PEMBUATAN WEB PORTAL SINDIKASI BERITA INDONESIA DENGAN KLASIFIKASI METODE SINGLE PASS CLUSTERING

Noor Ifada, Husni, Rahmady Liyantanto
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Truojoyo
Jl. Raya Telang PO. BOX 2 Kamal, Bangkalan, Madura, 691962
Telp : (031) 3011146, Fax : (031) 3011506
E-mail: noor.ifada@if.trunojoyo.ac.id, husni@if.trunojoyo.ac.id, liyantanto@gmail.com

ABSTRAK
Banyaknya berita yang terdapat pada media internet telah menyebabkan munculnya berbagai permasalahan, dalam hal teknologi penyimpanan, sistem temu balik, dan pengelompokkan berita itu sendiri. Pada umumnya, pembaca berita cenderung ingin dapat memperoleh inti sari dari berbagai macam berita yang disediakan oleh suatu porta berital. Meskipun web portal berita memiliki fasilitas RSS untuk mempermudah pembaca memperbaharui isinya, namun pada kenyataannya pembaca masih tetap memiliki kecenderungan untuk memperoleh berita yang sama dari setiap portal. Oleh karena itu perlu adanya web portal sindikasi berita yang mampu mengklasifikasi kemiripan berita sehingga tidak ada penggandaan berita dalam satu waktu. Untuk pengklasifikasian berita digunakan metode Single Pass Clustering sebagai algoritma untuk klasifikasi event. Klasifikasi dilakukan pada dokumen berita yang berdekatan untuk dicari kemiripannya. Klasifikasi ini ditekankan untuk dokumen berita berbahasa Indonesia. Kemiripan (similarity) antar berita dapat diukur dari judul dan deskripsi berita. Jika ditemukan dokumen yang memiliki kemiripan maka ia akan menjadi rekomendasi dari berita utamanya. Web portal sindikasi berita ini dapat menjadi referensi berita berbahasa Indonesia yang merupakan kumpulan berita dari beberapa portal berita.

Kata Kunci: single pass clustering, klasifikasi , web portal , berita

1. PENDAHULUAN

Jumlah dokumen berita yang beredar melalui media internet semakin hari semakin bertambah hingga mencapai milyaran dokumen. Ledakan jumlah informasi elektronik menyebabkan timbulnya permasalahan dalam teknologi penyimpanan dan teknologi temu balik (Arifin, 2009). Oleh karena itu pengelompokan dokumen sangat dibutuhkan untuk mempermudah pencarian informasi suatu kejadian atau topik berita tertentu.

Web portal sebagai sarana penyedia layanan informasi berita adalah situs *web* yang menyediakan kemampuan tertentu bagi para pengunjungnya. Setiap *web* portal berita pada umumnya telah diberi fasilitas *RSS* untuk memperoleh berita terbaru dengan tujuan untuk mempermudah pembaca mencari dan memilah berita. Berita yang disajikan dalam berbagai portal berita umumnya memiliki kesamaan topik satu dengan lainnya sehingga menyebabkan pembaca memiliki kecenderungan untuk membaca berita yang kemungkinan sama pada masing-masing portal berita. Jika hal ini terjadi, maka hilanglah salah satu dasar dari sebuah berita sebagai sesuatu yang baru yang diketengahkan bagi khalayak pembaca (Roy, 2007). Padahal kebutuhan pembaca yang hanya ingin membaca inti sari dari setiap kejadian seharusnya dipenuhi oleh *web* portal yang ada.

Oleh karena itu, sebuah *web* portal sindikasi berita dibutuhkan untuk menjawab permasalahan yang muncul akibat banyaknya informasi berita yang ada saat ini. *Web* portal ini akan menjadi referensi berita bagi pembaca dengan sumber portal berita yang sudah terpercaya tanpa menghilangkan kaidah berita itu sendiri.

