KLASTERING DOKUMEN MENGGUNAKAN HIERARCHICAL AGGLOMERATIVE CLUSTERING

Herny Februariyanti[1] **Edi Winarko**[2]

1) Sistem Informasi, Universitas Stikubank Semarang, email: herny@unisbank.ac.id
2) Ilmu Komputer, Universitas Gadjah Mada Yogyakarta, email: edwin@ugm.ac.id

Document retrieval process stored in document database often produces very large numbers of documents. And many documents are available is not relevant to the desired document. Clustering the documents in database before retrieval is one way to find relevant documents.

This study attempted to document be clustered using *Agglomerative Hierarchical Clustering* Algorithms. It emphasized clustering to documents written in Indonesian, because today, the needs of users in the homeland of information is increasing. The relationship between documents can be measured by the similarity between the documents (similarity).

This algorithm was tested by using the documents from UII SNATI publications from 2004-2009. The experimental results show that this algorithm can be applied to group documents written in Indonesian. The selection of appropriate keywords will increase the quality of information retrieval to the document. This quality is reflected in the *recall* rates 0.6 and 0.5 *precision*.

Keywords : Information Retrieval, Stemming, Cosine Similarity, Hierarchical Agglomerative Clustering.

Kemajuan yang pesat dibidang teknologi informasi terutama internet, telah menimbulkan lonjakan informasi yang hebat. Hal ini terjadi karena internet memungkinkan banyak orang untuk memproduksi, memanipulasi, mengakses dan menyebarluaskan informasi dengan mudah.

Salah satu cara untuk memperoleh informasi yang seimbang seperti apa yang diinginkan adalah dengan membaca beberapa dokumen yang membahas topik yang sama. Akan tetapi cara ini menyulitkan pembaca untuk menangkap topik bahasan utama dari dokumen - dokumen tersebut karena harus mengingat – ingat isi dokumen yang telah dibaca sebelumnya.

Dalam proses penelusuran informasi melalui internet sering diperoleh informasi yang sangat banyak, tetapi sebagian besar diantaranya adalah informasi yang tidak dibutuhkan. Oleh karena itu, dari sudut pandang temu kembali informasi (*information retrieval*), semakin banyaknya informasi yang tersedia di internet justru semakin mempersulit untuk menemukan kembali informasi yang relevan. Dalam suatu sistem temu kembali informasi, kemampuan untuk menemukan informasi yang tersedia diukur dengan *recall* dan kemampuan untuk

menemukan informasi yang relevan diukur dengan ketelitian, maka proses penelusuran dalam situasi seperti tersebut di atas akan menghasilkan *recall* yang tinggi tetapi ketelitian rendah.

Sistem yang tepat untuk masalah tersebut adalah sistem temu kembali informasi yang dapat menghasilkan integrasi dari beberapa dokumen elektronik yang berbeda dengan topik bahasan yang sama secara otomatis. Proses integrasi akan menghasilkan dokumen baru yang mengandung semua bagian dari dokumen – dokumen awal, namun memiliki susunan antar kalimat serta antar paragraf yang berbeda. Perbedaan ini karena saat proses integrasi topik – topik bahasan yang serupa (*similar*) dari semua dokumen dikumpulkan menjadi satu paragraf dan disusun ulang kalimat per kalimat sesuai dengan besarnya kesamaan (*similiarity*) antar kata (*term*). Dengan membaca hasil integrasi diharapkan pembaca dapat terbantu dalam menyerap informasi penting yang ada dalam kumpulan dokumen yang berbeda dan tidak perlu lagi membaca sekumpulan dokumen satu per satu.

METODOLOGI PENELITIAN

Penelitian ini menggunakan data yang diambil dari dokumen teks abstrak naskah publikasi SNATI dari tahun 2004-2009 Universitas Islam Indonesia Yogyakarta, data dalam bentuk format file teks sejumlah 468 dokumen abstrak. Untuk memvalidasi program aplikasi yang dibuat, koleksi data dikelompokkan menjadi 8 bidang kajian bidang yaitu bisnis, kesehatan, informatika, citra, spk (sistem penunjang keputusan), jaringan, pendidikan dan pemerintahan.

