Implementasi *Text Mining* Pengelompokkan Dokumen Skripsi Menggunakan Metode *K-Means Clustering*

Implementation Of Text Mining For Grouping Thesis Documents Using K-Means Clustering

Dezty Adhe Chajannah Rachman[1], Rito Goejantoro[2], dan Fidia Deny Tisna Amijaya[3]
[1,2]Laboratorium Statistika Komputasi, FMIPA, Universitas Mulawarman
[3]Laboratorium Matematika Komputasi, FMIPA, Universitas Mulawarman
E-mail: deztyadhechajannahr@gmail.com

Abstract

Text mining is the text analysis that automatically discover quality information from a series of texts that is summarized in a document. K-Means Clustering method is often used because of its ability to make a group of large amounts of data with relatively fast and efficient computing time. The purpose of this study is to determine the optimal number of the groups formed from the thesis documents and determine the results of the groups formed. This study is using Nazief and Adriani algorithms for the stemming step, Euclidean Similarity to calculate document distances, and Silhouette Coefficient to test the cluster validity. The sample in this study is 119 thesis documents of Statistics Study Program, Mathematics Department, Faculty of Mathematics and Natural Sciences, graduates of 2016-2018. Based on the results of the analysis, the optimal number of groups formed is two clusters with a silhouette coefficient of 0.12. The results of the grouping formed are two clusters with the total of the first cluster is 85 documents and the second cluster is 34 documents. The first cluster is dominated by studies with data mining especially classification, time series analysis, regression analysis, survival analysis, spatial analysis and operational research, and the second cluster is dominated by studies with multivariate analysis, quality control, and insurance mathematics.

Keywords : *documents, thesis, text mining, k-means clustering, silhouette coefficient.*

Pendahuluan

Menurut Han, Kamber, dan Pei (dalam Prilianti dan Wijaya, 2014), *text mining* adalah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. *Text mining* merupakan variasi dari *data mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar.

Menurut Rijbergen (1979), penerapan *clustering* dokumen dapat meningkatkan efektifitas temu kembali informasi. Dengan mengacu pada suatu hipotesis (*cluster-hypothesis*) bahwa dokumen yang relevan akan cenderung berada pada *cluster* yang sama jika sebuah koleksi dokumen telah dilakukan *clustering*. *Clustering* termasuk dalam teknik *unsupervised learning* dimana tidak memerlukan fase *training* (Suwrmayanti, 2014).

Menurut Alfiana, Santoso dan Ali Ridho B (2012) metode *K-Means* merupakan metode *clustering* yang cukup sederhana dan umum dalam penggunaannya. *K-Means* seringkali digunakan dalam permasalahan *clustering* dikarenakan mempunyai kemampuan mengelompokkan data dalam jumlah yang cukup besar dan dengan waktu komputasi yang relatif cepat serta efisien.

Penelitian sebelumnya oleh Indraloka dan Santoso tahun 2017, berhasil melakukan *clustering* data *tweet* Shopee Indonesia dengan metode *K-Means* yang menghasilkan 28 *cluster tweet*. Penelitian yang menggunakan dokumen skripsi oleh Prilianti dan Wijaya tahun 2014 (dalam Yudiarta, Sudarma, dan Ariastina, 2018) berhasil melakukan *clustering* menggunakan metode *K-Means* terhadap dokumen-dokumen skripsi mahasiswa yang ada di sebuah Universitas Ma Chung dengan nilai *purity* sebesar 76%, artinya sekitar 76% dokumen yang telah diolah telah berhasil dikelompokkan dengan benar oleh sistem. Pengelompokkan dokumen skripsi dapat dilakukan menggunakan sistem sehingga penelitian bisa lebih variatif setiap tahunnya. Dikarenakan penelitian ini menggunakan dokumen teks sebagai data penelitian, maka *text mining* adalah metode yang tepat untuk tahap *preprocessing*. Begitu juga karena metode *K-Means* telah dikenal sebagai metode *clustering* yang sangat efisien, maka *K-Means* menjadi metode yang digunakan dalam melakukan *clustering*.

