Comparative Analysis of Text Mining Results With Tf ldf Features and SQL Like Operator in Indonesian News Search

Analisa Perbandingan Hasil Text Mining Dengan Fitur Tf ldf dan SQL Like Operator Pada Pencarian Berita berbahasa Indonesia

Riwa Rambu Hada*, Enda Fajar Hariadi

Teknik Informatika, Universitas Kristen Wira Wacana Sumba, Indonesia

**Email Penulis Korespondensi:riwa@unikriswina.ac.id*

Abstract. *Research on the implementation of text mining uses the TF IDF method to be used in the Information retrieval / Indonesian news search feature. The dataset used was sourced from NewsAPI and built a Codeigniter based website named "News Plus Six Dua". This study also uses the Vertor Space Model (VSM) method to overcome the weaknesses of the TF IDF method at the time of the sorting process. The results of this study explain that the search by the TF IDF method has higher accuracy when compared to SQL like operators. TF IDF produces a percentage of precision 100% and recall (sensitivity) 66.7% on searches with the keyword "Indonesian soccer schedule" while SQL like operators do not display search results or equal to 0%. But the TF IDF method has the disadvantage of running slower than SQL like operators. This has been tested using either the number of words or terms entered, the number of datasets, and the location of access. At the location of access, access via hosting is monitored faster when compared via localhost*
Keywords- text mining; TF IDF; VSM; Indonesian News

Abstrak. *Penelitian implementasi text mining menggunakan metode TF IDF untuk digunakan pada fitur Informasi retrieval / pencarian berita berbahasa Indonesia. Dataset yang digunakan bersumber dari NewsAPI dan membangun website berbasis Codeigniter yang diberi nama "Berita Plus Enam Dua". Penelitian ini juga menggunakan metode Vertor Space Model (VSM) untuk mengatasi kelemahan metode TF IDF pada saat pada saat proses sorting. Hasil dari penelitian ini menjelakan bahwa pencarian dengan metode TF IDF memiliki keakurasian lebih tinggi jika dibandingkan dengan SQL like operator. TF IDF menghasilkan persentase precision 100% dan recall (sensitifitas) 66,7% pada pencarian dengan keyword "jadwal sepak bola indonesia" sedangkan SQL like operator tidak menampilkan hasil pencarian atau sama dengan 0%. Tapi metode TF IDF memiliki kekurangan yaitu berjalan lebih lambat dari pada SQL like operator. Hal ini telah diuji baik dengan menggunakan faktor jumlah kata atau term yang diinputkan, jumlah dataset, dan lokasi akses. Pada lokasi akses, akses melalui hosting dipantau lebih cepat jika dibandingkan melalui localhost.*
Kata kunci- text mining; TF IDF; VSM; Berita Berbahasa iNdonesia

PENDAHULUAN

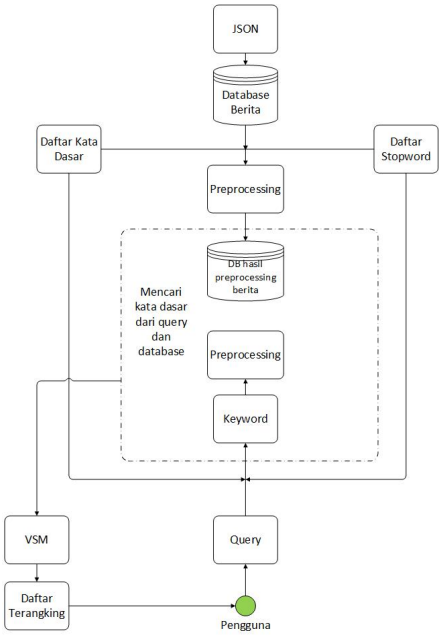
Dahulu kala berita hanya dihadirkan diatas kertas dan hanya waktu – waktu tertentu saja berita tersebut digunakan. Tapi saat ini berbeda, dengan semakin berkembangnya internet berita sekarang sudah digitalisasi [1]. Pencarian informasi bisa dilakukan dengan memanfaatkan mesin pencarian yang banyak terdapat di dunia maya. Tapi informasi yang didapatkan sering kali terlalu banyak, sehingga menyulitkan pengguna karena akan mendapatkan informasi yang kurang berguna. Hakikatnya kualitas informasi dipengaruhi oleh beberapa hal yaitu keakuratan, relevansi, dan ketepatan waktu [2].

Menurut Karter D. Putung (2016), Informasi retrieval merupakan solusi yang tepat untuk melakukan pencarian suatu dokumen. Dalam penelitiannya, Putung mengkombinasikan IR dengan pembobotan TF IDF untuk membangun sistem pencarian dokumen pada sistem penyimpanan skripsi [3]. Penelitian terkait TF IDF juga pernah dilakukan oleh Mochamad Alfan Rosid (2015), metode TF IDF digunakan untuk 133232 Univeristas Muhammadiyah sidoarjo [4]. Dari latar belakang diatas, peneliti melakukan penelitian tentang pemanfaatan text mining menggunakan metode TF IDF untuk digunakan pada fitur Informasi retrieval / pencarian berita berbahasa Indonesia. Peneliti menerapkannya pada website pencarian berita menggunakan framework codeigniter. Dataset pada penelitian ini menggunakan data dari NewsAPI yang merupakan layanan API yang berisi kumpulan berita. Peneliti juga menggunakan metode Vector Space Model (VSM) untuk menanggulangi kelemahan metode TF IDF pada saat sorting data [5]. Untuk mengetahui tingkat keakuratan dan kecepatan pencarian dengan metode ini, peneliti membandingkannya dengan pencarian SQL like operator yang dimiliki oleh DBMS MySQL.

