Kumpulan jurnaL Ilmu Komputer (KLIK)
Volume 06, No.01 Februari 2019
ISSN: 2406-7857

PENGUKURAN TINGKAT KEMIRIPAN DOKUMEN BERBASIS *CLUSTER* PADA CORPUS BESAR

Ibnu Santoso[1], Lya Hulliyyatus Suadaa[2]
Politeknik Statistika STIS
Jl Otto Iskandardinata no 64 C Jakarta Timur
[1]ibnu@stis.ac.id, [2]lya@stis.ac.id

Abstract

Document similarity can be measured and used to discover other similar documents in a document collection (corpus). In a small corpus, measuring document similarity is not a problem. In a bigger corpus, comparing similarity rate between documents can be time consuming. A clustering method can be used to minimize number of document collection that has to be compared to a document to save time. This research is aimed to discover the effect of clustering technique in measuring document similarity and evaluate the performance. Corpus used was undergraduate thesis of students from year 2007-2016 as many as 2.049 documents. These documents were represented as bag of words model and clustered using k-means clustering method. Measurement of similarity used is cosine similarity. From the simulation, clustering process for 3 clusters needs longer preparation time (17,32%) but resulting in faster query processing (77,88%) with accuracy of 0,98. Clustering process for 5 clusters needs longer preparation time (31,10%) but resulting in faster query processing (83,79%) with accuracy of 0,86. Clustering process for 7 clusters needs longer preparation time (45,10%) but resulting in faster query processing (85,30%) with accuracy of 0,98.

Keywords*: text processing, document similarity, k-means clustering, cosine similarity*

Abstrak

Tingkat kemiripan dokumen atau document similarity dapat diukur dan digunakan untuk menemukan dokumen-dokumen yang mirip dalam suatu koleksi dokumen (corpus). Dalam corpus yang kecil, hal ini tidak menjadi masalah. Namun dalam corpus yang besar, pembandingan tingkat kemiripan satu dokumen dengan seluruh dokumen dapat memakan waktu yang lama. Teknik clustering dapat digunakan untuk untuk meminimalisir jumlah koleksi dokumen yang harus dibandingkan dengan dokumen uji sehingga waktu yang dibutuhkan lebih sedikit. Penelitian ini dilakukan untuk melihat bagaimana teknik clustering memberikan pengaruh dalam pengukuran tingkat kemiripan dokumen dengan teknik pengolahan data text processing dan mengevaluasi kinerjanya. Corpus yang digunakan adalah skripsi mahasiswa dari tahun 2007-2016 sebanyak 2.049 dokumen. Dokumen skripsi direpresentasikan dalam model bag-of-words dan teknik clustering yang digunakan adalah k-means clustering. Ukuran kemiripan yang digunakan adalah cosine similarity. Dari simulasi yang dilakukan, proses clustering untuk 3 cluster membutuhkan waktu persiapan 17,32% lebih lama, namun dalam proses pengecekan/query dokumen uji menjadi 77,88% lebih cepat dengan akurasi yang 0,98. Proses clustering untuk 5 cluster membutuhkan waktu persiapan 31,10% lebih lama dan proses pengecekan dokumen uji menjadi 83,79% lebih cepat dengan akurasi 0,86. Proses clustering untuk 7 cluster membutuhkan waktu persiapan 45,10% lebih lama dan proses pengecekan dokumen uji menjadi 85,30% lebih cepat dengan akurasi 0.98

Kata kunci*: text* ***processing****, document similarity, k-means clustering, cosine similarity*

1. PENDAHULUAN

Tingkat kemiripan dokumen (*document similarity*) dapat diukur dan digunakan untuk menemukan dokumen-dokumen lain yang mirip dengan dokumen uji. Pengukuran tingkat kemiripan dokumen selain dapat dimanfaatkan untuk pengelompokan dokumen juga dapat digunakan misalnya sebagai langkah awal untuk mendeteksi kemungkinan adanya plagiarisme.

