

Pattern Matching pada Report View di Google Analytics

Nella Zabrina Pramata 13516025

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Bandung, Indonesia
sandhita.nellazp@gmail.com

Abstract—Bisnis merupakan kegiatan pembukaan usaha komersial dalam bidang perdagangan. Pada masa sekarang, bisnis sering dilakukan secara *online* dengan menggunakan platform yang terkoneksi internet. Dalam upaya untuk membuat bisnis berkembang, pebisnis perlu melakukan analisis terhadap laporan platform yang digunakan dalam memasarkan produk maupun jasanya. Google Analytics merupakan platform yang dapat membantu pebisnis dalam mengumpulkan data dan menampilkan olahan data dalam bentuk laporan. Akan tetapi, analisis perlu melihat laporan yang lebih spesifik. Oleh karena itu, perlu menggunakan pencarian keyword dengan menggunakan pattern matching sehingga dapat menampilkan laporan yang lebih spesifik.

Keywords—Google Analytics, keyword, report, pattern matching

I. PENDAHULUAN

Bisnis merupakan kegiatan seseorang atau sekelompok orang untuk membuka usaha komersial dalam bidang perdagangan. Dalam arti luas, bisnis merupakan aktivitas dan institusi yang memproduksi barang maupun jasa dalam kehidupan sehari-hari^[1]. Kegiatan ini sering dilakukan dengan cara membuka toko di rumah maupun menyewa suatu tempat atau dengan cara menawarkan produk dagangannya ke orang lain.

Seiring perkembangan zaman, bisnis yang dilakukan secara *door-to door* maupun membuka toko di suatu tempat dirasa kurang menguntungkan karena memiliki pelanggan yang terbatas. Dari permasalahan itu, muncul ide untuk membuka bisnis secara *online*. Bisnis secara online dianggap lebih menguntungkan karena pelanggan dari jauh dapat melihat dan memesan produk pebisnis. Selain itu, membuka suatu toko bukanlah keharusan karena dengan bisnis *online*, produk dapat disimpan di rumah. Hal lain yang membuat orang berlomba-lomba untuk membuka bisnis *online* adalah bisnis ini dapat dilakukan tanpa menggunakan jam kerja *nine-to-five*, sehingga bisnis ini memiliki kelonggaran waktu.

Dalam membuka bisnis *online*, pemilik bisnis akan menggunakan website maupun platform lain yang terkoneksi internet untuk memasarkan produk maupun jasanya. Namun, dengan menggunakan website saja, pebisnis tidak dapat mengetahui bagaimana kondisi orang yang mengunjungi

website tersebut. Dari kondisi ini, pebisnis mengalami kesulitan dalam memperbaiki pelayanan. Oleh karena itu, pebisnis perlu menganalisa website maupun platform yang digunakan untuk menawarkan dagangannya.

Berdasarkan permasalahan tersebut, Google membuat suatu platform yang membantu para pebisnis dengan membuat Google Analytics. Google Analytics digunakan untuk mengumpulkan data dari website dan mengolahnya sehingga dapat ditampilkan dalam bentuk laporan. Laporan ini kemudian digunakan oleh pebisnis untuk dianalisa. Namun, laporan terdiri dari banyak data sehingga analisis akan mengalami kesulitan dalam mendapatkan informasi penting. Oleh karena itu, perlu dilakukan pencarian keyword yang menggunakan pattern matching sehingga analisis dapat membaca report view sesuai dengan informasi spesifik yang dibutuhkan.

Pattern matching terdiri dari banyak metode seperti dengan menggunakan algoritma Brute Force, algoritma Knuth–Morris–Pratt, algoritma Boyer Moore, maupun Regular Expressions. Semua metode ini memiliki kelebihan dan kekurangan. Maka dari itu, penulis membuat makalah untuk menganalisa pattern matching yang berjudul “Pattern Matching pada Report View di Google Analytics”.

