algoritma Brute Force

Algoritma ini melakukan pencarian pola dengan memastikan bawa setiap huruf pada suatu teks cocok dengan pola yang dicari. Proses pencarian pola dilakukan dengan melakukan pencocokan dari kiri ke kanan dengan penggeseran satu persatu hingga ditemukan pola yang dicari. Penggeseran akan terus dilakukan selama pola yang dicari belum ditemukan atau selama teks belum mencapai akhir.

Kondisi paling menguntungkan terjadi ketika pola yang dicari berada pada bagian awal teks. Sementara itu, kemungkinan terburuk terjadi bila pola yang dicari berada di bagian akhir dari teks. Selain itu, algoritma ini dapat diandalkan pada teks yang memiliki jumlah alfabet yang cukup banyak, seperti huruf latin dengan alfabet yang berjumlah 26, dan tidak begitu dapat diandalkan untuk teks dengan jumlah alfabet yang relatif lebih sedikit, seperti pada DNA yang hanya memiliki 4 huruf atau pada teks binary yang hanya terdiri dari 2 huruf.

Tabel 2.2 Contoh *string matching* dengan Brute Force

T	S	t	r	a	t	e	g	i		a	l	g	o	r	i	t	m	a
P	a	l	g	o														
	S	t	r	a	t	e	g	i		a	l	g	o	r	i	t	m	a
	a	l	g	o														
		a	l	g	o													
			a	l	g	o												
				a	l	g	o											
					a	l	g	o										
						a	l	g	o									
							a	l	g	o								
								a	l	g	o							
									a	l	g	o						
										a	l	g	o					
											a	l	g	o				
												a	l	g	o			

D. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt (KMP), sesuai namanya, ditemukan oleh Donald E. Knuth. Algoritma ini merupakan salah satu algoritma *string matching* yang mencari pola dengan melakukan pencocokan dari kiri ke kanan. Meski sama-sama melakukan pencocokan dari kiri ke kanan, berbeda dengan Brute Force, algoritma KMP melakukan penggeseran dengan lebih efektif dan terkesan lebih “cerdas”.

Jika terjadi ketidakcocokan saat proses pencocokan pola P di teks T pada P[j] dan T[i], maka penggeseran dilakukan pada

prefiks terbesar pada $P[0..j-1]$ yang merupakan sufiks dari $P[1..j-1]$.

Tabel 2.3 Contoh algoritma KMP

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	a	b	a	c	a	b	a	b	a	a	b	c	b	a
P	a	b	a	a	b									
			a	b	a	a	b							
				a	b	a	a	b						
						a	b	a	a	b				

Banyak penggeseran yang dilakukan dapat dihitung menggunakan Fungsi Pinggiran KMP (*KMP Border Function*). Fungsi pinggiran melakukan praproses pada pola untuk mencari kecocokan pada prefiks pada pola yang dimiliki oleh pola itu sendiri.

Jika diambil,

- j = posisi ketidakcocokan pada $P[j]$
- k = posisi pas sebelum ketidakcocokan pada $P[j]$ ($k = j-1$)

maka fungsi pinggiran $b(k)$ didefinisikan sebagai ukuran prefiks terbesar dari $P[0..k]$ yang juga merupakan sufiks dari $P[1..k]$.

Tabel 2.4 Contoh fungsi pinggiran

j	0	1	2	3	4	5	6	7
$P[j]$	b	a	b	a	a	b	a	b
k	-	0	1	2	3	4	5	6
$b(k)$	-	0	0	1	2	0	0	2

Dengan menggunakan algoritma KMP, proses pencarian cenderung lebih cepat jika dibandingkan dengan algoritma *brute force*. Kelebihan dari algoritma KMP adalah bahwa algoritma KMP tidak memerlukan bergerak mundur untuk sebuah input. Oleh karena itu, algoritma ini dapat cukup diandalkan untuk proses pencarian pola pada file dengan ukuran besar atau file yang dibaca melalui gawai eksternal.