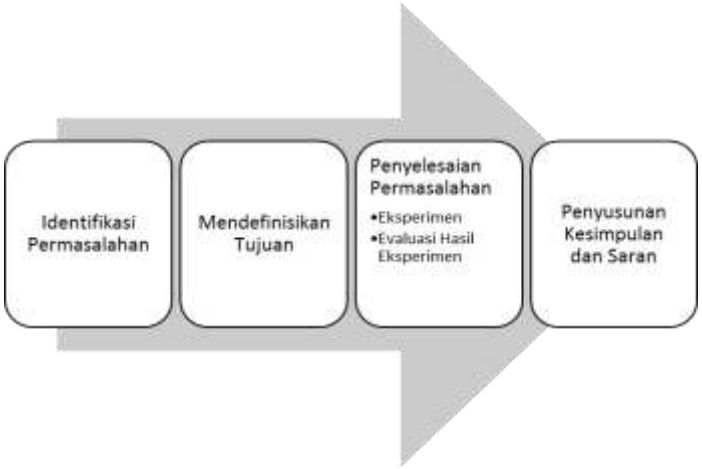
Untuk menemukan tingkat kemiripan suatu dokumen uji dengan koleksi dokumen (*corpus*) tentu harus dilakukan dengan cara membandingkan tingkat kemiripan satu dokumen uji tersebut dengan seluruh koleksi dokumen yang ada. Hal ini tidak menjadi masalah *corpus* yang ada tidak besar karena pembandingan satu dokumen uji dengan seluruh dokumen dalam corpus tidak akan memakan waktu lama. Namun untuk jumlah corpus yang besar, pembandingan tingkat kemiripan satu dokumen dengan corpus dapat menjadi tantangan tersendiri dari sisi waktu, apalagi jika dokumen uji juga banyak jumlahnya.

Dalam hal jumlah koleksi dokumen yang besar, teknik *clustering* dapat digunakan untuk untuk meminimalisir jumlah koleksi dokumen yang harus dibandingkan dengan dokumen yang akan diperiksa. Dalam penerapannya, teknik *clustering* ini akan membagi koleksi semua dokumen ke sejumlah *cluster* tertentu. Dokumen yang akan diperiksa ditentukan dahulu termasuk di *cluster* yang mana, dan hanya akan dibandingkan dengan sejumlah dokumen dalam *cluster* tersebut. Dokumen tidak perlu dibandingkan dengan dokumen di luar *cluster* sehingga tidak terjadi pembandingan yang tidak perlu dan dapat menyingkat waktu pemeriksaan.

Secara umum, penelitian ini dilakukan untuk melihat bagaimana teknik *clustering* memberikan pengaruh dalam pengukuran tingkat kemiripan dokumen dengan teknik pengolahan data pemrosesan teks. Harapannya, waktu yang diperlukan menjadi lebih singkat karena satu dokumen tidak perlu dibandingkan dengan seluruh dokumen yang ada, melainkan hanya dibandingkan dengan dokumen yang ada dalam *cluster*-nya saja, tanpa mengurangi terlalu banyak dari sisi akurasinya.

