PERANCANGAN DAN IMPLEMENTASI AUTOMATED DOCUMENT INTEGRATION DENGAN MENGGUNAKAN ALGORITMA COMPLETE LINKAGE AGGLOMERATIVE HIERARCHICAL CLUSTERING

Gede Aditra Pradnyana[1]**, Ngurah Agus Sanjaya ER**[2]
Program Studi Teknik Informatika, Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana
Email : gede.aditra@cs.unud.ac.id[1], agus.sanjaya@cs.unud.ac.id[2]

ABSTRAK

Salah satu cara yang umum digunakan untuk memperoleh informasi adalah dengan membaca beberapa dokumen yang membahas topik yang sama. Walaupun cara ini merupakan yang paling mudah namun pada pelaksanaannya banyak menghabiskan waktu. Penggunaan suatu sistem *automated document integration* yang membantu menemukan kalimat penting dari masing-masing dokumen akan menghemat waktu serta tenaga yang diperlukan. Keluaran dari sistem yang dikembangkan dalam penelitian ini adalah suatu dokumen yang dibentuk dari integrasi (*cluster*) kalimat-kalimat dari dokumen asli.

Kesamaan dokumen yang akan diintegrasikan ditentukan oleh *cosine similarity*. Sistem kemudian menghitung TF-IDF (*term frequency-inverse document frequency*) masing-masing kalimat pada dokumen. TF-IDF merupakan bobot dari suatu kalimat yang mencerminkan tingkat kepentingan dari kalimat pada suatu dokumen serta terhadap kalimat-kalimat lain pada dokumen yang berbeda. Kalimat-kalimat yang memiliki kesamaan yang tinggi kemudian digabungkan secara *agglomerative hierarchical* menggunakan metode *complete linkage*. Hasil uji coba memperlihatkan 75% responden menyatakan keluaran sistem adalah benar.

Kata kunci: *automated document integration, complete linkage agglomerative hierarchical clustering, cosine similarity*

ABSTRACT

A common way of gaining knowledge or information is by reading through documents which discuss the same topic. Despite being the easiest method to implement, it requires a lot of time and effort. Thus, the use of an automated text integration system which serves in finding important sentences from each original document, greatly reduces the time and effort needed. The research output which implements this system is a new document constructed from clustered sentences in the original documents.

The similarities of documents to be clustered is determined by using the cosine similarity. The system further calculates the TF-IDF (Term Frequency-Inverse Document Frequency) of each sentence in the documents. The TF-IDF serves as a weight of the sentence in a document to describe how important it is in comparison with other sentences in different documents. Sentences with high similarities are then clustered in an agglomerative hierarchical way using a complete linkage method. Experimental results show that 75% of respondents confirm that the output is correct.

Keywords: automated document integration, complete linkage agglomerative hierarchical clustering, cosine similarity

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat dewasa ini membuat para pengguna teknologi menuntut agar semua informasi dapat diperoleh dengan cepat, mudah, dan tidak membuang banyak waktu. Selain itu tingginya penggunaan internet telah memacu pesatnya pertumbuhan dan pertukaran

informasi sehingga informasi yang beredar pun semakin banyak. Salah satu cara untuk memperoleh informasi adalah dengan membaca beberapa dokumen yang pada kenyataannya banyak membahas topik yang sama. Namun hal ini akan sangat menyulitkan pembaca untuk menangkap topik bahasan utama dari dokumen-dokumen tersebut karena harus mengingat isi dokumen yang telah dibaca sebelumnya. Pembaca harus mengintegrasikan dahulu dokumen-dokumen yang dia baca di dalam pikirannya sebelum dapat merangkum maksud dan topik utama dokumen-dokumen tersebut secara keseluruhan. Selain itu, seringkali ada pembaca yang tidak ingin membaca seluruh dokumen tersebut karena faktor waktu yang dibutuhkan terlalu lama atau adanya keterbatasan waktu.

Seiring berkembangnya teknologi informasi, maka saat ini telah diperoleh suatu solusi yang memungkinkan pembaca dapat membuat integrasi dari beberapa dokumen tersebut menjadi suatu kesatuan dokumen dengan mudah. Maksud dari integrasi dokumen disini adalah sebuah proses untuk menghasilkan suatu dokumen baru dari beberapa dokumen dengan menggunakan bantuan komputer, tanpa menghilangkan arti dan bagian-bagian penting dari tiap dokumen tersebut. Tujuan dari integrasi dokumen ini adalah untuk mengambil sumber informasi dengan memperhatikan sebagian besar bagian-bagian berupa kalimat-kalimat yang penting dari setiap dokumen yang berbeda dan menampilkan kepada pembaca dalam bentuk suatu dokumen baru yang sesuai dengan kebutuhan pembaca. Setelah melalui proses ini, diharapkan pembaca dapat terbantu dalam menyerap informasi penting yang ada dalam kumpulan dokumen yang berbeda dengan topik bahasan yang sama, karena pembaca tidak perlu lagi membaca kumpulan dokumen satu per satu. Proses integrasi dokumen ini akan menghasilkan suatu produk teks yang memiliki atau mengandung semua bagian penting dari dokumen-dokumen awal, namun memiliki susunan antar kalimat atau antar paragraf yang berbeda.