Di sisi lain, algoritma KMP tidak bekerja terlalu baik seiring bertambahnya jumlah alfabet. Hal ini berkebalikan dengan algoritma *brute force* yang justru tidak begitu optimal pada teks dengan alfabet yang sedikit. Kondisi tersebut terjadi sebab ketidakcocokan antara pola dan teks akan semakin sering terjadi seiring bertambahnya alfabet.

E. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma *string matching* yang dilandaskan pada dua teknik. Teknik pertama disebut dengan teknik *looking-glass*. Teknik tersebut mencari

pola P pada teks T dimulai dari huruf terakhir pada P dengan bergerak mundur melalui P . Teknik kedua adalah teknik *character-jump*. Dengan teknik tersebut, dilakukan *jump* atau lompatan ketika ditemukan ketidakcocokan pada $T[i]=x$, dengan kata lain huruf atau karakter pada pola $P[j]$ tidak sama dengan yang ada pada $T[i]$.

Selain terdapat dua teknik, pada algoritma ini terdapat tiga kemungkinan kasus atau kondisi yang dapat terjadi. Kondisi pertama adalah kondisi ketika terdapat karakter x pada P dan akan dilakukan penggeseran pada P sesuai dengan kemunculan terakhir karakter x dengan $T[i]$.

Tabel 2.5 Contoh kasus 1

T			x	b	a	c	b
P	x	a	b	b			
			x	a	b	b	

Selanjutnya, kasus kedua terjadi ketika terdapat karakter x pada P , tetapi penggeseran sesuai dengan kemunculan terakhir tidak mungkin dilakukan, sehingga penggeseran P akan dilakukan sebanyak 1 karakter ke kanan menuju $T[i+1]$.

Tabel 2.6 Contoh kasus 2

T			x	b	a	c	b
P		b	b	x			
			b	b	x		

Sementara itu, kasus ketiga terjadi bila kasus pertama atau kedua tidak dapat dipenuhi. Jika kondisi ini dijumpai, maka penggeseran P dilakukan agar $P[0]$ sejajar dengan $T[i+1]$.

Tabel 2.7 Contoh kasus 3

T			x	b	a	c	b
P	b	a	b	b			
				b	a	b	b

Sebagaimana KMP, algoritma Boyer-Moore juga memiliki fungsi praproses yang disebut dengan *Last Occurrence Function* ($L()$). $L()$ memetakan seluruh karakter pada pola P ke dalam bilangan bulat sesuai dengan kemunculan terakhirnya pada P . Jika suatu karakter tidak ada pada P , maka $L()$ bernilai -1 . Fungsi ini akan dijalankan ketika pola P sudah ditentukan.

Algoritma Boyer-Moore akan bekerja lebih optimal untuk teks dengan alfabet yang banyak, ketimbang teks dengan alfabet yang sedikit. Hal ini berkebalikan dengan algoritma KMP dan bersesuaian dengan algoritma *brute force*, namun dengan kecepatan yang lebih tinggi.

Tabel 2.8 Contoh *Last Occurrence Function*

	0	1	2	3	4
P	a	b	a	c	b
x	a	b	c	d	
L(X)	2	1	3	-1	

Tabel 2.9 Contoh Algoritma Boyer-Moore

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
T	a	b	a	c	c	b	a	b	a	a	b	c	b	a
P	a	b	a	a	b									
					a	b	a	a	b					
						a	b	a	a	b				

III. PEMBAHASAN

A. Mengumpulkan Cuitan

Media sosial yang dipilih adalah Twitter dengan pertimbangan bahwa Twitter merupakan media sosial berbasis mikroblog. Artinya, konten utama dari Twitter berupa tulisan, sehingga sangat cocok dengan kebutuhan dan batasan pada eksperimen kali ini, yaitu *string matching*. Berbeda dengan Instagram yang didominasi oleh foto dan video. Proses pertama yang dilakukan adalah melakukan ekstraksi cuitan Twitter dari orang yang dituju. Cuitan yang diekstrak diambil dari akun @ainunnajib yang diunggah pada tanggal 17 Maret 2021.