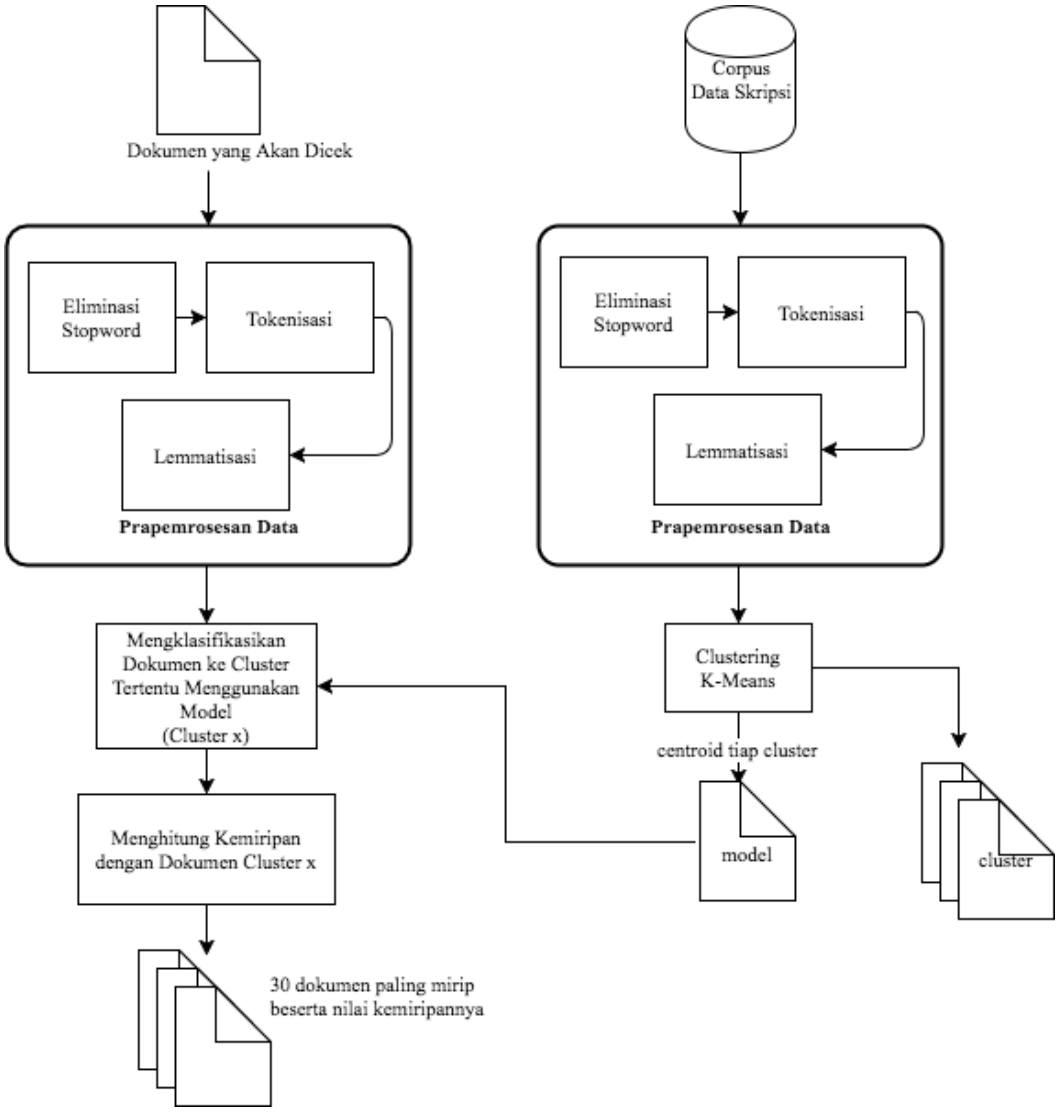
2. METODOLOGI PENELITIAN

Metodologi yang digunakan dalam penelitian ini yaitu metode eksperimental. Metode eksperimental adalah salah satu desain penelitian dari *scientific method*. Menurut [1], eksperimen adalah tes atau uji untuk menentukan karakteristik item yang diteliti dengan menggunakan suatu pengukuran. Tahapan penelitian yang akan dilakukan dirangkum dalam Gambar 1



Gambar 1. Tahapan Penelitian

a. Identifikasi Permasalahan
   Pada tahap ini, dilakukan identifikasi permasalahan dengan melakukan studi literatur mengenai metode *clustering* pada data teks, dan penelitian terkait. Dari hasil identifikasi tersebut, disimpulkan bahwa semakin banyak dokumen yang digunakan sebagai pembanding dalam sistem pendeteksian plagiarisme maka semakin lama waktu yang dibutuhkan untuk pemrosesannya.
b. Mendefinisikan Tujuan
   Tujuan didefinisikan berdasarkan permasalahan yang ada yaitu menerapkan *clustering* dalam pengelompokan dokumen untuk meningkatkan efisiensi query.
c. Penyelesaian Permasalahan
   Pada tahap ini, akan dilakukan analisis algoritma *clustering* yang cocok digunakan untuk mengurangi banyaknya dokumen yang akan diperbandingkan. Metode yang telah dikembangkan akan diujicoba dengan suatu eksperimen menggunakan dataset skripsi. Efisiensi metode *clustering* akan diukur berdasarkan waktu pemrosesan. Kualitas kemiripan dilihat berdasarkan anggota cluster yang terbentuk. Kerangka penelitian yang dilakukan digambarkan pada Gambar 2
d. Penyusunan Kesimpulan dan Saran
   Kesimpulan dan saran disusun berdasarkan hasil eksperimen.