Adapun algoritma yang digunakan dalam proses integrasi dokumen ini adalah *agglomerative hierarchical clustering* dengan metode *complete linkage*. *Agglomerative hierarchical clustering* adalah suatu metode *hierarchical clustering* yang bersifat *bottom-up* yaitu menggabungkan *n* buah *cluster* (beberapa dokumen) menjadi satu *cluster* tunggal (sebuah dokumen hasil integrasi). *Agglomerative hieararchical clustering* merupakan metode yang umum digunakan dalam *clustering* dokumen dan memiliki beberapa kelebihan, antara lain : tidak memperhitungkan initial *centroid* sehingga tepat digunakan dalam proses pengelompokan dokumen dan kinerja *information retrieval* berbasis *hierarchical clustering* memiliki hasil yang lebih baik jika dibandingkan dengan metode *partitional clustering* (Hamzah, 2009). Algoritma *agglomerative hierarchical clustering* dengan metode *complete linkage* memiliki hasil *clustering* yang lebih baik dibandingkan dengan metode *linkage* yang lainnya (Soebroto, 2005).

Dalam proses *clustering*, kesamaan antara satu dokumen dengan dokumen yang lain diukur dengan fungsi kesamaan *(similarity*) tertentu. Dalam sistem *automated document integration* ini digunakan algoritma *cosine similarity* dalam pengukuran kesamaan antar dokumen. Sebelum proses *clustering* dilakukan, suatu dokumen akan melalui proses *parsing*, *stemming*, dan pembobotan kalimat (*TF – IDF*) serta pembobotan relasi antar kalimat. Proses *stemming* dilakukan dengan menggunakan algoritma *porter stemmer for Bahasa Indonesia*.

2. TINJAUAN PUSTAKA

2.1 *Complete Linkage Agglomerative Hierarchical Clustering*

Salah satu kategori algoritma *clustering* yang banyak dikenal adalah *hierarchical clustering. Hierarchical clustering* merupakan salah satu algoritma *clustering* yang fungsinya dapat digunakan untuk pengelompokkan dokumen *(document clustering).* Dari teknik *hierarchical clustering*, dapat dihasilkan suatu kumpulan partisi yang berurutan, dimana dalam kumpulan tersebut terdapat :

- *Cluster-cluster* yang mempunyai poin-poin individu. *Cluster-cluster* ini berada di level yang paling bawah.
- Sebuah *cluster* yang didalamnya terdapat poin-poin yang dipunyai semua *cluster* didalamnya. *Single cluster* ini berada di level yang paling atas.

Dalam algoritma *hierarchical clustering, cluster* yang berada di level yang lebih atas *(intermediate level)* dari *cluster* yang lain dapat diperoleh dengan cara mengkombinasikan dua buah *cluster* yang berada pada level dibawahnya. Hasil keseluruhan dari algoritma *hierarchical clustering* secara grafik dapat digambarkan sebagai *tree,* yang disebut dengan *dendogram* (Tan, 2006).

Pada algoritma *agglomerative hierarchical clustering* ini, proses *hierarchical clustering* dimulai dari *cluster-cluster* yang memiliki poin-poin individu yang berada di level paling bawah. Pada setiap langkahnya, dilakukan penggabungan sebuah *cluster* dengan *cluster* lainnya, dimana *cluster-cluster* yang digabungkan berada saling berdekatan atau mempunyai tingkat kesamaan yang paling tinggi (Tan, 2006).

Salah satu metode yang digunakan dalam *Agglomerative Hierarchical Clustering* adalah *Complete linkage (furthest neighbor methods). Complete Linkage* adalah suatu metode yang menggunakan prinsip jarak minimum yang diawali dengan mencari jarak terjauh antar dua buah *cluster* dan keduanya membentuk *cluster* baru. Pada awalnya, dilakukan perhitungan jarak terpendek dalam $D = \{d_{ik}\}$ dan menggabungkan objek-objek yang bersesuaian misalnya, *U* dan *V* , untuk mendapatkan cluster (*UV*). Kemudian jarak-jarak antara (*UV*) dan *cluster W* yang lain dihitung dengan cara :

$$d_{(UV)W} = max\{d_{uw} , d_{vw}\}$$

2.2 Algoritma *Cosine Similarity*

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure.* Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vektor dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran.

