PENGELOMPOKKAN DOKUMEN MENGGUNAKAN K-MEANS DAN SINGULAR VALUE DECOMPOSITION: STUDI KASUS MENGGUNAKAN DATA BLOG

Munzir Umran[1] **Taufik Fuadi Abidin**[2]
Data Mining and Information Retrieval Research Group
Jurusan Matematika FMIPA
Universitas Syiah Kuala
Banda Aceh, 23111 Indonesia

E-mail: mu2n_jir@yahoo.co.id[1], taufik.abidin@unsyiah.ac.id[2]

Peningkatan jumlah dokumen dalam format teks yang cukup signifikan belakangan ini, seperti blogs dan website, membuat proses pengelompokkan dokumen (*document clustering*) menjadi semakin penting. Pengelompokkan dokumen bertujuan membagi dokumen dalam beberapa kelompok (*cluster*) sedemikian hingga dokumen-dokumen dalam cluster yang sama (*intra-cluster*) memiliki derajat kesamaan yang tinggi, sementara dokumen-dokumen dalam cluster yang berbeda (*inter-cluster*) memiliki derajat kesamaan yang rendah. Tulisan ini mendiskusikan dan memperlihatkan metode pengelompokkan dokumen yang dimulai dengan membangun matriks *terms-documents A* dan kemudian memecahnya menjadi tiga matriks *TSD* menggunakan *Singular Value Decomposition* (SVD). Selanjutnya, pengelompokkan dokumen dilakukan dengan *k*-means. *T* adalah matriks kata (*terms*) berukuran *t* x *r*, *S* adalah matriks diagonal berisi nilai skalar (*eigen values*) berdimensi *r* x *r*, dan *r* ditentukan sebelumnya, *D* adalah matriks dokumen berukuran *r* x *d*. Dekomposisi nilai singular dari matriks A dinyatakan sebagai $A = TSD^T$. Eksperimen dilakukan menggunakan data blog dan dekomposisi matrik menggunakan program *General Text Parser* (GTP). Hasil menunjukkan bahwa dekomposisi matrik *terms-documents A* dengan *Singular Value Decomposion* dapat mempercepat proses pengelompokkan dokumen karena dimensi dari setiap vector telah diperkecil tanpa mengurangi arti sebenarnya. Namun, karena metode clustering yang digunakan adalah k-means maka hasil cluster sangat sensitif terhadap dokumen yang diduga sebagai *outlier.*

Keywords: *Document clustering, Singular Value Decomposition, k-means, Latent Semantic Indexing*

1. PENDAHULUAN

Peningkatan jumlah dokumen dalam format teks yang cukup signifikan belakangan ini membuat proses pengelompokkan dokumen (*document clustering*) menjadi penting. Pengelompokkan dokumen bertujuan membagi dokumen dalam beberapa kelompok (*cluster*) sedemikian hingga dokumen-dokumen dalam cluster yang sama (*intra-cluster*) memiliki kesamaan yang tinggi, sementara dokumen-dokumen dalam cluster yang berbeda (*inter-cluster*) memiliki kesamaan yang rendah.

Berdasarkan Graepel [5], secara formal clustering dapat didefinisikan sebagai berikut: Jika $X \in R^{m+n}$ merupakan sejumlah data yang merepresentasikan *m* buah data poin x_i dalam ruang dimensi R^n, maka proses clustering akan mengelompokkan dataset *X* dalam *k* buah cluster C_k sedemikian hingga data-data dalam cluster C_k memiliki derajat kesamaan yang tinggi.

Beberapa algoritma clustering yang telah dikembangkan dan diuji diantaranya adalah *k-means*, *k-median*, DBSCAN, CLARAN, dan DENCLUE [6][7][8][9]. Masing-masing algoritma memiliki keunggulan dan kekurangan. Sebagai contoh: *k*-means menghasilkan cluster berkualitas untuk data yang tidak banyak memiliki outlier, sementara *k*-median tidak sensitif terhadap outlier, namun membutuhkan waktu yang lebih lama untuk mencari centroid (median).

Tulisan ini mendiskusikan dan memperlihatkan metode pengelompokkan dokumen yang dimulai dengan membangun matriks *terms-documents A* dan kemudian dipecah menjadi tiga matriks *TSD* menggunakan *Singular Value Decomposition*. SVD telah diimplementasikan dalam program *General Text Parser* [2]. Matriks *T* merupakan matriks kata (*term*), dan matriks *S* merupakan matriks diagonal yang berisi nilai skalar (*eigen values*), sedangkan matriks *D* merupakan matriks dokumen, sedemikian hingga $A = TSD^T$. Kemudian proses clustering dilakukan dengan

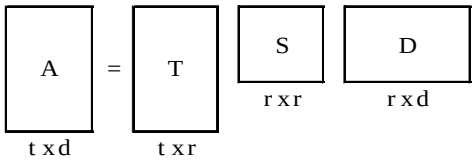
algoritma *k*-means. Data yang digunakan dalam eksperimen ini adalah data blog.