METODE PENELITIAN

Informasi Retrieval (IR)

Informasi Retrieval (IR) atau sistem temu kembali adalah suatu bagian dari ilmu komputer yang bekerja untuk mengambil informasi dari dokumen – dokumen dengan didasarkan pada konteks dan isi dari dokumennya sendiri. IR direpresentasikan sebagai mesin pencari dokumen atau data sesuai keinginan pengguna. Amin menjelaskan bahwa IR bertujuan untuk menjembatani sumber informasi yang tersedia dengan kebutuhan informasi pengguna [6]. IR secara umum digunakan pada mesin pencarian di jaringan internet [7]. Adapun gambar 1 mengilutrasikan cara kerja sistem IR yang akan dibuat pada penelitian ini.



Gambar 1. Ilustrasi dari Informasi Retrieval.

Data dari NewsAPI yang berbentuk JSON akan disimpan sebagai dataset pada tabel berita yang bernama news. Adapun attributenya ditunjukkan oleh tabel 1.

Tabel 1. Atributte tabel news

No	Nama	Tipe	Panjang	Keterangan
1	Id	Integer	11	Auto Increment
2	author	Varchar	40	
3	title	Text		
4	description	Longtext		
5	url	Text		
6	urltoimage	Text		
7	publishedat	Varchar	20	
8	content	Longtext		

Tabel news dan kata kunci pencarian dari pengguna (user) yang berbentuk query akan diproses melalui tahap preprocessing. Khusus untuk hasil preprocessing untuk tabel news akan disimpan pada tabel news_temp. Adapun attributenya ditunjukkan oleh tabel 2.

Tabel 2. Attribute tabel news_temp

No	Nama	Tipe	Panjang	Keterangan
1	id_doc	Integer	11	Auto Increment
2	dokumen	Text		

Proses diteruskan pada tahap VSM. Kemudian data akan ditampilkan sesuai dengan kecocokan terhadap kata kunci dari user.

NewsAPI

News API merupakan web API yang memberikan layanan berisi kumpulan berita yang nantinya dapat digunakan oleh developer untuk menggembangkan sebuah aplikasi. news API dapat diakses pada url https://newsapi.org/. Web API atau web service merupakan antar muka program yang dapat diakses lewat method dan protokol HTTP standar [8]. Web API bisa diartikan sebuah layanan yang dirancang untuk interaksi antar mesin melalui sebuah jaringan. Layanan jenis ini bersifat terbuka dan berfungsi untuk integrase data dan kolaborasi informasi menggunakan teknologi yang dimiliki oleh masing – masing pengguna [9]. Penelitian kali ini menggunakan data deri NewsAPI yang disimpan ke dalam basis data untuk nantinya digunakan sebagai dataset penelitian yang ditunjukkan oleh tabel 3.

Tabel 3. Dataset penelitian

Id. Dokumen	Judul Berita
D1	Calya Jadi Armada Taksi Express, Ini Kata Toyota - Kompas.com - Otomotif Kompas.com
D2	Gambir \'Pensiun\' Dikritik Jonan, Luhut: Pak Budi Karya Sudah Betul - detikFinance
D3	Intip Kekayaan Tambang Emas Grup Bakrie, Bisa Sampai 2050! - CNBC Indonesia
D4	Berbatov: Messi atau Ronaldo pun Akan Sulit Bikin Gol di MU - detikSport
D5	Tak Selamanya di Chelsea, Kepa Ingin Balik ke Bilbao - detikSport

Text Mining

Text Mining adalah sebuah informasi baru yang ditemukan oleh komputer dan secara otomastis akan ekstraksi ke dalam sumber daya tertulis yang berbeda [5]. Menurut Hearst (2003) dalam jurnal yang dia tulis yang berjudul "Untangling Text Data Mining", dia berpendapat tentang text mining untuk mengakses informasi. Tujuan pengguna mengakses informasi adalah untuk membantu pengguna temukan dokumen yang memuaskan kebutuhannya. Masalahnya tidak begitu banyak informasi yang ingin pengguna ketahui, melainkan hanya informasi yang valid dan diinginkan saja [10]. Contohnya, hanya karena pengguna saat ini ingin mencari SMARTWHEEL dan bukan berati HOT WHEEL harus ditampilkan. Jadi bisa disimpulkan bahwa tujuan dari text mining adalah untuk

memperoleh informasi baru dari data text, menemukan pola di seluruh dataset, dan memisahkannya menjadi kata baku. Menurut Mooney (2006) menjelaskan bahwa text mining dapat dilakukan dengan beberapa tahapan atau bisa disebut dengan preprocessing [5]. Proses preprocessing terdiri atas empat tahapan yang berurutan diantaranya yaitu case folding, tokenizing, filtering, stemming, dan analyzing.

Preprocessing

Disini peneliti membuat sebuah contoh kasus, ketika pengguna mencari berita dengan kata kunci (Q) pencarian yaitu "Kekayaan Ronaldo". Dimulai dari proses case folding yaitu sebuah tahapan dimana text atau kalimat yang sebelumnya huruf besar diubah menjadi huruf kecil [11]. Selain itu, Karakter selain huruf akan dianggap delimiter dan dihilangkan [12]. Proses case folding akan membuat dataset seperti pada tabel 1 dan kata kunci (Q) menjadi seperti yang terlihat pada tabel 4.

Tabel 4. Hasil proses case folding

Id. Dokumen	Kalimat
Q	kekayaan Ronaldo
D1	calya jadi armada taksi express, ini kata toyota - kompas.com - otomotif kompas.com
D2	gambir \'pensiun\' dikritik jonan, luhut: pak budi karya sudah betul - detikfinance
D3	intip kekayaan tambang emas grup bakrie, bisa sampai ! - cnbc indonesia
D4	berbatov: messi atau ronaldo pun akan sulit bikin gol di mu - detiksport
D5	tak selamanya di chelsea, kepa ingin balik ke bilbao - detiksport

Proses tokenizing dilanjutkan setelah itu, yaitu tahapan pemotongan kata dari apa yang diinput menjadi kata dasar dari penyusunnya. Tanda baca juga dihilangkan pada proses ini. Hasil dari proses ini dapat dilihat pada tabel 5.