II. LANDASAN TEORI

A. Algoritma Brute Force

Algoritma brute force atau algoritma naive adalah algoritma yang *straightforward* untuk memecahkan permasalahan. Algoritma ini memecahkan permasalahan dengan cara yang sangat sederhana dan jelas. Karakteristik algoritma Brute Force adalah

- Algoritma ini umumnya tidak cerdas dan tidak mangkus karena menggunakan jumlah langkah yang besar untuk menyelesaikan permasalahan.
- Seringkali merupakan pilihan yang kurang disukai karena tidak mangkus.
- Untuk masalah kecil, ketidakmangkusannya brute kurang diperhitungkan karena algoritma ini sederhana.

- Algoritma brute force sering digunakan sebagai basis untuk membandingkan dengan algoritma alternatif yang lebih mangkus.
- Hampir semua permasalahan dapat dipecahkan dengan algoritma brute force.
- Algoritma ini lebih mudah diimplementasikan daripada algoritma yang lebih canggih karena sederhana. Kadang, algoritma ini lebih mangkus jika ditinjau dari segi implementasi.

Metode brute force dalam *Pattern Matching* yaitu ^[3]

1. Mula-mula pattern dicocokkan pada awal teks.
2. Tiap karakter di dalam pattern dibandingkan dengan karakter yang sesuai di dalam teks dengan digerakkan dari kiri ke kanan sampai
 - a. Semua karakter yang dibandingkan sama sehingga pencarian berhasil
 - b. Terdapat karakter yang tidak cocok sehingga pencarian belum berhasil
3. Bila pattern belum ditemukan kecocokkannya dan teks belum habis, maka geser pattern satu karakter ke kanan dan ulangi dari langkah ke-2.

Pattern: NOT

Teks: NOBODY NOTICED HIM

```

NOBODY NOTICED HIM
1 NOT
2  NOT
3   NOT
4    NOT
5     NOT
6      NOT
7       NOT
4. 8        NOT

```

Gambar 1. *Pattern Matching Menggunakan Algoritma Brute Force*
 Sumber. Diklat IF2251 Strategi Algoritmik "Algoritma Brute Force (Lanjutan) dan Heuristik".

Pada algoritma brute force untuk *Pattern Matching* memiliki kompleksitas kasus terbaik $O(n)$. Kasus ini terjadi bila karakter pertama pattern P tidak pernah sama dengan karakter teks T yang dicocokkan. Sedangkan untuk kasus terburuk membutuhkan $O(mn)$. Hal ini terjadi saat hanya karakter terakhir pada pattern yang berbeda dari teks dan pattern tersebut ditemukan di akhir teks sehingga pencocokan selalu dilakukan hingga akhir pattern.

B. Algoritma Knuth–Morris–Pratt

Algoritma Knuth Morris Pratt merupakan algoritma yang mencocokkan pattern dengan teks dari kiri ke kanan seperti Brute Force. Algoritma ini menyimpan informasi yang tidak disimpan oleh algoritma Brute Force sehingga kompleksitas

waktunya sebesar $O(m+n)$. Informasi tersebut digunakan untuk melakukan jumlah pergeseran sehingga tidak menggeser satu karakter saja seperti pada algoritma Brute Force.

Algoritma ini menggunakan fungsi pinggiran (*border function*) yang memanfaatkan informasi prefix dan suffix.

Prefix merupakan awalan dari pattern P, dengan u sebagai upastring. P diawali dengan u. u diambil dari $P[0..j-1]$.

Suffix merupakan akhiran dari pattern P, dengan v sebagai upastring. P diakhiri dengan v. v diambil dari $P[1..j-1]$.

Fungsi pinggiran, atau border function sering disebut juga *failure function*. Pinggiran dari P adalah upastring yang kedua awalan dan akhiran sebenarnya dari P. Fungsi ini $b(j)$ didefinisikan sebagai ukuran terpanjang dari P yang merupakan akhiran dari $P[1..j]$. Apabila ada $b(4)$, maka artinya adalah prefix terbesar pada $P[0..4]$ dengan suffix $P[1..4]$.

