VOL. 17 NO. 4 DESEMBER 2016

ISSN : 1411-3201

Jurnal Ilmiah DASI

DATA MANAJEMEN DAN TEKNOLOGI INFORMASI

STMIK AMIKOM YOGYAKARTA

VOL. 17 NO. 4 DESEMBER 2016 ISSN:1411-3201

JURNAL ILMIAH DASI

DATA MANAJEMEN DAN TEKNOLOGI INFORMASI

STMIK AMIKOM
YOGYAKARTA

VOL. 17 NO. 4 DESEMBER 2016
JURNAL ILMIAH
Data Manajemen Dan Teknologi Informasi

Terbit empat kali setahun pada bulan Maret, Juni, September dan Desember berisi artikel hasil penelitian dan kajian analitis kritis di dalam bidang manajemen informatika dan teknologi informatika. ISSN 1411-3201, diterbitkan pertama kali pada tahun 2000.

PENANGGUNG JAWAB :
Ketua STMIK AMIKOM Yogyakarta, Prof. Dr. M. Suyanto, M.M.

ALAMAT PENYUNTING & TATA USAHA
STMIK AMIKOM Yogyakarta, Jl. Ring Road Utara Condong Catur Yogyakarta, Telp. (0274) 884201 Fax. (0274) 884208, Email : jurnal@amikom.ac.id

BERLANGGANAN
Langganan dapat dilakukan dengan pemesanan untuk minimal 4 edisi (1 tahun)
pulau jawa Rp. 50.000 x 4 = Rp. 200.000,00 untuk luar jawa ditambah ongkos kirim.

VOL. 17 NO. 4 DESEMBER 2016 **ISSN : 1411- 3201**

JURNAL ILMIAH

DASI

DATA MANAJEMEN DAN TEKNOLOGI INFORMASI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AMIKOM
YOGYAKARTA

JURNAL ILMIAH

DASI

KATA PENGANTAR

Puji syukur kehadlirat Tuhan Yang Maha Kuasa atas anugerahnya sehingga jurnal edisi kali ini berhasil disusun dan terbit. Beberapa tulisan yang telah melalui koreksi materi dari mitra bestari dan revisi redaksional dari penulis, pada edisi ini diterbitkan. Adapun jenis tulisan pada jurnal ini adalah hasil dari penelitian dan pemikiran konseptual. Redaksi mencoba selalu mengadakan pembenahan kualitas dari jurnal dalam beberapa aspek.

Beberapa pakar di bidangnya juga telah diajak untuk berkolaborasi mengawal penerbitan jurnal ini. Materi tulisan pada jurnal berasal dari dosen tetap dan tidak tetap STMIK AMIKOM Yogyakarta serta dari luar STMIK AMIKOM Yogyakarta.

Tak ada gading yang tak retak begitu pula kata pepatah yang selalu di kutip redaksi, kritik dan saran mohon di alamatkan ke kami baik melalui email, faksimile maupun disampaikan langsung ke redaksi. Atas kritik dan saran membangun yang pembaca berikan kami menghaturkan banyak terimakasih.

Redaksi

DAFTAR ISI

INFORMATION RETRIEVAL MENDETEKSI KONTEN ANARKIS PADA WEB KEAGAMAAN MENGGUNAKAN ALGORITMA RABIN KARP

Yuli Astuti [1] **, Sumarni Adi** [2]

[1] *Manajemen Informatika STMIK AMIKOM Yogyakarta*
[2] *Teknik Informatika STMIK AMIKOM Yogyakarta*
email : yuli@amikom.ac.id[1], sumarni.a@amikom.ac.id [2]

Abstraksi

Semakin meningkatnya teknologi di bidang komputer sangat mendukung juga dalam kegiatan share keilmuan khususnya bidang keagamaan terutama dengan penggunaan internet. Semakin banyaknya keilmuan yang diunggah melalui website dan disediakan berbagai kemudahan untuk mengakses membuat masyarkat yang ingin belajar agama lebih detail menjadi lebih mudah. Selain itu, penggunaan internet juga dapat diakses dimana saja jika ada jaringannya. Namun terkadang informasi keilmuan yang di publish terkadang "menyesatkan".

Hal ini merupakan dasar untuk melakukan penelitian mengenai pencegah situs anarkis berbasis website. Banyak website keagamaan yang beredar di internet dengan konten keilmuan namun keilmuan tersebut tidak ada jaminan kebenarannya bahkan menjurus pada ajakan berbuat anarki. Penelitian ini membahas tentang bagaimana melakukan indexing pada teks dan meretrieval teks berdasarkan tingkat similaritasnya. pencocokan salinan dokumen dengan konten website keagamaan menggunakan metode fingerprinting dengan teknik hasing dengan menerapkan algoritma Rabin Karp.