Tulisan ini terbagi dalam beberapa bagian. Bagian 2 menjelaskan tentang metodelogi, bagian 3 menjelaskan tentang metode yang diusulkan, bagian 4 mendiskusikan hasil dan bagian 5 menyimpulkan hasil percobaan.

2. METODELOGI

2.1 Singular Value Decomposition (SVD)

Singular Value Decomposition adalah metode aljabar linier [1] yang memecah matriks *A* (*terms-documents*) berdimensi *t* x *d* menjadi tiga matriks *TSD*. *T* adalah matriks kata (*terms*) berukuran *t* x *r*, *S* adalah matriks diagonal berisi nilai skalar (*eigen values*) berdimensi *r* x *r*, dan *r* ditentukan sebelumnya, dan *D* adalah matriks dokumen berukuran *r* x *d*. Dekomposisi nilai singular dari matriks *A* dinyatakan sebagai $A = TSD^T$, seperti yang diilustrasikan pada Gambar 1 berikut ini.



Gambar 1. Dekomposisi matriks A dengan SVD menjadi tiga matriks TSD^T.

SVD dapat mereduksi dimensi dari matriks *A* dengan cara mengurangi ukuran *r* dari matriks diagonal *S*. Pengurangan dimensi dari matriks *S* dilakukan dengan cara mengubah semua nilai diagonal matriks *S* menjadi nol, kecuali untuk nilai diagonal dari dimensi yang tersisa. Pengalian ketiga matriks TSD^T akan membentuk matriks *A* awal dengan nilai setiap elemennya mendekati nilai sebenarnya [4][10].

Untuk dapat memecah matriks *A* dengan SVD, vektor orthonormal harus diperoleh agar matriks *A* dapat didiagonalkan. Untuk itu, eigen-vektor $\vec{x}$ yang searah dengan $A\vec{x}$ perlu dicari. Perkalian eigen-vektor $\vec{x}$ dengan *A* akan menghasilkan $\lambda\vec{x}$ dimana λ adalah nilai eigen.

$$A\vec{x} = \lambda\vec{x} \qquad (1)$$

Persamaan 1 dapat diselesaikan dengan mengubahnya dalam bentuk sebagai berikut:

$$(A - \lambda I)\vec{x} = 0 \qquad (2)$$

E adalah matriks identitas dengan elemen diagonalnya bernilai 1. Jika persamaan 2 memiliki solusi maka $(A - \lambda I)$ tidak invertibel dan determinannya akan bernilai 0, nilai eigen λ dapat ditentukan.

$$\det(A - \lambda I)\vec{x} = 0 \qquad (3)$$

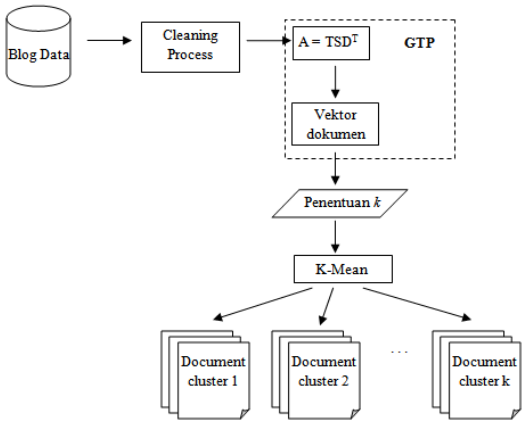
2.2 K-means

Jika diberikan sekumpulan data $X = \{x_1, x_2, \ldots, x_n\}$ dimana $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ adalah vektor dalam ruang real R^n, maka algoritma *k*-means akan mempartisi *X* dalam *k* buah cluster. Setiap cluster memiliki *centroid* (titik tengah) atau *mean* dari data-data dalam cluster tersebut.

Pada tahap awal, algoritma *k*-means memilih secara acak *k* buah data sebagai centroid. Kemudian, jarak antara data dan centroid dihitung menggunakan *Euclidian distance*. Data ditempatkan dalam cluster yang terdekat, dihitung dari titik tengah cluster. Centroid baru akan ditentukan bila semua data telah ditempatkan dalam cluster terdekat. Proses penentuan centroid dan penempatan data dalam cluster diulangi sampai nilai centroid konvergen (centroid dari semua cluster tidak berubah lagi).

