Tesis - KI142502

Reduksi Dimensi Fitur Menggunakan Algoritma ALOFT untuk Pengelompokan Dokumen

MAMLUATUL HANI'AH

5114201027

DOSEN PEMBIMBING

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom

Diana Purwitasari, S.Kom, M.Sc

PROGRAM MAGISTER

BIDANG KOMPUTASI CERDAS DAN VISUALISASI

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016

Tesis – KI142502

Feature Dimension Reduction Using Aloft Algorithm For Document Clustering

MAMLUATUL HANI'AH
5114201027

SUPERVISOR
Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
Diana Purwitasari, S.Kom, M.Sc

PROGRAM MAGISTER
DEPARTMENT OF INFORMATICS ENGINEERING
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
INTELLIGENT COMPUTING AND VISUALIZATION
SURABAYA
2016

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
MAMLUATUL HANI`AH
Nrp. 5114201027

Dengan judul :
REDUKSI DIMENSI FITUR MENGGUNAKAN ALGORITMA ALOFT UNTUK PENGELOMPOKAN DOKUMEN

Tanggal Ujian : 17-6-2016
Periode Wisuda : 2015 Genap

Disetujui oleh:

Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
NIP. 197512202001122002 (Pembimbing 1)

Diana Purwitasari, S.Kom, M.Sc
NIP. 197804102003122001 (Pembimbing 2)

Dr. Darlis Heru Murti, S.Kom, M.Kom
NIP. 197712172003121001 (Penguji 1)

Dr. Eng. Nanik Suciati, S.Kom, M.Kom
NIP. 197104281994122001 (Penguji 2)

Dini Adni Navastara, S.Kom, M.Sc
NIP. 198510172015042001 (Penguji 3)

Direktur Program Pasca Sarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., PhD.
NIP. 196012021987011001

(halaman ini sengaja dikosongkan)

REDUKSI DIMENSI FITUR MENGGUNAKAN ALGORITMA ALOFT UNTUK PENGELOMPOKAN DOKUMEN

Nama Mahasiswa : Mamluatul Hani'ah
NRP : 5114201027
Pembimbing : Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
Diana Purwitasari, S.Kom., M.Sc

ABSTRAK

Pengelompokan dokumen masih memiliki tantangan dimana semakin besar dokumen maka akan menghasilkan fitur yang semakin banyak. Sehingga berdampak pada tingginya dimensi dan dapat menyebabkan performa yang buruk terhadap algoritma *clustering*. Cara untuk mengatasi masalah ini adalah dengan reduksi dimensi. Metode reduksi dimensi seperti seleksi fitur dengan metode filter telah digunakan untuk pengelompokan dokumen. Akan tetapi metode filter sangat tergantung pada masukan pengguna untuk memilih sejumlah *n* fitur teratas dari keseluruhan dokumen, metode ini sering disebut *variable rangking* (VR). Cara mengatasi masalah ini adalah dengan Algoritma ALOFT (*At Least One FeaTure*) dimana ALOFT dapat menghasilkan sejumlah set fitur secara otomatis tanpa adanya parameter masukan dari pengguna. Algoritma ALOFT pada penelitian sebelumnya digunakan untuk klasifikasi dokumen, metode filter yang digunakan pada algoritma ALOFT adalah metode filter yang membutuhkan adanya label pada kelas sehingga metode filter tersebut tidak dapat digunakan untuk pengelompokan dokumen.

Oleh karena itu, pada penelitian ini diusulkan metode reduksi dimensi fitur dengan menggunakan variasi metode filter pada algoritma ALOFT untuk pengelompokan dokumen. Proses pencarian kata dasar pada penelitian ini dilakukan dengan menggunakan kata turunan yang disediakan oleh Kateglo (kamus, tesaurus, dan glosarium). Fase reduksi dimensi dilakukan dengan menggunakan metode filter seperti *Document Frequency* (DF), *Term Contributtion* (TC), *Term Variance Quality* (TVQ), *Term Variance* (TV), *Mean Absolute Difference* (MAD), *Mean Median* (MM), dan *Arithmetic Mean Geometric Mean* (AMGM). Selanjutnya himpunan fitur akhir dipilih dengan algoritma ALOFT. Tahap terakhir adalah pengelompokan dokumen menggunakan dua metode clustering yang berbeda yaitu *k-means* dan *hierarchical agglomerative clustering* (HAC).

Kualitas dari *cluster* yang dihasilkan dievaluasi dengan menggunakan metode *silhouette coefficient.* Pengujian dilakukan dengan cara membandingkan nilai *silhouette coefficient* dari variasi metode filter pada ALOFT dengan pemilihan fitur secara VR. Berdasarkan pengujian variasi metode filter pada ALOFT untuk pengelompokan dokumen didapatkan bahwa kualitas *cluster* yang dihasilkan oleh metode usulan dengan menggunakan algoritma *k-means* mampu memperbaiki hasil dari metode VR. Kualitas *cluster* yang didapat memiliki kriteria "Baik" untuk filter TC, TV, TVQ, dan MAD dengan rata – rata *silhouette* lebih dari 0,5.

Kata kunci : Reduksi dimensi, metode filter, ALOFT, pengelompokan dokumen

(halaman ini sengaja dikosongkan)

FEATURE DIMENSION REDUCTION USING ALOFT ALGORITHM FOR DOCUMENT CLUSTERING

Name	: Mamluatul Hani'ah
Student Identity Number	: 5114201027
Supervisor 1	: Dr. Eng. Chastine Fatichah, S.Kom, M.Kom
Supervisor 2	: Diana Purwitasari, S.Kom, M.Sc

ABSTRACT

Document clustering still have a challenge when the volume of document increases, the dimensionality of term features increases as well. this contributes to the high dimensionality and may cause deteriorates performance and accuracy of clustering algorithm. The way to overcome this problem is dimension reduction. Dimension reduction methods such as feature selection using filter method has been used for document clustering. But the filter method is highly dependent on user input to select number of n top features from the whole document, this method often called variable ranking (VR). ALOFT (At Least One feature) Algorithm can generate a number of feature set automatically without user input. In the previous research ALOFT algorithm used on classification documents so the filter method require labels on classes. Such filter method can not be used on document clustering.

This research proposed feature dimension reduction method by using variations of several filter methods in ALOFT algorithm for document clustering. Before the dimension reduction process first step that must be done is the preprocessing phase then calculate the weight of term using tfidf. filter method used in this study are such Document Frequency (DF), Term contributtion (TC), Term Variance Quality (TVQ), Term Variance (TV), Mean Absolute Difference (MAD), Mean Median (MM), and Arithmetic Mean geometric Mean (AMGM). Furthermore, the final feature set selected by the algorithm ALOFT. The last phase is document clustering using two different clustering methods, k-means and agglomerative hierarchical clustering (HAC).

