JURNAL TEKNOLOGI TECHNOSCIENTIA ISSN: 1979-8415
Vol. 2 No. 1 Agustus 2009

PENINGKATAN EFEKTIVITAS PENYAJIAN *SEARCH RESULT* DARI SISTEM TEMU KEMBALI INFORMASI MENGGUNAKAN CLUSTERING DOKUMEN

Amir Hamzah[1]

[1]*DosenTetap Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Sains & Teknologi AKPRIND Yogyakarta*

Masuk: 2 April 2009, revisi masuk: 11 Juli 2009, diterima: 15 Juli 2009

ABSTRACT

The fast expansion of text information volume has caused the difficulty of information retrieval process, mainly on the model of word-based matching. The synonymy factor of word has caused non relevant document to be retrieved, whereas the polisemy factor has caused relevant document not to be retrieved. The application of document clustering to the search results before presented to the user can increase the effectiveness of retrieval. This study elaborates the application of document clustering to improve the effectiveness of retrieval by clustering to the search result before presented to the user. Three clustering algorithms from partitional approach i.e. K-Means, Bisecting K-Mean and Buckshot, and hierarchical agglomerative approach with two cluster similarity function i.e. UPGMA and Complete Link were chosen. The performance parameter was measured using F-measure, a metric derived from Precision and Recall of retrieval. The document collections to be tested are 1000 news document and 350 academic abstract documents. The results show that the presentation of search results by using clustering has improved the number of relevant document in the up-level ranks. The improvement was statistically significant compare to the page-rank method. The improvement of F-measure as a performance metric is about 14,34% for news documents and 28,18% for abstract documents.

Keywords*: search result clustering, retrieval effectiveness, F-measure.*

INTISARI

Perkembangan volume informasi teks yang cepat telah menyebabkan kesulitan proses temu kembali informasi terutama pada model berbasis pencocokan kata. Faktor sinonim kata menyebabkan dokumen tidak relevan dipanggil sementara faktor polisemy menyebabkan dokumen tidak relevan dipanggil. Aplikasi clustering dokumen pada hasil pencarian sebelum disajikan kepada pengguna dapat meningkatkan efektivitas temu kembali. Kajian ini meneliti aplikasi clustering dokumen untuk meningkatkan efektivitas temu kembali dengan melakukan clustering pada hasil pencarian sebelum disajikan kepada pengguna. Dipilih tiga algoritma clustering pendekatan partisi, yaitu dari K-Means, Bisecting K-Mean dan Buckshot, serta pendekatan dari agglomerative menggunakan dua fungsi similaritas, yaitu UPGMA dan Complete Link. Parameter unjuk kerja ini diukur menggunakan parameter F-measure, suatu ukuran yang diturunkan pada Precisi on dan Recall dari temu kembali. Koleksi dokumen yang diuji terdiri dari 1000 dokumen berita dan 350 dokumen abstrak akademik. Hasil penelitian ini menunjukkan bahwa penyajian hasil pencarian dengan menggunakan clustering telah meningkatkan jumlah dokumen relevan pada ranking atas. Peningkatan signifikan secara statistik dibandingkan metode page-rank. Peningkatan nilai F-measure tersebut sebagai ukuran kira-kira sebesar 14,34% untuk dokumen berita dan 28,18% untuk dokumen abstrak akademik.

Kata kunci : clustering hasil pencarian, efektivitas temu kembali, F-measure

[1]*miramzah@yahoo.co.id*

PENDAHULUAN

Penerapan teknologi digital dan jaringan komputer telah menyebabkan terjadinya "ledakan" informasi yang berkembang eksponensial. Pada strategi ini pencarian *query* berbasis kata (*word-matching*) kesulitan yang dijumpai muncul dari aspek bahasa, yaitu faktor sinonim pada kata yang telah menyebabkan dokumen yang tidak relevan akan terus dipanggil hanya semata-mata karena dokumen ini mengandung kata yang ada dalam *query*. Sebaliknya faktor *polisemy*, yaitu keadaan di mana suatu kata dapat memiliki lebih dari satu makna, menyebabkan ada dokumen relevan dalam koleksi yang tidak dipanggil karena tidak memuat kata yang ada dalam *query*. Kesulitan ini semakin kompleks manakala pada kenyataannya koleksi dokumen cenderung bertambah besar dan akan menghasilkan (*search result*) yang berpresisi rendah dikatakan oleh (Zamir, 1999; Tombros, 2002).