3. METODE YANG DIUSULKAN

Gambar 2 memperlihatkan langkah-langah pengelompokkan dokumen menggunakan metode yang diusulkan. Proses dimulai dengan membersihkan dokumen-dokumen dalam database. Kemudian, matriks *terms-documents A* dibangun dan dipecah menjadi tiga matriks *TSD* menggunakan program General Text Parser [2]. Vektor dokumen dibangun dengan mengalikan matriks D^T dengan *S*. Selanjutnya, parameter *k* ditentukan dan proses pengelompokkan dilakukan dengan *k*-means.



Gambar 2. Flow chart dalam eksperimen

3.1 Dataset

Data yang digunakan dalam eksperimen ini adalah sebagian dari data blog yang dipersiapkan oleh Spinn3r.com untuk kompetisi pada *the*

International Conference on Weblogs and Social Media 2009 [3]. Jumlah total artikel dalam dataset tersebut adalah sebanyak 44 juta yang dipublikasi antara bulan Agustus sampai dengan Oktober 2008 dengan topik, diantaranya Olympics, nominasi calon presiden Amerika Serikat, dan awal dari krisis finansial di Amerika Serikat. Data ini disusun dalam format, seperti yang diperlihatkan pada Gambar 3.

```
<item>
<title>iMoneysoft 1.30 (Trial)</title>
<link>http://www.softpedia.com</link>
<guid>http://softpedia.com/isoft.shtml
</guid>
<pubDate>Fri, 01 Aug 2008GMT</pubDate>
<weblog:title>Softpedia - Windows -
All</weblog:title>
<weblog:description>Softpedia -
Windows - All</weblog:description>
<dc:lang>en</dc:lang>
<weblog:tier>3</weblog:tier>
<description>The personal finance
software provides clear navigation
designed to help you switch to
different financial operation
interfaces.
</description>
<weblog:publisher_type>WEBLOG</weblog:
publisher_type>
</item>
```

Gambar 3. Contoh data blog yang digunakan sebelum proses cleaning

Blog terdiri atas beberapa bagian seperti judul, tanggal publikasi, link, kode bahasa, dan deskripsi. Dalam penelitian ini, hanya bagian judul dan deskripsi yang digunakan karena kedua bagian ini menyimpan informasi penting dari artikel blog. Bagian judul menjabarkan ringkasan dari blog sedangkan bagian deskripi merupakan isi dari blog.

3.2 Proses Pembersihan (*Cleaning Process*)

Sebelum proses cleaning dilakukan, artikel blog dalam file XML dipecah dalam beberapa file. Bersamaan dengan proses itu, tag-tag HTML seperti `<title>` dan `</title>` dibuang. Selain itu, karakter dan simbol-simbol yang tidak bermakna, seperti `&lt; &gt;`, `&#0876;`, dan `&quot;` dihapus.

Dalam experimen ini, proses cleaning dilakukan menggunakan Perl. Perl adalah skrip bertipe data dinamis dan dapat dieksekusi secara langsung oleh interpreter Perl. Perl memiliki fasilitas *regular expression* (atau *regex*) dan diakui sebagai bahasa yang memiliki regex terlengkap, dibuktikan dengan adanya implementasi regex PCRE atau *Perl-compatible regular expression*. Gambar 4 merangkum Perl regex yang digunakan dalam proses cleaning.

```
# remove <title> and </title> tags
$content =~ s/<[\/]*title>//g;

# remove element in script tag
$content =~
s/&amp;lt;script.*?\/script&amp;gt;//gs;

# remove a href and its contents
$content =~ s/&amp;lt;a.*?\/a&amp;gt;//gs;

# remove all chars between &amp;lt; &amp;gt;
$content =~ s/&amp;lt;.*?&amp;gt;/ /gs;

# remove string in this form #0876;
$content =~ s/&amp;amp;#\d+;//gs;

# remove &amp;amp;lt;
$content =~ s/&amp;amp;lt;/ /gs;

# remove &amp;amp;nbsp;
$content =~ s/&amp;amp;nbsp;//gs;

# substitute /&amp;#039; menjadi quote
$content =~ s/&amp;#039;/'/gs;

$content =~ s/quote//gs;
$content =~ s/_/ /gs;
$content =~ s/&amp;\w+;/ /gs;

# remove #quot, replace with a space
$content =~ s/\s+/ /gs;

$content =~ s/<description>//g;
```