Metode pengukuran kesesuaian ini memiliki beberapa keuntungan, yaitu adanya normalisasi terhadap panjang dokumen. Hal ini memperkecil pengaruh panjang dokumen. Jarak *euclidean* (panjang) kedua vektor digunakan sebagai faktor normalisasi. Hal ini diperlukan karena dokumen yang panjang cenderung mendapatkan nilai yang besar dibandingkan dengan dokumen yang lebih pendek.

Perhitungan *cosine similarity* yang memperhitungkan perhitungan pembobotan kata pada suatu dokumen dapat dinyatakan dengan perumusan :

$$CosSim(d_i, q_i) = \frac{q_i \bullet d_i}{|q_i||d_i|} = \frac{\sum_{j=1}^{t}(q_{ij}.d_{ij})}{\sqrt{\sum_{j=1}^{t}(q_{ij})^2.\sum_{j=1}^{t}(d_{ij})^2}}$$

Keterangan :
q_{ij} = bobot istilah *j* pada dokumen $i = tf_{ij} * idf_j$
d_{ij} = bobot istilah *j* pada dokumen $i = tf_{ij} * idf_j$

2.3 Proses *Parsing* dan *Stemming*

Parsing adalah sebuah proses untuk membuat sebuah kalimat menjadi lebih bermakna. Hal ini dilakukan dengan cara memecah kalimat tersebut menjadi kata-kata atau frase-frase (Budhi, 2005). Proses *parsing* merupakan proses penguraian dokumen yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antar kata seperti titik (.), koma (,), spasi dan tanda pemisah lain menjadi kata-kata saja.

Stemming merupakan suatu proses yang terdapat dalam sistem IR (*Information Retrieval*) yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu untuk meningkatkan kualitas informasi yang didapatkan (Agusta, 2009). Proses tersebut akan menghilangkan imbuhan yang terkandung dalam suatu kata yang diproses. Sebagai contoh, kata bersama, kebersamaan, menyamai, setelah melaui proses *stemming* akan menjadi kata "sama". Proses ini akan mendukung tingkat ketelitian dalam perhitungan daftar *keyword* pada proses berikutnya. Algoritma *stemming* untuk Bahasa Indonesia yang dikenal antara lain Algoritma Porter Stemmer dan Algoritma Nazief & Adriani. *Porter Stemmer for Bahasa Indonesia* dikembangkan oleh Fadillah Z. Tala pada tahun 2003. Implementasi *Porter Stemmer for Bahasa Indonesia* berdasarkan *English Porter Stemmer* yang dikembangkan oleh W.B. Frakes pada tahun 1992.

2.4 Proses *Stopword Removal*

Kebanyakan bahasa resmi di berbagai negara memiliki kata fungsi dan kata sambung seperti preposisi dan kata hubung yang hampir selalu muncul pada dokumen-dokumen teks. Kata-kata ini umumnya tidak memiliki arti yang lebih untuk memenuhi kebutuhan seorang *searcher* dalam mencari informasi. Kata-kata tersebut (misalnya *a, an, the, on* pada bahasa Inggris) yang disebut sebagai kata tidak penting misalnya "di", "oleh", "pada", "sebuah", "karena", dan kata sambung lainnya. Sebelum proses *stopword removal* dilakukan, terlebih dulu dibuat daftar *stopword (stoplist)*. Preposisi, kata hubung dan partikel biasanya merupakan kandidat *stoplist*. *Stopword removal* merupakan proses penghilangan kata tidak penting pada suatu dokumen, melalui pengecekan kata-kata hasil *stemmer* dokumen tersebut apakah termasuk kata di dalam daftar kata tidak penting *(stoplist)* atau tidak

2.5 Perhitungan Bobot Kalimat dan Bobot Relasi Antar Kalimat

2.5.1 TF-IDF (*Terms Frequency-Inverse Document Frequency*)

Metode ini merupakan metode untuk menghitung nilai/bobot suatu kata *(term)* pada dokumen. Metode ini akan mengabaikan setiap kata-kata yang tergolong tidak penting. Oleh sebab itu, sebelum melalukan metode ini, proses *stemming* dan *stopword removal* harus dilakukan terlebih dahulu oleh sistem. Karena melakukan pembobotan suatu kalimat bukan kata, pada metode ini terdapat 5 proses yang berbeda untuk perhitungan nilai suatu kalimat, yaitu (Budhi, 2008):