Tabel 5. Hasil proses tokenizing

Id. Dokumen	Kalimat
Q	kekayaan | ronaldo
D1	calya | jadi | armada | taksi | express | ini | kata | toyota | kompas | com | otomotif | kompas | com
D2	gambir | pensiun | dikritik | jonan | luhut | pak | budi | karya | sudah | betul | detikfinance
D3	intip | kekayaan | tambang | emas | grup | bakrie | bisa | sampai | cnbc | indonesia
D4	berbatov | messi | atau | ronaldo | pun | akan | sulit | bikin | gol | di | mu | detiksport
D5	tak | selamanya | di | chelsea | kepa | ingin | balik | ke | bilbao | detiksport

Kata dasar yang sudah di tokenizing seperti yang ada di tabel 3, selanjutnya akan melalui proses filtering dimana hanya kata – kata baku saja yang diambil. Pengambilan ini disesuaikan dengan daftar stopword. Kata dasar sama dengan stopword maka kata tersebut akan dibuang [13]. Hasil prosesnya seperti yang ditunjukkan oleh tabel 6.

Tabel 6. Hasil proses filtering

Id. Dokumen	Kalimat
Q	kekayaan | ronaldo
D1	calya | armada | taksi | express | toyota | kompas | com | otomotif | kompas | com
D2	gambir | pensiun | dikritik | jonan | luhut | budi | karya | detikfinance
D3	intip | kekayaan | tambang | emas | grup | bakrie | cnbc | indonesia
D4	berbatov | messi | ronaldo | sulit | bikin | gol | mu | detiksport
D5	chelsea | kepa | bilbao | detiksport

Selanjutnya proses stemming akan dijalankan, yaitu sebuah proses pencarian kata dasar yang dilakukan dengan mereduksi kata – kata berimbuhan hasil proses filter manjadi kata dasar [14]. Hasil dari proses stemming diilustrasikan pada tabel 7.

Tabel 7. Hasil proses stemming

Id. Dokumen	Kalimat
Q	kaya | ronaldo
D1	calya | armada | taksi | express | toyota | kompas | com | otomotif | kompas | com
D2	gambir | pensiun | kritik | jonan | luhut | budi | karya | detikfinance
D3	intip | kaya | tambang | emas | grup | bakrie | cnbc | indonesia
D4	berbatov | messi | ronaldo | sulit | bikin | gol | mu | detiksport
D5	chelsea | kepa | bilbao | detiksport

Dari proses stemming kata dasar didapatkan. Kata dasar ini akan dijadikan bahan untuk proses analyzing menggunakan metode TF IDF. Analyzing dengan TF IDF dilakukan untuk mendapatkan nilai bobot.

TF IDF

Metode TF IDF (Term Frequency – Inverse Document Frequency) merupakan formula yang bisa dipakai untuk menghitung bobot (w) dari kata kunci terhadap masing – masing dokumen.

Proses TF IDF dimulai dengan mencari tf dan idf dari kata dasar / term yang telah didapatkan dari proses stemming pada tabel 7 sebelumnya. Adapun prosesnya diilustrasikan pada tabel 8 dan tabel 9.

Tabel 8. Mencari TF

TERM	TF						DF
	Q	D1	D2	D3	D4	D5	
armada	0	1	0	0	0	0	1
bakrie	0	0	0	1	0	0	1
berbatov	0	0	0	0	1	0	1
bikin	0	0	0	0	1	0	1
bilbao	0	0	0	0	0	1	1
budi	0	0	1	0	0	0	1
calya	0	1	0	0	0	0	1
chelsea	0	0	0	0	0	1	1
cnbc	0	0	0	1	0	0	1
com	0	2	0	0	0	0	1
detikfinance	0	0	1	0	0	0	1
detiksport	0	0	0	0	1	1	2
emas	0	0	0	1	0	0	1
express	0	1	0	0	0	0	1
gambir	0	0	1	0	0	0	1
gol	0	0	0	0	1	0	1
grup	0	0	0	1	0	0	1
indonesia	0	0	0	1	0	0	1
intip	0	0	0	1	0	0	1
jonan	0	0	1	0	0	0	1
karya	0	0	1	0	0	0	1
kaya	1	0	0	1	0	0	1
kepa	0	0	0	0	0	1	1
kompas	0	2	0	0	0	0	1
kritik	0	0	1	0	0	0	1
luhut	0	0	1	0	0	0	1
messi	0	0	0	0	1	0	1
mu	0	0	0	0	1	0	1
otomotif	0	1	0	0	0	0	1
pensiun	0	0	1	0	0	0	1
ronaldo	1	0	0	0	1	0	1
sulit	0	0	0	0	1	0	1
taksi	0	1	0	0	0	0	1
tambang	0	0	0	1	0	0	1
toyota	0	1	0	0	0	0	1

Tabel 9. Mencari IDF

TERM	IDF	
	D/df	log(D/df)
armada	5	0.699
bakrie	5	0.699
berbatov	5	0.699
bikin	5	0.699
bilbao	5	0.699
budi	5	0.699
calya	5	0.699
chelsea	5	0.699
cnbc	5	0.699
com	5	0.699
detikfinance	5	0.699
detiksport	2.5	0.398
emas	5	0.699
express	5	0.699
gambir	5	0.699
gol	5	0.699
grup	5	0.699
indonesia	5	0.699
intip	5	0.699
jonan	5	0.699
karya	5	0.699
kaya	5	0.699
kepa	5	0.699
kompas	5	0.699
kritik	5	0.699
luhut	5	0.699
messi	5	0.699
mu	5	0.699
otomotif	5	0.699
pensiun	5	0.699
ronaldo	5	0.699
sulit	5	0.699
taksi	5	0.699
tambang	5	0.699
toyota	5	0.699

Setelah TF dan IDF didapatkan, maka proses selanjutnya adalah mencari nilai pembobotannya. Adapun pembobotan dengan metode TF IDF ditulis seperti rumus 1 berikut ini.

$$w_{i,j} = tf_{i,j} * idf_i \quad (1)$$

Dimana i = kata dasar, j = dokumen, tf = banyaknya kata ditemukan dari i dalam j, $idf_i = \log\left(\frac{D}{df_i}\right)$, D = total jumlah dokumen, df_i = jumlah dokumen yang mengandung suatu term. Hasil dari pembobotan ditunjuukan oleh tabel 10.