Data mining

Data mining adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data dengan melakukan penggalian pola-pola dari data dengan tujuan untuk memanipulasi data menjadi informasi yang lebih berharga yang diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basis

data. Seperti halnya *data mining*, *text mining* adalah proses penemuan akan informasi atau *trend* baru yang sebelumnya tidak terungkap dengan memproses dan menganalisis data dalam jumlah besar. *Text mining* merupakan salah satu cabang ilmu *data mining* yang menganalisis data berupa dokumen teks. Menurut Han, Kamber, dan Pei (dalam Prilianti dan Wijaya), *text mining* adalah salah satu langkah otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. Seperti halnya dalam *data mining*, aplikasi *text mining* pada studi kasus, harus dilakukan sesuai prosedur analisis. Langkah awal sebelum suatu data teks dianalisis menggunakan metode-metode dalam *text mining* adalah melakukan *text preprocessing*.

1. *Text Preprocessing*

Text preprocessing menjadi tahap awal dalam *text mining*. Tujuan dari *text preprocessing* yakni menghasilkan sebuah set *term index* yang bisa mewakili dokumen (Sanjaya, 2015).

a. *Case Folding*

Case Folding merupakan proses untuk mengubah semua karakter pada teks menjadi huruf kecil. (Weiss, 2004).

b. *Tokenizing*

Tokenizing adalah proses pemotongan *string input* berdasarkan tiap kata penyusunnya. Pada proses ini juga dilakukan penghilangan angka, tanda baca dan karakter lain selain huruf alphabet (Asian, 2007).

c. *Filtering*

Filtering adalah tahap pemilihan kata-kata penting dari hasil *token*, yaitu kata-kata yang bisa digunakan untuk mewakili isi dari sebuah dokumen. Tahapan ini adalah dengan melakukan penghilangan *stopword* dan juga mengubah kata-kata ke dalam bentuk dasar terhadap kata yang berimbuhan. *Stopword* merupakan kosakata yang bukan merupakan kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat (Dragut, 2009).

d. *Stemming*

Stemming adalah proses pengubahan bentuk kata menjadi kata dasar atau tahap *root* kata dari setiap kata hasil *filtering*. *Stemming* yang digunakan yaitu *stemming* Nazief dan Adriani karena algoritma ini memiliki akurasi lebih besar dibandingkan dengan algoritma Porter (Agusta, 2009).

2. *Term Weighting*

Menurut Asian (2007), *term weighting* adalah suatu pembobotan kata dalam suatu dokumen. Perhitungan bobot tiap *term* dicari pada setiap dokumen bertujuan untuk dapat mengetahui ketersediaan dan kemiripan suatu *term* di dalam dokumen.

a. *Term Frequency*

Menurut Wurdianarto, Novianti dan Rosyidah (2014), *term frequency* adalah algoritma pembobotan heuristik yang menentukan bobot dokumen berdasarkan kemunculan *term*. Penelitian ini menggunakan algoritma *Raw* TF. *Raw* TF merupakan penentuan bobot suatu dokumen terhadap istilah dengan menghitung frekuensi kemunculan suatu istilah tersebut pada dokumen. Untuk mendapatkan nilai TF dapat digunakan persamaan:

$$tf_{t,x} = f_{t,x} \qquad (1)$$

di mana:
x = dokumen ke-*x*
t = *term* atau kata ke-*t*
$tf_{t,x}$ = banyaknya kemunculan *term* ke-*t* pada dokumen ke-*x*
$f_{t,x}$ = frekuensi kemunculan *term t* dalam seluruh dokumen

b. *Inverse Document Frequency*

Inverse Document Frequency (IDF) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Sebelum menentukan IDF, diperlukan untuk mencari *Document Frequency* (DF) yaitu banyaknya dokumen yang mengandung *term* atau token kata ke-t.Untuk mendapatkan nilai IDF dapat digunakan persamaan:

$$idf_t = log_{10}\frac{N}{df_t} \qquad (2)$$

di mana:
idf_t = *inverse* jumlah dokumen yang mengandung *term* ke-*t*
N = banyaknya seluruh dokumen
df_t = banyaknya dokumen yang mengandung *term* ke-*t*

c. TF-IDF

TF-IDF atau biasa disebut *Weight Term Document*. *TF-IDF* dari suatu *term* atau kata merupakan hasil perkalian antara tf *weight* dengan idf. *TF-IDF* digunakan untuk membangun matriks dokumen adalah mengatur angka-angka sesuai dengan pentingnya kata daripada menentukan frekuensi kata-kata dalam dokumen. Untuk mendapatkan *TF-IDF* dapat digunakan persamaan:

$$W_{t,x} = tf_{t,x} \times idf_t = tf_{t,x} \times log_{10}\frac{N}{df_t} \qquad (3)$$

di mana:
$W_{t,x}$ = bobot dari *term* ke-*t* terhadap dokumen ke-*x*
$tf_{t,x}$ = banyaknya kemunculan *term* ke-*t* pada dokumen ke-x
idf_t = *inverse* banyaknya dokumen yang mengandung *term* ke-*t*
N = banyaknya seluruh dokumen

df_t = banyaknya dokumen yang mengandung *term* ke-*t*

Nilai df menjelaskan banyaknya dokumen yang mengandung *term* dan membalikkan nilai skala**.** Berapapun besarnya nilai $tf_{t,x}$, apabila $N = df_t$ maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari $\log_{10}\frac{N}{N} = \log_{10} 1 = 0$, untuk perhitungan IDF. Untuk itu dapat ditambahkan nilai 1 pada sisi IDF, sehingga perhitungan bobotnya menjadi:

$$W_{t,x} = tf_{t,x} \times idf_t + 1 = tf_{t,x} \times log_{10}\frac{N}{df_t} + 1 \quad (4)$$

3. Analisis *Cluster*

Analisis *cluster* adalah suatu alat untuk mengelompokkan sejumlah objek berdasarkan p variabel yang secara relatif mempunyai kesamaan karakteristik diantara objek-objek tersebut, sehingga keragaman dalam suatu kelompok tersebut lebih kecil dibandingkan dengan keragaman antar kelompok. Tujuan analisis *cluster* adalah untuk mengidentifikasi sekelompok objek yang mempunyai kemiripan karakteristik tertentu yang dapat dipisahkan dengan kelompok lainnya (Supranto, 2010).

Menurut Santoso (2014), *cluster* yang baik adalah *cluster* yang mempunyai kesamaan (homogenitas) yang tinggi antar anggota dalam satu *cluster* (*within-cluster*) dan perbedaan (heterogenitas) yang tinggi antar *cluster* yang satu dengan *cluster* yang lainnya (*between-cluster*). Terdapat dua metode analisis *cluster* yaitu:

1. Pengelompokkan hirarki (*hierarchical clustering*) adalah metode analisis kelompok yang berusaha untuk membangun sebuah hirarki kelompok. Strategi untuk pengelompokkan hirarki pada umumnya dibagi menjadi dua jenis yaitu aglomeratif dan divisif.
2. Pengelompokkan non hirarki (*non hierarchical clustering*). Berbeda dengan metode hirarki, metode non hirarki dimulai dengan menentukan terlebih dahulu jumlah *cluster* yang diinginkan. Terdapat tiga metode dalam analisis *cluster* non-hirarki yaitu *K-Means*, *K-Harmonic Means*, dan *K-Modes*.

3. *K-Means Clustering*

K-Means Clustering pertama kali dipopulerkan oleh Hartigan pada tahun 1975. *K-Means Clustering* sangat cocok untuk data dengan ukuran yang besar karena memiliki kecepatan yang lebih tinggi. *K-Means Clustering* merupakan salah satu metode pengelompokkan data nonhierarki (sekatan) yang berusaha mempartisi data yang ada ke dalam bentuk dua atau lebih kelompok. Adapun tujuan pengelompokkan data ini adalah meminimalkan fungsi objektif yang diatur dalam proses pengelompokkan, yang pada umumnya berusaha meminimalkan variasi di dalam suatu kelompok dan memaksimalkan variasi antarkelompok.