1. Kecocokan kata-kata pada kalimat dengan daftar kata kunci/*keyword*. Idenya adalah semakin tinggi nilai suatu kalimat, maka kalimat tersebut semakin penting keberadaannya di dalam suatu dokumen.
2. Menghitung frekuensi kata-kata suatu kalimat terhadap keseluruhan dokumen dan hasilnya akan dibagi dengan jumlah kata pada dokumen tersebut.
3. Bagian ketiga ini sangat sederhana yaitu hanya melihat posisi kalimat di dalam suatu paragraf. Berdasarkan metode deduktif induktif sesuai kaidah Bahasa Indonesia, ide pokok suatu paragraf terdapat pada kalimat yang berada di awal dan atau akhir dari paragraf tersebut.
4. Bagian keempat ini sangat berhubungan dengan hasil pemetaan dokumen. Pada bagian keempat ini akan dihitung jumlah relasi (yang disimbolkan dengan *edge*) suatu kalimat di dalam dokumen. Idenya adalah semakin banyak relasi yang dimiliki suatu kalimat dengan kalimat lainnya di dalam suatu dokumen maka kalimat tersebut kemungkinan mendiskusikan topik utama suatu dokumen.
5. Bobot kelima ini merepresentasikan seberapa penting sebuah kalimat dibandingkan dengan kalimat-kalimat lain yang terdapat pada semua dokumen yang akan diintegrasikan.

Setelah mendapatkan hasil dari kelima bobo diatas, selanjutnya nilai *tf* akan dihitung dengan persamaan berikut:

$$tf = bobot_1 + bobot_2 + bobot_3 + bobot_4 + bobot_5$$

Faktor lain yang diperhatikan dalam pemberian bobot adalah kejarangmunculan kalimat *(sentence scarcity*) dalam koleksi. Kalimat yang muncul pada sedikit dokumen harus dipandang sebagai kata yang lebih penting (*uncommon sentences*) daripada kalimat yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kalimat (*inverse document frequency*).

Nilai dari *tf* akan dikalikan dengan nilai *idf* seperti pada persamaan dibawah ini (Intan, 2005):

$$w = tf\ x\ idf$$

$$idf = log\frac{N}{n}$$

Keterangan :
W= bobot kalimat terhadap dokumen

tf = jumlah kemunculan kata/*term* dalam dokumen
N= jumlah semua dokumen yang ada dalam database
n= jumlah dokumen yang mengandung kata/*term*
idf = inverse document frequency

2.5.3 Perhitungan Bobot *Edge*

Untuk perhitungan bobot *edge* akan digunakan persamaan berikut (Sjobergh, 2005) :

$$Cost_{i,j} = \frac{(i-j)^2}{overlap_{i,j}\, x\, weight_j}$$

Nilai *overlap* $_{i,j}$ diperoleh dengan menghitung jumlah kata yang sama antara kalimat ke-*i* dan kalimat ke-*j* dengan mengabaikan *stopword* yang ada di dalam kalimat-kalimat tersebut. Kemudian hasil dari persamaan diatas akan digunakan untuk menentukan nilai relasi dari setiap kalimat berdasarkan hasil pemetaan dari dokumen.

3. DESAIN AUTOMATED DOCUMENT INTEGRATION SYSTEM

Tahap awal yang dilakukan dalam pengembangan sistem adalah penentuan input, proses, dan output dari sistem yang akan dibuat. Input – input yang masuk dan akan diproses dalam sistem dapat dibagi menjadi 2 bagian yaitu :

1. Penentuan *input* sistem yang berupa kumpulan dokumen yang akan diintegrasikan. Dokumen disini berperan sebagai suatu kumpulan data-data mentah yang akan dijadikan objek pada penelitian ini. Dokumen berupa artikel-artikel mengenai teknologi informasi dalam Bahasa Indonesia dengan *format file* PDF.
2. Penentuan input yang kedua adalah input dari *user* yang berupa nilai toleransi kesamaan antar dokumen yang akan diintegrasikan (*similarity tolerance value*) ke sistem.

Setelah melakukan teknik kajian pustaka pada tahap sebelumnya, secara garis besar proses-proses yang ada pada sistem dapat dibagi ke dalam dua subsistem yaitu :

1. Subsistem *Pre-Integration*

Proses – proses yang ada pada subsistem ini adalah :

a. Proses upload dokumen ke dalam sistem.
b. Proses konversi dokumen dengan format file PDF menjadi file txt.
c. Proses *devide to word* atau *parsing* yaitu proses yang memecah kalimat-kalimat dalam file txt menjadi kata-kata.
d. Proses *stopword removal* atau menghilangkan kata-kata tidak penting.
e. Proses stemming dengan algoritma *Porter Stemmer for Bahasa Indonesia.*
f. Proses perhitungan kesamaan dokumen dengan algoritma *Cosine Similarity*.