Pengujian aplikasi dilakukan dengan menggunakan file abstrak SNATI sejumlah 468 file, telah mampu untuk tidak melakukan indeks-indeks kata umum (*stopword*) dan telah membentuk kata dasar dari tiap kata (*term*) yang ada dalam dokumen abstrak tersebut. Selanjutnya setiap *term* telah dihitung frekuensinya dan diberikan pembobotan menggunakan *cosine similaritas* dan selanjutnya *term* tersebut disimpan pada database korpus.

Selanjutnya penulis melakukan pengujian input string *query* dan kemudian hasil pengujian input *query* dilakukan pengukuran hasil *retrieval* (temu kembali informasi). Pengujaian hasil *query* dilakukan dengan menggunakan *recall precision* dan pengukuran *F-measure*.

Sistem Temu Kembali Informasi

Sistem temu-kembali informasi pada prinsipnya adalah suatu sistem yang sederhana. Misalkan ada sebuah kumpulan dokumen dan seorang user yang memformulasikan sebuah pertanyaan (*request* atau *query*). Jawaban dari pertanyaan tersebut adalah sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan. Secara matematis hal tersebut dapat dituliskan pada persamaan 1 berikut ini:

$$Q \xrightarrow{2^n} D \qquad (1)$$

Q = pertanyaan (*query*)

D = dokumen

n = jumlah dokumen

2^n = jumlah kemungkinan himpunan bagian dari dokumen yang ditemukan.

Sistem temu kembali akan mengambil salah satu dari kemungkinan tersebut. Sementara itu Salton (1989) menjelaskan bahwa secara sederhana temu kembali informasi merupakan suatu sistem yang menyimpan informasi dan menemukan kembali informasi tersebut. Secara konsep bahwa ada beberapa dokumen atau kumpulan *record* yang berisi informasi yang diorganisasikan ke dalam sebuah media penyimpanan untuk tujuan mempermudah ditemukan kembali. Dokumen yang tersimpan tersebut dapat berupa kumpulan *record* informasi bibliografi maupun data lainnya.

Sistem temu kembali informasi pada dasarnya dibagi dalam dua komponen utama yaitu sistem pengindeksan (*indexing*) yang menghasilkan basis data sistem dan temu-kembali yang merupakan gabungan dari *user interface* dan look-up-table. Pada bagian selanjutnya akan dijelaskan berbagai macam sistem pengindeksan dan teknik-teknik temu kembali informasi yang telah dikembangkan.

Salton (1989) juga mengemukakan fungsi utama Sistem Temu Kembali Informasi adalah sebagai berikut:

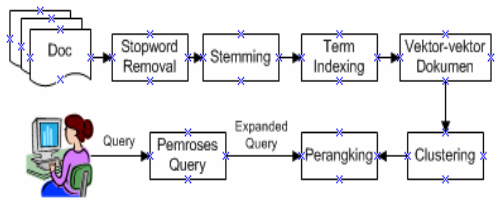
1. Mengidentifikasi sumber informasi yang relevan dengan minat masyarakat pengguna yang ditargetkan.
2. Menganalisis isi sumber informasi (dokumen)
3. Merepresentasikan isi sumber informasi dengan cara tertentu yang memungkinkan untuk dipertemukan dengan pertanyaan pengguna.
4. Merepresentasikan pertanyaan (*query*) pengguna dengan cara tertentu yang memungkinkan untuk dipertemukan sumber informasi yang terdapat dalam basis data.
5. Mempertemukan pernyataan pencarian dengan data yang tersimpan dalam basis data.
6. Menemu-kembalikan informasi yang relevan.
7. Menyempurnakan unjuk kerja system berdasarkan umpan balik yang diberikan oleh pengguna

Pada gambar 1 dijelaskan tentang gambaran proses temu kembali informasi, dijelaskan bahwa Modul *indexer* mengekstrak semua kata dalam tiap halaman, dan menyimpan dokumen dimana tiap kata muncul. Modul *query engine* bertanggung jawab untuk menerima dan melayani permintaan pencarian dari para pemakai. Mesin menyandarkan secara penuh pada indek-indek, dan kadang-kadang pada penyimpan halaman. Karena ukuran web dan fakta bahwa para pemakai umumnya hanya memasukkan satu atau dua kata kunci, maka himpunan hasil biasanya sangat besar.