Gambar 4. Proses cleaning dengan Perl regex

Gambar 5 memperlihatkan artikel blog yang sudah dibersihkan.yang sebelumnya ditampilkan pada Gambar 3.

```
The personal finance software provides
clear navigation designed to help you
switch to different financial
operation interfaces.
```

Gambar 5. Artikel blog yang sudah dibersihkan

3.3 General Text Parser (GTP)

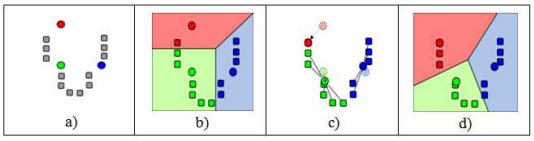
General Text Parser adalah suatu paket program dalam bahasa C yang mengimplementasikan SVD. GTP menghasilkan data vektor yang dapat digunakan untuk menyelesaikan masalah retrival informasi (IR). GTP versi 4.0, berjalan dalam sistem operasi Linux dan telah diuji pada Redhat versi 3.2.2-5 [2].

GTP dapat menerima input berupa sebuah file yang didalamnya terdapat beberapa data (*entry*) yang dipisah dengan dua buah baris baru atau sebuah direktori yang didalamnya terdapat beberapa file (dokumen). GTP menghilangkan *stopwords* atau kata-kata yang sangat sering muncul namun kurang bermakna yang ditemukan dalam dokumen. Kemudian, GTP membangun matriks *terms-documents* dan memecah matriks tersebut menjadi tiga matriks menggunakan *Singular Value Decomposition*.

GTP mengizinkan pengguna untuk menentukan ukuran dimensi dari matriks singular. Output dari GTP adalah sebuah file ascii yang terdiri dari vektor terms, vektor dokumen, dan nilai eigen.

3.4 Proses Clustering

K-means memilih secara acak *k* buah data sebagai centroid. Kemudian menempatkan data dalam cluster yang terdekat, dihitung dari titik tengah cluster (*centroid*). Centroid baru akan ditentukan bila semua data telah ditempatkan dalam cluster terdekat. Proses penentuan centroid dan penempatan data dalam cluster diulangi sampai nilai centroid konvergen. Gambar 6 memperlihatkan cara kerja *k*-means dan algoritma 1 memperlihatkan langkah-langkah proses *k*-means.



Gambar 6. Pengelompokkan menggunakan *k-means* [11].

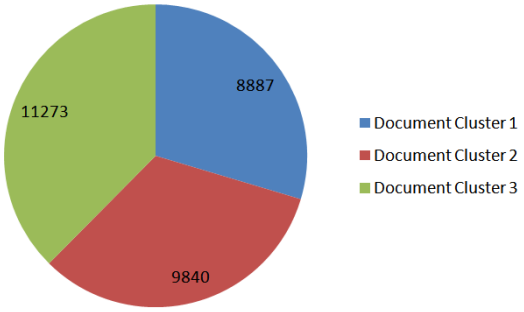
ALGORITMA 1: Proses k-means

```
Input: vektor dokumen D, k
Output: k cluster dokumen
1. Pilih secara acak k vektor
   sebagai centroid
2. repeat
3.     tempatkan data (vektor) dalam
       cluster atau centroid terdekat
4.     hitung centroid baru dari
       cluster yang terbentuk
5. until centroid tidak berubah lagi
```

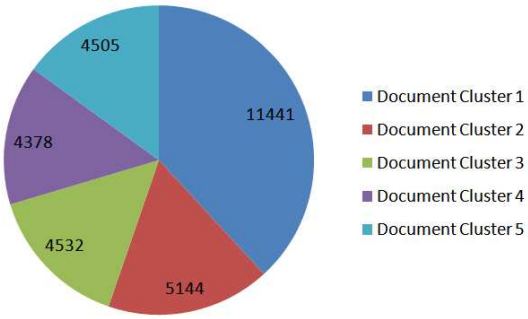
4. HASIL

Dari data blog yang digunakan, diperoleh sebuah matriks *terms-document A* berukuran 48,512 x 30,000. Nilai elemen dari matriks tersebut adalah total frekuensi dari kata (*terms*) dalam dokumen. Pemecahan matriks *A* menjadi tiga matriks *TSD* menggunakan *Singular Value Decomposition* via GTP. Dalam penelitian ini, dimensi matriks singular *S* adalah 600 sehingga vektor dokumen yang semestinya memiliki elemen sebanyak 48,512 dapat diperkecil menjadi 600 elemen. Pengurangan vektor elemen secara signifikan ini diyakini dapat mempercepat proses clustering, namun bila matriks *A* dibangun kembali dari perkalian matriks *TSD*, nilai elemen dari matriks *A* tersebut akan mendekati nilai aslinya.