Quality of cluster are evaluated using coefficient silhouette. Experiment is done by comparing value of silhouette coefficient from variation of filter method in ALOFT with feature selection in VR. Experiment results showed that the proposed method using k-means algorithm able to improve results of VR methods. This research resulted quality of cluster with criteria of "Good" for filter TC, TV, TVQ, and MAD with average silhouette width (ASW) *more than 0.5*

Keywords*: dimension reduction, filter method, ALOFT, document clustering*

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji syukur kepada Allah SWT atas limpahan rahmat dan karuniaNya sehingga penulis mampu menyelesaikan tesis dengan judul "Reduksi Dimensi Fitur Menggunakan Algoritma ALOFT untuk Pengelompokan Dokumen" ini. Shalawat serta salam disampaikan juga kepada Rasulullah Muhammad SAW yang dengan segala ketulusannya bersedia menyampaikan ajaran-ajaran Islam sehingga bisa sampai kepada penulis. Walaupun penulis belum pernah bertemu secara langsung, penulis mencoba menjadikan beliau sebagai rujukan dalam segala tindakan berdasarkan referensi-referensi yang ada dan dapat diterima.

Penulis juga mengucapkan terima kasih kepada seluruh pihak yang telah mendukung proses penyelesaian tesis ini, khususnya kepada:

1. Kedua orang tua, Bapak, Ibu serta Adek yang selalu memberikan dukungan moral, doa sampai finansial kepada penulis sehingga proses perkuliahan maupun penyelesaian tesis ini dapat terus dapat terus berjalan. Terima kasih Bapak dan Ibu atas segala dukungan terhadap anakmu ini.
2. Ibu Chastine Fatichah dan Ibu Diana Purwitasari yang telah memberikan bimbingan dan berbagai tantangan dalam menyelesaikan tesis ini sehingga tidak hanya sebagai syarat kelulusan studi S2, tetapi penulis juga mampu memahami bidang keilmuan yang dibahas pada tesis ini.
3. Bapak Darlis Heru Murti, Ibu Nanik Suciati, dan Ibu Dini Adni Navastara selaku penguji yang telah bersedia memberikan koreksi dan masukan terhadap tesis ini.
4. Saudara Yogi Kurniawan yang selalu memberikan dukungan dalam penyelesaian tesis ini.
5. Teman-teman S2 FTIF dan teman – teman GMT yang tidak dapat disebutkan satu per satu. Terima kasih atas segala pengalaman dan bantuan yang diberikan terhadap penulis.

Penulis menyadari bahwa tesis ini masih jauh dari kesempurnaan. Oleh karena itu, masukan dan saran yang bersifat membangun selalu dinantikan untuk perbaikan di masa mendatang. Akhirnya, penulis berharap agar tesis ini mampu memberikan kontribusi yang bermanfaat bagi bidang keilmuan di kemudian hari.

Surabaya, Juli 2016

Mamluatul Hani’ah

(halaman ini sengaja dikosongkan)

DAFTAR ISI

(halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

(halaman ini sengaja dikosongkan)

DAFTAR TABEL

(halaman ini sengaja dikosongkan)

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya Internet dan teknologi Web jumlah informasi digital yang dapat diakses oleh pengguna mengalami peningkatan yang pesat. Peningkatan jumlah informasi digital ini berakibat pada kesulitan pengguna untuk menemukan informasi yang relevan. Dengan demikian pengelolaan terhadap informasi digital menjadi sangat diperlukan, salah satu cara untuk mengelola informasi digital tersebut adalah dengan pengelompokan dokumen. Pengelompokan dokumen merupakan salah satu teknik dalam *text mining* yang bertujuan untuk mengelompokkan satu set dokumen ke dalam himpunan bagian atau *cluster* (Feldman & Sanger, 2007).

Pada *text mining* sebelum dilakukan proses pengelompokan dokumen terdapat beberapa proses yang harus dilakukan terlebih dahulu. Proses yang sering disebut dengan *preprocessing* dokumen ini terdiri dari *tokenizing*, *stopword removal*, dan *stemming*. *Stemming* merupakan proses pencarian kata dengan melakukan penghapusan awalan dan akhiran. Proses *stemming* pada dokumen berbahasa Indonesia masih memiliki kendala dimana tidak semua kata dapat terpotong dengan benar. Misalnya kata "penyidikan" setelah di *stemming* kata yang dihasilkan menjadi "sidi" padahal seharusnya kata dasar dari penyidikan adalah "sidik". Selain itu algoritma *stemming* bahasa Indonesia yang ada saat ini belum bisa mengatasi masalah pada kata bersisipan (Arifin, Mahendra, & Ciptaningtyas, 2009). Sehingga pada penelitian ini proses pembentukan kata dasar dilakukan dengan menggunakan produk Kateglo (kamus, tesaurus, dan glosarium) bahasa Indonesia (http://kateglo.com/api.php).

Setelah dilakukan *preprocesing* setiap dokumen direpresentasikan menjadi vektor menggunakan *Vector Space Model* (VSM) (Salton, Wong, & Yang, 1975). Secara tradisional didalam VSM setiap kata yang terdapat didalam dokumen merupakan representasi dari fitur yang berbeda. Semakin besar dokumen maka akan menghasilkan fitur yang semakin banyak, ratusan bahkan ribuan fitur. Jumlah fitur yang banyak dapat membuat tingginya komputasi, selain itu jika terdapat

banyak fitur yang tidak revelan dapat menyebabkan performa yang buruk dari algoritma *clustering*. Salah satu cara untuk mengatasi dimensi fitur yang tinggi adalah dengan reduksi dimensi (Bharti & Singh, A three-stage unsupervised dimension reduction method for text clustering, 2014) (Bharti & Singh, Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering, 2015) (Tabakhi, Moradi, & Akhlaghian, 2014) (Song & Park, 2009). Tujuan utama dari reduksi dimensi adalah memilih subset fitur dari fitur yang memiliki dimensi besar, tanpa mengurangi performa dari *machine learning*.

Salah satu metode reduksi dimensi adalah seleksi fitur, dimana metode ini berfungsi untuk menghapus fitur yang tidak relevan. Metode seleksi fitur secara konvensional dapat dikategorikan menjadi 2 model (Liu, Motoda, & eds, 2007): model *wrapper* dan model filter. Model *wrapper* akan menghasilkan akurasi yang bagus jika dibandingkan dengan model filter. Akan tetapi model *wrapper* memiliki kelemahan dimana model *wrapper* membutuhkan biaya yang tingggi karena harus berulang kali melakukan pengujian dengan *machine learning* untuk mendapatkan fitur yang cocok. Sedangkan model filter menggunakan analisis statistik untuk menentukan relevansi fitur tanpa berulang kali menguji dengan *machine learning*, model filter relatif cepat dan lebih efisien.