Tabel 10. Hasil dari pembobotan dengan TF IDF

TERM	W					
	Q	D1	D2	D3	D4	D5
Armada	0.000	0.699	0.000	0.000	0.000	0.000
bakrie	0.000	0.000	0.000	0.699	0.000	0.000
Berbatov	0.000	0.000	0.000	0.000	0.699	0.000
bikin	0.000	0.000	0.000	0.000	0.699	0.000
bilbao	0.000	0.000	0.000	0.000	0.000	0.699
budi	0.000	0.000	0.699	0.000	0.000	0.000
calya	0.000	0.699	0.000	0.000	0.000	0.000
chelsea	0.000	0.000	0.000	0.000	0.000	0.699
cnbc	0.000	0.000	0.000	0.699	0.000	0.000
com	0.000	1.398	0.000	0.000	0.000	0.000
detikfinance	0.000	0.000	0.699	0.000	0.000	0.000
detiksport	0.000	0.000	0.000	0.000	0.398	0.398
emas	0.000	0.000	0.000	0.699	0.000	0.000
express	0.000	0.699	0.000	0.000	0.000	0.000
gambir	0.000	0.000	0.699	0.000	0.000	0.000
gol	0.000	0.000	0.000	0.000	0.699	0.000
grup	0.000	0.000	0.000	0.699	0.000	0.000
indonesia	0.000	0.000	0.000	0.699	0.000	0.000
intip	0.000	0.000	0.000	0.699	0.000	0.000
jonan	0.000	0.000	0.699	0.000	0.000	0.000
karya	0.000	0.000	0.699	0.000	0.000	0.000
kaya	0.699	0.000	0.000	0.699	0.000	0.000
kepa	0.000	0.000	0.000	0.000	0.000	0.699
kompas	0.000	1.398	0.000	0.000	0.000	0.000
kritik	0.000	0.000	0.699	0.000	0.000	0.000
luhut	0.000	0.000	0.699	0.000	0.000	0.000
messi	0.000	0.000	0.000	0.000	0.699	0.000
mu	0.000	0.000	0.000	0.000	0.699	0.000
otomotif	0.000	0.699	0.000	0.000	0.000	0.000
pensiun	0.000	0.000	0.699	0.000	0.000	0.000
ronaldo	0.699	0.000	0.000	0.000	0.699	0.000
sulit	0.000	0.000	0.000	0.000	0.699	0.000
taksi	0.000	0.699	0.000	0.000	0.000	0.000
tambang	0.000	0.000	0.000	0.699	0.000	0.000
toyota	0.000	0.699	0.000	0.000	0.000	0.000

VSM

VSM merupakan metode yang bisa digunakan untuk melihat kesamaan atau tingkat kedekatan term dengan cara pembobotan term [2]. Salah satu jenis vsm adalah Cosine Similarity. Model jenis ini bekerja dengan mendapatkan similarity (kemiripan) query (Q) pada tiap - tiap dokumen (D_i). Dokumen hampir sama dengan query diindikasikan dari nilai cosine similarity yang cenderung besar. Jika cosine similarity = 1 maka dokumen sama dengan query [3]. VSM dapat ditulis seperti rumus 2.

$$CosSim\ Di = \frac{Q \bullet Di}{|Q||Di|} = \frac{\sum_{i=1}^{n} Q \times D_i}{\sqrt{\sum_{i=1}^{n}(Q)^2}\ x \sqrt{\sum_{i=1}^{n}(D_i)^2}} \quad (2)$$

Dimana $CosSim\ Di$ = cosine similarity yang didapatkan pada tiap dokumen, Q = Vektor Q (kata kunci), Di = Vektor Di, $Q \bullet Di$ = dot product antara vektor Q dan vektor Di, $|Q|$ = panjang vektor Q, $|Di|$ = panjang vektor Di, $|Q||Di|$ = cross product antara |A| dan |Di|, $\sum_{i=1}^{n} Q \times D_i$ = total penjumlahan dot product antara vektor Q dan vektor Di, $\sqrt{\sum_{i=1}^{n}(Q)^2}$ = total perpangkatan pada query kuadrat, $\sqrt{\sum_{i=1}^{n}(D_i)^2}$ = total perpangkatan pada Di kuadrat.

Sesuai dengan rumus 2, tahap selanjutnya adalah mencari total perpangkatan pada Q dan D_i kuadrat. Proses ini menggunakan pembobotan kata dasar sudah dilakukan seperti di tabel 10 sebelumnya. Adapun prosesnya dapat diilustrasikan seperti tabel 11.

Tabel 11. Mencari akar kuadrat Q dan D_i

(SQRT(SUM))