2. Subsistem *Integration Process*

Proses – proses yang ada pada subsistem ini adalah :

a. Proses perhitungan bobot kalimat dengan metode TF-IDF.
b. Proses perhitugan bobot relasi antar kalimat.
c. Proses *clustering* dengan algoritma *Complete Linkage Agglomerative Hierarchical Clustering*

Pada proses integrasi dengan algoritma *agglomerative hierarchical clustering*, awalnya semua kalimat yang terdapat dalam tabel kalimat dianggap sebagai *atomic cluster – atomic cluster*. Langkah pertama yang dilakukan adalah mencari cluster-cluster dengan jarak terdekat, atau pasangan kalimat yang memiliki bobot relasi antar kalimat yang paling kecil. Pencarian dilakukan dengan menggunakan perintah *query select* yang mengurutkan data-data pada tabel kalimat_relasi secara *ascending* berdasarkan bobot relasinya. Langkah selanjutnya adalah melakukan *update* jarak *cluster* yang baru terbentuk dengan *cluster-cluster* lainnya dengan metode *maximum distance.* Setelah semua kalimat telah tergabung menjadi sebuah *cluster*, dilakukan proses untuk memecah *cluster* tersebut menjadi paragraf – paragraf. Caranya adalah, kalimat – kalimat yang bergabung terlebih dahulu menjadi *cluster – cluster* besar dianggap sebagai sebuah paragraf tersendiri.

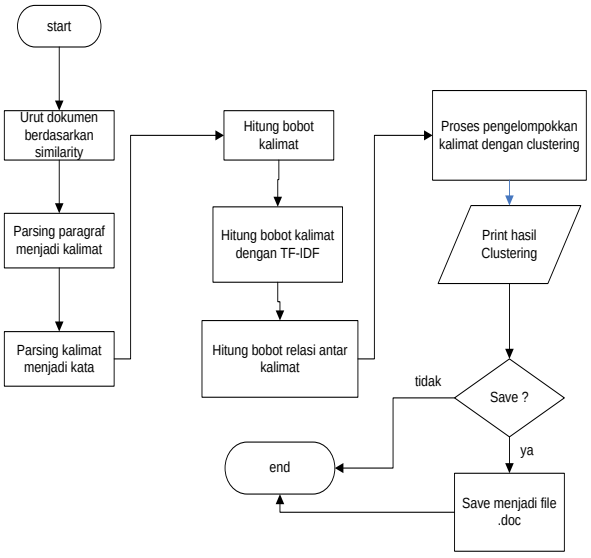
Asumsinya, bila secara *natural* kalimat – kalimat tersebut bergabung, dapat dianggap kalimat – kalimat tersebut memiliki *similarity* yang cukup tinggi dan membahas topik bahasan yang sama. Sementara untuk memproses kalimat – kalimat tersisa yang tidak mau bergabung kedalam *cluster – cluster* besar, dipakai aturan sebagai berikut:

- Bila hanya 1 kalimat akan digabungkan pada paragraf terakhir.
- Bila lebih dari satu kalimat, kalimat – kalimat yang tersisa tersebut akan dipaksakan bergabung menjadi satu paragraf tersendiri.

Sementara, *Output-output* yang dihasilkan sistem melalui pemrosesan input dari user adalah :

a. *Report* tingkat kesamaan antar dokumen yang akan diintegrasikan.
b. Dokumen hasil proses integrasi yang dapat disimpan dalam format file .doc.

Berikut ini adalah gambar rancangan alur pada subsistem *Integration* secara garis besar :



Gambar 1. Perancangan Alur pada Subsistem *Integration*

4. PENGUJIAN SISTEM

4.1 Pengujian Sistem dengan *White Box*

Pengujian *white box* yang digunakan dalam pembuatan *automated document integration system* ini adalah pengujian *basis path.* Pengujian tersebut ditujukan untuk mencari *path-path* yang dilalui saat program dijalankan. Pengujian *basis path* pada penelitian ini difokuskan pada proses perhitungan bobot relasi antar kalimat dan proses integrasi dengan algoritma *complete linkage agglomerative hierarchical clustering*. Alur logika dari sistem diujicoba dengan menyediakan kasus ujicoba yang melakukan semua kondisi atau perulangan yang ada pada sistem. Dimana setelah melakukan pengujian, setiap path yang ada dalam tiap proses dapat dijalan dengan baik.

4.2 Pengujian Proses Integrasi

Pengujian terhadap proses integrasi dilakukan dengan membandingkan proses integrasi dengan perhitungan manual dan dengan sistem. Dokumen yang dipergunakan adalah dokumen dengan judul *Data Mining* 1 dan *Data Mining* 2. Pada akhir pengujian didapatkan hasil bahwa hasil integrasi secara manual menghasilkan dokumen yang sama dengan dokumen yang diintegrasikan oleh sistem. Berikut ini adalah contoh dokumen yang akan diintegrasikan dengan *automated document integration system.*

"Data Mining 1"

Data Mining memang salah satu cabang ilmu komputer yang relatif baru. Dan sampai sekarang orang masih memperdebatkan untuk menempatkan data mining di bidang ilmu mana, karena data mining menyangkut database, kecerdasan buatan (artificial intelligence), statistik, dsb. Ada pihak yang berpendapat bahwa data mining tidak lebih dari machine learning atau analisa statistik yang berjalan di atas database. Namun pihak lain berpendapat bahwa database berperanan penting di data mining karena data mining mengakses data yang ukurannya besar (bisa sampai terabyte) dan disini terlihat peran penting database terutama dalam optimisasi query-nya.