2. *STEMMING* DENGAN ALGORITMA NAZIEF DAN ANDRIANI

Stemming merupakan bagian yang tidak terpisahkan dalam *Information Retrieval* (*IR*). Algoritma Nazief dan Adriani merupakan algoritma *stemming* untuk teks berbahasa Indonesia yang memiliki prosentase keakuratan lebih baik dari algoritma lainnya (Agusta, 2009). Algoritma ini memiliki tahap-tahap berikut:

1. Cari kata yang akan di-*stemming* dalam kamus kata dasar. Jika ditemukan maka kata diasumsikan sebagai kata dasar dan algoritma berhenti.
2. Jika kata mengandung *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), maka lakukan penghapusan. Kemudian lakukan penghapusan juga jika terdapat *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) dan *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”).
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

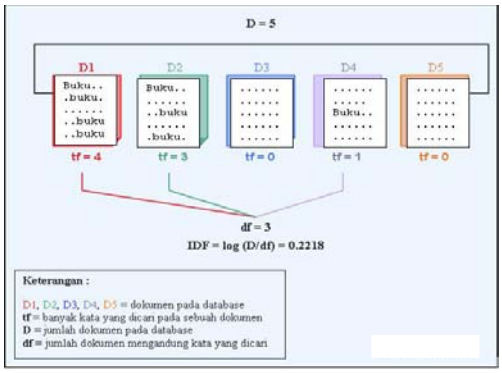
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
   b. For i ← 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika kata dasar belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama maka algoritma berhenti.
5. Melakukan *Recoding*.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil, maka kata awal diasumsikan sebagai kata dasar dan proses selesai.

Untuk mengatasi keterbatasan pada algoritma tersebut, maka ditambahkan aturan-aturan berikut:

1. Aturan untuk reduplikasi.
   a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka kata dasar adalah bentuk tunggalnya, contoh : “buku-buku” maka kata dasarnya adalah “buku”.
   b. Kata lain, misalnya “bolak-balik”, “berbalas-balasan, dan ”seolah-olah”. Untuk mendapatkan kata dasarnya, kedua kata diartikan secara terpisah.
2. Tambahan bentuk awalan dan akhiran serta aturannya.
   a. Untuk tipe awalan “mem-“, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”.
   b. Tipe awalan “meng-“, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

3. KLASIFIKASI KEMIRIPAN BERITA DENGAN *SINGLE PASS CLUSTERING*

Proses klasifikasi dimulai dengan ektraksi untuk mendapatkan kumpulan *term* dari tiap dokumen. Setiap *term* dicari bentuk kata dasarnya dan dilakukan penyaringan terhadap kata-kata yang tidak layak untuk dijadikan pembeda (*stoplist*) dan kemudian dilakukan pencarian representasi nilai dari tiap-tiap dokumen menggunakan *TF/IDF*. *TF/IDF* adalah metode pembobotan yang merupakan integrasi antar *term frequency (tf)*, dan *inverse document frequency (idf)* (Arifin dan Setiono, 2002). Penggunaan metode *TF/IDF* dijelaskan dalam Gambar 1 sedangkan bentuk persamaannya dirumuskan dalam Persamaan (1). Selanjutnya dibentuk suatu vektor antara dokumen dengan kata (*documents with terms*) yang kemudian untuk kesamaan antar dokumen dengan *cluster* akan ditentukan oleh *Single Pass Clustering*.



Gambar 1. Penerapan TF/IDF (Hayatin, 2007)

$$w(t,d) = tf(t,d) * \log(N / nt) \quad (1)$$

Keterangan:

w(t,d) = bobot dari *term* (*t*) dalam dokumen (*d*).
tf(t,d) = frekuensi *term* dalam dokumen (*tf*).
N = banyaknya dokumen yang akan di-*training* untuk penghitungan *TF/IDF*.
nt = jumlah *term* dari semua dokumen yang di-*training*.

Algoritma *Single Pass Clustering* dapat dilakukan dengan langkah berikut:

1. Masukkan (dokumen pertama) D_1 representasi (*cluster* pertama) C_1.
2. Untuk (dokumen ke-*i*) dihitung kesamaan (*similarity*) dengan setiap wakil dari masing-masing *cluster* menggunakan *standard cosine similarity* pada Persamaan (2).

$$S_{D_i D_j} = \frac{\sum_{k=1}^{L} (weight_{ik} weight_{jk})}{\sqrt{\sum_{k=1}^{L} weight_{ik}^2 \sum_{k=1}^{L} weight_{jk}^2}} \quad (2)$$

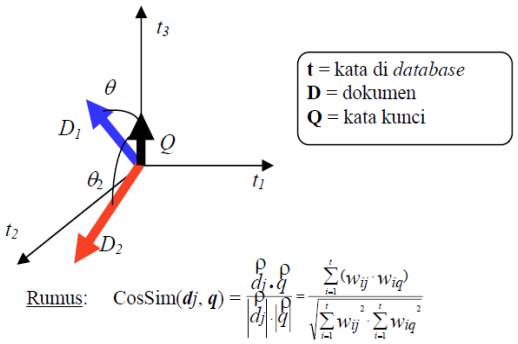
3. Jika (*Maximum Similarity*) S_{max} lebih besar dari batas nilai (*threshold value*) S_T, tambahkan item kepada *cluster* yang bersesuaian dan hitung kembali representasi *cluster*, sebaliknya gunakan D_i untuk inisialisasi *cluster* baru.
4. Jika masih ada sebuah item D_i yang belum dikelompokkan, kembali ke langkah ke-2.