Term	Q^2	D1^2	D2^2	D3^2	D4^2	D5^2

armada	0.000	0.489	0.000	0.000	0.000	0.000
bakrie	0.000	0.000	0.000	0.489	0.000	0.000
berbatov	0.000	0.000	0.000	0.000	0.489	0.000
bikin	0.000	0.000	0.000	0.000	0.489	0.000
bilbao	0.000	0.000	0.000	0.000	0.000	0.489
budi	0.000	0.000	0.489	0.000	0.000	0.000
calya	0.000	0.489	0.000	0.000	0.000	0.000
chelsea	0.000	0.000	0.000	0.000	0.000	0.489
cnbc	0.000	0.000	0.000	0.489	0.000	0.000
com	0.000	1.954	0.000	0.000	0.000	0.000
detikfinance	0.000	0.000	0.489	0.000	0.000	0.000
detiksport	0.000	0.000	0.000	0.000	0.158	0.158
emas	0.000	0.000	0.000	0.489	0.000	0.000
express	0.000	0.489	0.000	0.000	0.000	0.000
gambir	0.000	0.000	0.489	0.000	0.000	0.000
gol	0.000	0.000	0.000	0.000	0.489	0.000
grup	0.000	0.000	0.000	0.489	0.000	0.000
indonesia	0.000	0.000	0.000	0.489	0.000	0.000
intip	0.000	0.000	0.000	0.489	0.000	0.000
jonan	0.000	0.000	0.489	0.000	0.000	0.000
karya	0.000	0.000	0.489	0.000	0.000	0.000
kaya	0.489	0.000	0.000	0.489	0.000	0.000
kepa	0.000	0.000	0.000	0.000	0.000	0.489
kompas	0.000	1.954	0.000	0.000	0.000	0.000
kritik	0.000	0.000	0.489	0.000	0.000	0.000
luhut	0.000	0.000	0.489	0.000	0.000	0.000
messi	0.000	0.000	0.000	0.000	0.489	0.000
mu	0.000	0.000	0.000	0.000	0.489	0.000
otomotif	0.000	0.489	0.000	0.000	0.000	0.000
pensiun	0.000	0.000	0.489	0.000	0.000	0.000
ronaldo	0.489	0.000	0.000	0.000	0.489	0.000
sulit	0.000	0.000	0.000	0.000	0.489	0.000
taksi	0.000	0.489	0.000	0.000	0.000	0.000
tambang	0.000	0.000	0.000	0.489	0.000	0.000
toyota	0.000	0.489	0.000	0.000	0.000	0.000
SUM	0.977	6.840	3.908	3.908	3.578	1.624
SQRT (SUM)	0.988	2.615	1.977	1.977	1.892	1.274

Dilanjutkan dengan mencari total dot product antara Q dan D_i. proses ini diilustrasikan pada tabel 12.

Tabel 12. Mencari dot product Q dan D_i (SUM(Q * D_i))

Term	Q * D1	Q * D2	Q * D3	Q * D4	Q * D5
armada	0.000	0.000	0.000	0.000	0.000
bakrie	0.000	0.000	0.000	0.000	0.000
berbatov	0.000	0.000	0.000	0.000	0.000
bikin	0.000	0.000	0.000	0.000	0.000
bilbao	0.000	0.000	0.000	0.000	0.000
budi	0.000	0.000	0.000	0.000	0.000
calya	0.000	0.000	0.000	0.000	0.000
chelsea	0.000	0.000	0.000	0.000	0.000
cnbc	0.000	0.000	0.000	0.000	0.000
com	0.000	0.000	0.000	0.000	0.000
detikfinance	0.000	0.000	0.000	0.000	0.000
detiksport	0.000	0.000	0.000	0.000	0.000
emas	0.000	0.000	0.000	0.000	0.000
express	0.000	0.000	0.000	0.000	0.000
gambir	0.000	0.000	0.000	0.000	0.000
gol	0.000	0.000	0.000	0.000	0.000
grup	0.000	0.000	0.000	0.000	0.000
indonesia	0.000	0.000	0.000	0.000	0.000
intip	0.000	0.000	0.000	0.000	0.000
jonan	0.000	0.000	0.000	0.000	0.000
karya	0.000	0.000	0.000	0.000	0.000
kaya	0.000	0.000	0.489	0.000	0.000
kepa	0.000	0.000	0.000	0.000	0.000
kompas	0.000	0.000	0.000	0.000	0.000
kritik	0.000	0.000	0.000	0.000	0.000
luhut	0.000	0.000	0.000	0.000	0.000

messi	0.000	0.000	0.000	0.000	0.000
mu	0.000	0.000	0.000	0.000	0.000
otomotif	0.000	0.000	0.000	0.000	0.000
pensiun	0.000	0.000	0.000	0.000	0.000
ronaldo	0.000	0.000	0.000	0.489	0.000
sulit	0.000	0.000	0.000	0.000	0.000
taksi	0.000	0.000	0.000	0.000	0.000
tambang	0.000	0.000	0.000	0.000	0.000
toyota	0.000	0.000	0.000	0.000	0.000
SUM	0.000	0.000	0.489	0.489	0.000

setelah mendapatkan total perpangkatan antara Q dengan Di pada tabel 11 serta mendapatkan total dot product antara Q dan D_i pada tabel 12, proses dilanjutkan dengan menghitung nilai cosine similarity. Proses ini menggunakan implementasi pada rumus 2. Adapun proses perhitungannya adalah sebagai berikut:

CosSim (D1) = sum(Q*D1) / [sqrt(Q^2) * sqrt(D1^2)]
= 0 / [0.98849286 * 2.61530628]
= 0

CosSim (D2) = sum(Q*D2) / [sqrt(Q^2) * sqrt(D2^2)]
= 0 / [0.98849286 * 1.97698572]
= 0

CosSim (D3) = sum(Q*D3) / [sqrt(Q^2) * sqrt(D3^2)]
= 0.488559067 / [0.98849286 * 1.97698572]
= 0.977118134

CosSim (D4) = sum(Q*D4) / [sqrt(Q^2) * sqrt(D4^2)]
= 0.488559067/[0.98849286 * 1.891631497]
= 0.934932114

CosSim (D5) = sum(Q*D5) / [sqrt(Q^2) * sqrt(D5^2)]
= 0 / [0.98849286 * 1.2743757]
= 0

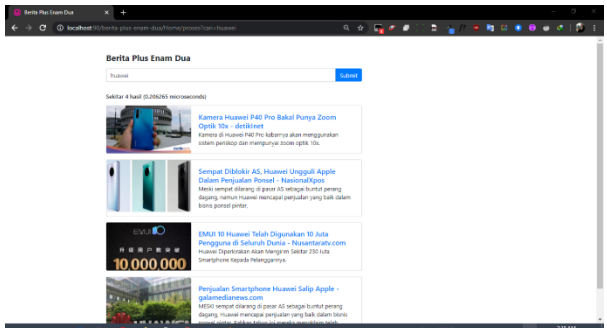
Jadi dari hasil perhitungan diatas dapat diketahui bahwa urutan dokumen yang dihasilkan yaitu urutan pertama D3, dan urutan ke dua D4. Sedangkan dokumen berita D1, D2, dan D5 tidak ditampilkan karena bernilai 0.