Semakin baik indexing maka semakin tinggi nilai similaritas, sedangkan Hasing-modulo hanya menentukan waktu proses namun tidak menentukan tinggi rendahnya nilai similaritas. Penentuan k pada k-Gram menentukan nilai similaritas, semakin kecil nilai k maka nilai similaritas semakin baik.

Kata Kunci :
konten anarkis, indexing, retrieval, similaritas, Rabin Karp

Abstract

The rise of technology computer field is very supportive also in the activities of the scientific share particular religious field, especially with the use of the internet. Increasing number of scientific uploaded through the website and provided various facilities for accessing make the community who want to learn more detail religion becomes easier. In addition, the use of the Internet can also be accessed anywhere if there is network. But sometimes publish scientific information that sometimes "mislead".

This is the basis for research on the prevention of anarchist sites based websites. Many religious websites are circulating on the internet with scientific content but there is no guarantee that scientific truth even lead to calls to do anarchy. This study discusses how to match a copy of the document with the religious website content using a similarity level. The method for indexing and query using model approach finger printing. This approach uses a method Rabin Karp.

The better indexing the higher the value of similarity, whereas hasing-modulo only specify the time the process but does not determine the level of value similaritas. Determining k on k-Gram determine similarity value, the smaller the value of k, the similarity value the better.

***Keywords* :**
Indexing, Finger Printing, Rabin Karp

Pendahuluan

Saat ini dunia internet sedang berada pada fase *user generated content*, yang berarti seluruh konten yang berada di internet adalah buatan pengguna secara umum. Dengan demikian, internet seperti gudang super besar yang diisi oleh pengguna dan pengguna juga dapat menggunakan isi gudang tersebut. Salah satu aplikasi internet yang mendukung *user generated content* adalah website. Konten yang berada di website ada yang bersifat fakta, opini, hasil penelitian, tanggapan, asumsi pribadi dan sebagainya yang tidak ada jaminan kebenarannya. Trend penggunaan website untuk menyajikan informasi telah dijalani masyarakat Indonesia untuk berbagi keilmuan, salah satunya adalah *share* keilmuan agama. Namun adakalanya ketika menyajikan tidak sesuai dengan kondisi sebenarnya atau bahkan "dilebih-lebihkan" yang menjurus pada sikap anarkisme.

Permasalahan yang muncul dalam website keagamaan menimbulkan motivasi khusus untuk mengembangkan suatu penelitian. Penelitian ini akan membahas proses indexing dan retrieval teks dengan mengimplementasikan algoritma rabin karp untuk mencocokkan *keyword* terkait anarkisme dengan konten website keagamaan menggunakan tingkat similaritasnya, sehingga website keagamaan yang mengandung anarkisme dapat ditinjau kembali**.**

Penelitian ini dilakukan hanya pada situs keagamaan di indonesia yang kontennya adalah teks yang sesuai dengan KBBI. *Keyword* anarkis yang digunakan adalah teks yang diinputkan oleh pengguna sistem. Teks dokumen akan dilakukan proses indexing, dimana pembobotan term menggunakan *tf-idf* dan proses retrieval dilakukan dengan menghitung *similarity* antara keyword dan teks dokumen menggunakan algoritma Rabin Karp.

Pada penelitian sebelumnya, informasi retieval digunakan untuk mencari dan mengambil informasi iklan baris dari web, untuk katalog dan menyusun materi yang berhubungann dengan topik yang satu sumber. Sedangkan pada penelitian ini akan membahas pencocokan antara salinan dokumen yang ada di website keagamaan dan konten anarkis dengan memberikan persentase nilai similaritasnya.