Gambar 3.1 Cuitan yang diekstrak

Sumber: twitter.com/ainunnajib

B. Perancangan Program

Algoritma yang digunakan adalah algoritma Boyer-Moore dengan pertimbangan bahwa algoritma tersebut relatif lebih efektif ketimbang algoritma *Brute Force* maupun KMP. Algoritma KMP diketahui kurang efektif untuk teks dengan alfabet yang banyak. Mengingat teks dan pola yang dicari ditulis dengan bahasa Indonesia yang memiliki alfabet sebanyak 26, kandidat algoritma yang mungkin digunakan adalah *brute force* dan Boyer-Moore. Tetapi, sudah diketahui bahwa Boyer-Moore jauh lebih efektif ketimbang *brute force*.

Sementara itu, implementasi program ditulis dengan bahasa Python. Implementasi yang dibuat adalah fungsi *Last Occurrence Function* dan fungsi algoritma Boyer-Moore itu sendiri. Program fungsi L() mengembalikan *dictionary* dengan *key* merupakan huruf yang berkaitan, dan *value* berupa kemunculan terakhir dari huruf tersebut. Sementara itu, fungsi BM() akan mengembalikan -1 bila tidak ditemukan pola pada teks yang dicari, sebaliknya akan mengembalikan indeks huruf pertama dari pola yang dicari pada teks.

```

1 def L(pattern, text):
2     x={}
3     for i in range(len(text)):
4         x[text[i]] = -1
5     for i in range(len(pattern)):
6         x[pattern[i]] = i
7     return x

```

Gambar 3.2 Implementasi *Last Occurrence Function*

Sumber: Penulis

```

1 def BM(text, pattern):
2     last = L(text,pattern)
3
4     n = len(text)
5     m = len(pattern)
6     i = m-1
7
8     if(i>n-1):
9         return -1
10
11    j = m-1
12
13    while True:
14        if (text[i] == pattern[j]):
15            if(j == 0):
16                return i
17            else:
18                i -= 1
19                j -= 1
20        else:
21            lo = last[text[i]]
22            i = i + m - min(j, 1+lo)
23            j = m - 1
24
25        if(i>(n-1)):
26            break;
27
28    return -1

```

Gambar 3.3 Implmentasi algoritma Boyer-Moore

Sumber: Penulis

Di samping itu, telah disediakan sebuah *dictionary* dengan *key* berupa id iklan dan *value* berupa kata kunci yang relevan terhadap konteks iklan yang bersesuaian dengan id tersebut. *Dictionary* tersebut diberi nama “*keywords*”. Kemudian, proses dilakukan dengan melakukan iterasi pada setiap *value* di seluruh *key* pada *keywords*. Setiap *keyword* tersebut dijadikan sebagai pola, dan cuitan yang diekstrak sebagai teks yang dicari polanya. Pola dan teks tersebutlah yang akan dijadikan sebagai masukan pada fungsi BM().

Apabila ditemukan sebuah kata yang cocok, maka fungsi BM() akan mengembalikan indeks huruf pertama dari kata yang berseuaian. Sebaliknya, akan dikembalikan nilai -1 bila tidak ditemukan kata yang cocok. Setelah didapat id iklan dari hasil pencocokan, id tersebut ditampung pada sebuah variabel. Selanjutnya, diambil seluruh iklan dengan id yang didapat. Iklan yang didapat tersebut merupakan hasil rekomendasi yang didapat.