Menurut Han dan Kamber (2006), algoritma *K-Means* bekerja dengan cara membagi data ke dalam k buah *cluster* yang telah ditentukan. Perhitungan jarak yang digunakan dalam penelitian ini adalah *euclidean similarity*. Tahap-tahap algoritma dasar *K-Means* seperti berikut:

1. Menentukan banyaknya k sebagai *cluster* atau kelompok yang akan dibentuk.
2. Menentukan pusat *cluster* atau *centroid* secara acak sebanyak k.
3. Menentukan jarak setiap objek atau dokumen terhadap dokumen lain. Untuk menghitung jarak tiap dokumen dengan dokumen lain menggunakan *Euclidian Distance*, dengan persamaan sebagai berikut:

$$d(p,q) = \sqrt{\sum_{t=1}^{nt}\left(W_{t,p} - W_{t,q}\right)^2} \quad (5)$$

di mana:

$d(p,q)$ = jarak antara dokumen ke-p dengan dokumen ke-q ($p \neq q$)

$W_{t,p}$ = bobot dari *term* ke-t pada dokumen ke-p ($p = 1,2,\ldots,N$. N = jumlah dokumen)

$W_{t,q}$ = bobot dari *term* ke-t pada dokumen ke-q ($q = 1,2,\ldots,N$. N = jumlah dokumen)

t = *term* ke-t ($t = 1,2,\ldots,nt$. nt = jumlah term)

4. Mengelompokkan setiap dokumen berdasarkan kedekatannya dengan *centroid* (jarak terkecil).
5. Menentukan pusat *cluster* baru. Memperbaharui nilai *centroid* dari rata-rata *cluster* yang bersangkutan. Untuk menentukan *centroid* baru dapat menggunakan persamaan berikut:

$$c_{k,t} = \frac{\sum_{x=1}^{n} W_{t,x}}{n} \quad (6)$$

di mana:

$c_{k,t}$ = nilai *centroid* ke-k pada *term* ke-t

$W_{t,x}$ = bobot suatu dokumen ke-x pada *term* ke-t yang menjadi anggota *cluster*

n = banyaknya anggota *cluster*

6. Mengulangi langkah 3 hingga 5 sampai anggota yang ada pada tiap *cluster* tidak berubah.
7. Jika langkah 6 sudah terpenuhi, maka *centroid* pada perulangan terakhir akan digunakan sebagai parameter untuk pengelompokkan dokumen. Kemudian menghitung *Silhouette Coefficient* dengan k dan anggota-anggota *cluster* yang didapat.
8. Mengulangi langkah 1 hingga 6 dengan k yang berbeda untuk menghitung *Silhouette Coefficient*.

Pada langkah 4 pada algoritma *K-Means Clustering*, pengalokasian kembali data ke dalam masing-masing kelompok dalam metode *K-Means Clustering* didasarkan pada perbandingan jarak

antara data dengan *centroid* setiap kelompok yang ada. Menurut MacQueen (1967) dalam Prasetyo (2012) pengalokasian ini dapat dirumuskan sebagai berikut:

$$a_{x,k} = \begin{cases} 1, D = \min\{d(p,q)\} \\ 0, lainnya \end{cases} \quad (7)$$

di mana:

$a_{x,k}$ = nilai keanggotaan dokumen ke-*x* terhadap *centroid* ke-*k*

D = jarak terpendek dari dokumen ke semua *cluster*

4. *Silhouette Coefficient*

Silhouette coefficient merupakan salah satu metode evaluasi yang digunakan untuk menguji kualitas dan kekuatan dari sebuah *cluster*. Tahap perhitungan *silhouette coefficient* adalah:

1. Menghitung rata-rata jarak tiap dokumen ke-*i* dengan semua dokumen yang berada dalam satu *cluster*. Nilai ini disebut $a(i)$.
2. Kemudian menghitung rata-rata jarak tiap dokumen ke-*i* dengan semua dokumen di *cluster* lain. Mengambil nilai terkecil dari semua jarak rata-rata tersebut. Nilai ini disebut $b(i)$.
3. Kemudian menghitung nilai *silhouette coefficient* dengan menggunakan persamaan:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (8)$$

di mana:

$a(i)$ = rata-rata dokumen ke-*i* dengan semua dokumen pada satu *cluster* yang sama

$b(i)$ = rata-rata dokumen ke-*i* dengan semua dokumen pada *cluster* yang berbeda

$S(i)$ = nilai *silhouette coefficient*

Nilai *silhouette coefficient* berkisar antara -1 dan 1. Hasil *cluster* dikatakan baik jika nilai *silhouette coefficient* adalah 1, berarti dokumen ke-*i* sudah berada dalam *cluster* yang tepat. Jika nilai *silhouette coefficient* adalah 0, maka dokumen ke-*i* berada di antara dua *cluster*. Jika nilai *silhouette coefficient* adalah -1, artinya struktur *cluster* yang dihasilkan tidak baik, sehingga dokumen ke-*i* lebih tepat dimasukkan ke dalam *cluster* yang lain.

5. Skripsi

Skripsi adalah karya ilmiah dari hasil penelitian mahasiswa program sarjana sebagai salah satu syarat mendapatkan gelar sarjana. Program sarjana adalah pendidikan akademik bagi lulusan menengah atau sederajat sehingga mampu mendapatkan ilmu pengetahuan dan teknologi melalui penalaran ilmiah (Sugiyono, 2013).

Hasil Penelitian dan Pembahasan

1. Data Penelitian

Data yang digunakan dalam penelitian ini adalah judul dokumen skripsi mahasiswa Program Studi Statistika Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Mulawarman tahun kelulusan 2016-2018. Banyaknya skripsi pada tahun kelulusan 2016, 2017 dan 2018 berturut-turut adalah 24, 44 dan 51 dokumen skripsi.

2. *Text Preprocessing*

Data awal yang telah di-*input* akan memasuki tahap *case folding*.

a. *Case Folding*

Pada tahap *case folding*, setiap karakter pada dokumen akan diubah menjadi huruf kecil. Di setiap tahap *preprocessing* juga akan dilakukan penghilangan spasi yang berlebih. Pada proses ini, terjadi perubahan semua huruf pada dokumen menjadi huruf kecil, seperti huruf P pada "Pengendalian", "Perbandingan" dan "Peramalan" menjadi "pengendalian", "perbandingan" dan "peramalan".

b. *Tokenizing*

Pada tahap *tokenizing*, dilakukan proses penghilangan angka, tanda baca dan karakter lain selain huruf *alphabet* dikarenakan karakter-karakter tersebut tidak memiliki pengaruh terhadap pemrosesan teks.

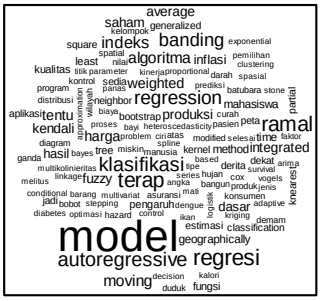
c. *Filtering*

Pada tahap *filtering*, dilakukan proses penghilangan *stopword*. *Stopword* merupakan kosakata yang bukan merupakan kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat, seperti kata "menggunakan", "analisis", "dan", "untuk", "kasus" dan lain sebagainya.

d. *Stemming*

Pada tahap *stemming*, dilakukan proses pengubahan kata menjadi kata dasar atau tahap *root* kata dari setiap kata hasil *filtering* dengan cara menghilangkan imbuhan yang jika tidak dihilangkan akan menjadi beban di *database*. Penelitian ini menggunakan *stemming* algoritma Nazief dan Adriani. Setelah itu, kata-kata yang tersisa akan dipisahkan atau di-*token* menjadi *term-term* sebagai variabel yang akan digunakan pada tahapan-tahapan selanjutnya. Kata-kata yang telah dipisah dapat dibuat menjadi sebuah awan kata atau *wordcloud*. Kata yang paling banyak muncul akan tampil lebih besar dibanding kata-kata lainnya.