Contoh perhitungan fungsi pinggiran pada

Teks: abcabcabd

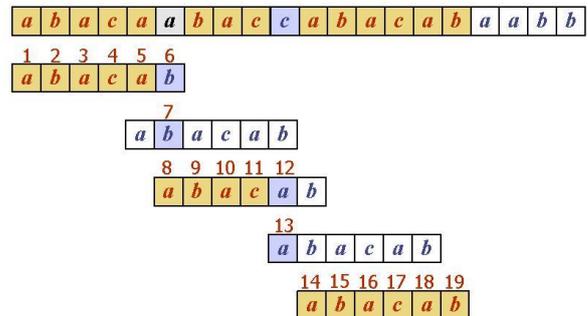
Pattern: abcabd

<i>j</i>	1	2	3	4	5	6
<i>P[j]</i>	a	b	c	a	b	d
<i>b(j)</i>	0	0	0	1	2	0

Gambar 2. Perhitungan Fungsi Pinggiran

Sumber. Diklat IF2251 Strategi Algoritmik "Algoritma Pencarian String (String Matching)".

Setelah menghitung fungsi pinggiran, maka teks dicocokkan dengan pattern. Apabila pattern tidak cocok dengan teks maka pattern digeser hingga indeks $b(k)$ dari pattern sebelumnya.



<i>x</i>	0	1	2	3	4	5
<i>P[x]</i>	a	b	a	c	a	b
<i>f(x)</i>	0	0	1	0	1	2

Gambar 3. *Pattern Matching Menggunakan Algoritma Knuth Morris Pratt*
 Sumber. <http://i-satyam.blogspot.co.id/2015/12/knuth-morris-pratt-algorithm.html>

Kompleksitas waktu untuk menghitung fungsi pinggiran adalah $O(m)$. Sedangkan untuk pencarian string membutuhkan waktu $O(n)$. Sehingga, waktu algoritma KMP adalah $O(m+n)$.

Biasanya algoritma ini cepat bila digunakan untuk ukuran alfabet kecil seperti biner. Semakin besar ukuran alfabetnya, maka semakin besar kemungkinan ketidakcocokkannya.

C. Algoritma Boyer Moore

Algoritma Boyer Moore adalah pencocokan string dimulai dari belakang pattern. Algoritma ini akan bekerja dengan cepat apabila alfabet yang digunakan besar ukurannya (beragam) dan

pattern yang digunakan relative panjang. Pencocokan ini didasari dengan dua teknik, yaitu

1. Teknik Looking-Glass

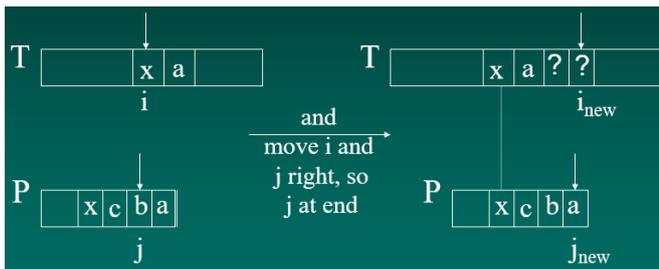
Pencocokan dengan teknik looking-glass dilakukan dengan cara memindahkan pattern string dari alfabet terbelakang kemudian mundur sampai di alfabet terdepan dalam teks.

2. Teknik Character Jump

Teknik character-jump dilakukan ketika huruf pada teks $T[i] \neq P[j]$. Teknik ini dibagi dalam tiga kasus, yaitu

a. Kasus 1

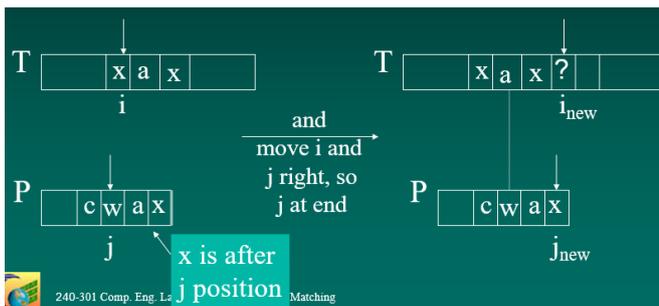
Jika P memiliki huruf x di indeks keberapapun (backward), maka geser pencocokan agar sesuai dengan last occurrence dari x di P dengan menggunakan $T[i]$