Menurut Rijbergen (1979), *clustering* dokumen telah lama diterapkan untuk meningkatkan efekifitas temu kembali informasi. Penerapan *clustering* ini bersandar pada suatu hipotesis (*cluster-hypothesis*) bahwa dokumen yang relevan akan cenderung berada pada kluster yang sama jika pada koleksi dokumen dilakukan *clustering*. Beberapa penelitian ini untuk dokumen berbahasa Inggris menerapkan *clustering* dokumen untuk memperbaiki kinerja dalam proses *searching oleh* (Frakes and Baeza-Yates, 1992; Salton, 1989; dan Tombros, 2002). Sedangkan perbaikan dalam penyajian hasil *search* ini dilakukan oleh antara lain Cutting et.al.(1992), Zamir (1999), Osinki(2004) dan Widyantoro (2007). Untuk dokumen berbahasa Indonesia penelitian bidang IR adalah oleh Vega (2001) dan Tala (2004) yang meneliti efek *stemming* pada hasil pencarian. Penelitian penerapan *clustering* untuk perbaikan kinerja perolehan informasi untuk dokumen berbahasa Indonesia belum pernah dilakukan. Hal ini mengingat secara umum penelitian tentang komputasi bahasa untuk dokumen Bahasa Indonesia juga masih sangat minim (Nazief, 2000). Permasalahan dalam penelitian ini adalah bagaimana mengemas suatu hasil pencarian sedemikan sehingga dokumen yang relevan terhadap *query* akan mengelompok dalam kelompok teratas.Metode yang diajukan adalah de-ngan cara melakukan *clustering* pada hasil pencarian linear dan mencari label pada tiap-tiap kluster kemudian menyajikan kepada pengguna dengan petunjuk label kluster tersebut. Penelitian ini memiliki batasan model yaitu model ruang vektor dengan uji coba sistem berupa dokumen teks berita berbahasa Indonesia. Dari penelitian ini diharapkan dapat dirancang suatu sistem temu kembali informasi yang memiliki kinerja yang lebih baik dibandingkan dalam menyajikan informasi yang selama ini berupa *list* dokumen yang sangat panjang dan membosankan bagi pengguna untuk melakukan *browsing* secara satu-persatu guna menemukan dokumen relevan yang dicari. Model ruang vektor untuk koleksi dokumen mengandaikan dokumen sebagai sebuah vektor dalam ruang kata (*feature*). Jika koleksi n buah dokumen dapat diindeks oleh t buah *term*/ *feature* maka suatu dokumen dapat dipandang sebagai vektor berdimensi t dalam ruang term tersebut. Koleksi dokumen diwakili matrik kata-dokumen X:

$$X = \{x_{ij}\}\ i= 1,2,..t\ ;\ j =1,2,..\ n \qquad (1)$$

x_{ij} adalah bobot term i dalam dokumen ke j. Pembobotan dasar yang dilakukan dengan menghitung frekuensi kemunculan *term* dalam dokumen karena dipercaya bahwa frekuensi kemunculan *term* merupakan petunjuk sejauh mana *term* tersebut mewakili isi dokumen.

$$x_{ij}= tf_i * \log(n/df_i);\ i=1,2,.,t\ ;\ j=1,2,.,n \qquad (2)$$

Menurut Luhn (1958), kekuatan pembeda terkait dengan frekuensi term (*term-frequency*, *tf*), di mana *term* yang memiliki kekuatan diskriminasi adalah *term* dengan frekuensi sedang. Pembobotan baku yang digunakan adalah *term-frequency invers-document freqeuency* (TF-IDF) [1] sebagai berikut:

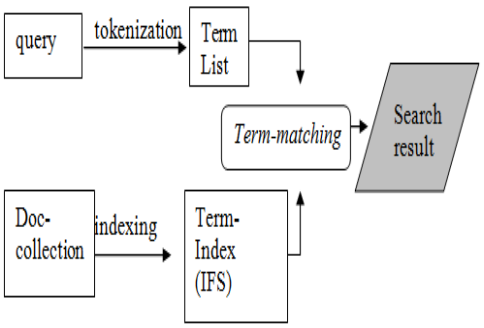
dengan t=total *term* dalam index, n=total dokumen dalam koleksi, df_i=total dokumen yang mengandung *term* ke-i.