Berdasarkan Gambar 1, dapat dilihat bahwa kata yang muncul paling besar adalah kata "model", maka dapat diduga bahwa kata "model" adalah kata yang memiliki frekuensi terbanyak. Setelah itu diikuti oleh kata "klasifikasi", "ramal", "terap", "regresi", "banding", "regression", "autoregressive", dan lain-lain.



Gambar 1 *Wordcloud* Kata-Kata Hasil Token

Hasil inspeksi dari setiap tahap *text preprocessing* adalah sebagai berikut:

Tabel 1 Hasil Inspeksi Tahap *Text Processing*

Tahap *Text Preprocessing*	Banyak Dokumen	Banyak Karakter	Banyak *Term*
Data Awal	119	17.465	-
Case Folding	119	17.460	-
Tokenizing	119	16.789	-
Filtering	119	12.790	-
Stemming (Akhir)	119	-	539

Tahap *text preprocessing* menghasilkan kumpulan kata sebanyak 539 kata yang menjadi variabel dalam penelitian ini.

3. *Term Weighting*

Selanjutnya adalah proses pemberian bobot dimana bobot tersebut mengindikasikan pentingnya sebuah *term* terhadap sebuah dokumen. Tahapan yang dilakukan adalah:

a. *Term Frequency*

Penelitian ini menggunakan algoritma Raw TF, yaitu penentuan bobot suatu dokumen terhadap *term* dengan menghitung frekuensi kemunculan suatu *term* tersebut pada suatu dokumen. Pada penelitian ini, terdapat 539 kata yang akan dihitung frekuensi kemunculannya terhadap 119 dokumen. Dapat ditampilkan dalam bentuk tabel sebagai berikut.

Tabel 2 TF pada Tiap Dokumen

No.	*Term*	TF				
		D1	D2	D3	⋯	D119
1.	Adaptive	0	0	0		0
2.	Age	0	0	0		0
3.	Aggregating	0	0	0		0
4.	Air	0	0	0	⋯	0
5.	Ajar	0	0	0		0
⋮	⋮	⋮	⋮	⋮		⋮
539.	With	0	0	0		0

b. *Inverse Document Frequency*

IDF merupakan perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen. Sebelum menghitung IDF, perhitungan DF (*Document Frequency*) yaitu banyaknya dokumen yang mengandung suatu *term*. Selanjutnya, menghitung IDF menggunakan persamaan (2).

Tabel 3 IDF pada Tiap *Term*

No.	*Term*	TF				DF	IDF
		D1	D2	⋯	D119		
1.	Adaptive	0	0		0	3	1,598
2.	Age	0	0		0	1	2,076
3.	Aggregating	0	0		0	1	2,076
4.	Air	0	0	⋯	0	2	1,775
5.	Ajar	0	0		0	1	2,076
⋮	⋮	⋮	⋮		⋮	⋮	⋮
539.	With	0	0		0	1	2,076

c. TF-IDF

TF-IDF merupakan hasil perkalian antara TF dengan IDF.

Tabel 4 TF-IDF Tiap *Term* pada Tiap Dokumen

No.	*Term*	TF-IDF				
		D1	D2	D3	⋯	D119
1.	Adaptive	0	0	0		0
2.	Age	0	0	0		0
3.	Aggregating	0	0	0		0
4.	Air	0	0	0	⋯	0
5.	Ajar	0	0	0		0
⋮	⋮	⋮	⋮	⋮		⋮
539.	With	0	0	0		0

4. Analisis *Cluster* Menggunakan Algoritma *K-Means Clustering*

Langkah selanjutnya adalah mengelompokkan dokumen-dokumen menggunakan *K-Means Clustering* dengan *k* sebagai *cluster* atau kelompok yang digunakan adalah 2, 3, 4, 5, 6.