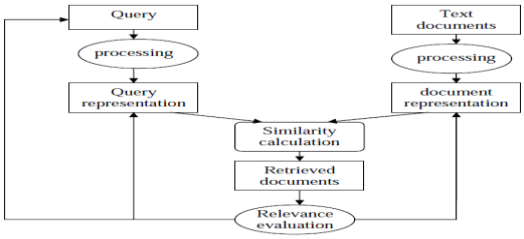
Tinjauan Pustaka

Menurut [1] berdasarkan penelitiannya menghasilkan sistem yang secara otomatis mencocokkan ketertarikan pengguna pada sistem iklan baris online. Sistem ini bekerja dengan mendefinisikan jenis iklan baris sesuai dengan struktur dan semantiknya, mengembangkan ontologi bebas domain dari iklan baris untuk pencocokan ketertarikan pengguna, mengembangkan ontologi yang *extendable* bagi domain-domain berbeda dan informasi detail mengenai pengguna. Mekanisme ekstraksi dari sistem tersebut terdiri dari 2 langkah, yaitu (1) Pembagian teks ke dalam blok-blok. Setiap *keyword* diperiksa, dilakukan pencocokan pola untuk mendapatkan beberapa bagian di atas. Hasilnya adalah suatu *hashtable* berisi pengenal bagian dan informasi (blok) yang mungkin mewakilinya.(2) Pencocokan pola terhadap blok-blok. Jika pola cocok maka disimpan di dalam *hashtable* baru. Hasilnya adalah *hashtable* yang seperti sebelumnya tetapi blok hanya diwakili oleh *keyword* yang paling relevan.

Penelitian serupa juga dilakukan oleh[2] membuat sistem ekstraksi informasi iklan khusus lowongan kerja dalam bahasa Perancis. Informasi bebas mengenai lowongan kerja tersebut diekstrak untuk menghasilkan 6 potongan informasi yang diperlukan oleh pencari kerja, yaitu subyek (posisi pekerjaan), durasi, tingkat pendidikan yang diperlukan, nama perusahaan, kota penempatan dan waktu kerja dimulai.

[3] dalam penelitiannya mengusulkan metode *web-based similarity kernel* untuk mengekspasi teks pendek menggunakan hasil pencarian web. Kemiripan antara dua teks pendek dapat dihitung dalam ruang representasi yang diekspansi. [4] mengusulkan ukuran kemiripan *web-relevance* yang tetap memanfaatkan *search engine* untuk mendapatkan *n-top* dokumen web tetapi hanya mengambil judul dan rangkumannya untuk membangun dokumen baru. Kemiripan antar dokumen dihitung menggunakan pendekatan *tf.idf* dalam ruang vektor. [5] mengusulkan metode baru dalam perhitungan kemiripan antar teks pendek dengan membandingkan topik probabilitasnya. Metode ini dimulai dengan menghimpun *term-term* pembeda antara dua teks pendek dan membandingkan keduanya dengan serangkaian topik yang diekstrak dengan algoritma *Gibbs sampling*. Kedekatan antara term pembeda ditentukan oleh probabilitasnya di dalam setiap topik. Kemiripan antara dua teks dihitung berdasarkan pada *common term* dan kedekatan dari *term-term* pembedanya.

ISO 2382/1 mendefinisikan *Information Retrieval* (IR) sebagai tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan informasi mengenai subyek yang dibutuhkan. Tindakan tersebut mencakup *text indexing*, *inquiry analysis*, dan *relevance analysis***.** Gambaran umum IR dapat ditunjukkan pada Gambar 1.



Gambar 1 Gambaran Umum IR

Pembuatan *index* dari koleksi dokumen merupakan tugas pokok pada tahapan *preprocessing* didalam IR. Kualitas *index* mempengaruhi efektifitas dan efisiensi sistem IR [6]. *Index* dokumen adalah himpunan *term* yang menunjukkan isi atau topik yang dikandung oleh dokumen. *Index* akan membedakan suatu dokumen dari dokumen lain yang berada di dalam koleksi. Menurut [7] terdapat 5 langkah pembuatan *inverted index*, contohnya ditunjukkan pada Gambar 2, yaitu:

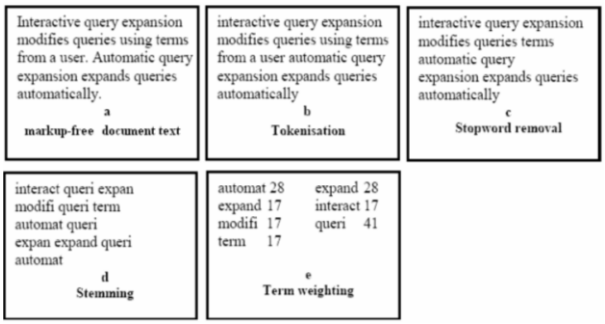
1. Penghapusan format dan *markup* dari dalam dokumen. Tahap ini menghapus semua *tag markup* dan format khusus dari dokumen, terutama pada dokumen yang mempunyai banyak *tag* dan format seperti dokumen (X)HTML. Jika isi dokumen telah berada di

dalam database maka tahapan ini sering dilewatkan.