Penelitian (Liu, Kang, Yu, & Wang, 2005) memperkenalkan beberapa metode filter untuk pengelompokan dokumen yaitu *Document Frequency* (DF), *Term Contributtion* (TC), *Term variance quality* (TVQ), dan *Term Variance* (TV). (Ferreira & Figueiredo, 2012) mengusulkan seleksi fitur dengan model filter pada metode *unsupervised* dan *supervised* untuk data dengan dimensi yang besar. Penulis mengusulkan 2 tahapan filter, filter pertama mengatasi masalah relevansi dengan perhitungan statistik *Mean Absolute Difference* (MAD), *Mean Median* (MM), *Arithmetic Mean Geometric Mean* (AMGM) dan filter kedua adalah *absolute cosine* (AC) yang berguna untuk mengatasi masalah redudansi fitur dengan cara menghitung kemiripan antar fitur yang relevan dari tahap pertama.

Akan tetapi pemilihan fitur yang umum digunakan pada model filter adalah dengan perangkingan fitur berdasarkan nilai relevansi dari fitur, kemudian memilih sejumlah *n* fitur teratas dari keseluruhan dokumen, cara ini sering disebut dengan *Variabel Rangking* (VR) (Liu, Motoda, & eds, 2007). Karena VR memilih *n* fitur

teratas berdasarkan masukan pengguna maka nilai *n* menjadi sangat penting karena jumlah fitur yang berbeda mungkin akan menghasilkan kelompok dokumen yang berbeda. Selain itu fitur yang dipilih mungkin tidak mencakup keseluruhan dokumen karena fitur yang ada dalam dokumen tersebut tidak masuk kedalam *n* fitur teratas.

Penelitian yang dilakukan oleh (Pinheiro, Cavalcanti, Correa, & Ren, 2012) mengusulkan sebuah metode seleksi fitur yang diberi nama ALOFT (*At Least One FeaTure*) untuk pengklasifikasian dokumen. Ide utama dari metode ini adalah untuk mencari satu set fitur yang pasti mencakup semua dokumen. Setidaknya terdapat satu fitur pada setiap dokumen yang harus menjadi fitur akhir. Kemudian selanjutnya di implementasikan algortima *machine learning*. ALOFT menghasilkan sejumlah set fitur secara otomatis tanpa adanya parameter masukan dari pengguna dengan cara memilih fitur tertinggi dari setiap dokumen. Jika dibandingkan dengan memilih fitur menggunakan VR, ALOFT memiliki performa yang lebih bagus.

ALOFT yang merupakan metode seleksi fitur dengan model filter membutuhkan filter untuk penentuan relevansi dari setiap fitur. Metode filter digunakan pada ALOFT adalah *Bi-Normal Separation (BNS), Class Discriminating Measure (CDM), Chi-Squared (CHI), Information Gain (IG),* dan *Multiclass Odds Ratio (MOR).* Akan tetapi metode filter tersebut membutuhkan adanya label pada kelas sehingga tidak dapat digunakan untuk pengelompokan dokumen.

Oleh karena itu, pada penelitian ini diusulkan metode reduksi dimensi fitur dengan menggunakan beberapa metode filter pada algoritma ALOFT untuk pengelompokan dokumen. Diharapkan metode yang diusulkan dapat meningkatkan performa dan efisiensi dari algoritma *clustering*. Selain itu dengan penggunaan Kateglo untuk pembentukan kata dasar diharapkan akan terbentuk kata dasar yang sesuai, sehingga hasil *cluster* yang diperoleh akan lebih berkualitas.

1.2 Perumusan Masalah

Berdasarkan uraian yang telah dipaparkan pada latar belakang, maka masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut:

1. Bagaimana pengaruh kata dasar dengan menggunakan kata turunan pada Kateglo terhadap hasil *cluster*.
2. Bagamaina reduksi dimensi fitur yang menggunakan variasi metode filter pada ALOFT.
3. Bagaimana kualitas *cluster* yang dihasilkan setelah dilakukan reduksi dimensi.

1.3 Batasan Masalah

Permasalahan yang dibahas pada penelitian ini memiliki beberapa batasan sebagai berikut :

1. Dokumen yang digunakan pada penelitian ini adalah dokumen berita online berbahasa Indonesia.
2. Dokumen diambil dari situs berita online www.kompas.com.
3. Dari komponen berita hanya digunakan judul dan isi berita.
4. Sistem dibangun menggunakan bahasa python.

1.4 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mengelompokkan dokumen berita dengan memanfaatkan teknik reduksi dimensi fitur menggunakan variasi metode filter pada algoritma ALOFT. Metode yang diusulkan diharapkan dapat meningkatkan performa dan efisiensi dari algoritma *clustering*.

Manfaat yang didapat dari penelitian ini adalah untuk mengelola penumpukan informasi digital yang ada di internet dengan cara pengelompokkan dokumen. Hasil dokumen yang sudah terkelompokkan dapat memperbaiki efektifitas dan efisiensi dalam pencarian informasi yang relevan.

1.5 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah reduksi dimensi fitur dengan menggunakan kombinasi metode filter pada algoritma ALOFT untuk mengelomppokan dokumen.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penelitian, secara garis besar yang meliputi beberapa bab, sebagai berikut :

BAB 1 : PENDAHULUAN

Bagian pendahuluan mnguraikan tentang latar belakang masalah yang kemudian dirumuskan kedalam rumusan masalah dalam bentuk uraian terstruktur dan dilengkapi dengan, batasan masalah, tujuan, manfaat, kontribusi serta sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA

Bagian tinjauan pustaka melakukan pengkajian mengenai teori-teori dan referensi yang berkaitan dan menunjang penelitian.

BAB 3: METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan mengumpulkan data permasalahan. Data selanjutnya diolah dengan menggunakan metode yang diajukan untuk mendapatkan pengetahuan serta informasi yang bisa dimanfaatkan oleh pengguna.

BAB 4 : HASIL DAN PEMBAHASAN

Bab ini akan dijabarkan dan dijelaskan hasil yang diperoleh dari hasil penelitian berupa analisa terhadap pengetahuan yang dihasilkan dari pengguna menggunakan metode yang diusulkan

BAB 5 : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diambil berdasarkan analisa setelah pengujian serta saran untuk pengembangan lebih lanjut.

(halaman ini sengaja dikosongkan)

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dipaparkan penelitian sebelumnya dan konsep dasar tentang teori – teori yang dipakai sebagai dasar dalam melakukan penelitian.

2.1 Kajian Pustaka

Penelitian mengenai reduksi dimensi pada pengelompokan dokumen telah dilakukan oleh beberapa peneliti. Penelitian yang dilakukan oleh (Liu, Kang, Yu, & Wang, 2005) melakukan reduksi dimensi dokumen dengan menggunakan seleksi fitur dimana model seleksi fitur yang digunakan adalah model filter. Penelitian tersebut memperkenalkan beberapa model filter seperti *Document Frequency* (DF), *Term Contributtion* (TC), *Term Variance Quality* (TVQ), dan *Term Variance* (TV) yang merupakan usulan dari penelitian tersebut. Selain itu dalam penelitiannya Luying LIU juga menganalisis kelebihan dan kelemahan serta pengaruhnya terhadap algoritma *cluster*ing. Dari hasil evaluasi didapatkan bahwa filter TC dan TV memiliki performa yang lebih bagus dibandingkan dengan TVQ dan DF.