Kehadiran data mining dilatar belakangi dengan problema data explosion yang dialami akhir-akhir ini dimana banyak organisasi telah mengumpulkan data sekian tahun lamanya (data pembelian, data penjualan, data nasabah, data transaksi dsb.). Hampir semua data tersebut dimasukkan dengan menggunakan aplikasi komputer yang digunakan untuk menangani transaksi sehari-hari yang kebanyakan adalah OLTP (On Line Transaction Processing). Bayangkan berapa

transaksi yang dimasukkan oleh hypermarket semacam Carrefour atau transaksi kartu kredit dari sebuah bank dalam seharinya dan bayangkan betapa besarnya ukuran data mereka jika nanti telah berjalan beberapa tahun. Pertanyaannya sekarang, apakah data tersebut akan dibiarkan menggunung, tidak berguna lalu dibuang, ataukah kita dapat menambangnya untuk mencari emas, berlian yaitu informasi yang berguna untuk organisasi kita. Banyak diantara kita yang kebanjiran data tapi miskin informasi.

"*Data Mining 2*"

Data Mining (DM) adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari database skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Definisi umum dari DM itu sendiri adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data. Dalam review ini, penulis mencoba merangkum perkembangan terakhir dari teknik-teknik DM beserta implikasinya di dunia bisnis.

Perkembangan data mining(DM) yang pesat tidak dapat lepas dari perkembangan teknologi informasi yang memungkinkan data dalam jumlah besar terakumulasi. Sebagai contoh, toko swalayan merekam setiap penjualan barang dengan memakai alat POS(point of sales). Database data penjualan tsb. bisa mencapai beberapa GB setiap harinya untuk sebuah jaringan toko swalayan berskala nasional. Perkembangan internet juga punya andil cukup besar dalam akumulasi data.

Tetapi pertumbuhan yang pesat dari akumulasi data itu telah menciptakan kondisi yang sering disebut sebagai "rich of data but poor of information" karena data yang terkumpul itu tidak dapat digunakan untuk aplikasi yang berguna. Tidak jarang kumpulan data itu dibiarkan begitu saja seakan-akan "kuburan data" (data tombs).

Hasil dari proses integrasi kedua dokumen diatas adalah sebagai berikut :

Kehadiran data mining dilatar belakangi dengan problema data explosion yang dialami akhir-akhir ini dimana banyak organisasi telah mengumpulkan data sekian tahun lamanya (data pembelian, data penjualan, data nasabah, data transaksi dsb). Hampir semua data tersebut dimasukkan dengan menggunakan aplikasi komputer yang digunakan untuk menangani transaksi sehari-hari yang kebanyakan adalah OLTP (On Line Transaction Processing). Bayangkan berapa transaksi yang dimasukkan oleh hypermarket semacam Carrefour atau transaksi kartu kredit dari sebuah bank dalam seharinya dan bayangkan betapa besarnya ukuran data mereka jika nanti telah berjalan beberapa tahun.

Pertanyaannya sekarang, apakah data tersebut akan dibiarkan menggunung, tidak berguna lalu dibuang, ataukah kita dapat menambangnya untuk mencari emas, berlian yaitu informasi yang berguna untuk organisasi kita. Banyak diantara kita yang kebanjiran data tapi miskin informasi.

Dan sampai sekarang orang masih memperdebatkan untuk menempatkan data mining di bidang ilmu mana, karena data mining menyangkut database, kecerdasan buatan (artificial intelligence), statistik, dsb. Ada pihak yang berpendapat bahwa data mining tidak lebih dari machine learning atau analisa statistik yang berjalan di atas database. Namun pihak lain berpendapat bahwa database berperanan penting di data mining karena data mining mengakses data yang ukurannya besar (bisa sampai terabyte) dan disini terlihat peran penting database terutama dalam optimisasi query-nya. Data Mining memang salah satu cabang ilmu komputer yang relatif baru.

Tetapi pertumbuhan yang pesat dari akumulasi data itu telah menciptakan kondisi yang sering disebut sebagai "rich of data but poor of information" karena data yang terkumpul itu tidak dapat digunakan untuk aplikasi yang berguna. Tidak jarang kumpulan data itu dibiarkan begitu saja seakan-akan "kuburan data" (data tombs). Perkembangan internet juga punya andil cukup besar dalam akumulasi data.