Modul *rangking* mempunyai tugas untuk mengurutkan hasil sedemikian sehingga hasil yang dekat diatas adalah yang paling sesuai dengan yang diinginkan oleh pemakai.

Modul query menjadi perhatian khusus, karena terdapat beberapa masalah ketika teknik perolehan informasi tradisional

tanpa modifikasi diterapkan dalam pencarian web. Kebanyakan teknik tradisional menyandarkan pada pengukuran keserupaan dari *query* teks dengan teks-teks dalam koleksi dokumen. *Query* yang kecil diatas koleksi yang besar seperti pada mesin pencari web menyebabkan beberapa pendekatan berbasis keserupaan menghasilkan halaman-halaman yang tidak *relevan*.



Gambar 1 Proses Temu Kembali Informasi (Salton, 1998)

Index Inverted

Inverted file atau index inverted adalah mekanisme untuk pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Struktur *inverted file* terdiri dari dua elemen, yaitu: kata (*vocabulary*) dan kemunculan (*occurences)*. Kata-kata tersebut adalah himpunan dari kata-kata yang ada pada teks, atau merupakan ekstraksi dari kumpulan teks yang ada.

Dan tiap kata terdapat juga informasi mengenai semua posisi kemunculannya (*occurences*) secara rinci. Posisi dapat merefer kepada posisi kata ataupun karakter.

Stemming

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Dengan menghilangkan semua imbuhan (*affixes)* baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar.

Penilitian terhadap *stemming* untuk *text retrieval*, machine translation, document summarization dan *text classification* sudah pernah dilakukan sebelumnya. Untuk *stemming* yang dilakukan pada *text retrieval*, *stemming* ini meningkatkan kesensitivan *retrival* dengan meningkatkan kemampuan untuk menemukan document yang relevan, tetapi hal itu terkait dengan pengurangan pada pemilihan dimana pengelompokkan menjadi kata dasar menyebabkan penghilangan makna kata. Pada *text retrival, stemming* diharapkan dapat meningkatkan *recall*, tetapi memungkinkan untuk menurunkan *precision*.

Teknik Boolean Temu Kembali Informasi

Model *Boolean* dalam sistem temu kembali merupakan model yang paling sederhana. Model ini berdasarkan teori himpunan dan aljabar *Boolean*. Dokumen adalah himpunan dari istilah (*term*) dan *query* adalah pernyataan *Boolean* yang ditulis pada term. Dokumen diprediksi apakah relevan atau tidak. Model ini menggunakan operator *Boolean*. Istilah dalam sebuah *query* dihubungkan dengan menggunakan operator AND, OR atau NOT. Metode ini merupakan

metode yang paling sering digunakan pada`mesin penelusur (*search engine*) karena kecepatannya.

Keuntungan menggunakan model Boolean (Baeza, 1999) :

1. Model *Boolean* merupakan model sederhana yang menggunakan teori dasar himpunan sehingga mudah diimplementasikan.
2. *Query* sederhana dan mudah dimengerti.
3. Operator *Boolean* bisa mendekati bahasa alami. Operator AND dapat menemukan hubungan antar konsep, OR dapat menemukan terminologi alternatif, NOT dapat menemukan arti alternatif.