Tabel 3.1 *Dictionary keywords* dan iklan.

```

iklan = {
    1: "masker",
    2: "face shield",
    3: "mobil",
    4: "nastar"
}

```

```

}
keywords = {
    1: ['corona', 'covid', 'flu'],
    2: ['corona', 'covid', 'flu'],
    3: ['mobil', 'kendaraan'],
    4: ['lebaran', 'puasa', 'idul fitri']
}

```

C. Percobaan

Setelah didapat cuitan dari pengguna yang dituju, cuitan tersebut dapat langsung dimasukkan pada program yang telah dibuat.

```

@ainunnajib:
potensi pengobatan covid-19 dengan getaran?
https://news.mit.edu/2021/ultrasound-coronaviruses-damage-0316

Rekomendasi iklan:
1 masker
2 face shield

```

Gambar 3.4 Hasil percobaan

Sumber: Penulis

Setelah program dijalankan, ditampilkan keluaran berupa cuitan dan rekomendasi iklan yang bersesuaian. Terlihat pada gambar 3.4 cuitan yang diekstrak memiliki rekomendasi iklan berupa iklan produk masker dan *face shield*. Hasil itu didapat dari data bahwa, iklan masker dan *face shield* memiliki keyword “covid” yang juga terdapat pada cuitan yang diekstrak.

Dengan demikian, dapat ditampilkan iklan sesuai hasil rekomendasi. Rekomendasi tersebut diharapkan dapat meningkatkan potensi calon konsumen, yang dalam hal ini adalah pemilik akun @ainunnajib, untuk membeli produk masker atau *face shield* dari pihak pengiklan. Dengan cara ini, pihak pengiklan dapat menghemat biaya sponsor agar tidak perlu memasang iklan kepada orang yang memang sedari awal tidak memiliki ketertarikan pada produk mereka. Atau dalam hal ini, pengiklan masker atau *face shield* dapat menghemat biaya dari mengiklankan produk kepada orang-orang yang tidak percaya terhadap Covid-19, sebab orang-orang tersebut tidak akan tertarik untuk membeli produk mereka.

IV. SIMPULAN DAN SARAN

Perekomendasi iklan sebagai implementasi *targeted advertising* dapat dilakukan dengan sederhana melalui algoritma pencocokan *string* khususnya pada media sosial yang berbasis teks, seperti Twitter. Kelebihan dari metode ini adalah bahwa tidak ada batasan privasi yang dilanggar sebagaimana perekomendasi iklan yang sekarang sering dipakai. Namun, dengan dipakainya metode ini, dibutuhkan riset dan proses pengembangan program yang jauh lebih lama untuk megumpulkan *keyword* yang bersesuaian sebanyak mungkin agar program benar-benar efektif.

LINK VIDEO YOUTUBE

https://youtu.be/_DSPZ80dHoA

REFERENCES

- [1] Kotler, Philip. 2005. Manajemen Pemasaran. Jilid 1 dan 2. Jakarta: PT Indeks Kelompok Gramedia.
- [2] Djaslim, Saladin. 2002. Manajemen Pemasaran: Analisis, Pelaksanaan, dan Pengendalian. Bandung: Linda Karya.
- [3] Dawkins, Richard. 2017. The Ancestor's Tale: A Pilgrimage to the Dawn of Life. London: Weidenfeld & Nicolson.
- [4] Supriyanto, Sugeng. 2008. Meraih Untung dari Spanduk hingga *Billboard*. Yogyakarta: Pustaka Girhatama.
- [5] Kristiani, Nunung. 2007. Analisa Pengaruh Iklan di Media Sosial dan Jenis Media Sosial terhadap Pembentukan Perilaku Konsumtif Mahasiswa Di Yogyakarta. Yogyakarta: Juran Bisnis dan Ekonomi (JBE).
- [6] Munir, Rinaldi. 2021. Pencocokan String (String/Pattern Matching). <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>. Diakses pada 10 Mei 2021.

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.

Gresik, 11 Mei 2021



Muhammmad Fawwaz Naabigh 13519206