Dalam proses *clustering*, kesamaan antara dokumen D_i dengan dokumen D_j umumnya diukur dengan fungsi similaritas tertentu. Menurut Chisholm and Kolda(1999) untuk tujuan *clustering* dokumen fungsi yang baik adalah fungsi similaritas Cosine, berikut:

$$Cosine\text{-sim}(D_i,D_j)= \frac{\sum_{k=1}^{t} D_{ik}D_{j_k}}{\sqrt{\sum_{k=1}^{t}(D_{ik})^2\sum_{k=1}^{t}(D_{jk})^2}} \quad .. \quad (3)$$

Jika vektor Di dan Dj masing-masing ternormalisasi sehingga masing-masing panjangnya satu, maka fungsi *cosine* menjadi:

$$Cosine\text{-sim}(D_i,D_j) = \sum_{k=1}^{t} D_{ik}D_{jk} \quad .... \quad (4)$$



Gambar 1. Pencarian *query* berbasis kata model IFS

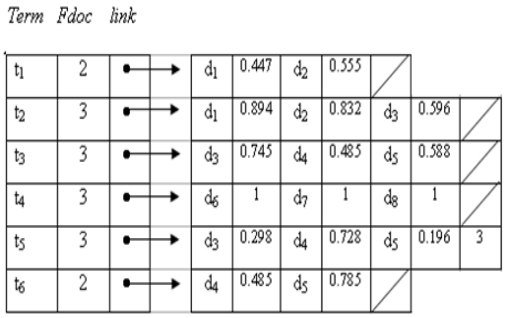
SRClus ini dimaksudkan untuk meningkatkan efektifitas *retrieval* dari mesin pencari. Pada model IFS hasil pencarian disajikan berupa lajur daftar panjang dokumen yang "dianggap" relevan oleh sistem. Dan pada kenyatannya karena pengukuran similaritas hanya dilakukan antara *query* dengan dokumen dan *ran-king* dalam daftar jawaban *search result* adalah didasarkan pada tingkat similaritas *query*-dokumen tanpa melihat similaritas antar dokumen maka kasus yang sering terjadi adalah dokumen yang sebenarnya relevan terhadap *query* karena kebetulan frekuensi kata *query*nya kecil akan berada pada ranking bawah. Sebaliknya suatu dokumen yang sebenarnya tidak relvan terhadap

Dalam pemrosesan *query*, simi-laritas antara *query* Q dengan dokumen Di juga dapat digunakan formula pada persamaan (4), yaitu:

$$Cosine\text{-sim}(Q,D_i) = \sum_{k=1}^{t} Q_k D_{ik} \quad ..... \quad (5)$$

Ada berbagai strategi pencarian (*search strategies*) dalam IR antara lain: *boolean search*, *inverted file search*, *probabilistic search, extended boolean search* (Salton, 1989). Dari model-model *search* tersebut yang banyak digunakan adalah *inverted files search* (IFS) karena alasan efisiensi.

Sekema IR model IFS dapat dilihat seperti pada Gambar 1. Dalam *indexing* model IFS *term* terindeks akan menunjuk pada *list* yang memuat daftar dokumen yang mengandung *term* tersebut pada Gambar 2, sehingga jika suatu *query* diberikan akan dengan cepat diberikan jawaban daftar dokumen yang memuat *term* tersebut.



Gambar 2. Struktur data pada pencarian *query* model IFS

query karena kebetulan mengandung kata *query* dengan frekuensi besar akan berada paaa ranking atas.