Single Pass clustering cukup handal digunakan sebagai algoritma klasifikasi *event.* Nilai *threshold* yang paling bagus (0.0175) akan menghasilkan nilai *recall-precision* 79 %, dengan nilai *recall* rata-rata 76 % dan *precision* rata-rata 87%.

4. PENCARIAN DENGAN *VECTOR SPACE MODEL*

Vector Space Model (Gambar 2) digunakan untuk mengukur kemiripan antara dokumen dengan *query* (Harjono, 2005). *Query* dan dokumen dianggap sebagai vektor-vektor pada ruang *n*-dimensi, dimana *t* adalah jumlah dari seluruh *term* yang ada dalam leksikon. Leksikon adalah daftar semua *term* yang ada dalam indeks. Selanjutnya dihitung nilai *cosinus* sudut dari dua vektor, yaitu *w* dari tiap dokumen dan *w* dari kata kunci.

 ISSN: 1907-5022



Gambar 2. *Vector Space Model* (Christanty, 2007)

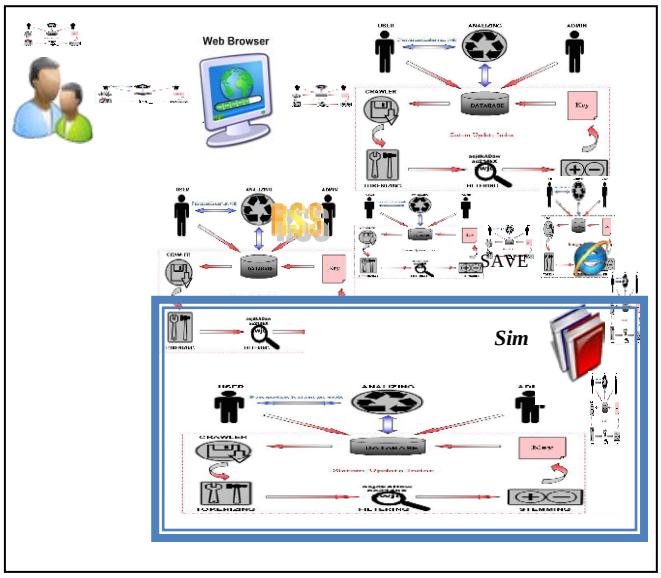
5. PERANCANGAN SISTEM

Sistem yang dirancang pada penelitian ini adalah *web* portal sindikasi berita Indonesia yang akan mengklasifikasi kemiripan antar berita dengan menggunakan Algoritma *Single Pass Clustering*. Aplikasi ini membaca sepuluh *web* berita terpopuler di Indonesia yang dijadikan sebagai sumber berita. Proses klasifikasi kemiripan dokumen dilakukan di belakang sistem utama sehingga tidak tampak dan dijalankan secara otomatis menggunakan *crontab*. Klasifikasi kemiripan berita dilakukan pada berita yang berdekatan sehingga menghasilkan sebuah rekomendasi berita bagi pembaca.

Proses indeksing berita juga dilakukan di belakang sistem dengan menggunakan *crontab* yang telah dikonfigurasi waktunya dengan tujuan untuk mempercepat proses pencarian berita dengan *vector space model*. Indeksing dilakukan setelah berita selesai di-*stemming* dan pengurangan *stoplist*, untuk memperoleh *term* yang menjadi kata kunci berita. Kata tersebut dihitung nilai frekuensi kemunculannya dalam dokumen untuk dilakukan proses *vector space model* nanti pada saat pencarian.