Gambar 2. Kerangka Penelitian

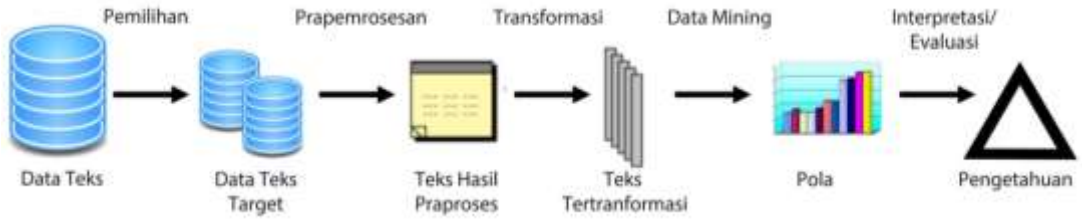
Dalam mengukur kemiripan dokumen, sebenarnya pengetahuan-pengetahuan dalam bentuk data terstruktur seperti sumber data penelitian yang dilakukan dapat lebih mudah diolah menggunakan bantuan *Database Management System* (DBMS). Namun, pengetahuan mengenai skripsi dalam bentuk data yang tidak tersktruktur seperti abstrak dan konten dari skripsi yang merupakan pengetahuan utama dalam pengukuran tingkat kemiripan dokumen memerlukan teknik pengolahan data yang khusus, yaitu pemrosesan teks. Secara sederhana, suatu teks direpresentasikan sebagai sekumpulan kata tanpa dipertimbangkan urutannya (*bag of words*) yang kemudian dapat diolah berdasarkan kemunculan katanya. Tabel 1 menunjukkan proporsi skripsi mahasiswa Jurusan Statistika tahun 2011 sampai dengan 2015 berdasarkan sumber data penelitiannya.

Tabel 1. Proporsi Skripsi Mahasiswa Jurusan Statistika Tahun 2011-2015 berdasarkan Sumber Data

Tahun	Sumber Data	
	Primer	Sekunder
2011	82	155
2012	75	204
2013	68	154
2014	51	293
2015	38	336
Jumlah	314	1.142

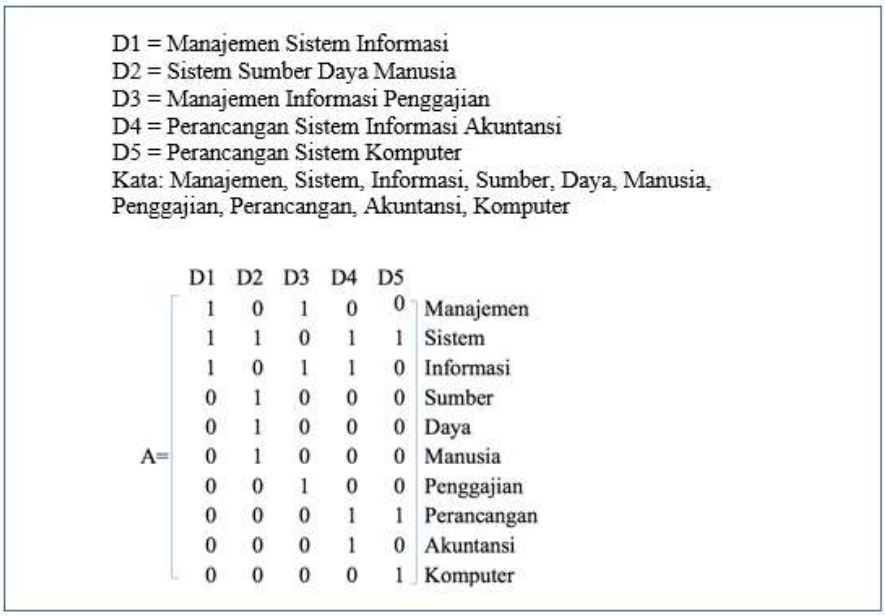
2.1. Pemrosesan Teks

Data teks terdiri dari data tidak terstruktur berupa sekumpulan kata-kata yang saling terkait yang dapat memiliki arti berbeda. Oleh karena itu, metode data mining dalam data teks lebih kompleks terutama dalam tahap prapemrosesan data. Proses data mining untuk data teks diadaptasi dari langkah-langkah pada proses *Knowledge Discovery* dalam basisdata yang disampaikan oleh [2], seperti yang ditunjukkan pada Gambar 3.



Gambar 3. *Knowledge Discovery* pada Data Teks

Pertama, dilakukan pemilihan data teks yang akan diolah menjadi data teks target. Kemudian, prapemrosesan dilakukan untuk membersihkan data teks yang akan diolah. Teks hasil prapemrosesan ditransformasi menjadi *bag of words*. Dalam model *bag of words*, teks direpresentasikan sebagai vektor kata berdasarkan jumlah kemunculan katanya tanpa memperhatikan urutan kata [3]. Gambar 4 menunjukkan contoh representasi dokumen dalam model *bag of words*. Teks yang telah ditransformasi diolah dengan algoritma data mining tertentu sesuai kebutuhan dan kemudian hasilnya diinterpretasikan sebagai pengetahuan.