Gambar 4. Teknik Character-jump Kasus 1
Sumber. 240-301 Comp. Eng. Lab III (Software), Pattern Matching

b. Kasus 2

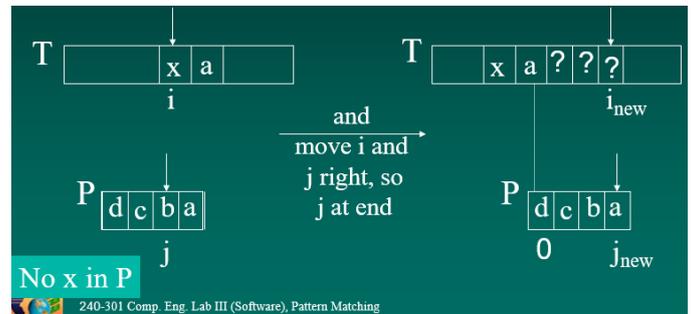
Jika P memiliki huruf x akan tetapi melakukan shift right ke last occurrence tidak dimungkinkan, maka shift P kekanan 1 karakter ke $T[i+1]$



Gambar 5. Teknik Character-jump Kasus 2
Sumber. 240-301 Comp. Eng. Lab III (Software), Pattern Matching

c. Kasus 3

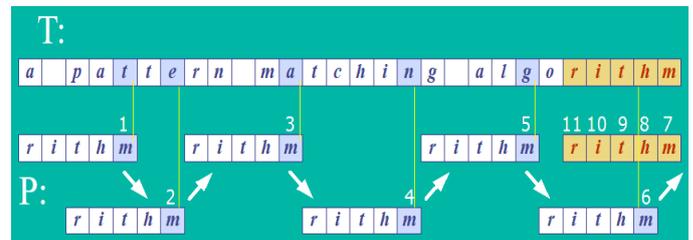
Jika kasus 1 dan 2 tidak dimungkinkan, maka shift P menjadi $P[0]$ dengan $T[i+1]$



Gambar 6. Teknik Character-jump Kasus 3
Sumber. 240-301 Comp. Eng. Lab III (Software), Pattern Matching
Untuk melakukan teknik characterjump maka menggunakan fungsi last occurrence. Fungsi ini mencatat kemunculan posisi terakhir alphabet A pada pattern P dan dimasukkan dalam fungsi $L()$. $L()$ memetakan semua huruf A dalam integers.

$L(x)$ didefinisikan sebagai:

- Indeks terbesar sehingga $P[i] == x$
- -1 jika tidak terdapat indeks yang memiliki alphabet tersebut



Gambar 7. Pattern Matching Menggunakan Algoritma Boyer Moore
Sumber. 240-301 Comp. Eng. Lab III (Software), Pattern Matching

Kompleksitas algoritma Boyer-Moore pada kasus terburuk adalah $O(nm + A)$. Boyer-Moore akan cepat apabila alphabet A besar dan lambat bila alphabet tersebut kecil. Contohnya pada kasus terburuk adalah $T: "aaaa...a"$ dan $P: "baaaaa"$.

D. Regular Expressions

Regular expressions atau yang sering disingkat regex merupakan pattern yang dapat cocok dengan perulangan karakter maupun string. Selain itu, dapat digunakan untuk mengekspresikan alternative karakter maupun string. Perulangan dan alternatif diekspresikan dengan menggunakan metacharacters seperti *, +, dan |. Sedangkan untuk menuliskan ekspresi dari karakter metacharacter perlu menggunakan \ sebagai escape character, seperti *.

Regex ini sering digunakan dalam pencarian teks dengan melakukan pattern matching. Pattern yang dicari dapat berupa pattern yang case-sensitive maupun case-insensitive bergantung pada regular expressions yang didefinisikan oleh pengguna.