Rancangan aplikasi *web* portal sindikasi berita Indonesia ini diperlihatkan pada Gambar 3. Proses-proses yang terjadi di dalamnya adalah:

1. *User* adalah pembaca yang akan melihat referensi berita.
2. *Admin* merupakan pihak yang berhak melakukan *update* dan kontrol pada *database*.
3. *Database*, sebagai tempat penyimpanan *URL web* portal berita dan data penting lainnya, seperti konfigurasi *web* portal, eleman dari berita yang ada, dan kumpulan kata yang digunakan untuk proses *stemming*.
4. *Analizing*, proses menampilkan berita dari yang ada *database* ke user.
5. *Crawler*, proses membaca berita dari *web portal* melalui *URL* yang telah tersimpan pada *database*. Pembacaan berita menggunakan RSS yang ada pada *web* portal, untuk kemudian disindikasi *title*, deskripsi , tanggal *update*, dan *link* halaman berita.
6. *Tokenizing*, proses pemenggalan kata pada dokumen berdasarkan *spasi* dan tanda penghubung (-).
7. *Filtering*, proses penghilangan kata-kata (*stoplist*) sebagai kata yang jarang dicari atau jarang digunakan sebagai *keywords* pada proses pencarian (Wibisono, 2008). Proses ini dilakukan untuk mengurangi waktu yang dibutuhkan pada saat penghitungan frekuensi tiap kata pada dokumen.
8. *Stemming*, proses mengubah kata menjadi bentuk kata dasarnya berdasarkan kamus kata dasar (Lanin dan Hardiyanto, 2009).
9. *Similarity* merupakan proses mencocokkan keterkaitan antar berita yang satu dengan berita lainnya dengan melihat kemiripan antar dokumen. Paramater yang digunakan dalam penentuan kemiripan tersebut adalah judul dan deskripsi dari berita itu sendiri.
10. *Key*, proses ini menyimpan *term title*, dan deskripsi berita, sebagai kata kunci dokumen untuk disimpan pada *database*.



Gambar 3. Proses Pada *Web* Portal

6. HASIL DAN PEMBAHASAN

6.1 Klasifikasi Kemiripan Dokumen Berita

Web portal akan melakukan klasifikasi kemiripan antara dua dokumen yang berdekatan. Klasifikasi menggunakan Algoritma *Single Pass Clustering* dengan *threshold* sebesar 0.0175 untuk melihat kemiripan berita (Arifin, 2009). Jika dokumen berita tersebut memiliki kesamaan, maka berita pertama akan menjadi berita utama dan berita kedua menjadi sebuah rekomendasi berita utamanya. Klasifikasi kemiripan melihat dari judul dan diskripsi berita. Perhatikan contoh Dokumen 1 dan 2 berikut:

Dokumen 1 (D_1): **"**Polres Kudus Tangkap Penjual DVD Mesum Ariel-Luna Aparat Polres Kudus, Jawa Tengah, menangkap penjual cakram dijital (DVD) mesum yang diperankan oleh artis mirip Ariel dan Luna Maya, Selasa (67)."

 ISSN: 1907-5022

Dokumen 2 (D_2): **"**Alat Kelamin Ariel Juga Diukur Polisi Pemeriksaan fisik terhadap para pelaku yang diduga menjadi pemain dalam video porno yang melibatkan Ariel, Luna Maya dan Cut Tari telah selesai dilakukan. Tidak ada satu bagian pun dari tubuh vokalis Peterpan itu yang luput dari pemeriksaan polisi."

Dokumen D_1 dan D_2 akan melalui proses *token* untuk menghilangkan tanda baca, angka, dan lainnya. Kemudian dilakukan pembuangan kata-kata yang termasuk *stoplist* dan proses *stemming* untuk mencari kata dasarnya. Masing-masing kata akan dihitung bobotnya dengan menggunakan *TF/IDF*.

Tabel 1. Pembobotan Kata dengan *TF/IDF* pada D_1 dan D_2.