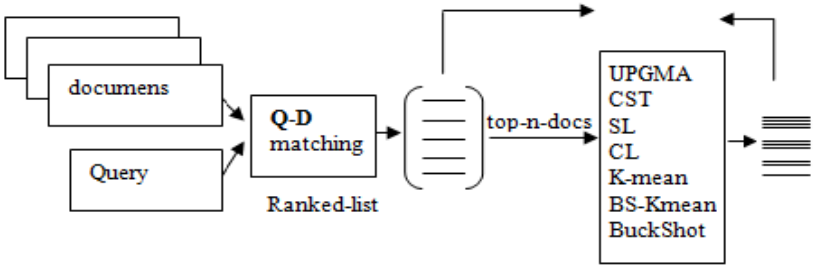
Pada SRClus tersebut dapat diasumsikan bahwa hasil pemrosesan IFS masih merupakan kumpulan dokumen yang secara ideal semua relevan terhadap *query*, tetapi secara kenyataan tetap mengandung dokumen yang tidak relevan. Dokumen tidak relevan ini jika jumlahnya banyak dan berada pada ranking atas dalam daftar *search result* maka pengguna akan menemukan kesulitan mencari dokumen yang relevan sesungguhnya yang kebetulan ada di ranking bawah. Dengan asumsi jika dilakukan *clustering* pada hasil ini maka sesuai dengan *cluster hypothesis* bahwa

dokumen relevan akan mengelompok dengan dokumen yang relevan dan sebaliknya. Pusat kluster ini selanjutnya dapat digunakan sebagai representasi kluster tersebut yang dapat diukur pada tingkat similaritasnya terhadap *query*. Pusat kluster yang merupakan rata-rata vektor dokumen dalam kluster tersebut juga dapat digunakan mengekstrak kata-kata yang dapat dijadikan sebagai label kluster yang mewakili tentang apa kluster tersebut. Algoritma SRCLus dapat dituliskan sebagai berikut:

Algoritma SRCLus:

- Lakukan pencarian *query* dengan metode IFS.
- Lakukan clustering pada hasil pencarian
- Cari pusat-pusat kluster dari hasil clustering
- Cari label-label kluster berdasarkan informasi pusat kluster
- Tentukan similaritas pusat kluster terhadap *query* yang diberikan
- Sajikan hasil pencarian perkluster terranking berdasar tingkat similaritas pusat kluster terhadap *query*, dengan bantuan label kluster

Secara sekema pemrosesan SRClus dapat digambarkan dalam Gambar 3.



Gambar 3. Skema penyajian hasil dengan SRClus

Evaluasi suatu model *retrieval* oleh suatu sistem IR yang paling umum adalah ukuran *Recall* dan *Precision* (Rijsbergen, 1979). *Recall* didefinisikan sebagai rasio cacah dokumen relevan terpanggil dengan cacah total dokumen terpanggil, sedangkan *Recall* didefinisikan sebagai rasio antara cacah dokumen relevan terpanggil dengan total cacah dokumen relevan dalam koleksi. Parameter tunggal ukuran keberhasilan *retrieval* yang menggabungkan *Recall* dan *Precision* adalah parameter F-*measure* (Rijsbergen, 1979):

$$\text{F-}measure = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad \ldots\ldots (6)$$

dengan β merupakan nilai parameter kepentingan antara aspek Precision dan Recall. Jika *Recall* (R) dan *Precision* (P) memiliki bobot yang sama penting, β =1, maka parameter F-*measure* menjadi seperti persamaan (7) berikut.

$$\text{F-}measure = \frac{2PR}{P+R} \quad \ldots\ldots\ldots (7)$$

Bahan penelitian ini berupa koleksi dokumen teks berbahasa Indonesia, yang terdiri dari dua buah koleksi berita dan koleksi abstrak , yaitu seperti tersaji dalam Tabel 1 berikut.

Adapun daftar *query* untuk masing-masing koleksi dan informasi relevansi dengan pemeriksaan manual untuk tiap *query* adalah seperti pada Tabel 2 dan Tabel 3 .