HASIL PENELITIAN DAN PEMBAHASAN

Hasil

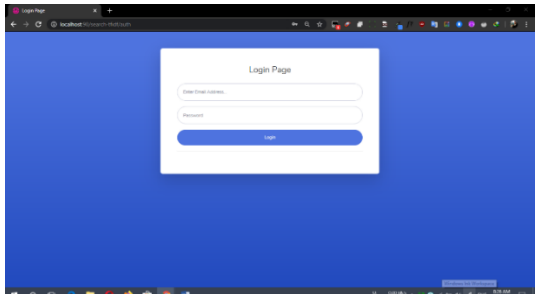
Berita Plus Enam Dua merupakan website yang dibuat peneliti pada penelitian ini. Website ini dibuat dengan framework Codeigniter sebagai implementasi dari metode TF IDF dalam pencarian berita berbahasa Indonesia. Website berita plus enam dua memiliki 2 hak akses yaitu user dan administrator.

User berhak melakukan pencarian berita menggunakan form yang disediakan. Berita – berita dimunculkan menggunakan metode TF IDF. User juga bisa membuka website berita yang dimunculkan dengan cara mengklik salah satu berita. Terdapat penghitung kecepatan pencarian dan jumlah data yang ditampilkan dibawah form pencarian. Fitur ini akan ditampilkan setelah proses pencarian selesai.



Gambar 2 Tampilan dari sisi user

Pada gambar 3 ditunjukkan bahwa pengguna harus login terlebih dahulu untuk mengakses administrator. Mengambil data dari NewsAPI, mengelola dataset, dan melakukan uji coba merupakan tugas utama administrator. Pada sisi ini juga memiliki fitur pendukung yaitu menampilkan profil, edit profil, dan ubah password seperti yang terlihat pada gambar 4.



Gambar 3 Form login untuk mengakses

Gambar 4 Tampilan dashboard setelah login administrator

Pembahasan

Ada 2 uji coba yang dilakukan di penelitian ini, yaitu pertama pengujian kinerja pencarian TF IDF dibandingankan dengan SQL Like Operator menggunakan Precission dan Recall. kedua membandingkan kecepatan pencarian antara TF IDF

dengan SQL Like Operator.

SQL Like Operator

Pencarian dengan memanfaatkan salah satu fitur dari DBMS MySQL yaitu SQL Like Operator. Pada codeigniter, fitur ini peneliti terapkan pada model yang berinteraksi langsung ke tabel news. Fitur ini bekerja jika form pencarian sudah terisi dengan kata kunci pencarian. Kata kunci pencarian selanjutnya akan dibawah ke model dengan menggunakan parameter $query yang selanjutnya akan diproses oleh MySQL dengan operator Like terhadap kolom tittle pada database news. Adapun source code pencarian_model.php adalah sebagai berikut:

Model_pencarian.php

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Pencarian_model extends CI_Model
{
   public function searchByQuery($query)
   {
      $this->db->like('title', $query);
      return $this->db->get('news')->result_array();
   }
}
```

Precission dan Recall

Ukuran efektivitas sebuah sistem temu balik (Information Retrieval) dapat dilihat dari seberapa banyak dokumen yang relevan dengan query dari sistem tersebut hasilkan. Tentunya dengan ditampilkan secara terurut dari dokumen yang memiliki tingkat relevansi paling tinggi sampai yang paling rendah. Pengujian efektivitas kinerja dapat diukur dengan pengujian Precision dan Recall [15]. Precision merupakan perbandingan antara jumlah prediksi benar dengan keseluruhan hasil prediksi yang muncul. Perhitungan precission dirumuskan seperti rumus 3.

$$Precision = \frac{TP}{P}\ x\ 100\% \quad (3)$$

Dimana TP = prediksi muncul benar yang sesuai dengan prediksi yang diharapkan peneliti (PP), P = seluruh prediksi yang dimunculkan sistem.

Recall (sensitivitas) merupakan perbandingan antara jumlah prediksi benar dengan keseluruhan data yang diharapkan peneliti muncul. Perhitungan recall dirumuskan seperti rumus 4.

$$Recall = \frac{TP}{PP}\ x\ 100\% \quad (4)$$

Dimana TP = prediksi muncul benar yang sesuai dengan PP = prediksi yang diharapkan peneliti.

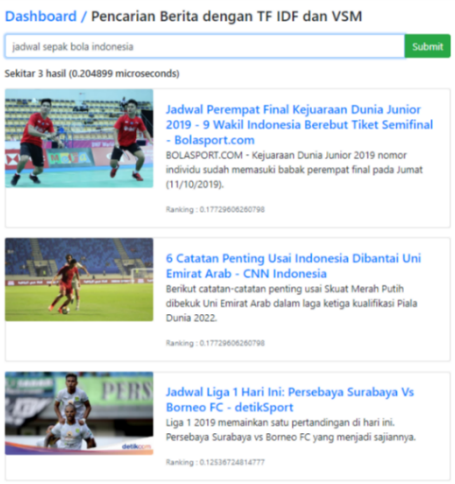
Tabel 13. Lima dataset untuk uji coba.

Id. Dokumen	Judul Berita
D1	Jadwal Perempat Final Kejuaraan Dunia Junior 2019 - 9 Wakil Indonesia Berebut Tiket Semifinal - Bolasport.com
D2	Tinggalkan Real Madrid, Luka Modric Gabung Klub Inggris Ini?
D3	Jadwal Liga 1 Hari Ini: Persebaya Surabaya Vs Borneo FC - detikSport
D4	'Marquez Menurunkan Moral Lorenzo' - detikSport
D5	6 Catatan Penting Usai Indonesia Dibantai Uni Emirat Arab - CNN Indonesia

Tabel 13 merupakan dataset yang digunakan pada penelitian ini. Peneliti mencoba menginputkan pencarian dengan kata kunci (Q) “jadwal sepak bola indonesia” pada website. Peneliti mengharapkan data yang muncul adalah D1 dan D3 (PP =2). Adapun hasilnya dapat dilihat pada tabel 14.

Tabel 14. TF IDF pada uji coba precission dan recall.

TF IDF		
Prediksi (P)	Precission	Recall
3 (D1, D3, dan D5)	Precision = $\frac{2}{3}\ x\ 100\% =$ **66,7%** Dimana prediksi muncul benar (TP) berjumlah 2 (D1 dan D3), dan seluruh prediksi (P) berjumlah 3 (D1, D3, dan D5)	Recall = $\frac{2}{2}\ x\ 100\% =$ **100%** Dimana prediksi muncul benar (TP) berjumlah 2 (D1 dan D3), dan seluruh data yang diharapkan muncul (PP) berjumlah 2 (D1 dan D3)



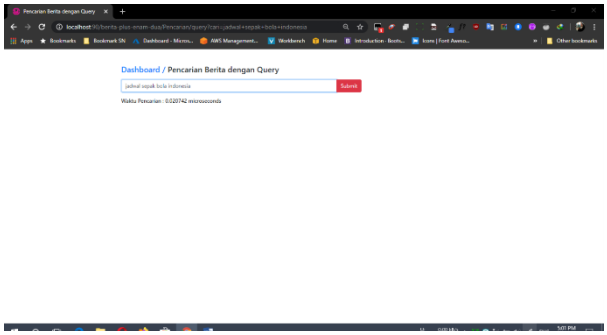
Gambar 5 Hasil pencarian dengan TF IDF

Gambar 5 menunjukkan hasil pencarian menggunakan metode TF IDF.

Tabel 15. SQL Like Operator pada uji coba precission dan recall.

SQL Like Operator		
Prediksi (P)	Precission	Recall
0 (Tidak ada)	Karena prediksi berjumlah 0, maka Precision = **0**%	Karena prediksi berjumlah 0, maka Recall = **0**%

Hasil pencarian dengan SQL Like Operator ditunjukkan oleh gambar 6.



Gambar 6 Hasil pencarian dengan SQL like operator

Kecepatan Pencarian

Peneliti berfokus untuk membandingkan kecepatan 2 macam pencarian yaitu metode TF IDF dan SQL Like Operator (SLO). Peneliti menggunakan 297 dataset yang didapatkan dari NewsAPI pada 24 Desember 2019. Adapun spesifikasi perangkat keras pada localhost yaitu Laptop Acer Aspire E1-451G, Processor AMD A8-4500M APU dengan grafik Radeon HD 1.90 GHz, Memory RAM 8 GB, HDD 500 GB dan hosting yang digunakan yaitu Professional 250 MB, *Unlimited traffic limit*, *Unlimited email account*, Dua domain, *Unlimited* MySQL/MariaDB, pusat data Singapura.

Penelitian tentang kecepatan pencarian berfokus pada beberapa faktor seperti panjang kata pencarian, jumlah dataset dengan term pencarian yang sama, dan perbedaan akses localhost dengan hosting. Adapun hasil pengujiannya dapat dilihat pada tabel 16 dan tabel 17.

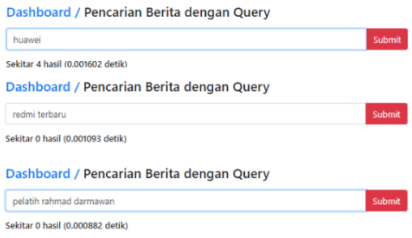
Tabel 16. Hasil uji coba dengan faktor panjang kata (per detik).

Panjang Kata	Rata – rata kecepatan (per detik)			
	TF IDF (Local)	TF IDF (Hosting)	SLO (Local)	SLO (hosting)
Satu kata (huawei)	0.71712	0.38039	0.00248	0.00132
Dua kata (redmi terbaru)	0.74280	0.39980	0.00238	0.00108
Tiga kata (pelatih rahmad darmawan)	0.78651	0.42761	0.00235	0.00098

Gambar 6 TF IDF faktor jumlah term input di localhost

Gambar 7 *TF IDF faktor jumlah term input di hosting*

Gambar 8 SLO faktor jumlah term input



Gambar 9 SLO faktor jumlah term input di hosting di localhost

.

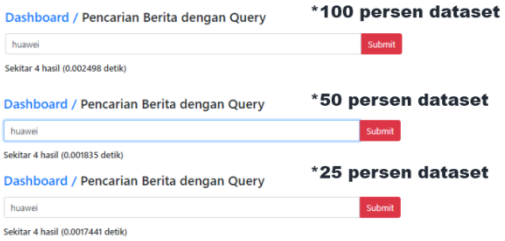
Tabel 17 Hasil uji coba dengan faktor jumlah dataset (per detik).

Jumlah Dataset	Rata – rata kecepatan (per detik)			
	TF IDF (Local)	TF IDF (Hosting)	SLO (Local)	SLO (hosting)
297 (100%)	0.71712	0.38039	0.00248	0.00132
149 (50%)	0.66664	0.11836	0.00195	0.00082
74 (25%)	0.22103	0.06038	0.00165	0.00082