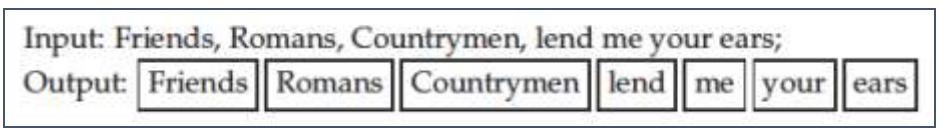
Gambar 4. Contoh Representasi Dokumen dalam Model *Bag of Words*

2.2 Prapemrosesan Teks

Prapemrosesan teks merupakan salah satu task penting yang perlu dilakukan dalam text mining untuk membersihkan data teks yang akan diolah. Hal tersebut dikarenakan karakteristik data teks yang biasanya berupa data tidak terstruktur. Terdapat beberapa proses prapemrosesan teks yang dapat dilakukan, diantaranya eliminasi kata-kata tidak berarti (*stopwords*), tokenisasi dan lematisasi.

2.2.1 Tokenisasi

Tokenisasi adalah proses memenggal suatu kalimat menjadi beberapa bagian yang disebut[4]. Token adalah sekumpulan karakter dalam beberapa dokumen tertentu yang dikelompokkan bersama-sama sebagai unit semantik yang selanjutnya akan diolah. Token yang dihasilkan dapat berupa kata ataupun frasa. Token biasanya disebut juga sebagai term. Gambar 5 menunjukkan contoh sederhana dari proses tokenisasi[3].



Gambar 5. Contoh Tokenisasi

2.2.2 Eliminasi *Stopwords*

Stopwords adalah term yang sering muncul dalam suatu dokumen namun tidak berarti sehingga dapat dieliminasi. Contoh stopwords diantaranya adalah preposisi. Tala [5] melakukan penelitian mengenai pengaruh eliminasi stopwords pada sistem temu balik informasi dan membuktikan bahwa eliminasi stopwords

dapat meningkatkan precision dan recall. Dalam penelitian tersebut, [5] menyajikan kata-kata yang termasuk daftar *stopwords* Bahasa Indonesia.

2.2.3 Lematisasi

Lematisasi adalah pengubahan/konversi bentuk jamak/variannya ke bentuk dasar, yaitu bentuk yang digunakan saat pencarian kamus. misalnya memakan dan dimakan akan diubah ke bentuk makan. Lematisasi menyiratkan perlakukan reduksi "tepat" ke bentuk kata dasar dari kamus.

2.3 Teknik *Clustering*

Secara umum, teknik *clustering* adalah salah satu metode analisa data yang tujuannya adalah mengelompokkan objek-objek dengan karakteristik yang mirip ke dalam satu cluster dan objek dengan karakteristik yang berbeda ke *cluster* yang lain. Prinsipnya adalah memaksimalkan kesamaan antar objek dalam satu *cluster* dan meminimalkan kesamaan antar *cluster*.

Terdapat beberapa pendekatan yang digunakan untuk melakukan *clustering*. Dua pendekatan utama yang banyak digunakan adalah pendekatan partisi (*partition based clustering*) dan pendekatan hirarkis (*hierarchical clustering*). *Partition based clustering* melakukan pengelompokan objek dengan cara memilah-milah objek-objek ke dalam *cluster-cluster* yang ada. Contoh algoritma yang menggunakan *partition based clustering* ini adalah *k-means clustering* seperti yang digunakan dalam penelitian ini. Sedangkan *hierarchical clustering* melakukan pengelompokan objek dengan membuat suatu hirarki berupa dendogram. Objek yang mirip akan ditempatkan pada hirarki yang berdekatan sedangkan objek yang tidak mirip akan ditempatkan pada hirarki yang berjauhan.

2.3.1 *K-Means Clustering*

K-means clustering merupakan metode untuk melakukan pengelompokan objek ke sejumlah K *cluster*. *K-means clustering* adalah metode pengelompokan data yang sederhana dan banyak digunakan dalam aplikasi data mining. Aplikasi *K-means clustering* misalnya dapat digunakan untuk melakukan segmentasi pasar sesuai edengan karakteristik pelanggan dan juga memberikan rekomendasi untuk mengelompokkan objek-objek yang saling terkait.

Input dari *k-means clustering* adalah data/objek dan jumlah cluster(k) yang diinginkan. Setiap *cluster* direpresentasikan oleh sebuah titik pusat (*centroid*). Setiap data akan dikelompokkan pada *cluster* dengan titik pusat yang terdekat dari data tersebut.