2. Pemisahan rangkaian kata (*tokenization*). *Tokenization* adalah tugas memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi *token* atau potongan kata tunggal atau *termmed word*. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua *token* ke bentuk huruf kecil (*lower case*).
3. Penyaringan (*filtration*). Pada tahapan ini ditentukan *term* mana yang akan digunakan dengan Penghapusan *stop-word* dari dalam suatu koleksi dokumen .
4. Konversi *term* ke bentuk akar (*stemming*). *Stemming* adalah proses konversi term ke bentuk dasarnya.
5. Pemberian bobot terhadap term (*weighting*). Setiap term diberikan bobot sesuai dengan skema pembobotan yang dipilih, Banyak aplikasi menerapkan pembobotan kombinasi berupa perkalian bobot lokal *term frequency* dan global *inverse document frequency*, ditulis *tf .idf*. Adapun rumus untuk menghitung *tf.idf* dapat dilihat di Persamaan 1.

$$Tf.idf\ (d, w) = tf\ (d, w) \times \log N/dfw \quad (1)$$

Dimana: tf(d,w) adalah frekuensi kemunculan term w pada dokumen d, n adalah jumlah keseluruhan dokumen dan dfw adalah jumlah dokumen yang mengandung term w.



Gambar 2 Contoh lima tahap *indexing* pada sistem berbasis content secara urut
mulai dari *markup removal* (a), *tokenization* (b), *stopwords filtration* (c), *stemming* (d)
dan *weighting* (e)

Inquiry analysis merupakan langkah berikutnya dalam IR. Pada tahap ini dilakukan pencocokan string dengan menggunakan algoritma *Rabin Karp*. Adapun langkah-langkah algoritma *Rabin Karp* sebagi berikut :

1. *Parsing*, yaitu term yang sudah melalui proses indexing dipotong-potong per karakter huruf. Pemotongan per karakter menggunakan metode *k-Grams.* Cara kerj*a* metode *k-grams* dengan mengambil potongan-potongan karakter huruf sejumlah *k* dari sebuah kata yang secara kontinyu dibaca dari teks sumber hingga akhir dari dokumen. contoh *k-grams* dengan *k*=4:
   Teks : `evenifnone`
   Hasil 4-*grams* dari teks : `even veni enif nifn ifno fnon none`
2. *Hashing*, yaitu mengkonversi potongan huruf sejumlah *k* kedalam nilai *hash*.
3. *Rabin Karp.* Pada bagian ini akan membandingkan nilai *hash* dari *string* masukan dan *hash substring* pada teks. Apabila sama, maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya. Apabila tidak sama, maka *substring* akan bergeser ke kanan. Rabin Karp merepresentasikan setiap karakter ke dalam bentuk desimal digit (*digit radix-d*) ⩙ = {0, 1, 2, 3, …, *d*}, dimana d = |⩙|. Sehingga didapat masukan *string k* berturut-turut sebagai perwakilan panjang *k* desimal. Kemudian pola *p* dihash menjadi nilai desimal dan *string* direpresentasikan dengan penjumlahan *digit-digit* angka menggunakan aturan Horner's, misal (Elchison, 2004) :
   { A, B, C, ..., Z } → { 0, 1, 2, ..., 26 }
   • BAN → 1 + 0 + 13 = 14
   • CARD → 2 + 0 + 17 + 3 = 22
   Untuk pola yang panjang dan teks yang besar, algoritma ini menggunakan operasi *mod*, setelah dikenai operasi *mod q*, nilainya akan menjadi lebih kecil dari *q*, misal: • BAN = 1 + 0 + 13 = 14
   = 14 *mod* 13 = 1
   = BAN → 1
   • CARD= 2 + 0 + 17 + 3 = 22
   = 22 *mod* 13 = 9
   = CARD → 9
   Secara Matematis dapat dituliskan pada Persamaan 2:

$$t_{s+1} = (d\,(t_s - T[s+1]\,h) + T[s+m+1]) \bmod q \quad (2)$$

dimana

t_s = nilai desimal dengan panjang m dari *substring* $T[s+1 .. s+m]$, untuk $s = 0, 1, ..., n-m$

t_{s+1} = nilai desimal selanjutnya yang dihitung dari t_s

d = *radix* desimal (bilangan basis 10)

h = d^{m-1}

n = panjang teks

m = panjang pola

q = nilai *modulo*

4. Peningkatan performa. Menurut [8] agar performa *rabin karp* meningkat maka diberikan solusi agar tidak membandingkan sisa hasil bagi (modulo) saja, tetapi membandingkan hasil baginya juga. Seperti yang ditunjukkan pada Persamaan 3 :

Rem (n1/q) = rem (n2/q)
And
Quotient (n1/q) = Quotient (n2/q) (3)

Relevance analysis merupakan langkah terakhir dalam teks IR. Proses ini adalah mencari relevansi antar query dengan teks dokumen dengan cara mengukur *similarity* (kemiripan) dan jarak antara dua entitas informasi. Menurut [9] Karena parsing dilakukan dengan pendekatan *k-gram* maka untuk mengukur nilai similarity menggunakan *Dice's Similarity Coefficient*, rumus matematisnya ditunjukkan pada Persamaan 4.