Data Mining (DM) adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari database skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Definisi umum dari DM itu sendiri adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan

yang selama ini tidak diketahui secara manual dari suatu kumpulan data. Dalam review ini, penulis mencoba merangkum perkembangan terakhir dari teknik-teknik DM beserta implikasinya di dunia bisnis. Perkembangan data mining(DM) yang pesat tidak dapat lepas dari perkembangan teknologi informasi yang memungkinkan data dalam jumlah besar terakumulasi. Sebagai contoh, toko swalayan merekam setiap penjualan barang dengan memakai alat POS(point of sales). Database data penjualan bisa mencapai beberapa GB setiap harinya untuk sebuah jaringan toko swalayan berskala nasional.

4.3 Pengujian Terhadap Waktu Proses Integrasi

No.	No Urut	Judul Dokumen	Ukuran Dokumen			Ukuran Hasil Integrasi			Waktu *Complete Linkage Agglomerative Hierarchical Clustering*	Waktu Proses Total
			Paragraf Jumlah	Kalimat Jumlah	Jumlah Kata	Paragraf Jumlah	Kalimat Jumlah	Jumlah Kata		
1	1	Algoritma Genetik 1	1	5	84	5	13	176	0,291 Detik	3,602 Detik
	2	Algoritma Genetik 2	2	8	92					
2	1	HTML 1	2	7	73	8	26	250	1,315 Detik	8,212 Detik
	3	HTML 2	2	10	89					
	2	HTML 3	2	9	88					
3	1	*Data Mining* 1	3	11	132	6	25	266	1,195 Detik	8,996 Detik
	2	*Data Mining* 2	4	14	134					
4	3	Algoritma Genetik 1	1	5	84	4	13	176	0,293 Detik	3,510 Detik
	1	Algoritma Genetik 2a	1	3	43					
	2	Algoritma Genetik 2b	1	5	49					

Untuk mendapatkan hasil uji coba yang terbaik mengenai lama proses *clustering* dan proses integrasi keseluruhan yang dilakukan sistem maka perlu dilakukan uji coba terhadap berbagai dokumen yang berbeda-beda. Dari dokumen yang berbeda-beda tersebut yang perlu diperhatikan adalah dari segi jumlah dokumen yang diintegrasi, jumlah paragraf dari setiap dokumen asli, dan jumlah kalimat dari setiap dokumen asli. Pengujian waktu proses ini diujikan dengan keadaan *hardware* dan *software* sebagai berikut : *Processor Intel Pentium Dual Core T2390 1,86 Ghz, RAM 1 GB.* Hasil pengujian sistem terhadap waktu dapat dilihat pada tabel 1.

4.4 Evaluasi Relevansi Sistem

Pengujian ini dilakukan dengan cara meminta bantuan 100 orang responden yang merupakan mahasiswa Jurusan Ilmu Komputer Universitas Udayana untuk membaca dokumen – dokumen asal yang berjenis eksposisi dan narasi serta membaca dokumen hasil integrasi, kemudian menjawab 3 pertanyaan berikut:

1) Menurut anda, apakah kata-kata pada dokumen hasil integrasi tersebut telah terorganisir dengan baik (tiap paragraf memberikan arti yang jelas dan dapat dipahami) ? A. Ya B. Tidak

2) Menurut anda, apakah dokumen hasil integrasi tersebut telah memberikan gambaran secara umum dari keseluruhan dokumen yang ada sebelumnya ? A. Ya B. Tidak

3) Menurut anda, apakah dokumen hasil integrasi dapat memberikan informasi - informasi penting yang terdapat pada dokumen sebelumnya secara jelas? A. Ya B. Tidak

Berdasarkan hasil survei relevansi didapatkan kesimpulan bahwa sistem akan bekerja lebih baik pada dokumen yang bertipe eksposisi dibandingkan dengan dokumen yang bertipe narasi. Dimana lebih dari 75 persen mengatakan bahwa hasil integrasi dokumen yang bertipe eksposisi dapat mewakili dokumen asli, memiliki susunan antar paragraf yang jelas, dan tetap mengandung informasi penting dari dokumen awal.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

- ✓ Secara umum sistem *automated document integration* dapat dibagi menjadi dua subsistem yaitu subsistem *pre-integration* dan subsistem *integration*. Adapun proses yang termasuk dalam subsistem *pre-integration* adalah proses *devide to word*, proses *stopword removal*, proses *stemming*, dan proses perhitungan kesamaan. Proses integrasi dokumen berada dalam subsistem *integration* menggunakan algoritma *complete linkage agglomerative hierarchical clustering*

yang sebelumnya didahului dengan proses pembobotan kalimat dengan metode TF-IDF(*term frequency-inverse document frequency*) dan pembobotan relasi antar kalimat.