Tabel 1. Koleksi dokumen untuk tes

Koleksi	∑ **doc**	∑ **Term**	∑ **term index**	∑**cluster**	∑**Query**
News1009	1009	18.255	5.233	21	10
Abstract	350	5.110	1.119	30	10

Tabel 2. Daftar *query* untuk koleksi dokumen berita Nws1009.dok

No	Query	Num of Rel Doc
1	Pemberangkatan jamaah haji	38
2	Pertandingan piala dunia	183
3	Pasar uang dollar	67
4	Penumpasan gam aceh	61
5	Kerusuhan ambon maluku	51
6	Pertandingan tinju tyson lewis	21
7	Tki indonesia di malaysia	30
8	Penyelesaian kasus tommy Suharto	67
9	Pertandingan tenis junior	30
10	Penyelesiaian kasus bulog akbar tanjung	83

Tabel 3. Daftar *Query* untuk koleksi dokumen akademik abstrak

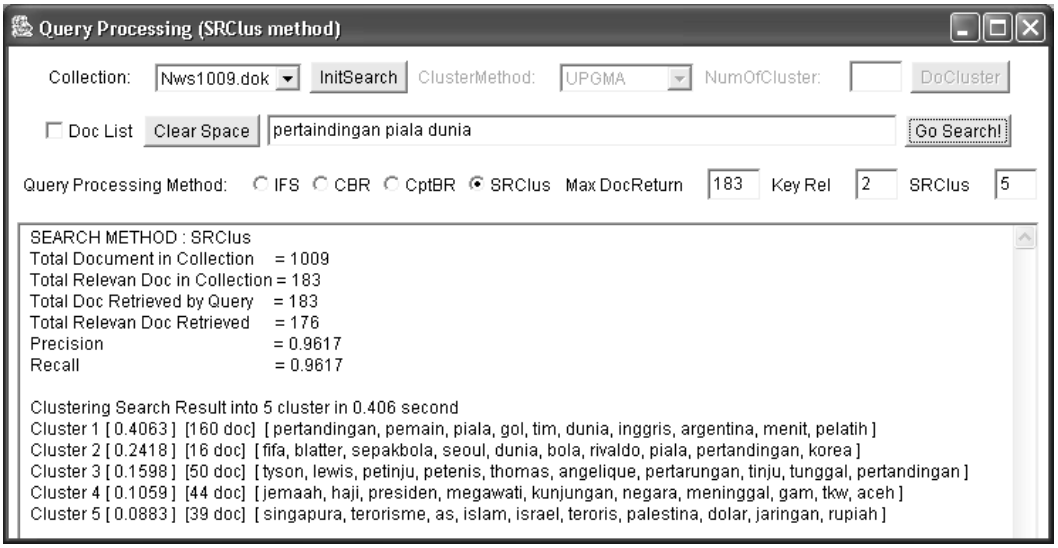
No	Query	Num of Rel Doc
1	Aplikasi logika fuzy	20
2	Sistem informasi	45
3	Jaringan syaraf tiruan	17
4	Pengolahan citra	10
5	Algoritma genetika	17
6	Database	15
7	Sistem pendukung keputusan	16
8	GPS GPRS komunikasi data	29
9	Rekayasa perangkat lunak	24
10	Keamanan system informasi	21

Proses *pre-processing* berupa ekstrak kata, perancangan kode untuk IFS maupun SRClus dan perancangan antar muka grafis dilakukan dengan kode program JAVA (jdk1.4.2). Analisis dilakukan dengan cara pemanggilan *query* secara linear (IFS) dengan mengambil informasi banyak dokumen yang relevan sebagai nilai *cut-off* (batasan jumlah dokumen yang harus dikembalikan). Hasilnya dievaluasi dengan menghitung F-*measure*nya. Selanjutnya dengan algoritma SRCLus diterapkan dengan nilai *cut-off* yang sama dengan IFS, hasil dokumen yang dikembalikan diambilkan dari kluster dengan ranking teratas jika telah memenuhi atau ditambahkan kluster dibawahnya dan seterusnya. Kemudian nilai untuk *F-measure* dari SRClus juga ditentukan. Hasil pengujian statistik digunakan untuk mem-bandingkan F-*measure* dari IFS dengan SRClus. Uji statistik hasil dengan uji t wilcoxon signrank untuk berpasangan.