Cosine Simmilarity

Kesamaan antar dokumen dapat diukur dengan fungsi similaritas (mengukur kesamaan) atau fungsi jarak (mengukur ketidaksamaan). Beberapa fungsi similaritas atau fungsi jarak yang dapat dijumpai adalah *Disk, Jaccard, Overlap, Asimmetric, Minowski distance, Euclidean distance, Pearson Correlation, Cosine.*

Untuk tujuan klastering dokumen fungsi yang baik adalah fungsi *Cosine Similaritas*.

$$Similarity(X,Y) = \frac{X \bigcap Y}{|X|^{\frac{1}{2}} . |Y|^{\frac{1}{2}}} \qquad (2)$$

Dimana :

$X \bigcap Y$ adalah jumlah term yang ada di dokumen X dan yang ada di dokumenY

$|X|$ adalah jumlah term yang ada di dokumen X

$|Y|$ adalah jumlah term yang ada di dokumen Y

Clustering hierarchical

Metode pembentukan klaster biasanya dikategorikan menurut tipe dari struktur klaster yang dihasilkan. Secara umum metode klaster terbagi menjadi dua, yaitu metode *Non-Hierarchical* C*lustering* (klastering non-hirarkhis) dan metode *Hierarchical Clustering* (klastering hirarkhis).

Metode non-hirarkhis disebut juga metode partisi, yaitu membagi serangkaian data yang terdiri dari n obyek ke dalam k klaster (k<n) yang tidak saling tumpang-tindih (*overlap*), dimana nilai *k* telah ditentukan sebelumnya. Salah satu prosedur pengelompokkan pada non-hirarkhis adalah dengan menggunakan metode *k-means*. Metode ini merupakan metode pengelompokkan yang bertujuan untuk mengelompokkan objek sedemikian hingga jarak tiap-tiap objek kepusat kelompok didalam suatu kelompok adalah minimum.

Metode klaster yang kedua adalah metode *Hierarchical Clustering* (klastering hirarkhis). Metode pengelompokkan hirarkhis biasanya digunakan apabila belum ada informasi jumlah kelompok yang akan dipilih. Arah pengelompokkan bisa bersifat *divisive* (*top to down*) artinya dari 1 klaster sampai

menjadi k buah klaster atau bersifat *agglomerative* (*bottom up*) artinya dari n klaster (dari n-buah data yang ada) menjadi k buah klaster. Teknik hirarkhis (*hierarchical methods*) adalah teknik klastering membentuk kontruksi hirarki atau berdasarkan tingkatan tertentu seperti struktur pohon. Dengan demikian proses pengelompokkannya dilakukan secara bertingkat atau bertahap.

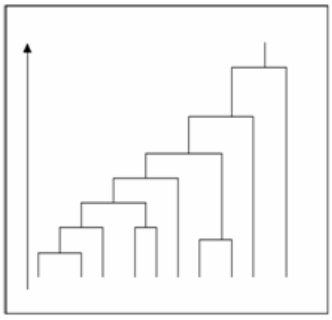
Hierarchical Clusstering adalah salah satu algoritma klastering yang dapat digunakan untuk meng-*klaster* dokumen *(document clustering*). Dari teknik *Hierarchical Clustering*, dapat dihasilkan suatu kumpulan partisi yang berurutan, dimana dalam kumpulan tersebut terdapat:

1. Klaster-klaster yang mempunyai poin – poin individu. *Klaster-klaster* ini berada di level yang paling bawah.
2. Sebuah klaster yang didalamnya terdapat poin – poin yang dipunyai semua *klaster* didalamnya. *Single klaster* ini berada di *level* yang paling atas.

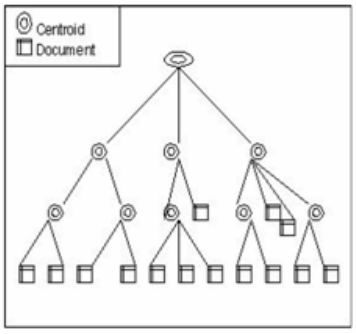
Pembentukan klaster dokumen dalam sistem temu kembali informasi dengan metode hirarkhis adalah sebagi berikut:

1. Mengidentifikasi dua dokumen yang paling mirip dan menggabungkannya menjadi sebuah klaster.
2. Mengidentifikasi dan menggabungkan dua dokumen yang paling mirip berikutnya menjadi sebuah klaster sampai semua dokumen tergabung dalam klaster-klaster yang terbentuk.
3. Proses penelusuran dokumen dilakukan dengan cara mencocokkan *query* dengan *centroid. Centroid* merupakan dokumen parent pada masing-masing klaster dokumen. Berikutnya dokumen yang berada dalam satu klaster dengan *centroid* akan ditampilkan sebagai hasil *query*.