Character classes	
.	any character except newline
\w \d \s	word, digit, whitespace
\W \D \S	not word, digit, whitespace
[abc]	any of a, b, or c
[^abc]	not a, b, or c
[a-g]	character between a & g
Anchors	
^abc\$	start / end of the string
\b	word boundary
Escaped characters	
\. * \	escaped special characters
\t \n \r	tab, linefeed, carriage return
\u00A9	unicode escaped ©
Groups & Lookaround	
(abc)	capture group
\1	backreference to group #1
(?:abc)	non-capturing group
(?=abc)	positive lookahead
(?!abc)	negative lookahead
Quantifiers & Alternation	
a* a+ a?	0 or more, 1 or more, 0 or 1
a{5}	exactly five
a{2,}	two or more
a{1,3}	between one & three
a+? a{2,}?	match as few as possible
ab cd	match ab or cd

Gambar 8. Notasi Regular Expressions
 Sumber: <https://www.regexpal.com/>

E. Google Analytics

Google Analytics merupakan platform dari Google yang dikeluarkan pada tahun 2005. *Google Analytics* dapat digunakan untuk mengumpulkan data dari website dan mengumpulkan *behavioral data* dari berbagai system yang merupakan *internet-connected platform*. Analytics ini digunakan untuk menganalisa market dari bisnis sehingga dapat mengenal user lebih baik. *Web analytics service* ini digunakan untuk mengumpulkan data dan memprosesnya hingga menjadi laporan yang berguna.

Pada Google Analytics, pemilik bisnis dapat melihat laporan yang terdiri dari

- **Real-time report**
 Merupakan report yang digunakan untuk menampilkan tingkah laku user secara live.
- **Audience reports**
 Merupakan report yang digunakan untuk menampilkan karakter dari user seperti umur, jenis-kelamin, asal, user baru atau lama, dll.
- **Acquisition reports**
 Merupakan report yang digunakan untuk menampilkan channel yang membuat user datang ke situs bisnis yang dianalisa. Channel itu terdiri dari organic, CPC, referral, social, dll.
- **Behavior reports**
 Merupakan report yang digunakan untuk menampilkan tingkah laku user pada situs bisnis yang dianalisa.

- **Conversion reports**
 Merupakan report yang digunakan untuk menampilkan keterhubungan tingkat pencapaian website dengan objektif dari bisnis yang telah dibuat.

Dari tiap-tiap laporan terdapat laporan perkembangan user dari berbagai aspek dan rangkuman dari semua aspek tersebut. Laporan tersebut dapat dilihat dalam bentuk grafik maupun table. Dalam menggunakan laporan, analis sering menggunakan pencarian keyword sehingga data yang ditampilkan lebih jelas. Keyword digunakan untuk melacak aktivitas dari website secara spesifik.

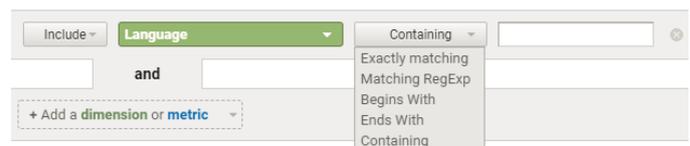
III. PATTERN MATCHING PADA REPORT VIEW

Dalam pencarian keyword pada report, terdapat pencarian biasa dan pencarian *advanced*. Dalam pengaturan *advanced*, pencarian dapat dilakukan dengan mode *include* dan mode *exclude*. Pada mode *include*, pencarian digunakan untuk mencari data yang cocok sesuai dengan masukan analis sedangkan pada mode *exclude*, pencarian digunakan untuk mencari data yang tidak mengandung keyword yang sesuai dengan masukan analis.



Gambar 9. Pilihan Metode Include dan Exlude Keyword Pencarian
 Sumber: <https://analytics.google.com/>

Keyword tersebut dimasukkan oleh analis dan kemudian dilakukan Pattern Matching pada data yang ada. Pada pengaturan lebih lanjut terdapat pilihan *exactly matching*, *matching RegExp*, *begins with*, *ends with*, dan *containing*.



Gambar 10. Pilihan Tipe Pencarian
 Sumber: <https://analytics.google.com/>

Exactly matching digunakan untuk mencocokkan pattern sama persis (*case sensitive*) sesuai masukan analis. Lalu, untuk *matching RegExp* digunakan untuk mencocokkan pattern menggunakan notasi regex. Selanjutnya, *begins with* digunakan untuk mencari data yang memiliki awalan sesuai keyword. Selain itu, *ends with* digunakan untuk mencari data yang memiliki akhiran sesuai keyword. Terakhir, *containing* digunakan untuk mencari data yang memiliki upastring yang sesuai keyword.