PEMBAHASAN

Hasil perancangan antar muka grafis untuk membandingkan kinerja IFS dengan SRClus disajikan seperti dalam Gambar 4 dan Gambar 5. Dari Gambar 4 terlihat bahwa pada IFS, dalam 183 dokumen *ranking* teratas hanya 153 yang relevan. Jika hasil pencarian dikluster terlebih dahulu dengan SRClus, dalam contoh dikluster menjadi 5 buah kluster, terlihat bahwa 2 kluster pertama berkaitan dengan piala dunia seperti tersaji dalam 10 kata yang dipilih sebagai label dari masing-masing kluster, yaitu label kluster 1 [pertandingan, pemain, dst...] dan label kluster 2 [fifa, blatter, dst ...], sedangkan pada kluster ke 3,4 dan 5 berhubungan dengan dokumen lain. Setelah dilakukan *clustering* ternyata dalam 183 dokumen pertama terdapat 176 dokumen relevan. Terjadinya peningkatan dokumen relevan sebanyak 23 buah dokumen.

Gambar 4. Pencarian *query* "pertandingan piala dunia" dengan model IFS



Gambar 5. Pencarian *query* "pertandingan piala dunia" dengan model SRClus

Untuk perbandingan hasil evaluasi temu kembali metode SRClus dengan temu kembali metode IFS untuk seluruh *query* pada koleksi dokumen berita Nws1009.dok disajikan dalam Tabel 4. Adapun hasil analisis statistik perbedaan rerata kinerja IFS dan kinerja SRCLus disajikan dalam Tabel 5.

Tabel 4. Hasil *retrieval* koleksi News1009 dengan model **IFS** dan **SRClus**

No	Query	F-IFS	F-SRClus
1	Pemberangkatan…	0.4079	0.4342
2	Pertandingan ..	0.4290	0.4836
3	Pasar uang dolar…	0.4925	0.4925
4	Penumpasan …	0.4836	0.4918
5	Kerusuhan …	0.4412	0.4608
6	Pertandingan…	0.5000	0.5000
7	Tki indonesia …	0.3833	0.4667
8	Penyelesaian …	0.3806	0.4851
9	Pertandingan...	0.2000	0.4667
10	Penyelesiaian …	0.4157	0.4458
	Average	0.4134	0.4727

Dari Tabel 4 dapat ditunjukkan bahwa kinerja *retrieval* dengan melakukan *clustering* pada *search result* dapat meningkatkan temu kembali dari cara *linear* (IFS), yaitu rerata 0.4772 untuk SRClus dan dibandingkan 0.4134 untuk IFS. Peningkatan ini setelah diuji secara statistik adalah signifikan pada taraf alpa 5% (lihat Tabel 5). Sehingga dapat disimpulkan bahwa penerapan clustering akan meningkatkan kinerja temu kembali dibandingkan dengan penyajian langsung hasil pencariannya.

Tabel 5. Hasil uji statistik beda *retrieval* **SRClus** dan **IFS** koleksi Nws1009.dok

Mean	Std.Deviation	T	df	Sig(2-tailed)
-0,05934	0,08084	-2,321	9	0,045

Untuk koleksi abstrak, kinerja **SRClus** disajikan dalam Tabel 6 dengan uji statistik hasil perbandingan pada Tabel 7. Dari Tabel 6 terlihat bahwa kinerja SRClus memberikan rata-rata F-*measure* sebesar 0.3248 dibandingkan dengan kinerja IFS sebesar 0.2534. Dari uji statistik juga terlihat bahwa perbaikan kinerja SRClus terhadap IFS signifikan hampir mencapai taraf 1% , yaitu pada taraf signifikansi 1,1%.