(Ferreira & Figueiredo, 2012) mengusulkan metode filter yang efisien untuk metode *unsupervised* and *supervised* pada data dengan dimensi yang besar. Penulis mengusulkan dua tahapan filter, filter pertama mengatasi masalah relevansi dengan perhitungan statistik *Mean Absolute Difference* (MAD), *Mean Median* (MM), *Arithmetic Mean Geometric Mean* (AMGM). Filter kedua adalah *Absolute Cosine* (AC) yang berguna untuk mengatasi masalah redundansi fitur dengan cara menghitung kemiripan antar fitur yang relevan dari tahap pertama. Hasil evaluasi yang dilakukan pada penelitian ini menyebutkan bahwa MAD, MM, dan AMGM cocok untuk data yang *sparse.*

Penelitian lain yang melakukan reduksi dimensi adalah penelitian (Bharti & Singh, A three-stage unsupervised dimension reduction method for text clustering, 2014). Penelitian tersebut melakukan reduksi dimensi dengan tiga tahapan tahap pertama seleksi fitur menggunakan filter MAD dan MM. Tahap kedua adalah melakukan ekstraksi fitur menggunakan dengan *Principal component analysis*

(PCA) terhadap hasil seleksi fitur. Kemudian tahap ketiga merupakan tahap untuk mengurangi redudansi fitur dengan model filter *Absolute Cosine* (AC).

Pada tahun 2015 (Bharti & Singh, Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering, 2015) mengusulkan metode reduksi dimensi dengan melakukan modifikasi pendekatan union untuk penggabungan fitur dari dua buah metode filter (DF dan TV). Metode ini diusulkan karena jika menggabungkan fitur dari dua buah metode filter yang berbeda dengan union maka akan akan menghasilkan fitur yang terlalu banyak, sedangkan jika digabungkan dengan intersection maka fitur yang dihasilkan akan sangat sedikit. Pada Tabel 2.1 dapat dilihat ringkasan tentang beberapa penelitian dalam reduksi dimensi.

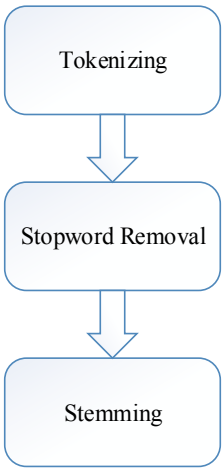
Tabel 2. 1 Beberapa Penelitian Reduksi Dimensi

Penulis	Pengukuran	Metode	Algoritma	Kelemahan	Kelebihan	
(Liu, 2005)	FS (DF/TC/TVQ/TV)	FS (filter)	*k-means clustering*	Tergantung pada inputan pengguna *n* fitur teratas	meningkatkan efisiensi dan akurasi *clustering*	Pengelompokan dokumen
(Ferreira, 2012)	FS (MAD / MM / AMGM) + FS (Absolute cosine (AC))	FS + FS (filter)	*k-means clustering*, SVM	Tergantung pada inputan pengguna *n* fitur teratas	dapat digunakan untuk supervised atau unsepervised	*Clustering* dan klasifikasi di data
(Bharti, 2014)	FS(MAD / MM) + FS (AC) + PCA	FS+FS+ FE	*k-means clustering*	Tergantung pada inputan pengguna *n* fitur teratas dan untuk AC	menghapus fitur- fitur yang tidak relevan, redundansi, dan fitur noise	Pengelompokan dokumen
(Bharti, 2015)	FS (TV +DF) + PCA	FS (filter) + FE	*k-means clustering*	tergantung pada inputan pengguna dan interaksi antar term tidak dipetimbangkan	mengatasi masalah penggabungan filter dengan union dan intersection	Pengelompokan dokumen
(Pinheiro, 2012)	FS (IG / GHI/ MI /BNS)	FS (filter)	SVM,KNN, *Naive Bayes*	Fitur yang dihasilkan sedikit	Tidak bergantung inputan pengguna, perfoma lebih bagus dibanding VR	Klasifikasi dokumen

2.2 *Text Mining*

Text mining merupakan proses penemuan informasi yang berguna dari sumber data melalui proses identifikasi dan eksplorasi pola. Dalam kasus *text mining* sumber data yang digunakan adalah koleksi dokumen dan explorasi pola ditemukan dari data teks yang tidak terstruktur (Feldman & Sanger, 2007). *Text mining* sendiri meliputi klasifikasi dokumen, pengelompokan dokumen, *sentiment analysis*, peringkasan dokumen, dan lain sebagainya (Han, Kamber, & Pei, 2012). Secara tradisional *framework* dari *text mining* berisi 3 tahap berurutan yaitu *Text Preprocessing*, *Text Representation*, dan *Knowledge Discovery* (Aggarwal & Zhai, 2012).

2.2.1 *Text Preprocessing*



Gambar 2. 1 Tahapan *Preprocessing*

Text preprocessing bertujuan untuk membuat dokumen masukan lebih konsisten dan mempermudah representasi teks. *Text processing* secara tradisional dapat dilihat pada Gambar 2.1 dimana tahap ini berisi tiga proses utama yaitu *tokenizing*, *stopword removal*, dan *stemming*. Tahap *tokenizing* adalah tahap pemotongan *string* berdasarkan tiap kata yang menyusunnya. Selanjutnya *stopword removal* mengeliminasi kata – kata yang tidak penting berdasarkan daftar st*opword*. Contoh *stopword* adalah "yang", "dan", "di", "dari" dan seterusnya. Tahap yang terakhir adalah *stemming* dimana tahap ini dilakukan untuk mencari *root* kata. Setiap kata yang memiliki imbuhan seperti imbuhan awalan dan akhiran maka akan diambil kata dasarnya.

2.2.2 *Text Representation*

Cara yang paling umum untuk memodelkan dokumen adalah mengubah setiap kata (term) menjadi vektor numerik. Representasi ini disebut "*Bag Of Words*" (BOW) atau "*Vector Space Model*" (VSM). Dalam VSM setiap kata yang terdapat didalam dokumen merupakan representasi dari fitur yang berbeda tanpa dipertimbangkan hubungan semantik antar kata yang ada di dalam dokumen. Selanjutnya setiap kata akan direpresentasikan dengan bobot. Metode pembobotan kata yang paling populer adalah *Term frequency-inverse document frequency* (*tf idf*) pada persamaan 2.1

$$tfidf(t_k) = tf * \log \frac{N}{df(t_k)} \quad (2.1)$$

Dimana *tfidf(w)* merupakan bobot term *w* didalam dokumen, *tf* merupakan frekuensi kemunculan term *tk* didalam sebuah dokumen, dan *df* adalah jumlah dokumen yang memiliki kata *tk*.