Gambar 10 TF IDF faktor jumlah dataset localhost

Gambar 11 TF IDF faktor jumlah dataset di hosting.

Gambar 12 SLO faktor jumlah dataset di localhost



Gambar 13 SLO faktor jumlah dataset di hosting.

KESIMPULAN

Dengan melihat hasil pada tabel 14 dan tabel 15, pada pengukuran kinerja menggunakan precission dan recall menghasilkan pernyataan bahwa pencarian dengan metode TF IDF lebih baik jika dibandingkan dengan menggunakan SQL Like Operator.

Dengan melihat pada tabel 16 yaitu faktor panjang kata untuk pencarian menghasilkan 2 pernyataan. Pertama, pencarian dengan metode TF IDF dinilai lebih lambat dari SQL Like Operator. Kedua, semakin panjang kata yang diinputkan maka akan memperlambat proses pencarian pada TF IDF. Sedangkan, berbanding terbalik jika menggunakan SQL Like Operator.

Dengan melihat pada tabel 17 yaitu faktor jumlah dataset, maka menghasilkan 2 pernyataan. Pertama, pencarian dengan metode TF IDF lebih lambat dari SQL Like Operator. Kedua, semakin besar dataset maka akan semakin lambat pula proses pencarian pada TF IDF. Hal ini juga berlaku jika menggunakan SQL Like Operator. Dengan melihat faktor akses, membuktikan bahwa akses pencarian melalui hosting dinilai lebih cepat jika dibandingkan jika diakses melalui localhost.

5. Daftar Notasi

i : kata dasar,

j : dokumen,

tf : banyaknya kata ditemukan dari i dalam j,

$idf_i = \log\left(\frac{D}{df_i}\right)$,

D : total jumlah dokumen,

df_i : jumlah dokumen yang mengandung suatu term

$CosSim\ Di$: cosine similarity yang didapatkan pada tiap dokumen,

Q : Vektor Q (kata kunci),

Di : Vektor Di,

$Q \bullet Di$: dot product antara vektor Q dan vektor Di,

$|Q|$: panjang vektor Q,

$|Di|$: panjang vektor Di,

$|Q||Di|$: cross product antara |A| dan |Di|,

$\sum_{i=1}^{n} Q \times D_i$: total penjumlahan dot product antara vektor Q dan vektor Di,

$\sqrt{\sum_{i=1}^{n}(Q)^2}$: total perpangkatan pada query kuadrat,

$\sqrt{\sum_{i=1}^{n}(D_i)^2}$: total perpangkatan pada Di kuadrat.

TP : prediksi muncul benar yang sesuai dengan prediksi yang diharapkan peneliti (PP),

P : seluruh prediksi yang dimunculkan sistem.

TP : prediksi muncul benar yang sesuai

PP : prediksi yang diharapkan peneliti.

REFERENSI

[1] K. Frinta and P. P. Adikara, "Pencarian Berita Berbahasa Indonesia Menggunakan Metode BM25," vol. 3, no. 3, pp. 2589–2595, 2019.

[2] F. Amin, "Implementasi Search Engine (Mesin Pencari) Menggunakan Metode Vector Space Model," *Din. Tek.*, vol. 1, no. 1, pp. 45–58, 2011.

[3] K. D. Putung, A. S. M. Lumenta, and A. Jacobus, "Penerapan Sistem Temu Kembali Informasi Pada Kumpulan Dokumen Skripsi," *J. Tek. Inform.*, vol. 8, no. 1, 2016.

[4] M. A. Rosid, G. Gunawan, and E. Pramana, "Centroid Based Classifier With TF – IDF – ICF for Classfication of Student's Complaint at Appliation E-Complaint in Muhammadiyah University of Sidoarjo," *J. Electr. Electron. Eng.*, vol. 1, no. 1, p. 17, 2016.

[5] R. J. Mooney, "Machine Learning Text Categorization," *Mach. Learn.*, pp. 1–6, 2006.

[6] A. Harahap and M. Agung, *Jurnalistik Televisi: Teknik Memburu dan Menulis Berita*. 2012.

[7] H. Bunyamin, "Algoritma Umum Pencarian Informasi Dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen," *J. Inform. UKM*, vol. 2, no. Mesin Pencari, pp. 85–91, 2005.

[8] G. Block, P. Cibraro, P. Felix, H. Dierking, and D. Miller, *Designing Evolvable Web APIs with ASP.NET*. 2014.

[9] K. Arianto, M. A., Munir, S., dan Khotimah, "Analisis Dan Perancangan Representational State Transfer (Rest) Web Service Sistem Informasi Akademik Stt Terpadu Nurul Fikri Menggunakan Yii Framework," *J. Teknol. Terpadu*, vol. 2, no. 2, 2016.

[10] M. A. Hearst, "Untangling text data mining," pp. 3–10, 1999.

[11] H. Manning, C. D., Raghavan, P., dan Schütze, *An Introduction to Information Retrieval*, no. c. 2009.

[12] Y. Wibisono and M. L. Khodra, "Clustering Berita Berbahasa Indonesia," *Univ. Pendidik. Indones.*, pp. 1–4, 2005.

[13] G. Karyono, F. S. Utomo, A. Sistem, and T. Balik, "Temu Balik Informasi Pada Dokumen Teks Berbahasa Indonesia Dengan Metode Vector Space Retrieval Model," *Semin. Nas. Teknol. Inf. dan Terap. 2012*, vol. 2012, no. Semantik, pp. 282–289, 2012.

[14] F. Sanjaya, "Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model," *J. Inf. Technol.*, vol. 53, no. 9, pp. 1689–1699, 2018.

[15] A. Indriani, Gunawan, and E. Novianto, "Weight Adjusted K-Nearest Neighbor dan Minimum Spanning Tree untuk Information Retrieval System di Perpustakaan STMIK PPKIA Tarakanita Rahmawati Tarakan," *Semin. Nas. Apl. Teknol. Inf. 2013*, pp. 18–22, 2013.

Conflict of Interest Statement:
The author declares that the research was conducted in the absence of any commercial or financialrelation- ships that could be construed as a potential conflict of interest.

Article History:

Received: 2020-01-23 | Accepted: 2020-03-30 | Published: 2020-04-29