Hasil keseluruhan dari algoritma *Hierarchical Clustering* secara grafik dapat digambarkan sebagai tree, yang disebut dengan dendogram. Tree ini secara grafik menggambarkan proses penggabungan dari klaster-klaster yang ada, sehingga menghasilkan klaster dengan level yang lebih tinggi. Cabang-cabang dalam pohon menyajikan cluster. Kemudian cabang- cabang bergabung pada node yang posisinya sepanjang sumbu jarak (similaritas) menyatakan tingkat di mana penggabungan terjadi.Gambar 2.a dan.b memperlihatkan struktur dendogram dan diagram pohon untuk klastering hirarkhis.



(a)



(b)

Gambar 2 Dendogram dan Struktur Pohon dari *Hierarchical Clustering* (Salton, 1989)

Kemiripan antar dokumen ditentukan dengan mengukur jarak antar dokumen. Dua dokumen yang mempunyai jarak paling kecil dikatakan mempunyai kemiripan paling tinggi, dan dikelompokkan ke dalam satu klaster yang sama. Sebaliknya dua dokumen yang mempunyai jarak paling besar dikatakan mempunyai kemiripan paling rendah, dan dimasukkan ke dalam klaster yang berbeda.

Metode Hierarchical Agglomerative Clustering

Metode *Hierarchical Agglomerative Clustering* adalah metode yang menggunakan strategi disain *Bottom-Up* yang dimulai dengan meletakkan setiap obyek sebagai sebuah klaster tersendiri (atomic klaster) dan selanjutnya menggabungkan atomic klaster – atomic klaster tersebut menjadi klaster yang lebih besar dan lebih besar lagi sampai akhirnya semua obyek menyatu dalam sebuah klaster atau proses dapat pula berhenti jika telah mencapai batasan kondisi tertentu.

Langkah-langkah dalam algoritma *Hierarchical Agglomerative Clustering* untuk mengelompokkan *N* objek (item/variabel) adalah sebagai berikut :

1. Mulai dengan *N klaster*, setiap klaster mengandung entiti tunggal dan sebuah matriks simetrik dari jarak (similarities) $D = \{d_{ik}\}$ dengan tipe matrik adalah *NxN*.
2. Cari matriks jarak untuk pasangan klaster yang terdekat (paling mirip), yaitu dengan mencari similaritas terbesar.

   Misalkan jarak antara klaster *U* dan *V* yang paling mirip adalah d_{uv}.
3. Gabungkan klaster *U* dan *V*. Label klaster yang baru dibentuk dengan (*UV*).

   Update entries pada matrik jarak dengan cara :
   a. Hapus baris dan kolom yang bersesuaian dengan klaster *U* dan *V*
   b. Tambahkan baris dan kolom yang memberikan jarak-jarak antara klaster (*UV*) dan klaster-klaster yang tersisa.

Metode single-linkage hierarchical clustering

Ada 3 (tiga) metode klaster hirarkhis yaitu *metode single linkage, metode complete linkage, metode average linkage*. *Single linkage* memberikan hasil bila kelompok-kelompok digabungkankan menurut jarak antara anggota-anggota yang paling dekat, *complete linkage* terjadi bila kelompok-kelompok digabungkan menurut jarak antara anggota-anggota yang paling jauh. Untuk

average linkage, digabungkan menurut jarak rata-rata antara pasangan-pasangan anggota masing-masing pada himpunannya.