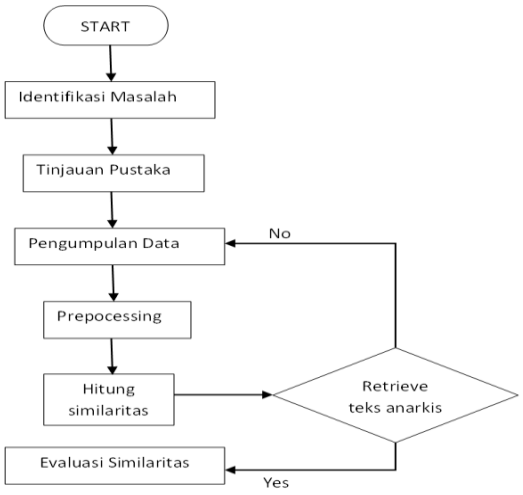
$$S = \frac{K \times C}{A + B} \qquad (4)$$

Dimana S adalah nilai *similarity*, A dan B adalah jumlah dari kumpulan *k-grams* dalam teks 1 dan teks 2. C adalah jumlah dari *k-grams* yang sama dari teks yang dibandingkan

Metode Penelitian

Adapun metode penelitian yang dilakukan seperti yang ditunjukkan pada Gambar 3, yaitu sebagai berikut :

1. Dalam tahap ini dilakukan mengidentifikasian permasalahan yang ada dari mulai latar belakang, perumusan, pembatasan masalah, tujuan, manfaat, sampai pada metodologi yang digunakan
2. Pada tahapan ini dilakukan studi literatur dan kajian pustaka terhadap beberapa referensi yang relevan dengan topik penelitian. Adapun referensi yang dirujuk dalam penelitian ini adalah dasar-dasar *information retrieval*, proses retrieval, stemming, indexing, dan pemodelan *fingerprinting* dengan algoritma Rabin Karp.
3. Melakukan pengumpulan teks dokumen pada website keagamaan yang direkomendasikan oleh Departemen Agama RI secara realtime. Pengumpulan data ini dilakukan dengan memanfaatkan JSOUP yang ada di library java. Kemudian hasil ekstraksi teks dokumen website ini disimpan di database. Konten teks yang dikumpulkan antara lain user, judul, isi teks, alamat website, tanggal dan waktu posting.
4. Pada tahap preprocessing dilakukan Pembuatan *index* dari koleksi dokumen
5. Pada Tahap hitung Similaritas adalah melakukan pencocokan string dengan menggunakan algoritma *Rabin Karp.* Jika cocok maka akan lanjut pada proses evaluasi similaritas, namun jika tidak maka teks dokumen akan dikembalikan ke database.
6. Tahap Evaluasi similaritas akan dilakukan perhitungan similarity (kemiripan) menggunakan *Dice's Similarity Coefficient*



Gambar 3 Alur Metode Penelitian

Hasil dan Pembahasan

Pengumpulan data

Data yang digunakan dalam penelitian ini adalah data teks website keagamaan yang diambil secara realtime dengan memanfaatkan API JSOUP yang ada pada library java. Sistem memberikan request ke server alamat website untuk mengambil dokumen XML yang berisi text dokumen yang kemudian disimpan di database. Adapun struktur teks dokumen yang dikumpulan dalam sistem adalah sebagai berikut :

1. User adalah data pengguna yang memosting text dokumen.
2. Judul adalah judul dari teks dokumen
3. Isi adalah data konten text dokumen
4. Date adalah tanggal teks dokumen tersebut diposting.
5. Time adalah waktu teks dokumen tersebut diposting.

Indexing

Berikut adalah dua teks dokumen hasil ekstraksi dari website yang ditunjukkan pada Tabel 1.