2.2.3 *Knowledge Discovery*

Proses *knowledge discovery* dapat dilakukan ketika teks telah berhasil dirubah ke dalam bentuk vektor numerik. Proses ini dilakukan dengan cara menerapkan algoritma *machine learning* seperti klasifikasi atau *clustering*.

2.3 Pengelompokan Dokumen

Pengelompokan dokumen merupakan cara untuk mengelompokan dokumen secara otomatis dari satu set dokumen ke dalam himpunan bagian atau *cluster*. *Cluster* yang dibentuk menggambarkan isi dari setiap dokumen. Dokumen dalam sebuah *cluster* sebaiknya semirip mungkin dan dokumen dalam satu *cluster* harus berbeda dari dokumen dalam *cluster* lain (Manning, Raghavan, & Schütze, 2008). Pengelompokan dokumen termasuk metode *unsupervised* dimana pengelompokan dokumen dilakukan tanpa adanya informasi lain seperti data pelatihan, pengelompokan dilakukan hanya berdasarkan pada data yang akan diproses.

Menurut (Chen, Tseng, & Liang, An integration of WordNet and fuzzy association rule mining for multi-label document clustering, 2010) dalam

meningkatkan kualitas algoritma *clustering* masih terdapat beberapa tantangan yaitu :

a. Mengatasi Dimensi yang Tinggi

   Semakin meningkatnya volume dari dokumen maka akan menghasikan fitur yang banyak sehingga meningktakan dimensi.

b. Meningkatkan Skalabilitas

   Banyak algoritma *clustering* yang mampu bekerja dengan baik dalam dokumen yang berskala kecil akan tetapi tidak mampu bekerja secara efisien pada dokumen yang besar.

c. Meningkatkan Akurasi

   Algoritma *clustering* yang sudah ada banyak yang membutuhkan peran user sebagai masukan untuk menentukan jumlah *cluster*, akan tetapi penentuan jumlah *cluster* merupakan hal yang sulit karena jika estimasi jumlah *cluster* tersebut salah, maka akan menyebabkan rendahnya akurasi.

d. Memberikan Label *Cluster*

   Pemberian label pada *cluster* dapat meningkatkan proses pencarian, akan tetapi sebagian besar algoritma *clustering* tidak dapat memberikan label pada *cluster*.

e. Memungkinkan *Overlapping Cluster*

   Beberapa algoritma *clustering* fokus pada *hard cluster* dimana setiap dokumen tepat dimiliki sebuah *cluster,* namun dapat memungkinkan bahwa didalam sebuah dokumen dapat berisi beberapa topik hal ini dapat diatasi dengan *soft clustering*. *Soft clustering* memungkinkan sebuah dokumen dapat muncul di beberapa *cluster.*

f. Mengekstrak Sematik Teks

   Metode *bag of words* yang digunakan untuk pengelompokan dokumen seringkali tidak mempertimbangkan interaksi (kemiripan) antar term.

2.4 Kateglo

Kateglo adalah aplikasi dan layanan web, sumber dan isinya terbuka untuk kamus, tesaurus, dan glosarium bahasa Indonesia. Namanya diambil dari akronim

unsur dari layanannya: ka(mus), te(saurus), dan glo(sarium). (Kateglo, 2009) Kateglo menyediakan beberapa hal berikut ini :

1. Memeriksa ejaan dan makna baku suatu lema.
2. Sinonim suatu lema untuk meningkatkan ragam pilihan kata dalam suatu wacana.
3. Kata turunan yang merupakan kata-kata berimbuhan dari suatu lema.
4. Gabungan kata atau yang sering disebut kata majemuk merupakan sebuah istilah khusus. Contoh : ibu kota.
5. Mencari padanan istilah asing dalam bidang tertentu, sekaligus memeriksa. apakah istilah tersebut telah dibakukan di kamus, memiliki ejaan dan makna yang tepat, dan sesuai dengan pedoman pembentukan istilah.

2.5 Algoritma *Clustering*

Berdasarkan pengelompokan data menjadi *cluster*, metode *clustering* dibedakan menjadi dua jenis yaitu *partitioning* dan *hierarchical*. *Partitioning* (*flat clustering*) merupakan metode yang paling mendasar dan sederhana dimana data dikelompokkan menjadi *cluster* tanpa struktur yang eksplisit. Metode *patitioning* yang populer dan sering digunakan adalah *k-means*. Sedangkan *Hierarchical Clustering* menghasilkan output sebuah struktur hierarki dari *cluster*. *Algortima agglomerative hierarchical clustering* merupakan salah satu metode *Hierarchical clustering* yang sering digunakan pada data teks. (Han, Kamber, & Pei, 2012)

2.4.1 *K-means*

K-means diperkenalkan oleh (MacQueen, 1967) merupakan salah satu teknik *partitioning* berbasis *centroid* dimana teknik ini menggunakan *centroid* dari *cluster* untuk mewakili *cluster* tersebut. Pada *k-means* penentuan *centroid* dilakukan dengan cara menghitung rata-rata dari dokumen yang ditetapkan sebagai anggota *cluster*. Langkah pertama adalah untuk memilih jumlah k *centroid* sesuai *cluster* yang akan dibentuk. Selanjutnya dokuemn – dokumen selain *centroid* akan dihitung kemiripannya terhadap *centroid*. Algoritma *k-means* kemudian menentukan anggota *cluster* dengan cara memilih dokuemn – dokumen yang memiliki kemiripan tertinggi antara *centroid* dan *non-centroid* dokumen. Selanjutnya untuk

setiap *cluster* dihitung rata-rata dari anggota *cluster* untuk menentukan *centroid* baru. iterasi berlanjut sampai tidak ada perubahan anggota *cluster*. Prosedur dari algortima *k-means* dapat dilihat pada Gambar 2.2.

Algoritma: *k-means*

Input :

D = { d1, d2, d3 ,d4.... dn }

K = jumlah dari *cluster*

Output :

himpunan k *cluster* C= (c1,c2,...ck)

Metode :

1. Pilih k dokumen secara random pada sekumpulan n dokumen sebagai centroid
2. Hitung kemiripan antara centroid dengan dokuemn non-centroid
3. Dapatkan anggota *cluster* dengan memilih dokumen yang mempunyai kemiripan yang paling besar antara dokumen non centroid dengan centroid
4. Update centroid dengan cara menghitung rata-rata dari anggota *cluster*
5. Ulangi langkah 2 sampai 5 sampai tidak ada perubahan dari anggota *cluster*.