Tabel 5 Hasil Pengelompokkan *K-Means Clustering* dengan *k* Bervariasi

Banyaknya *k* (*cluster*)	*Cluster* ke-	Banyaknya Anggota
2	1	85
	2	34
3	1	49
	2	27
	3	43
4	1	18
	2	43
	3	23
	4	35
5	1	17
	2	31
	3	10
	4	29
	5	32
6	1	19
	2	10
	3	31
	4	17
	5	15
	6	27

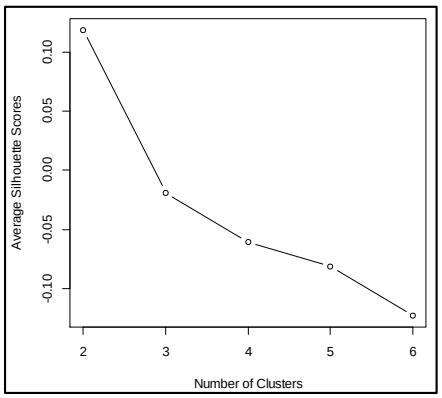
5. Validitas *Cluster* Menggunakan *Silhouette Coefficient*

Pada pengujian ini, nilai *silhouette coefficient* digunakan untuk menentukan nilai *k* yang terbaik dalam mengelompokkan dokumen-dokumen skripsi.

Tabel 6 Pengujian Validitas *Cluster* dengan *Silhouette Coefficient*

Nilai *k*	*Silhouette Coefficient*	Hasil *Structure*
2	0,12	*No Structure*
3	−0,02	*No Structure*
4	−0,06	*No Structure*
5	−0,08	*No Structure*
6	-0,12	*No Structure*

Dari Tabel 6, dapat dilihat bahwa nilai *silhouette coefficient* optimal didapatkan ketika *k* bernilai 2. Berdasarkan Kaufman dan Rousseeuw, hasil *structure* dari kelima nilai *k* yang digunakan menghasilkan *no structure* atau tidak ada struktur. Hal ini dapat dikarenakan penelitian ini menggunakan *dataset* dokumen skripsi sebanyak 119, sehingga *term-term* yang didapatkan tidak sepenuhnya mewakili dan menjadi ciri dari dokumen-dokumen yang mengandung *term-term* tersebut. Kemudian ketika nilai *k* yang dimasukkan semakin banyak, maka dokumen yang memiliki similaritas tinggi dan seharusnya berada pada satu *cluster* akan terpecah dan berada pada *cluster* yang berbeda. Hasil ini juga dapat ditampilkan dengan menggunakan grafik seperti berikut.



Gambar 2 Grafik Hasil Pengujian Nilai *Silhouette Coefficient*

Berdasarkan Gambar 2, dapat dilihat bahwa nilai *silhouette coefficient* dari 5 nilai *k* yang berbeda adalah menurun artinya semakin besar nilai *k* yang dimasukkan, semakin kecil nilai *silhouette coefficient* yang didapatkan. Maka dari itu, pengelompokkan yang optimal adalah yang menghasilkan nilai *silhouette coefficient* terbesar yaitu $k = 2$ dengan nilai $S(2) = 0{,}12$.

6. Interpretasi Hasil Cluster

Dari pengujian validitas *cluster* menggunakan nilai *silhouette coefficient*, dapat disimpulkan bahwa *k* terbaik untuk mengelompokkan dokumen skripsi adalah 2. Anggota *cluster* ke-1 adalah sebanyak 85 dokumen dan anggota *cluster* ke-2 adalah sebanyak 34 dokumen.

Berdasarkan judul dari dokumen skripsi pada masing-masing *cluster*, dokumen-dokumen skripsi yang masuk ke *cluster* 1 adalah dokumen yang lebih dominan pada penelitian *data mining* terutama tentang klasifikasi sebanyak sekitar 16 dokumen. Kemudian banyak juga terdapat penelitian analisis runtun waktu sebanyak sekitar 15 dokumen, analisis regresi sebanyak sekitar 12 dokumen, analisis data uji hidup sebanyak sekitar 5 dokumen, analisis spasial sebanyak 10 dokumen, dan operasi riset sebanyak 5 dokumen.