Untuk melakukan pencarian biasa, ketik keyword pada kotak search yang disediakan kemudian akan tampil laporan yang diinginkan. Sedangkan untuk melakukan pencarian *advanced*, maka dilakukan pengaturan pada metode *include* dan *exclude* serta pilihan tipe pencarian. Lalu masukkan keyword yang diinginkan. Berikut adalah contoh pencarian data *audience* yang menggunakan Bahasa EN pada akunnya dengan konfigurasi metode *include* dan tipe *containing*.

Language	Acquisition		Behavior		Conversion				Ecommerce Conversion Rate
	Users	New Users	Sessions	Bounce Rate	Pages / Session	Avg. Session Duration	Transactions	Revenue	
	13,496	11,282	16,745	39.52%	4.66	00:02:43	589	\$127,099.39	3.52%
1. en-us	11,793	9,630	14,725	37.34%	4.83	00:02:49	587	\$126,501.48	3.99%
2. en-gb	1,165	1,076	1,337	54.39%	3.23	00:01:54	1	\$47.99	0.07%
3. en	204	193	219	51.14%	3.02	00:01:40	1	\$549.92	0.46%
4. en-ca	132	120	169	43.79%	4.20	00:01:59	0	\$0.00	0.00%
5. en-au	113	107	128	49.61%	4.43	00:02:43	0	\$0.00	0.00%
6. en-in	66	64	77	57.14%	2.53	00:01:15	0	\$0.00	0.00%
7. en-sg	39	37	45	53.33%	5.07	00:02:10	0	\$0.00	0.00%
8. en-ae	11	11	12	58.33%	3.00	00:01:58	0	\$0.00	0.00%
9. en-za	5	5	6	50.00%	2.17	00:00:26	0	\$0.00	0.00%
10. en-ec	4	4	5	60.00%	6.80	00:01:16	0	\$0.00	0.00%

Gambar 11. Laporan Audience yang Menggunakan Bahasa EN
 Sumber: <https://analytics.google.com/>

IV. ANALISA IMPLEMENTASI PATTERN MATCHING

Pattern matching dapat dilakukan dengan berbagai metode seperti dengan menggunakan algoritma Brute Force, algoritma Knuth–Morris–Pratt, algoritma Boyer Moore, maupun Regular Expressions.

Pada bagian ini, akan dilakukan pengujian untuk tiap metode agar dapat diketahui lama waktu eksekusi program. Pengujian dilakukan dengan membuat program berbahasa Python3 yang terdiri dari teks, yaitu string yang dicocokkan dengan lima pattern.

Teks

Aplikasi dengan system Web, yaitu mdkamdkask.com ialah aplikasi yang dapat memperlihatkan event - event apa saja yang akan dilaksanakan dalam beberapa waktu kedepan di Kota Bandung. Event tersebut dapat berupa event musik dan sukarela. Panitia - panitia dari event tersebut terdiri dari pengguna yang telah mendaftarkan diri sebagai member. Event tersebut dibuat oleh komunitas - komunitas di Bandung yang telah mendaftarkan diri kepada admin. erangkat lunak yang akan dikembangkan ialah aplikasi Web mdkamdkask, yaitu perangkat lunak yang dapat memperlihatkan dan mengorganisir event - event apa saja yang bakal dilaksanan dalam waktu kedepan di kota bandung. Web mdkamdkask memiliki fungsi layanan pendaftaran untuk siapa saja yang ingin mendaftar sebagai member dalam mdkamdkask. Untuk member-member yang telah mendaftar dapat ditampilkan profil dari member tersebut. Dalam perangkat lunaknya, terdapat beberapa event apa saja yang akan diadakan dalam Kota Bandung dalam waktu dekat. Terdapat fitur untuk menampilkan event - event tersebut dalam aplikasi. xxxxxxxxxx Member yang telah mendaftar, dapat mendaftarkan diri untuk menjadi partisipan event. Untuk mencari event tersebut, terdapat fitur mencari event sesuai dengan keyword pencarian pengguna ataupun menggunakan beberapa kategori. Ketika event tersebut sudah mau dilaksanakan, terdapat notifikasi kepada member tersebut tentang pemberitahuan tentang event tersebut. Member tersebut dapat melihat event apa saja yang diikuti. Member juga dapat melakukan sharing event ke berbagai social media. Selain itu, terdapat komunitas yang hanya dapat didaftarkan oleh admin. Komunitas tersebut dapat melakukan xxxxxxxxxx penambahan event dalam