Gambar 2. 2 *Pseudocode* Algoritma *K-means*

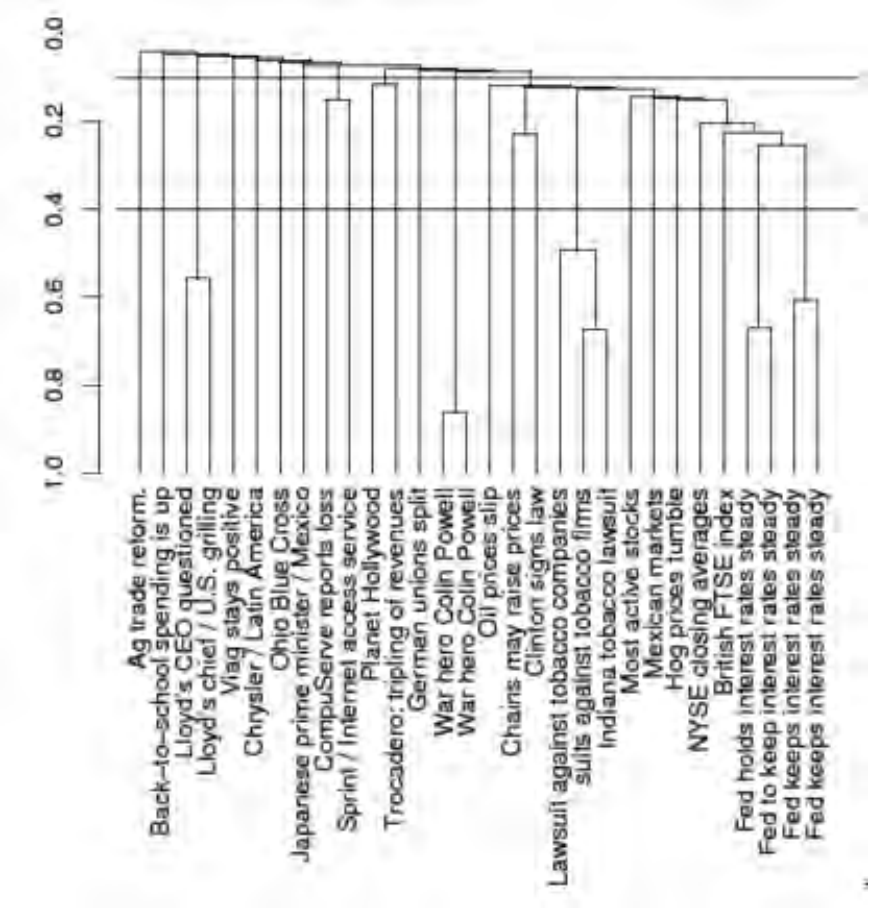
2.4.2 *Hierarchical Agglomerative Clustering*

Hierarchical agglomerative clustering (HAC) merupakan salah satu algoritma *Hierarchical clustering* yang bersifat *bottom-up* dimana proses awal setiap dokumen dianggap sebagai *cluster* tunggal dan kemudian berturut – turut digabungkan menjadi satu *cluster*. (Manning, Raghavan, & Schütze, 2008) HAC biasanya divisualisasikan sebagai dendrogram seperti yang ditunjukkan pada gambar 2.3. Setiap penggabungan *cluster* akan diwakili oleh garis horisontal, misal pada “Lloyd's CEO questioned” and “Lloyd's chief / U.S.” penggabungan untuk dua *cluster* ini pada 0,5. Penentuan jumlah *cluster* tidak dibutuhkan pada HAC akan tetapi HAC membutuhkan pemotongan tingkat kemiripan pada level tertentu. Pada

gambar 2.3 jika dipotong pada level 0,4 maka akan terbentuk 24 *cluster* dan jika dipotong pada level 0,1 maka terbentuk 12 *cluster*s. Prosedur dari algortima HAC dapat dilihat pada Gambar 2.4.

Pada HAC dalam menggabungkan *cluster* terdapat beberapa kriteria pengukuran kemiripan antar cluster yaitu (Manning, Raghavan, & Schütze, 2008):

1. *Single-Linkage : Maximum Similarity*
   kemiripan antara dua *cluster* pada *single-linkage clustering* adalah kemiripan dari anggota yang paling mirip. Semakin besar nilai kemiripan antar dua *cluster* maka semakin mirip *cluster* tersebut.
2. *Complete-Linkage Clustering : Minimum Similarity*
   kemiripan antara dua *cluster* pada *complete - linkage clustering* berdasarakan pada ketidakmiripan antar dokumen. Sehingga dokumen yang memiliki nilai kemiripan paling kecil akan digabungkan.
3. *Centroid Linkage: Average Inter-Similarity*
   pembentukan *cluster* yang didasarkan pada kemiripan antar centroidnya. Kemiripan antar centroid adalah rata-rata kemiripan dokumen yang ada dalam suatu *cluster* dengan kemiripan seluruh dokumen dalam *cluster* yang lain.
4. *Group-Average Agglomerative Clustering* (GAAC)
   Metode ini mengevaluasi kualitas *cluster* berdasarkan semua kemungkinan kemiripan antara dokumen.



Gambar 2. 3 Ilustrasi Dendrogram pada HAC (Manning, Raghavan, & Schütze, 2008)

Algoritma: Hierarchical Agglomerative *Cluster*ing

Input :

D = { d1, d2, d3 ,d4.... dn }

Output :

himpunan *cluster* C= (c1,c2,...ck)

Metode :

1. Pada tahap awal asumsikan setiap dokumen adalah *cluster*
2. Hitung kemiripan antar *cluster*
3. Merge / gabungkan dua *cluster* yang memiliki kemiripan tertinggi
4. Hitung kemiripan antara *cluster* yang baru dengan *cluster* lainnya
5. Ulangi langkah 3 dan 4 sampai hanya satu *cluster* yang terbentuk.

Gambar 2. 4 *Pseudocode* HAC

2.6 Reduksi Dimensi

Reduksi dimensi merupakan salah satu teknik yang populer untuk mengurangi *noise* (fitur yang tidak relevan) dan fitur redundan. Tujuan utama dari reduksi dimensi adalah memilih subset fitur dari fitur yang memiliki dimensi besar tanpa mengurangi performa dari *machine learning*. Teknik reduksi dimensi dapat dibagi menjadi dua yaitu ekstraksi ftur dan seleksi fitur. (Alelyani, Tang, & Liu, 2013)

a. Ekstraksi fitur

Ekstraksi fitur atau yang bisa disebut transformasi fitur seringkali digunakan pada data yang memiliki dimensi besar. Ekstraksi fitur mengubah fitur dengan dimensi besar menjadi fitur dengan dimensi yang lebih rendah melalui proses kombinasi atau transformasi. Contoh ekstraksi fitur adalah *Latent Semantic Indexing* (LSI), *Principal Component Analysis* (PCA), *Semantic Mapping* (SM), *Independent Component Analysis* (ICA). Meskipun metode ekstraksi fitur seringkali bermaanfaat tetapi teknik ini seringkali tidak dapat menghapus fitur yang tidak relevan selain itu karena proses kombinasi fitur sulit diinterpretasikan, hal ini menyebabkan hasil pengelompokan kurang bermanfaat. Karena alasan tersebut ekstraksi fitur / transformasi fitur paling sesuai untuk data yang memiliki banyak fitur yang relevan tetapi fitur tersebut redundan. (Parsons, Haque, & Liu, 2004)

b. Seleksi Fitur

Dalam pengolahan teks jumlah term yang berbeda yang muncul dalam koleksi dokumen bersakla besar dapat menjadi fitur cukup besar maka teknik seleksi fitur digunakan untuk memilih satu set term yang paling relevan untuk digunakan sebagai vektor fitur (Berry, 2004). Secara tradisional seleksi fitur dapat dibagi menjadi dua, yaitu model *wrapper* dan model filter.