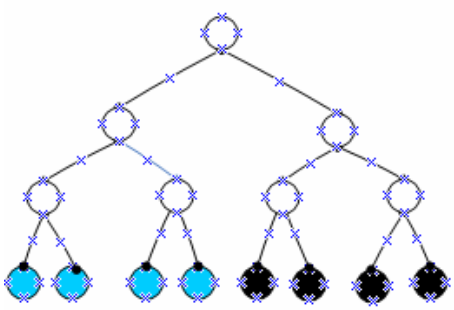
Pada penelitian ini digunakan metode s*ingle linkage* (Salton, 1998) untuk pembentukkan klaster dokumen. Input untuk algoritma *single linkage* merupakan jarak atau similaritas antara pasangan-pasangan dari objek-objek. Kelompok-kelompok dibentuk dari entiti tunggal dengan menggabungkan jarak paling pendek atau similaritas (kemiripan) yang paling besar. Pada awalnya, kita harus menemukan jarak terpendek dalam $D = \{d_{ik}\}$ dan menggabungkan objek-objek yang bersesuaian misalnya, U dan V , untuk mendapatkan klaster (UV). Untuk langkah (3) dari algoritma di atas jarak-jarak antara (UV) dan klaster W yang lain dihitung dengan cara :

$$d_{(uv)w} = min\{ d_{uw}, d_{vw} \}$$

Di sini besaran-besaran d_{UW} dan d_{VW} berturut-turut adalah jarak terpendek antara klaster-klaster U dan W dan juga klaster-klaster V dan W .

Temu Kembali Berbasis Klaster

Sistem temu kembali berbasis klaster (*cluster based Retrieval*) dikemukakan oleh Rijbergen (1971) sebagai alternative terhadap temu kembali linear. Dengan temu kembali berbasis klaster sebuah klaster terbaik akan dipanggil jika klaster tersebut paling *match* dengan *query*. Similaritas klaster dengan query diwakili oleh similaritas pusat klaster dengan *query*. Sebuah sekenario ideal jika klastering hirarkhis (*hierarchical clustering*) dapat secara sempurna memisahkan dokumen relevan dan tidak relevan (disajikan dalam gambar 3 maka temu kembali akan memiliki efektifitas yang sangat tinggi.



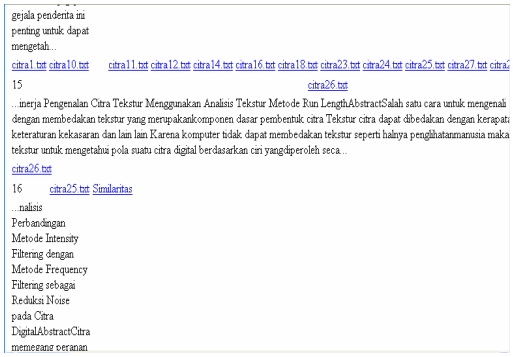
Gambar 3. Pemisahan Sempurna Dokumen Relevan dan non-Relavan

Temu kembali berbasis klaster dapat diimplementasikan dengan langkah wal melakukan pencarian *query* berbasis klaster(*cluster based search*). Metode pencarian akan menyesuaikan dengan struktur klaster, apakah hirarkhis atau flat (*partitional*). Kebanyakan penelitian awal klastering dokumen untuk sistem temu kembali informasi menggunakan algoritma hirarkhis (Rijsbergen, 1979). Pada pendekatn *search* secara hirarkhis ada dua pendekatan *search* yang terkenal yaitu *Top Down Search* dan pendekatan *Bottom Up Search*.

HASIL DAN PEMBAHASAN

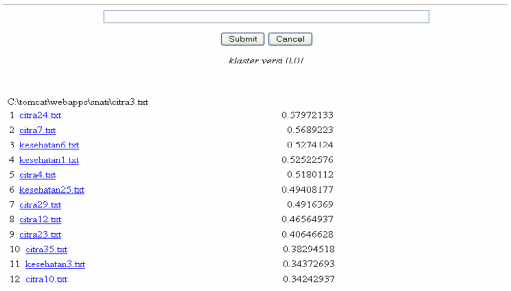
Pada tahap pengujian input *query* dilakukan dengan cara memasukkan *query* “citra”, “pencitraan”, “pendidikan”, “kesehatan”, “penyakit”, “jaringan”, “bisnis”, “penyakit”, “pengolahan citra”, “pemerintahan daerah”, “manajemen bisnis”, dan penunjang keputusan”. Terlihat pada gambar 4 adalah salah satu contoh hasil tampilan dari input

query "citra". Hasil proses dari *query* akan ditampilkan dokumen-dokumen yang berada dalam klaster yang sama. Dokumen yang ditampilkan adalah dokumen sebagai parent dan jika dokumen parent memiliki child akan ditampilkan dokumen childnya.