- ✓ Hasil dari pengujian terhadap waktu dari proses integrasi dan proses *hierarchical clustering*, menunjukkan bahwa semakin banyak dokumen paragraf dan kalimat yang yang diproses maka akan membutuhkan *running time* yang semakin besar pula.
- ✓ Berdasarkan hasil survei relevansi sistem kepada 100 orang responden, didapatkan kesimpulan bahwa sistem akan bekerja lebih baik pada dokumen yang bertipe eksposisi dibandingkan dengan dokumen yang bertipe narasi. Dimana lebih dari 75 persen mengatakan bahwa hasil integrasi dokumen yang bertipe eksposisi dapat mewakili dokumen asli, memiliki susunan antar paragraf yang jelas, dan tetap mengandung informasi penting dari dokumen awal.

5.2 Saran

- ✓ Sistem *automated document integration* memiliki kelemahan dalam menentukan urutan dokumen yang diintegrasi. Dalam sistem *automated document integration* ini urutan dokumen ditentukan berdasarkan tingkat kesamaannya dengan metode *cosine similarity*. Padahal *cosine similarity* merupakan metode yang bersifat simetris, sehingga tidak dapat menentukan suatu urutan. Untuk pengembangan selanjutnya disarankan agar menggunakan metode pengukuran kesamaan yang asimetris.
- ✓ Pada pengembangan selanjutnya, sistem dapat dikembangkan untuk melakukan integrasi dokumen-dokumen dengan bahasa lain, tidak terbatas pada dokumen yang berbahasa Indonesia saja.
- ✓ Pada pengembangan selanjutnya, sistem dapat dikembangkan dengan menggunakan algoritma lain sehingga dapat mengintegrasi dokumen narasi dengan baik, disamping itu juga dapat diteliti hasilnya dalam mengintegrasi jenis dokumen selain dokumen berjenis eksposisi dan narasi.

6. DAFTAR PUSTAKA

Agusta, Ledy. 2009. *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia*. Bali : Fakultas Teknologi Informasi Universitas Kristen Satya Wacana.

Budhi, Gregorius S, Arlinah I. Rahardjo, dan Hendrawan Taufik. 2008. *Hierarchical Clustering Untuk Aplikasi Automated Text Integration*. Surabaya : UK Petra Jurusan Teknik Informatika.

Budhi, Gregorius S, Ibnu Gunawan dan Ferry Yuwono. 2005. *Algoritma Porter Stemmer For Bahasa Indonesia Untuk Pre-Processing Text Mining Berbasis Metode Market Basket Analysis.* Surabaya : UK Petra Jurusan Teknik Informatika.

Hamzah, Amir. 2009. *Temu Kembali Informasi Berbasis Kluster Untuk Sistem Temu Kembali Informasi Teks Bahasa Indonesia*. Yogyakarta : Jurusan Teknik Informatika, Fakultas Teknologi Industri Institut Sains & Teknologi AKPRIND.

Hartini, Entin. 2004. *Metode Clustering Hirarki*. Pusat Pengembangan Teknologi dan Komputasi Batan.

Intan, Rolly dan Andrew Defeng. 2005. *HARD: Subject-based Search Engine menggunakan TF-IDF dan Jaccard's Coefficient*. Surabaya : UK Petra Jurusan Teknik Informatika.

Kendall, Kenneth E dan Julie E. Kendall. 2006. *Analisis dan Perancangan Sistem.Edisi kelima*. Indeks. Jakarta.

Mandala, Rila dan Hendra Setiawan. 2002. Peningkatan Performansi Sistem Temu-Kembali Informasi dengan Perluasan Query Secara Otomatis. Bandung : Departemen Teknik Informatika Institut Teknologi Bandung.

Pramudiono, Iko. 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. Tokyo : The University of Tokyo.

Pressman, Roger S. 2005. *Software Engineering A practitioner's Approach Sixth Edition*. New York : Mc-Graw-Hill.

Sjobergh, Jonas dan Kenji Araki. 2005. *Extraction Based Summarization Using Shortest Path Algorithm.* Sweden : KTH Nada.

Soebroto dan Arief Andy. 2005. *Hybrid Average Complete Clustering sebagai Algoritma Kompromi antara Kualitas dan Waktu Komputasi Proses Clustering*. Surabaya : Institut Teknologi Sepuluh November.

Steinbach, Michael, George Karypis dan Vipin Kumar. 2005. *A Comparisont of Document Clustering Techniques.* Minnesota : University of Minnesota, Department of Computer Science and Engineering.

Tan, P. N., M. Steinbach dan V. Kumar. 2005. *Introduction to Data Mining*. New York : Addison Wesley.