1. Model *Wrapper*

Model *wrapper* memilih subset fitur secara acak yang kemudian diterapkan algoritma *machine learning* untuk mengevaluasi kualitas dari fitur. Proses ini akan terus menerus diulang sampai tercapai kualitas yang diinginkan. Pada dimensi yang tinggi proses evaluasi kualtias fitur dengan menggunakan metode *wrapper* menjadi tidak mungkin dilakukan.

2. Model Filter

Model filter menggunakan beberapa kriteria dari data untuk memilih fitur tanpa menggunakan *machine learning*. Kriteria data dihitung menggunakan analisis statistik untuk menentukan relevansi fitur. Karena model filter hanya mengevaluasi fitur dalam kaitannya dengan fitur-fitur lainnya tanpa berulang kali menguji dengan *machine learning*, model filter relatif cepat dan lebih efisien. Pemilihan fitur pada model filter yang paling umum adalah dengan memilih sejumlah *n* fitur teratas dari keseluruhan dokumen.

2.7 Metode Filter

Metode filter digunakan untuk menghitung / menentukan skor relevansi dari sebuah fitur. beberapa metode filter yang dapat digunakan untuk pengelompokan dokumen adalah sebagai berikut :

a. *Document Frequency* (DF)

Dokumen frequency (DF) adalah jumlah dokumen dimana sebuah term muncul. Dalam seleksi fitur metode DF adalah kriteria yang paling sederhana dan mudah untuk dataset yang besar. Asumsi dalam DF adalah bahwa term yang memiliki nilai DF rendah maka dianggap term yang tidak penting atau tidak akan berpengaruh ke dalam proses *clustering*.

b. *Term Contribution* (TC)

Hasil pengelompokan dokumen sangat bergantung pada kemiripan antar dokumen. Jadi kontribusi dari term dapat dilihat sebagai kontribusinya terhadap kemiripan dokumen sehingga kontribusi dari term dapat didefinisikan sebagai kontribusi secara keseluruhan untuk kesamaan dokumen. Jika kemiripan dokumen biasanya dihitung dengan perkalian *dot product* antar dua dokumen maka TC dapat dihitung dengan cara perkalian *dot product* dari kesuluruhan dokumen yang memiliki term tersebut. Perhitungan TC dapat dilihat pada persamaan 2.2

$$TC\ (tk) = \sum_{i,j \cap i \neq j} tfidf(t_k, D_i) * tfidf(t_k, D_j) \quad (2.2)$$

Dimana $tfidf(t_k, D_i)$ merupakan *Tfidf* dari term tk pada dokumen D_i. Penggunaan *Tfidif* bertujuan untuk mengatasi masalah pada DF dimana DF mudah bias karena setiap term dianggap sama penting di dalam dokumen yang berbeda.

Term yang sering muncul memiliki nilai DF yang tinggi akan tetapi memiliki distribusi yang seragam di kelas yang berbeda. Pada *Tfidf* term yang terlalu banyak muncul dalam berbagai dokumen dianggap terlalu umum dan tidak penting untuk *clustering.*

c. *Term Variance Quality* (TVQ)

Term Variance Quality (TVQ) diperkenalkan oleh (Dhillon, Kogan, & Nicholas, 2004) relevansi dari fitur dihitung dengan persamaan (2.3)

$$q(t_k) = \sum_{j=1}^{n} f_{ij}^2 - \frac{1}{n} \left(\sum_{j=1}^{n} f_{ij}\right) \quad (2.3)$$

Dimana n adalah jumlah dari dokumen dimana term t_k muncul minimal satu kali sedangakan f_{ij} adalah frekuensi dari kemunculan term t_k dalam dokumen D_j.

d. *Term Variance* (TV)

Term variance menghitung varian dari semua term yang ada pada dataset, ide dari TV seperti DF dimana setiap term yang memiliki nilai *document frequency* rendah dianggap sebagai term yang tidak penting. Suatu term yang muncul dalam sedikit dokumen atau memiliki distribusi umum di seluruh dokumen akan memiliki nilai yang rendah. TV dihitung dengan persaman 2.4

$$v\,(t_k) = \sum_{j=1}^{n} (f_{kj} - \bar{f}_k)^2 \quad (2.4)$$

Dimana f_{kj} adalah frekuensi term t_k pada dokumen D_j dan $\overline{f_k}$ adalah rata-rata dari frekuensi term t_k dalam koleksi dokumen.

e. *Mean Absolute Difference* (MAD)

Metode ini adalah bentuk sederhana dari *Term Variance*. Metode ini memberikan nilai relevansi dari setiap fitur dengan menghitung perbedaan sampel dari nilai rata-rata. Formula untuk menghitung MAD dapat dilihat pada persamaan 2.5

$$MAD\,(t_k) = \frac{1}{n} \sum_{j=1}^{n} |tfidf_{kj} - \overline{tfidf_k}| \quad (2.5)$$

MAD didefinisikan sebagai menghitung selisih mutlak dari nilai rata-rata. Dimana $tfidf_{ij}$ merupakan *tfidf* dari tem t_k pada dokumen j. Dan $\overline{tfidf_k}$ adalah rata – rata *tfidf* dari tem t_k.

f. *Mean Median* (MM)

Metode ini adalah bentuk sederhana dari *skewness*. Nilai yang dihasilkan memberikan nilai relevansi untuk setiap fitur berdasarkan perbedaan mutlak antara mean dan median dari *tfidf*. MM dihitung dengan persaman 2.6

$$MM(t_k) = |\overline{tfidf_k} - median\ (tfidf_k)| \quad (2.6)$$

Dimana $\overline{tfidf_k}$ adalah rata – rata *tfidf* dari tem t_k dan $tfidf_k$ merupakan *tfidf* dari tem t_k.

g. *Arithmetic Mean Geometric Mean* (AMGM)

AMGM yang diusulkan oleh (Ferreira & Figueiredo, 2012) merupakan filter diusulkan untuk mengatasi masalah pada *Arithmetic Mean* dan *Geometric Mean* dengan menerapkan fungsi eksponensial untuk setiap fitur. Formula untuk menghitung MAD dapat dilihat pada persamaan 2.7

$$AMGM(t_k) = \frac{\frac{1}{n}\sum_{j=1}^{n} \exp(tfidf_{kj})}{(\prod_{j=1}^{n} \exp(tfidf_{kj}))^{\frac{1}{n}}} \quad (2.7)$$

Dimana $tfidf_{ij}$ merupakan *tfidf* dari tem t_k pada dokumen j.