Tabel 1 Teks dokumen hasil ekstraksi

Teks dokumen 1	<Document doc=”Ajaran keagamaan membawa pada ketenangan dan kebahagiaan hidup”.get()>
Teks Dokumen 2	<Document doc=”Dengan hidup beragama akan memberikan rasa tenang dan bahagia.get()>

Tabel 1 mengalami proses indexing, yaitu hapus format dan markup, tokenizing, filtering, stemming. Hasilnya ditunjukkan pada Tabel 2.

Tabel 2 Teks dokumen hasil indexing mulai dari hapus format dan markup, tokenizing, filtering, stemming

File	Hapus Markup dan format	Tokenizing	Filtering	Stemming	Term-Weighting
Teks dokumen 1	Ajaran keagamaan membawa pada ketenangan dan kebahagiaan hidup	ajaran	ajaran	ajar	
		keagamaan	keagmaan	agama	
		pada	ketenangan	tenang	
		ketenangan	kebahagiaan	bahagia	
		dan	hidup	hidup	
		kebahagiaan			
		hidup			
Teks dokumen 2	Dengan hidup beragama akan memberikan rasa tenang dan bahagia	dengan	hidup	hidup	
		hidup	beragama	agama	
		beragama	memberikan	beri	
		akan	rasa	rasa	
		memberikan	tenang	tenang	
		rasa	bahagia	bahagia	
		tenang			
		dan			
		bahagia			

Dan dilanjutkan dengan proses term weighting dengan rumus *tf.idf* pada Persamaan 1.

Jumlah dokumen = 2, tf ("ajar" pada teks 1) = 1, df ("ajar") = 1

$$tf\text{-}idf \text{ (teks 1,"ajar")} = 1 \times log\frac{2}{1} = 0{,}301$$

Apabila proses perhitungan tersebut dilakukan untuk semua dokumen dan semua kata maka akan dihasilkan matrik perhitungan tf-idf yang terdapat pada Tabel 3.

Tabel 3 Teks dokumen hasil indexing

File	Hapus Markup dan format	Tokenizing	Filtering	Stemming	Term-Weighting
Teks dokumen 1	Ajaran keagamaan membawa pada ketenangan dan kebahagiaan hidup	ajaran	ajaran	ajar	0,301
		keagamaan	keagmaan	agama	0
		pada	ketenangan	tenang	0
		ketenangan	kebahagiaan	bahagia	0
		dan	hidup	hidup	0
		kebahagiaan			
		hidup			
Teks dokumen 2	Dengan hidup beragama akan memberikan rasa tenang dan bahagia	dengan	hidup	hidup	0
		hidup	beragama	agama	0
		beragama	memberikan	beri	0,301
		akan	rasa	rasa	0,301
		memberikan	tenang	tenang	0
		rasa	bahagia	bahagia	0
		tenang			
		dan			
		bahagia			

Pencocokan Similaritas dengan Algoritma *Rabin Karp*

Parsing menggunakan *k-gram* dengan *k=4*, pada setiap teks dokumen. Hasil parsing ditunjukkan pada Tabel 4.

Teks Dokumen 1 : ajaragamatenangbahagiahidup

Teks Dokumen 2 : agamaberirasatenangbahagia

Tabel 4 Hasil parsing 4-*gram*

No	Parsing Teks 1	Parsing teks 2
1	ajar	agam
2	jara	gama
3	arag	amab
4	raga	mabe
5	agam	aber
6	gama	beri
7	amab	erir
8	maba	rira
9	abaw	iras
10	bawa	rasa
11	awat	Asat
12	wate	Sate
13	aten	Aten
14	tena	Tena
15	enan	Enan
16	nang	nang
17	angb	angb
18	ngba	ngba
19	gbah	gbah
20	baha	baha
21	ahag	ahag
22	hagi	hagi
23	agia	agia
24	giah	
25	iahi	
26	ahid	
27	hidu	
28	idup	

Dan dilanjutkan proses *hashing* dengan mengkonversi string menjadi nilai ASCII. a-z = 97-122), dan dilanjutkan dengan proses *rabin karp* menggunakan Persamaan 2 dan peningkatan performa *rabin karp* dengan Persamaan 3. Penghitungan nilai *hash* dengan modulo 101:

ajar $= (97*10^3) + (106*10^2) + (97*10^1) + (114*10^0)$
$= 97{,}000 + 10{,}600 + 970 + 114$
$= 108{,}684$
Mod $= 108{,}684 \ mod \ 101 = 8$
Rem $= 108{,}684 \ / \ 101 = 1076.07920792079$

jara $= (106*10^3) + (97*10^2) + (114*10^1) + (97*10^0)$
$= 106{,}000 + 9{,}700 + 1{,}140 + 97$
$= 116{,}937$
Mod $= 116{,}937 \ mod \ 101 = 80$
Rem $= 116{,}937 \ / \ 101 = 1157.79207920792$

Apabila proses perhitungan tersebut dilakukan untuk semua dokumen dan semua kata maka akan dihasilkan matrik perhitungan *hasing* modulo dan *hasing* reminder yang terdapat pada Tabel 5.