Secara umum, cara kerja dari algoritma *k-means clustering* adalah sebagai berikut:

a. Tentukan jumlah *cluster* (k)
b. Pilih k buah titik centroid secara acak
c. Kelompokkan objek-objek ke dalam sejumlah k *cluster. Cluster* yang terpilih adalah yang jarak *centroid*-nya dengan objek yang akan dikelompokkan paling kecil.
d. Setelah terbentuk k *cluster*, perbarui nilai titik *centroid*

e. Ulangi langkah 3 dan 4 sampai nilai titik centroid tidak berubah lagi.

Untuk menghitung jarak antara objek dengan titik *centroid* dapat menggunakan penghitungan jarak Minkowski dengan formula sebagai berikut:

$$d(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \cdots + |x_{ip} - x_{jp}|^g)^{1/g}$$

Dimana:
g = 1, untuk menghitung jarak Manhattan
g = 2, untuk menghitung jarak Euclidean
g = ∞, untuk menghitung jarak Chebychev
x_i , x_j adalah dua buah data yang akan dihitung jaraknya
p = atribut ke-p dari sebuah objek

Untuk memperbarui nilai titik *centroid* pada langkah 4 dapat dihitung dengan formula berikut ini:

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$$

Dimana:
μ_k = titik *centroid* dari *cluster* ke-K
N_k = banyaknya data pada *cluster* ke-K
x_q = data ke-q pada *cluster* ke-K

Terdapat juga ukuran jarak yang lain untuk menghitung kemiripan objek yaitu yaitu *cosine similarity* sebagaimana yang digunakan dalam penelitian ini.

2.3.2 *Cosine Similarity*

Tingkat kesamaan (*similarity*) dari dua buah objek dapat diukur dengan menggunakan metode *cosine similarity*. Dalam penelitian ini, kemiripan antar dokumen diukur berdasarkan sudut kosinus yang dibentuk dua vektor dokumen. Perhitungan metode *cosine similarity* berdasarkan *vector space similarity measure*. Formula untuk menghitung *cosine similarity* adalah sebagai berikut:

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{d}_j||\vec{d}_k|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

Dimana:
d_j = vektor dokumen pertama
d_k = vektor dokumen kedua

Skor maksimal untuk *cosine similarity* adalah 1 yang artinya dua buah dokumen tersebut identik satu sama lain, dan skor minimal adalah 0 yang artinya dua buah dokumen saling berbeda satu sama lain.

3. HASIL DAN PEMBAHASAN

3.1. Dataset

Dataset yang digunakan pada penelitian ini terdiri dari dokumen skripsi mahasiswa tahun 2007-2016 sebanyak 2.049. Pada awalnya dokumen belum terbagi per jurusan dengan format skripsi dalam satu folder dengan format [nim]-skripsiLengkap.pdf. Oleh karena itu, dilakukan pemilahan manual sesuai jurusan terlebih dahulu. Tabel 2 menunjukkan statistik dataset yang digunakan.

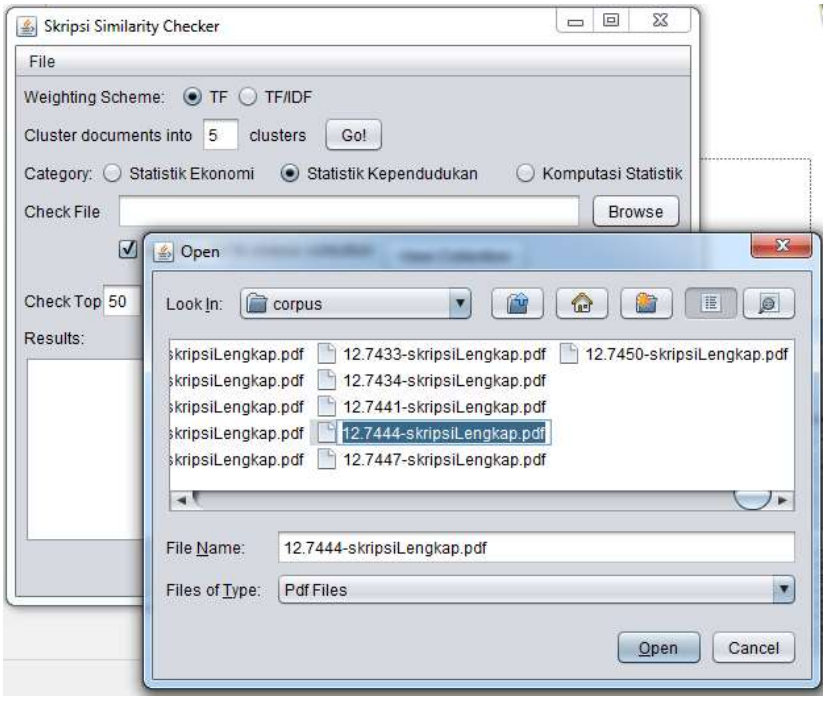
Tabel 2. Statistik Dataset yang Digunakan