Gambar 4. Contoh Hasil Pengujian dengan Menggunakan Query "citra"

Dalam implementasi sistem juga disediakan menu similaritas, yaitu aplikasi memberikan fasilitas untuk menampilkan dokumen-dokumen yang similar dengan dokumen yang dipilih. Terlihat pada gambar 5 dokumen yang similar akan ditampilkan dan user dapat melihat isi abstrak dokumen yang similar.



Gambar 5. Tampilan Menu Similaritas

Pengujian Recall dan Precision

Kriteria yang digunakan untuk menilai kualitas sistem temu kembali informasi dalam penelitian ini adalah terpenuhinya kebutuhan pengguna. Hal ini dapat dilihat dari *recall* dan *precision.* (Rijsbergen, 1979). *Recall* dan *Precision* adalah pengukuran yang sering digunakan untuk mengukur kualitas hasil proses dari hasil proses sistem temu kembali informasi. Secara singkat, precision dapat dianggap sebagai ukuran ketepatan/ketelitian, sedangkar *recall* adalah ukuran kesempurnaan. Dalam penggunaannya pada sistem temu kembali informasi, nilai *precision* yang sempurna (1) berarti semua hasil yang keluar adalah relevan. Nilai *recall* yang sempurna (1) berarti semua dokumen yang relevan telah berhasil didapatkan.

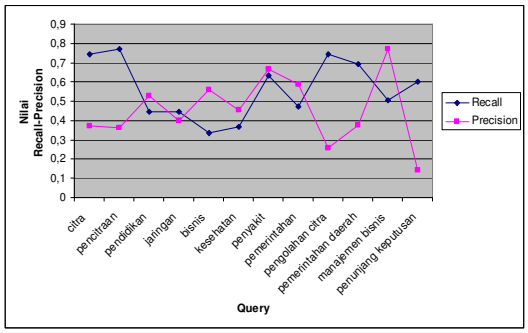
Dari hasil pengujian yang dilakukan terhadap data training yang diambil dari dokumen abstrak naskah pubikasi SNATI Universitas Islam Indonesia Yogyakarta dari tahun 2004-2009 dengan sejumlah 468 dokumen didapatkan nilai *recall* dan *precision* berdasarkan beberapa contoh *query* yang diinput user dapat dilihat pada tabel 1.

Tabel 1 Tabel Pengujian Recall dan Precision

No	Query	Recall	Precision
1	Citra	0,7428571	0,3714286
2	Pencitraan	0,7714286	0,3648649
3	Pendidikan	0,4444444	0,5283019
4	Jaringan	0,4431818	0,3979592
5	Bisnis	0,3333333	0,5625
6	Kesehatan	0,3658537	0,4545455

7	Penyakit	0,6341463	0,6666667
8	Pemerintahan	0,4722222	0,5862069
9	pengolahan citra	0,7428571	0,2574257
10	pemerintahan daerah	0,6944444	0,3787879
11	manajemen bisnis	0,5061728	0,7735849
12	penunjang keputusan	0,6	0,1411765
	Rata-rata	**0,5625785**	**0,456954**

Grafik perbandingan antara nilai *recall* dan *precision* dapat dilihat pada gambar 6, dari grafik terlihat bahwa nilai *recall* dan *precision* berbanding terbalik.