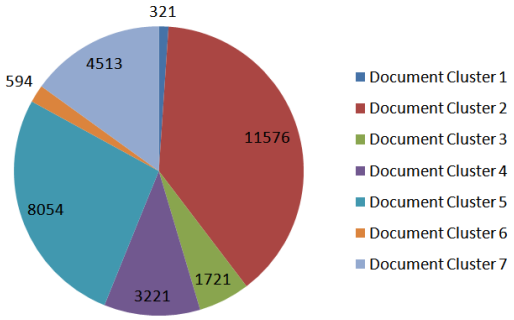
Dalam penelitian ini, pengelompokkan dokumen dilakukan dengam $k = 3$, $k = 5$, dan $k = 7$. Untuk setiap *k*, 10 kali pengujian dilakukan dan jumlah dokumen pada setiap cluster dicatat. Terlihat bahwa dalam setiap pengujian, jumlah dokumen dalam cluster berubah-ubah. Hal ini disebabkan karena metode *k*-means sangat sensitif terhadap *outlier* dan sangat dipengaruhi oleh nilai centroid awal. Gambar 7a, 7b, dan 7c memperihatkan rata-rata jumlah dokumen dalam setiap cluster.



Gambar 7a. Jumlah dokumen per cluster untuk k = 3



Gambar 7b. Jumlah dokumen per cluster untuk k = 5



Gambar 7c. Jumlah dokumen per cluster untuk k = 7

5. SIMPULAN

Pengelompokkan dokumen dapat dilakukan dengan membangun matriks *terms-documents A* yang nilai elemennya merupakan frekuensi dari kata dalam dokumen. Pemecahan matriks *A* menjadi tiga matriks *TSD* menggunakan *Singular Value Decomposition* dapat memperkecil ukuran dimensi vektor dokumen dan mempercepat proses clustering.

Hasil cluster yang diperoleh sangat tergantung pada nilai centroid awal dan sangat dipengaruhi

oleh vektor *outlier* yang mungkin ada karena proses clustering menggunakan algoritma *k*-means. Hal ini terlihat dari perubahan jumlah dokumen dalam cluster pada setiap pengujian.

6. DAFTAR PUSTAKA

[1]. Bau III, David, Lloyd N. Trefethen, 1997. *Numerical Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics.

[2]. Berry, W. Michael, 2006. *General Text Parser for Linux*, Department of Computer Science: University of Tennessee.

[3]. Burton, K., Burton, A. Java, and I. Soboroff. The ICWSM 2009 Spinn3r Dataset. *In Proceedings of the Third Annual International Conference on Weblogs and Social Media (ICWSM 2009)*, San Jose, CA. May 2009.

[4]. Geiß, Johanna, 2008. *Latent Semantic Indexing and Information Retrieval – Aquest with Bosse*. Saarbrücken: VDM Verlag Dr. Müller Aktiengesll schaft & Co. KG.

[5]. Graepel, T., 1998. *Statistical Physics of Clustering Algortihms*. Technical Report 171822, FB Physik, Institut fur Theoretische Physic.

[6]. Grira, N., CrucianuM, Boujemaa N, 2005. *Unsupervised and Semi-Supervised Clustering: a Brief Survey*. In: 7th ACM SIGMM international workshop on multimedia information retrieval, pp 9–16.

[7]. Han, J., and Kamber, M., 2006. *Data Mining: Concepts and Techniques*, 2nd edition. San Francisco, CA: Morgan Kaufmann Publishers.

[8]. Jain, AK., Murty MN., Flynn PJ., 1999. *Data Clustering: a Review*. ACM Comput Surv 264–323. CSUR. doi: 10.1145/ 331499.331504.

[9]. Liu, B., 2007. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Berlin Heidelberg: Springer-Verlag.

[10]. Ratna, Anak Agung Putri, Bagio Budiardjo, Djoko Hartanto, 2007. *Simple: Sistim Penilai Esei Otomatis untuk Menilai Ujian dalam Bahasa Indonesia*. Departemen Elektro, Fakultas Teknik, Universitas Indonesia. Depok, Indonesia April 2007: 5-11: Makara.

[11]. Świniarski, Roman, Cios, Krzysztof J., Pedrycz, Witold, 1998. *Data Mining Methods for Knowledge Discovery*. Kluwer Academic.