mdkamdkask dan juga dapat melihat siapa saja yang sudah melakukan pendaftaran pada event tersebut. Member dapat mengikuti beberapa komunitas yang menurut mereka menarik. Untuk menghubungkan antar member, dan juga member dan komunitas, terdapat fitur messaging. Untuk orang awam sebagai calon pengguna website tidak akan diberikan pelatihan khusus. Namun pengguna akan diberikan pengenalan melalui tombol help untuk membantu kebutuhan pengguna. Sedangkan untuk debugger juga tidak akan diberikan pelatihan sehingga dapat diketahui apakah pengguna dapat mengoperasikan aplikasi web mdkamdkask.com dengan bantuan tombol help. Akan diberikan pelatihan kepada yang melakukan maintenance yaitu administrator aplikasi web mdkamdkask.com. Selain itu, akan diberikan pelatihan untuk mengoperasikan server kepada server operator. Baik administrator dan server operator akan diberikan pelatihan selama tiga hari sebelum dilakukan pengujian. xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

1. Pengujian ke-1

Pattern yang digunakan:

xxxxa

Hasil pengujian: ditemukan

Metode Patern Matching	Waktu Eksekusi
Algoritma Brute Force	0,0008432865142822266 s
Algoritma Knuth–Morris–Pratt	0,000692606 s
Algoritma Boyer Moore	0,000627518 s
Regular Expressions	0,000228643 s

2. Pengujian ke-2

Pattern yang digunakan:

selama tiga hari sebelum

Hasil pengujian: ditemukan

Metode Patern Matching	Waktu Eksekusi
Algoritma Brute Force	0,0005433559417724609 s
Algoritma Knuth–Morris–Pratt	0,000275135 s
Algoritma Boyer Moore	0,000552654 s
Regular Expressions	0,000356674 s

3. Pengujian ke-3

Pattern yang digunakan:

tidak perlu

Hasil pengujian: tidak ditemukan

Metode Patern Matching	Waktu Eksekusi
Algoritma Brute Force	0,0005867481231689453 s
Algoritma Knuth–Morris–Pratt	0,000391722 s
Algoritma Boyer Moore	0,000567436 s
Regular Expressions	0,000262499 s

4. Pengujian ke-4

Pattern yang digunakan:

green tea

Hasil pengujian: tidak ditemukan

Metode Patern Matching	Waktu Eksekusi
Algoritma Brute Force	0,00054168701171875 s
Algoritma Knuth–Morris–Pratt	0,000449896 s
Algoritma Boyer Moore	0,000662565 s
Regular Expressions	0,000248432 s

5. Pengujian ke-5

Pattern yang digunakan:

xxx kategori.

Hasil pengujian: tidak ditemukan

Metode Patern Matching	Waktu Eksekusi
Algoritma Brute Force	0,0005388259887695312 s
Algoritma Knuth–Morris–Pratt	0,00036335 s
Algoritma Boyer Moore	0,00058198 s
Regular Expressions	0,00028348 s

Dari pengujian keempat metode pattern matching, dapat diketahui bahwa 4 dari 5 waktu eksekusi regular expressions

paling cepat diantara yang lainnya. Selain itu, tidak seperti algoritma Brute Force, Knuth–Morris–Pratt, maupun Boyer Moore yang melakukan exact matching, Regular Expressions dapat mencocokkan teks secara *case insensitive*. sehingga diharapkan analisis tidak perlu pusing memikirkan huruf besar dan kecil saat mencari keyword untuk menampilkan laporan yang lebih spesifik.

Setelah melakukan analisa ini, penulis akhirnya mendapatkan kesimpulan untuk menggunakan metode Regular Expressions dibandingkan metode lainnya dalam pencarian keyword di Google Analytics merupakan ide yang baik sehingga dapat menampilkan Report View yang lebih spesifik dalam waktu yang lebih singkat.