Token	***TF*(D_1)**	***TF*(D_2)**	***IDF***	***w*(D_1)**	***w*(D_2)**
polres	2	0	0	0	0
Kudus	2	0	0	0	0
tangkap	2	0	0	0	0
jual	2	0	0	0	0
dvd	2	0	0	0	0
mesum	2	0	0	0	0
ariel	2	2	-1	-2	-2
luna	2	1	-0.5849	-1.1698	-0.5849
aparat	1	0	1	1	0
jawa	1	0	1	1	0
cakram	1	0	1	1	0
dijital	1	0	1	1	0
peran	1	0	1	1	0
artis	1	0	1	1	0
mirip	1	0	1	1	0
maya	1	1	0	0	0
alat	0	1	1	0	1
kelamin	0	1	1	0	1
ukur	0	1	1	0	1
polisi	0	2	0	0	0
pemeriksaan	0	2	0	0	0
fisik	0	1	1	0	1
pela	0	1	1	0	1
main	0	1	1	0	1
video	0	1	1	0	1
porno	0	1	1	0	1
libat	0	1	1	0	1
cut	0	1	1	0	1
tari	0	1	1	0	1
selesai	0	1	1	0	1
tubuh	0	1	1	0	1
vokalis	0	1	1	0	1
peterpan	0	1	1	0	1
luput	0	1	1	0	1

Berdasarkan Persamaan (1), nilai *TF* didapatkan dari frekuensi *term* dalam satu dokumen dan nilai *IDF* merupakan log dari hasil bagi *DF* terhadap *D*. Nilai *D* itu sendiri merupakan banyaknya dokumen sedangkan nilai *DF* didapatkan dari jumlah *term* seluruh dokumen. Bobot (*w*) diperoleh dari hasil perkalian *TF* dan *IDF*.

Setelah diketahui nilai *TF/IDF* suatu berita, maka seluruh *token* akan disimpan di dalam tabel khusus untuk indeksing agar nantinya dapat digunakan untuk mempercepat proses pencarian berita. Hasil pembobotan kata untuk D_1 dan D_2 diperlihatkan pada Tabel 1**.**

Similarity antara D_1 dengan D_2 dihitung dengan menggunakan Persamaan (2). Maka diperoleh $Sim(D_1,D_2)=0.2953$. Oleh karena $Sim(D_1,D_2)$ melebihi *threshold* (0.0175) maka kedua dokumen dinyatakan memiliki kemiripan dan D_2 akan menjadi rekomendasi berita terhadap D_1. Gambar 4 memperlihatkan contoh dokumen berita yang memiliki kemiripan dengan berita utamanya dan kemudian menjadi sebuah rekomendasi.

Gambar 4. Contoh Implementasi Klasifikasi Kemiripan Berita (D_1 dan D_2).

Sebagai contoh lainnya adalah Dokumen 3 dan 4 yang memperlihatkan ketidakmiripan berita:

Dokumen 3 (D_3) : "Cici Tegal Dipanggil KPK KPK akan memeriksa pelawak Cici Tegal dan pesinetron Meidiana Hutomo terkait dugaan korupsi pengadaan alat rontgen portable di Kementerian Kesehatan pada 2007."

Dokumen 4 (D_4) : "Aktivis ICW Dianiaya Aktivis ICW, Tama Satya Langkun, dianiaya oleh segerombolan orang yang tak dikenal di kawasan Duren Tiga, Jakarta Selatan, Kamis dini hari."

Pembobotan kata untuk D_3 dan D_4 dengan menggunakan *TF/IDF* diperlihatkan pada Tabel 2.

Similarity antara D_3 dengan D_4 diperoleh $Sim(D_3,D_4)=0$. Oleh karena $Sim(D_3,D_4)$ kurang dari *threshold* yang sudah ditetapkan, kedua dokumen dinyatakan tidak memiliki kemiripan. Jadi, D_3 dan D_4 akan menjadi sebuah berita utama. Gambar 5 memperlihatkan contoh dokumen berita utama sebagai hasil ketidakmiripan berita antara keduanya.

ISSN: 1907-5022

Tabel 2. Pembobotan Kata dengan *TF*/*IDF* pada D_3 dan D_4.

Token	*TF*(D_3)	*TF*(D_4)	*IDF*	*w*(D_3)	*w*(D_4)
cici	2	0	0	0	0
tegal	2	0	0	0	0
panggil	1	0	1	1	0
kpk	2	0	0	0	0
memeriksa	1	0	1	1	0
lawak	1	0	1	1	0
sinetron	1	0	1	1	0
meidiana	1	0	1	1	0
hutomo	1	0	1	1	0
duga	1	0	1	1	0
korupsi	1	0	1	1	0
ada	1	0	1	1	0
alat	1	0	1	1	0
rontgen	1	0	1	1	0
portable	1	0	1	1	0
menteri	1	0	1	1	0
sehat	1	0	1	1	0
aktivis	0	2	0	0	0
icw	0	2	0	0	0
aniaya	0	2	0	0	0
tama	0	1	1	0	1
satya	0	1	1	0	1
langkun	0	1	1	0	1
gerombol	0	1	1	0	1
kenal	0	1	1	0	1
kawasan	0	1	1	0	1
duren	0	1	1	0	1
jakarta	0	1	1	0	1
dini	0	1	1	0	1

Gambar 5. Contoh Implementasi Ketidakmiripan Berita (D_3 dan D_4).