Sedangkan pada *cluster* 2 dokumen yang lebih dominan pada penelitian analisis multivariat sebanyak sekitar 12 dokumen, diantaranya adalah mengenai *structural equation modeling*, analisis korespondensi dan lain-lain. Selain itu, terdapat juga penelitian pengendalian mutu dan matematika asuransi.

Kesimpulan

Dari hasil analisis, maka dapat diambil kesimpulan sebagai berikut:

1. Banyaknya kelompok optimal yang terbentuk dari dokumen skripsi menggunakan metode *K-Means Clustering* adalah 2 *cluster* dengan nilai *silhouette coefficient* 0,12 yang berarti *no structure*. Hal ini dapat dikarenakan penelitian ini menggunakan *dataset* dokumen skripsi sebanyak 119, sehingga *term-term* yang didapatkan tidak sepenuhnya mewakili dan menjadi ciri dari dokumen-dokumen yang mengandung *term-term* tersebut.
2. Hasil pengelompokkan yang terbentuk dari dokumen skripsi menggunakan metode *K-Means Clustering* adalah sebanyak 2 kelompok dengan anggota *cluster* ke-1 sebanyak 85 dokumen dan anggota *cluster* ke-2 sebanyak 34 dokumen. Dokumen-dokumen skripsi yang masuk ke *cluster* 1 didominasi penelitian dengan metode *data mining* terutama tentang klasifikasi, analisis runtun waktu, analisis regresi, analisis data uji hidup, analisis spasial dan operasi riset. Sedangkan dokumen-dokumen skripsi yang masuk ke *cluster* 2 didominasi penelitian dengan metode analisis multivariat, pengendalian mutu dan matematika asuransi.

Daftar Pustaka

Agusta, Ledy. (2009). *Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks*

Bahasa Indonesia. Bali: Konferensi Nasional Sistem dan Informatika.

Alfiana, T., Santosa, B. dan Ridho, A. B. (2012). "Analisa Perbandingan Metode Hierarchical Clustering, K-Means dan Gabungan Keduanya dalam Cluster Data". *Jurnal Teknik ITS*, 1.

Asian, J. (2007). *Effective Techniques for Indonesian Text Retrieval*. Melbourne: PhD. Royale Melbourne Institute of Technology University.

Dragut, E., Fang, F., Sistla, P., Yu, C., dan Meng, W. (2009). *Stop Word and Related Problems in Web Interface Integration*. Chicago: Computer Science Department, University of Illinois.

Han, J., dan Kamber, M. (2006). *Data Mining Concept and Techniques Second Edition*. Burlington: Morgan Kaufman Publishers.

Rijbergen, C. J. (1979). *Information Retrieval*. Scotland: Information Retrieval Group, University of Glasgow.

Sanjaya, dan Absar. (2015). "Pengelompokkan Dokumen Menggunakan Winnowing Fingerprint dengan Metode K-Nearest Neighbour". *Jurnal CorelIT*, 1 (2).

Santoso, S. (2014). *Statistik Multivariat Edisi Revisi Konsep dan Aplikasi dengan SPSS*. Jakarta: Elex Komputindo.

Sugiyono. (2013). *Cara Mudah Menyusun Skripsi, Tesis dan Disertasi*. Yogyakarta: ALFABETA.

Supranto, J. (2010). *Analisis Multivariat: Arti dan Interpretasi*. Jakarta: Rineka Cipta.

Suwrmayanti, P., Putra, I. K. G. D. dan Kumara, I. N. S. (2014). "Optimasi Pusat Cluster K-Prototype dengan Algoritma Genetika". *Jurnal Teknologi Elektro*, 13 (2).

Weiss, Sholom, Indurkhya, N., Zhang, T., dan Damerau, F. J. (2004). *Text Mining: Predictive Methods for Analyzing Unstructured Information*. New York: Springer.

Wurdianarto, S. R., Novianti, S., dan Rosyidah, U. (2014). "Perbandingan Euclidean Distane Dengan Canberra Distance". *Face Recognition TECHNO.COM*, 13 (1), 31-37.