Jurusan	Jumlah Dokumen
Statistik Ekonomi (SE)	822
Statistik Kependudukan (SK)	671
Komputasi Statistik (KS)	539
Password Protected	2
File tidak sesuai format	6
Error Saat Dibaca Aplikasi	9

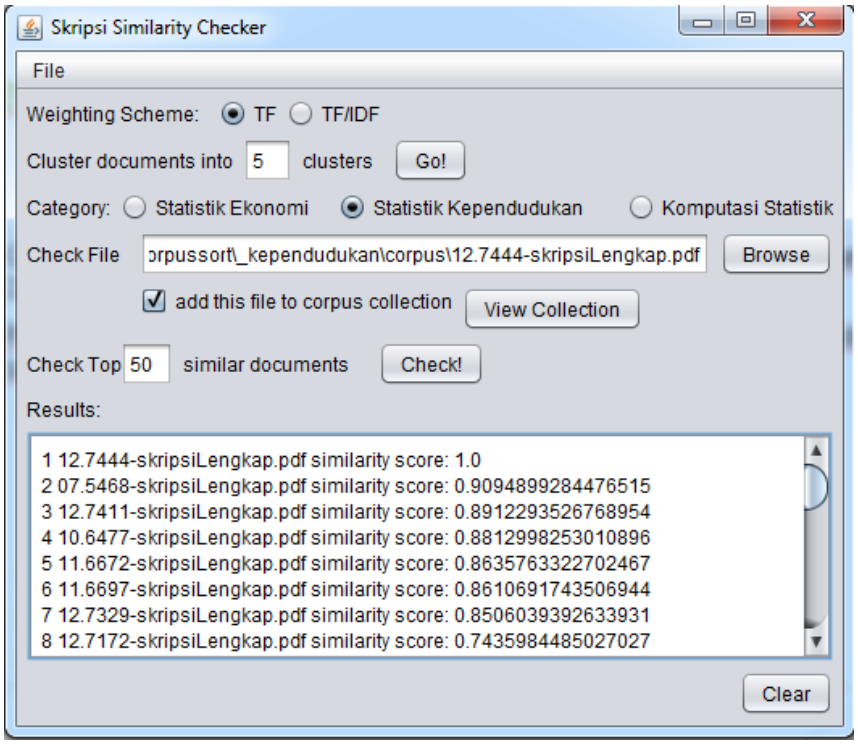
3.2 *Prototype* Aplikasi

Prototype aplikasi dikembangkan menggunakan bahasa pemrograman Java. Input dokumen dalam bentuk pdf diklasifikasikan terlebih dahulu berdasarkan jurusan, yaitu SE, SK, dan KS. Model representasi teks yang digunakan yaitu *bag of words*. Teknik prapemrosesan teks yang diterapkan diantaranya eliminasi *stopwords*, tokenisasi dan lematisasi. Gambar 6 menunjukkan tampilan *protoype* aplikasi saat pemilihan input dokumen yang akan diperiksa tingkat kemiripannya.

Teknik *clustering* yang diterapkan yaitu *K-Means* dengan menggunakan ukuran *cosine similarity*. Corpus yang ada akan di-*cluster* terlebih dahulu, dan nilai centroid untuk setiap *cluster* disimpan agar model *cluster* yang dihasilkan dapat digunakan kembali pada penggunaan aplikasi yang selanjutnya. Dokumen yang diinput hanya akan diperiksa tingkat kemiripannya dengan dokumen yang ada dalam satu *cluster* dengan dokumen input. Output yang dihasilkan yaitu nilai kemiripan dokumen input dengan dokumen dalam *cluster* tersebut, dengan rentang nilai 0 hingga 1, diinterpretasikan sebagai tingkat kemiripan rendah hingga tinggi. Gambar 7 menunjukkan tampilan prototype aplikasi pada saat proses pengecekan kemiripan telah selesai. Gambar 8 menunjukkan bahwa dokumen-dokumen yang memiliki skor similarity yang tinggi ternyata membahas tentang topik-topik yang mirip, misalnya tentang topik "*unmet needs*".