6.2 Pencarian Dokumen Berita

Vector space model (*VSM*) digunakan sebagai metode pencarian berita. Metode ini akan mencari bobot kata pada masing-masing dokumen dan mengurutkan berdasarkan nilai tertinggi. Pencarian dimulai pada saat *user* memasukkan *keyword*. Pertama kali yang dilakukan adalah melakukan pengecekan pada tabel *cache* yang merupakan tempat penyimpanan kumpulan *keyword* yang pernah dimasukkan oleh *user*. Jika keyword tidak terdapat di dalam tabel, maka proses *VSM* dimulai dengan melakukan *tokenizing*, *filtering*, dan *stemming* terhadap *keyword*. Tabel *token* akan mencocokkan *keyword* yang dicari dengan *term* yang ada pada berita. Nilai *VSM* dihitung dengan melihat banyaknya kemunculan *term* pada dokumen dikalikan dengan *keyword* yang dicari. Kemiripan berita akan diperoleh dari perbandingan bobot *query* dengan bobot kata pada dokumen. Selanjutnya, berita yang mengandung *keyword* pencarian akan ditampilkan dan *keyword* disimpan dalam tabel *cache*. Perhatikan contoh pencarian dengan *keyword* "kpk dan icw" berikut:

Dokumen 5 (D_5): "aktivis icw dianiaya aktivis icw, tama satya langkun, dianiaya oleh segerombolan orang yang tak dikenal di kawasan duren tiga, jakarta selatan, kamis dini hari."

Dokumen 6 (D_6) : "cici tegal dipanggil kpk kpk akan memeriksa pelawak cici tegal dan pesinetron meidiana hutomo terkait dugaan korupsi pengadaan alat rontgen portable di kementerian kesehatan pada 2007."

Tabel 3. Kemunculan Kata pada Dokumen Pencarian D_5 dan D_6.

Token	*Keyword*(kk)	D_5	D_6
icw	1	2	0
kpk	1	0	2
aktivis	0	2	0
aniaya	0	2	0
tama	0	1	0
satya	0	1	0
langkun	0	1	0
gerombol	0	1	0
kenal	0	1	0
kawasan	0	1	0
duren	0	1	0
jakarta	0	1	0
dini	0	1	0
cici	0	0	2
tegal	0	0	2
panggil	0	0	1
memeriksa	0	0	1
lawak	0	0	1
sinetron	0	0	1
meidiana	0	0	1
hutomo	0	0	1
duga	0	0	1
korupsi	0	0	1
ada	0	0	1
alat	0	0	1
rontgen	0	0	1
portable	0	0	1
menteri	0	0	1
sehat	0	0	1
$\|q\|$ dan $\|d\|$	1.4142	4.58	5.09
$q * d$		4	2
$\|q\| * \|d\|$		6.4806	7.211
$q * d/\|q\| * \|d\|$		0.6172	0.2774
θ		51.888	73.8949

Keterangan:
- q = jumlah *term* k*eyword* pencarian
- d = jumlah *term* pada dokumen
- $|q|$ = akar dari jumlah *term keyword* pencarian
- $|d|$ = akar dari jumlah *term* pada dokumen
- θ = sudut *tetha*

Setelah proses *token* D_5 dan D_6, diperoleh daftar kemunculan kata yang diperlihatkan dalam Tabel 3. Contoh hasil pencarian diperlihatkan oleh Gambar 6.

Berdasarkan perhitungan *VSM*, *cosine* D_5 adalah 0.6172 sedangkan *cosine* D_6 adalah 0.2774. Dari hasil akhir *cosine* tersebut dapat diketahui bahwa D_5 memiliki tingkat *similarity* lebih tinggi dibandingkan D_6. Sehingga apabila diurut berdasarkan kemunculannya maka yang pertama muncul adalah D_5 dan kemudian disusul oleh D_6.

Gambar 6. Contoh Implementasi Pencarian Dokumen Berita (D_5 dan D_6).