Tabel 5 Hasil perhitungan *Hashing* modulo dan *Hashing* Div (Reminder)

No	Teks Dokumen 1			Teks Dokumen 2			cocok
	Parsing	Hash	Remainder	Parsing	Hash	Remainder	
1	ajar	8	1076.07920792079	agam	6	1073.05940594059	Tidak
2	jara	80	1157.79207920792	gama	60	1127.59405940594	Tidak
...	...	...	...	...	...	...	...
5	agam	6	1073.05940594059	agam	6	1073.05940594059	Ya
6	gama	60	1127.59405940594	gama	60	1127.59405940594	Ya
7	amab	90	1078.89108910891	amab	90	1078.89108910891	Ya
8	aten	34	1086.33663366336	aten	34	1086.33663366336	Ya
9	tena	37	1260.36633663366	tena	37	1260.36633663366	Ya
10	enan	61	1119.60396039603	enan	61	1119.60396039603	Ya
11	nang	6	1197.05940594059	nang	6	1197.05940594059	Ya
12	angb	48	1080.47524752475	angb	48	1080.47524752475	Ya
13	ngba	76	1201.75247524752	ngba	76	1201.75247524752	Ya
14	gbah	47	1127.46534653465	gbah	47	1127.46534653465	Ya
15	baha	60	1077.59405940594	baha	60	1077.59405940594	Ya
16	ahag	98	1073.99009900990	ahag	98	1073.99009900990	Ya
17	hagi	99	1136.98019801980	hagi	99	1136.98019801980	Ya
18	agia	74	1073.73267326732	agia	74	1073.73267326732	Ya
...	...	...	...	...	...	...	...

Dari Tabel 5 dapat dilihat bahwa baris satu tidak cocok karena nilai hash dan reminder antara teks

dokumen 1 dan teks dokumen 2 tidak sama. Sedangkan pada baris 3 cocok karena nilai hash dan remindernya sama. Jadi teks dokumen cocok ketika nilai hash modulo dan hash div sama.

Untuk menghitung similarity (kecocokan) menggunakan Persamaan 4. Jadi similarity teks dokumen 1 dan 2 adalah

Similarity = ((2*14) / (28+23)) * 100%

= (28/51) * 100%

= 0.5490 * 100%

= 54.90%

Pengujian dengan besarnya ukuran konten

File uji terdiri dari tiga buah *file* yang masing-masing telah dilakukan proses indexing. Informasi file uji ditunjukkan pada Tabel 6.

Tabel 6 File Uji

Id	Nama_file	∑ kata	Ukuran(byte)
1	1_web_50_kata	50	428
2	2_web_100_kata	100	861
3	3_web_150_kata	150	1293

File uji pada Tabel 6 akan dilakukan pengujian terhadap waktu, hasilnya ditunjukkan pada Tabel 7

Tabel 7 File Uji terhadap waktu

Id	Teks Dokumen 1	Teks Dokumen 1	Similarity (%)	Waktu
1	1_web_50_kata	1_web_50_kata	100	0.321823
2	2_web_100_kata	1_web_50_kata	36	0.379652
3	3_web_150_kata	1_web_50_kata	54	0.411584

Dari Tabel 7 dapat dilihat bahwa semakin besar ukuran file maka waktu yang diperlukan juga semakin besar untuk proses mencari similarity file.

Jika file tidak mengalami proses indexing, maka waktu yang diperlukan semakin sedikit namun nilai similaritasnya berkurang. Hal ini ditunjukkan pada Tabel 8.

Tabel 8 File Uji tanpa indexing terhadap waktu dan similarity

Id	Teks Dokumen 1	Teks Dokumen 1	Similarity (%)	Waktu
1	1_web_50_kata	1_web_50_kata	100	0.078856
2	2_web_100_kata	1_web_50_kata	34	0.154483
3	3_web_150_kata	1_web_50_kata	49	0.198553

Pengujian Modulo dan *k-gram* pada *Rabin Karp*

Berikut informasi dokumen teks yang akan digunakan untuk pengujian, terdapat pada Tabel 9.