V. KESIMPULAN

Pebisnis *online* menggunakan platform yang terkoneksi internet untuk memasarkan produk dan jasanya. Dalam upaya memperbaiki pelayanan, pebisnis biasa melakukan Analisa bisnis. Salah satu platform yang digunakan untuk menganalisa bisnis adalah Google Analytics. Platform ini digunakan untuk menampilkan laporan. Pada laporan ini, disediakan fitur pencarian keyword dengan menggunakan Pattern Matching sehingga laporan yang ditampilkan menjadi lebih spesifik. Dalam melakukan Pattern Matching dapat dilakukan dengan menggunakan berbagai metode seperti algoritma Brute Force, algoritma Knuth-Morris-Pratt, algoritma Boyer Moore, dan Regular Expressions. Setelah melakukan analisa, penulis mendapatkan hasil bahwa Pattern Matching dengan menggunakan Regular Expressions dapat menampilkan Report View lebih cepat dibandingkan algoritma lainnya. Hal membuat Regular Expressions menjadi ide yang tepat untuk diaplikasikan pada pencarian keyword di Google Analytics sehingga mempersingkat waktu yang dibutuhkan untuk menampilkan pencarian data yang sesuai.

UCAPAN TERIMA KASIH

Pertama, Penulis mengucapkan syukur kepada Tuhan, sehingga Penulis dapat menyelesaikan makalah ini. Penulis juga berterima kasih kepada Dr. Masayu Leylia Khodra ST, MT yang telah mengajarkan saya mengenai *Pattern Matching* dalam mata kuliah IF2211 Strategi Algoritma sehingga ilmu yang telah didapatkan dapat digunakan untuk menyelesaikan makalah ini. Selain itu, Penulis juga berterima kasih kepada keluarga Penulis yang telah mendukung Penulis dalam penulisan makalah ini. Penulis juga mengucapkan terima kasih kepada Suhendi yang telah memberikan semangat dan memberikan masukan serta dorongan sehingga saya dapat menyelesaikan makalah ini. Tak lupa, Penulis mengucapkan terima kasih rekan-rekan yang telah membantu saya dalam pembuatan dan penyelesaian makalah ini.

REFERENCES

[1] Universitas Negeri Yogyakarta. <http://eprints.uny.ac.id/7990/3/BAB%202-05404241009.pdf>. Diakses tanggal 13 Mei 2018.

[2] Rinaldi Munir (2004). "Strategi Algoritmik dan Algoritma Brute Force". Diktat IF2251 Strategi Algoritmik.

[3] Rinaldi Munir (2004). "Algoritma Brute Force (lanjutan) dan Heuristik". Diktat IF2251 Strategi Algoritmik.

- [4] Rinaldi Munir (2004). "Algoritma Pencarian String (*String Matching*)". Diktat IF2251 Strategi Algoritmik.
- [5] Dr. Andrew Davison (2006-2007). "*Pattern Matching*". <http://fivedots.coe.psu.ac.th/Software.coe/LAB/PatMatch/PatternMatching.ppt>. Diunduh tanggal 13 Mei 2018.
- [6] Rashid Bin Muhammad. "Boyer-Moore Algorithm". <http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/StringMatch/kuthMP.htm>. Diakses tanggal 13 Mei 2018.
- [7] Donald Bren School of Information & Computer Sciences. <https://www.ics.uci.edu/~alspauh/cls/shr/regularExpression.html>. Diakses tanggal 13 Mei 2018.
- [8] Rashid Bin Muhammad. "Boyer-Moore Algorithm". <http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/StringMatch/boyerMoore.htm>. Diakses tanggal 13 Mei 2018.
- [9] Sriharsha Oddiraju (16 Desember 2011). "Boyer Moore". *Computer Science, Indiana State University*. <http://cs.indstate.edu/~soddiraju/abstract.pdf>. Diakses tanggal 13 Mei 2018.
- [10] Google Analytics Course. <https://analytics.google.com/analytics/academy>. Diakses tanggal 13 Mei 2018.

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, 14 Mei 2018



Nella Zabrina Pramata
13516025