Gambar 6. Grafik Nilai Recall dan Precision

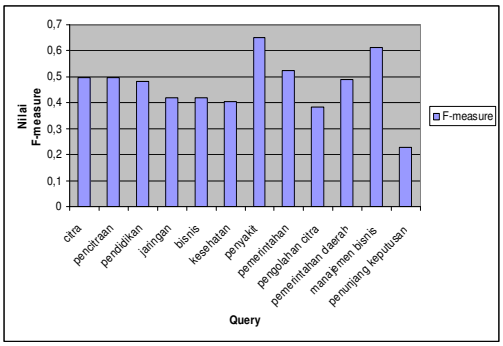
Untuk kinerja algoritma klastering yang dibangun digunakan pengukuran sistem temu kembali informasi dengan menggunakan ukuran *F-measure*.

Dari pengukuran *F-measure* yang dilakukan terhadap hasil kinerja kinerja algoritma dari sistem temu kembali informasi yang dibangun didapatkan nilai seperti dalam tabel 2.

Tabel 2 Hasil Perhitungan F-measure

No	Query	Recall	Precision	F-measure
1	Citra	0,7428571	0,3714286	0,4952381
2	Pencitraan	0,7714286	0,3648649	0,4954128
3	Pendidikan	0,4444444	0,5283019	0,4827586
4	Jaringan	0,4431818	0,3979592	0,4193548
5	Bisnis	0,3333333	0,5625	0,4186047
6	Kesehatan	0,3658537	0,4545455	0,4054054
7	Penyakit	0,6341463	0,6666667	0,65
8	pemerintahan	0,4722222	0,5862069	0,5230769
9	pengolahan citra	0,7428571	0,2574257	0,3823529
10	pemerintahan daerah	0,6944444	0,3787879	0,4901961
11	manajemen bisnis	0,5061728	0,7735849	0,6119403
12	Penunjang keputusan	0,6	0,1411765	0,2285714
	Rata-rata	**0,5625785**	**0,456954**	**0,4669093**

Hasil perhitungan *F-measure* terhadap algoritma kinerja sistem temu kembali informasi yang dibangun ditunjukkan pada grafik gambar 7, terlihat pada grafik bahwa kinerja sistem stabil. Selisih hasil antara *query* yang satu dengan *query* yang lain rata-rata sama. Hanya *query* "penunjang keputusan" menghasilkan *F-measure* rendah. Hal ini dipengaruhi juga karena dokumen spk (sistem penunjang keputusan) yang ada di dalam database jumlahnya sangat kecil dibandingkan dengan dokumen-dokumen yang lain.



Gambar 7 Grafik Kinerja Sistem Temu Kembali Informasi (*F-measure*)

KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan hal-hal sebagai berikut:

1. Pembobotan term frekuensi dan cosine similaritas digunakan untuk menunjukkan kemiripan antar dokumen.
2. Sistem dapat menampilkan dokumen yang mempunyai kedekatan similaritas dari query yang diinputkan user.
3. Dokumen yang membahas topik yang sama cenderung untuk mengelompok menjadi satu klaster.
4. Klaster dapat membantu menemukan dokumen yang ada dalam satu klaster dengan *query* yang diinputkan user.
5. Klaster dapat membantu mendapatkan dokumen yang relevan.
6. Hasil pengujian dengan *query* yang diinputkan user menunjukkan rata-rata recall = 0,6 dan precision = 0.5 dan F-measure = 0,5.

DAFTAR PUSTAKA

Baeza-Yates, R. & Ribeiro-Neto, B., 1999, *Modern Information Retrieval*, Addison-Wesley.

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

Salton, G., 1989, *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*, Addison – Wesly Publishing Company, Inc. All rights reserved.

Salton, G. and Buckley, 1988, *Term Weigting Approaches in Automatic Text Retrieval, Department of Computer Science*, Cornell University, Ithaca, NY 14853, USA.

Steinbach, M., Xiong H., Ruslim A., Kumar V., 2007, *Characterizing Pattern Preserving Clustering, Department of Management Science and Information Systems Rutgers*, the State University of New Jersey, USA.

Yue, W., 2005, *Using Query Expansion and Classification for Information Retrieval, College of Computer and Communication*, Hunan University ChangSha, Hunan Province, 410082,China.