Tabel 9 informasi File Uji

Id	Nama_file	∑ kata	Ukuran(byte)
1	1_web_50_kata	50	428

Tabel 9 akan dilakukan uji modulo terhadap waktu, hasilnya akan ditunjukkan pada Tabel 10.

Tabel 10 File Uji modulo terhadap waktu

Id	K-Gram	Modulo	Similarity (%)	Waktu
1	1	13	109.854	0.854227
2	1	23	109.854	0.283314
3	1	43	109.854	0.151186
4	1	73	109.854	0.321828
5	1	101	109.854	0.089997
6	1	151	109.854	0.186377
7	1	173	109.854	0.275583

Dari Tabel 10 dapat lihat bahwa semakim besar nilai modulo maka waktu yang dibutuhkan semakin sedikit.

Tabel 10 File Uji *K-Gram* terhadap waktu

Id	K-Gram	Modulo	Similarity (%)	Waktu
1	1	101	109.854	0.050045
2	2	101	60.7698	0.110385
3	3	101	41.2881	0.104722
4	4	101	40.9256	0.285242
5	5	101	40.6540	0.089997
6	6	101	40.4421	0.110583
7	7	101	40.0058	0.195587

Sedangkan semakin kecil nilai *k* pada *k-gram* maka semakin tinggi nilai similaritasnya. Hal ini seperti yang ditunjukkan pada Tabel 11.

Kesimpulan dan Saran

Berdasarkan percobaan-percobaan yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Sistem yang dibangun berbentuk website dimana web tersebut akan melakukan ekstraksi konten teks website keagamaan yang ada di indonesia dengan menggunakan API pada JSOUP yang terdapat di library java dan konten teks web akan dikonversi ke file word
2. Sistem dalam membandingkan teks dokumen memberikan hasil berupa prosentase *similarity.*
3. Semakin besar ukuran file maka waktu yang diperlukan juga semakin besar untuk proses mencari similarity file
4. Jika file tidak mengalami proses indexing, maka waktu yang diperlukan semakin kecil namun nilai similaritasnya berkurang
5. Nilai *modulo* berpengaruh pada waktu proses, tetapi tidak pada nilai *similarity*.
6. Semakin kecil *k-gram* menghasilkan akurasi nilai *similarity* yang lebih baik dibandingkan *k-gram* yang lebih besar.

Penelitian ini masih sangat jauh dari kesempurnaan, oleh karena itu diberikan saran-saran sebagai berikut :

1. Sistem yang dibangun tidak dapat mendeteksi 100% konten teks pada website karena keterbatasan database dan algoritma yang digunakan
2. Sistem ini hanya mampu mendeteksi teks berbahasa indonesia yang sesuai dengan KBBI
3. Semakin banyak teks yang diolah maka semakin lama beban komputasinya maka

diperlukan teknik untuk memanjamen waktu eksekusi file

4. Sistem hanya dapat membandingkan *file* uji dengan semua *file* sumber pada *database* sistem.
5. Sistem diharapkan dapat mendeteksi thesaurus yang meliputi sinonim, homonon, homograf dan lainnya.
6. Penelitian selanjutnya bisa dikembangkan untuk menganalisis sentimen per kalimat sehingga proses penemuan konten anarkisme lebih valid

Daftar Pustaka

[1] Gribova, Valeriya, Kachanov, Pavel (2009) An Approach to Automated User Interest Matching in Online Classified Advertising Systems, *ICIC* 2009, pp. 665-673, Springer-Verlag Berlin

[2] Faure, David, Morlon, Claire, *Information Extraction for Classified Advertising*, 2005.

[3] Sahami, M, Heilman, T, A web-based kernel function for measuring the similarity of short text snippets, *In Proceedings ofWWW 2006*, pages 377-386, 2006.

[4] Meek, Christopher, Yih, Wen-tau, *Improving Similarity Measures for Short Segments of Text*, 2007.

[5] Quan, Xiaojun. Etc (2009) Short text similarity based on probabilistic topics, *Knowledge Information Systems Regular Paper*, 17 September 2009

[6] Chu W. Liu Z, Mao W. *Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining*, 2002.

[7] Manning, C.D, Raghavan, P, & Schutze, H, 2008, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge.

[8] Singh, Rajender Chillar, Barjesh Kochar. 2008. RB-Matcher: String Matching Technique. *World Academy of Science, Engineering and Technology*, 42.

[9] Kosinov, Serhiy. 2001. Evaluation of n-grams conflation approach in text-based information retrieval. *Unpublished journal*. Computing Science Department, University of Alberta, Canada.