Gambar 6. *Protoype* Aplikasi : Pemilihan Input Dokumen



Gambar 7. *Protoype* Aplikasi : Output Pengecekan Kemiripan Dokumen Input

Gambar 8. Contoh dokumen-dokumen yang memiliki similarity score yang tinggi

4.3 Hasil Eksperimen

Eksperimen dilakukan dengan membuat 3 *cluster*, 5 *cluster*, dan 7 *cluster* dokumen dan dibandingkan dengan kondisi jika tidak menggunakan *cluster*. Kondisi tidak menggunakan *cluster* artinya dokumen dibandingkan dengan seluruh *corpus* yang ada. Jika terdapat *cluster*, dokumen hanya dibandingkan dengan dokumen dalam *cluster*-nya saja. Tabel 3 dan tabel 4 menampilkan perbandingan jumlah *cluster*, waktu persiapan, waktu *query*, jumlah iterasi dan akurasi yang diperoleh untuk setiap percobaan. Akurasi untuk top 50 dokumen artinya berapa % dokumen yang sama jika dibandingkan dengan top 50 dokumen tanpa *cluster*. Akurasi untuk skor similarity > 0,8 artinya semua dokumen yang memiliki skor similarity 0,8 ke atas yang diambil untuk dibandingkan dengan dokumen tanpa *cluster*.

Tabel 3. Perbandingan Penggunaan *Cluster* dan tanpa *cluster*

Jumlah Cluster	Waktu Persiapan (ms)	Waktu Query (ms)	Jumlah iterasi (SE,SK,KS)	Akurasi untuk top 50 dokumen	Akurasi untuk skor similarity > 0,8
0	3730135	660	-	-	-
3	4376286	146	9,13,7	0.98	1.0

Jumlah Cluster	Waktu Persiapan (ms)	Waktu Query (ms)	Jumlah iterasi (SE,SK,KS)	Akurasi untuk top 50 dokumen	Akurasi untuk skor similarity > 0,8
5	4890229	107	19,11,7	0.86	1.0
7	5412573	97	10,19,7	0.98	1.0

Tabel 4. Perbandingan Persentase Penggunaan *Cluster* dan tanpa C*luster*

Jumlah *Cluster*	Waktu persiapan (% lebih lama) dibandingkan tanpa *cluster*	Waktu Query(% lebih singkat) dibandingkan tanpa *cluster*	Rata-rata jumlah iterasi (SE,SK,KS)	Akurasi untuk top 50 dokumen	Akurasi untuk skor similarity > 0,8
3	17,32	77,88	9,66	0.98	1.0
5	31,10	83,79	12,33	0.86	1.0
7	45,10	85,30	12,00	0.98	1.0

Dari tabel terlihat bahwa secara umum, waktu persiapan memang lebih lama dengan semakin banyaknya cluster yang dibentuk, namun waktu query akan menjadi lebih singkat juga dengan tidak mengubah dokumen-dokumen yang mirip dengan skor similarity di atas 0,8.

4. SIMPULAN

Berdasarkan pembahasan sebelumnya maka dapat ditarik kesimpulan sebagai berikut:

a. Teknik *k-means clustering* dengan penghitungan jarak *cosine similarity* dapat digunakan untuk mengelompokkan *corpus* dokumen
b. Kinerja *k-means clustering* yang diterapkan dalam corpus cukup baik untuk mendapatkan dokumen-dokumen yang memiliki tingkat kemiripan yang tinggi di atas 0,8.
c. Semakin banyak jumlah *cluster*, semakin lama waktu yang dibutuhkan untuk persiapan, namun waktu *query* menjadi semakin singkat.

DAFTAR PUSTAKA

[1] R. Maxion, "**Making Experiments Dependable**", Dependable and Historic Computing, Halaman 344-357, 2011.
[2] U. Fayad, G. Piatetsky-Shapiro, dan P. Smyth, "**From Data Mining to Knowledge Discovery in Databases**", American Association for Artificial Intelligence, 1996.

[3] C. D. Manning, P. Raghavan and H. Schütze, "**Introduction to Information Retrieval**", Cambridge University Press. 2008.
[4] W. B. Frakes dan R. Baeza, "**Information Retrieval, Data Structures and Algorithms**", Prentice Hall, 1992.
[5] F. Z. Tala, "**A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia**", M.Sc. Thesis, Institute for Logic, Language and Computation, Universiteti van Amsterdam The Netherlands, 2003.