7. KESIMPULAN

Berdasarkan hasil dan pembahasan dapat ditarik kesimpulan sebagai berikut:

1. *Web* portal ini mampu mengklasifikasi dokumen yang berdekatan untuk dicari kemiripannya. Dengan menggunakan Algoritma Nazief dan Adriani untuk s*temming* kata, dan Algoritma *Single Pass Clustering* untuk mencari kemiripan berita (dengan batas *threshold* 0.0175)
2. Dokumen berita D_1 dan D_2 memperlihatkan contoh kemiripan berita dengan hasil *similarity* yang melebihi *threshold* yaitu *Sim*(D_1,D_2) = 0.2953. Dengan demikian dapat D_2 menjadi rekomendasi berita bagi D_1.
3. Dokumen berita D_3 dan D_4 memperlihatkan contoh ketidakkemiripan berita dengan hasil *similarity* yang kurang dari *threshold* yaitu Sim(D_3,D_4) = 0. Dengan demikian, baik D_3 maupun D_4 menjadi suatu berita utama.
4. Dengan proses indeksing dan pembuatan *cache*, pencarian dengan *Vector Space Model* (*VSM*) dapat berjalan lebih cepat dan efisien. *VSM* menjawab permasalahan yang dihadapi pencarian menggunakan *TF/IDF* karena dalam *VSM* tidak ada bobot yang sama pada dokumen.
5. Pencarian berita dengan menggunakan *keyword* "kpk dan icw" merujuk ke dokumen berita D_5 dan D_6. D_5 dimunculkan di posisi atas karena memiliki *similarity* yang lebih tinggi daripada D_6, yaitu 0.6172 terhadap 0.2774.

PUSTAKA

Agusta, L. (2009). Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief dan Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia. *Konferensi Nasional Sistem dan Informatika 2009*. Diakses pada 10 Juli 2010 dari http://yudiagusta.files.wordpress.com/2009/11/196-201-knsi09-036-perbandingan-algoritma-stemming-porter-dengan-algoritma-nazief-adriani-untuk-stemming-dokumen-teks-bahasa-indonesia.pdf .

Arifin, A. Z. (2009). *Penggunaan Digital Tree Hibrida pada Aplikasi Information Retrieval untuk Dokemen Berita*. Diakses pada 10 Maret 2010 dari http://www.its.ac.id/personal/files/pub/669-agusza-DigitalTreePaper.pdf .

Arifin, A. Z., dan Setiono, A. N. (2002). Klasifikasi Dokemen Berita Kejadian Berbahasa Indonesia dengan Algoritma *Single Pass Clustering*. *Proceeding of Seminar on Intelligent Technology and Its Applications (SITIA)*, Teknik Elektro, ITS. Diakses pada 10 Maret 2010 dari http://www.its.ac.id/personal/files/pub/667-agusza-SITIAKlasifikasiEvent.pdf .

Christanty, M. H. (2007). *Penerapan Web-Based Knowledge Management System Untuk Manajemen Pengalaman Dan Logistik Pasca Bencana Alam*. Tugas Akhir. Surabaya: Jurusan Teknologi Informasi ITS. Diakses pada 2 Juli 2010 dari http://student.eepis-its.edu/~okoj/New%20Folder/7403040021.pdf.

Harjono, K. D. (2005). Perluasan Vektor Pada Metode *Search Vector Space. Integral, 10(2).*. Diakses pada 15 Juli 2010 dari http://home.unpar.ac.id/~integral/Volume%2010/Integral%2010%20No%202/Perluasan%20Vektor.pdf.

Hayatin, N. (2007). *Pembuatan Mesin Pencari Berdasarkan Kata Kunci Menggunakan Metode Tf/Idf.* Tugas Akhir. Surabaya: Jurusan Teknologi Informasi ITS.

Lanin, I dan Hardiyanto, R. (2009). *Bahasa dan Terjemahan Indonesia*. Diakses pada 20 Juni 2010 dari http://www.bahtera.org/kateglo/?mod=dictionary&action=view&phrase=kamus.

Roy. (2007). *Berita*. Diakses pada 15 Maret 2010 dari http://www.beritanet.com/Education/Berita-Jurnalistik/berita.html.

Wibisono, Y. (2008). *Stop Words Untuk Bahasa Indonesia*. Diakses pada 20 Juli 2010 dari http://yudiwbs.wordpress.com/2008/07/23/stop-words-untuk-bahasa-indonesia.