Tabel 6. Hasil *retrieval* **IFS** dan **SRClus** untuk koleksi abstrak

No	**Query**	**F-IFS**	**F-SRClus**
1	Aplikasi logika fuzy	0.3250	0.3750
2	Sistem informasi	0.3000	0.3000
3	Jaringan syaraf…	0.4118	0.4118
4	Pengolahan citra	0.2500	0.4500
5	Algoritma genetika	0.1765	0.2647
6	Database	0.3000	0.3667
7	Sistem pendukung ..	0.2500	0.3125
8	GPS GPRS komu…	0.1552	0.1897
9	Rekayasa perang…	0.2708	0.2917
10	Keamanan system…	0.0952	0.2857
	Average	0.2534	0.3248

Tabel 7. Hasil uji statistik *retrieval* **SRClus** dan **IFS** koleksi abstrak

Mean	Std.Deviation	T	df	Sig (2-tailed)
-0,07133	0,07124	-3,166	9	0,011

KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari penelitian ini adalah bahwa penerapan *clustering* dokumen ternyata mampu meningkatkan kinerja *retrieval* meskipun ia bekerja pada level penyajian. Peningkatan F-*measure* sebagai kinerja ukuran efektivitas adalah sebesar 14,34% untuk koleksi berita dan 28,18% untuk koleksi abstrak akademik. Dengan demikian dapat disarankan untuk perbaikan perancangan suatu mesin pencari dalam skala data yang besar, penerapan *clustering* dilakukan untuk upaya penyajian hasil pencarian yang lebih efisien dari pada penyajian berbasis *page-rank* seperti yang telah umum diterapkan saat ini.

Penerapan clustering dokumen ini dapat juga digunakan untuk merancang system temu kembali yang lebih efektif karena saat ini volume data yang ada dalam bentuk digital seperti pada web perkembangannya sangat pesat sehingga mengakibatkan terus membesarnya ukuran koleksi. Ukuran koleksi yang besar pada gilirannya akan menyebabkan hasil pencarian yang cenderung membesar pula. Untuk seleksi dokumen tidak relevan dapat dirancang

dengan menerapkan clustering, yaitu dengan menghapuskan saja hasil pencarian yang pusat klusternya kurang relevan dengan *query* yang diberikan.

DAFTAR PUSTAKA

Chisholm, E. and T. G. Kolda, 1999, New Term Weighting Formula for the Vector Space Method in Information Retrieval, *Research Report*, Computer Science and Mathematics Division, Oak Rid - ge National Library, Oak Ridge, TN 3781-6367.

Cutting, D. R., D. R. Karger, J. O. Pederson, and J. W. Tukey,1992, Scatter/Gather:A Cluster-based Approach to Browsing Large Document Collection, *Proceding 15th Annual Int 7ACM SIGIR Conference on R&D in IR*, June 1992.

Frakes, W.B., and Baeza-Yates, R., *Information Retrieval , Data Structures and Algorithm*, Prentice Hall, Englewood New Jersey, 1992.

Luhn, H.P., 1958, The Automatic Creation of Literature Abstracts *IBM Journal of Research and Development*, 2:159-165.

Nazief, B., 2000, Development of Computational Linguistic Research: a Challenge for Indonesia, *Computer Science Center*, University of Indonesia

Salton, G., 1989, Automatix Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer, *Cornell University*, Addison Wisley Publishing Comp, New York.

Tala, F. Z., 2004, A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia, *Master Thesis*, Universiteit van Amsterdam, The Netherlands.

Osinki, S. , 2004, Dimensionality Reduction Techniques for Search Engine Results Clustering, *Master Thesis*, University of Sheffield, UK.

Rijsbergen, C. J.,1979, *Information Retrieval*, Information Retrieval Group, University of Glasgow .

Tombros, A., 2002, The Effectiveness of Query-Based Hierarchic Clustering of Documents for Information Retrieval, *PhD Thesis*, Univerity of Glasgow.

Vega, V. B. , 2001, Information Retrieval for the Indonesian Language, *Master Thesis*, National University of Singapore.

Widyantoro, D., H.,2007, Toward the Development of The Next Generation Search Engine, *Proceeding of The International Conference on Electrical Engineering and Informatics*, ICEEI2007, Bandung.

Zamir, O.E., 1999, Clustering Web Document : A Phrase-Based Method for Grouping Search Engine Result, *PhD. Dissertation*, University of Washington.