2.8 ALOFT (At Least One FeaTure)

ALOFT merupakan salah satu metode seleksi fitur untuk pengkategorian teks yang diusulkan oleh (Pinheiro, Cavalcanti, Correa, & Ren, 2012). Algoritma yang diusulkan adalah menggunakan model filter dengan memastikan bahwa setiap dokumen akan berkontribusi untuk pemilihan akhir himpunan fitur. Setidaknya terdapat satu fitur yang mewakili dokumen. Algoritma ALOFT dikenalkan untuk mengatasi masalah pada pemilihan fitur secara *Variabel Rangking* (VR) dimana VR memilih *n* fitur teratas dari keseluruhan dokumen. Karena VR memilih *n* fitur teratas berdasarkan masukan pengguna maka nilai *n* menjadi sangat penting karena jumlah fitur yang berbeda mungkin akan menghasilkan kelas yang berbeda. Sedangkan dengan menggunakan ALOFT tidak diperlukan masukan untuk pemilihan fiturnya.

Kriteria fitur pada penelitian yang diusulkan oleh (Pinheiro, Cavalcanti, Correa, & Ren, 2012) dievaluasi dengan metode *Bi-Normal Separation* (BNS),

Class Discriminating Measure (CDM), *Chi-Squared* (CHI), *Information Gain* (IG), dan *Multiclass Odds Ratio* (MOR). Prosedur dari algoritma ALOFT dapat dilihat pada Gambar 2.5

Algoritma: ALOFT

Input :

D = { d1, d2, d3 ,d4.... dn }

Output :

himpunan fitur

Metode :

1. for h =1 sampai V hitung nilai filter untuk setiap term
2. S_h = filter (Wh)
3. End for
4. m=0
5. set VF sebagai vektor kosong
6. for semua $Di \in D$ pilih nilai filter tertinggi untuk setiap dokumen
7. nilai terbaik = 0,0
8. for h=1 sampai V
9. if $W_{h,i} > 0$ & $S_h >$ nilai terbaik maka
10. nilai terbaik = S_h
11. fitur terbaik = h
12. end if
13. end for
14. if fitur terbaik bukan anggota VF maka
15. $m = m+1$
16. VF_m = fitur terbaik
17. End if
18. End for

Gambar 2. 5 *Pseudocode* ALOFT *(Pinheiro, Cavalcanti, Correa, & Ren, 2012)*

Penjelasan :

- Baris 1- 3

Setiap fitur akan dihitung nilai relevansi dari fitur menggunakan metode filter kemudian disimpan dalam S_h. *V* merupakan jumlah fitur dari keseluruhan dokumen.

- Baris 4-18

  Set baru fitur *VF* dihitung. Fitur ke *h* dimasukkan di *VF* jika nilai S_h tertinggi di antara semua fitur. Namun, jika fitur ini sudah terdapat di *VF*, fitur tersebut akan diabaikan dan algoritma berjalan ke dokumen berikutnya. Pada akhir fase ini, *VF* harus menjadi vektor dengan nilai-nilai *m*, dan nilai-nilai ini merupakan indeks dari fitur yang dipilih.

Contoh :

Tabel 2. 2 Contoh dokumen *(Pinheiro, Cavalcanti, Correa, & Ren, 2012)*

D	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	C
1	0	1	0	0	0	0	1	0	0	A
2	0	0	0	1	1	0	0	1	0	A
3	1	0	0	1	0	0	0	1	1	A
4	0	1	0	0	0	0	0	0	0	A
5	0	0	0	0	0	0	1	1	0	A
6	0	0	0	0	0	1	0	0	0	B
7	0	0	0	1	0	1	0	0	1	B
8	1	0	1	1	1	0	0	0	1	B
9	0	0	0	0	0	1	0	0	0	B
10	0	0	0	0	0	1	1	0	0	B
11	1	0	1	1	1	0	1	0	0	B
12	0	0	0	1	1	0	0	0	1	B
13	1	0	0	1	0	1	0	0	0	B
S	11	7	4	15	10	8	2	5	13	-

Terdapat dokumen seperti Tabel 2.2 diatas.

- Nilai *S* merupakan hasil perhitungan nilai relevansi dari fitur menggunakan filter.
- Kolom D merupakan indeks dari dokumen.
- Kolom W_i merupakan fitur.
- Kolom C adalah kategori dari dokumen.

Cara kerja ALOFT

- Pertama adalah mencari nilai *S* dari hasil perhitungan nilai relevansi dari fitur menggunakan filter.
- Tahap kedua adalah mencari fitur terbaik. Untuk setiap dokumen fitur terbaik berdasarkan nilai *S* yang tertinggi.
- Untuk dokumen D_1 memiliki W_2 dengan nilai $S = 7$, W_7 dengan nilai $S= 2$. Maka W_2 akan dipilih karena memiliki nilai *S* tertinggi sehingga vektor $VF = \{2\}$
- Untuk dokumen D_2 memiliki w_4 dengan nilai $S = 15$, W_5 dengan nilai $S=10$ dan W_8 dengan nilai $S=5$. Maka W_4 akan dipilih karena memiliki nilai *S* tertinggi sehingga vektor $VF = \{2,4\}$
- Untuk dokumen *D3*, W_4 terpilih lagi akan tetapi indeks tersebut sudah terdapat di dalam *VF* sehingga diabaikan
- Untuk dokumen D_4 hal yang sama terjadi seperti pada dokumen D_3. Indeks dokumen dari Nilai *S* terbesar (W_2) sudah terdapat di *VF*.
- Untuk dokumen D_5 nilai FEF terbesar dimiliki W_8 sehingga $VF = \{2,4,8\}$
- Untuk dokumen D_6 nilai FEF terbesar dimiliki W_6 sehingga $VF = \{2,4,8,6\}$
- Dari dokumen D_7 sampai D_{13} tidak ada fitur baru yang ditambahkan ke *VF*. Sehingga final $VF = \{2,4,8,6\}$

Keuntungan ALOFT

- Satu dokumen diwakili oleh vektor fitur, setidaknya satu fitur dalam satu dokumen
- ALOFT secara otomatis menemukan fitur dengan jumlah paling sedikit yang telah mencakup semua dokumen di data training
- Jika dibandingkan dengan VR, ALOFT tidak memerlukan parameter masukan dari pengguna.

2.9 Metode Evaluasi

2.9.1 *Silhouette Coefficient*

Silhouette Coefficient yang dikenalkan oleh (Rousseeuw, 1987) merupakan salah satu cara untuk mengevaluasi kualitas *cluster* yang dihasilkan. Selain itu

s*ilhouette coefficient* juga mengindikasikan derajat kepemilikan derajat kempemilikan setiap objek yang berada di dalam *cluster*. Dokumen O_j yang berada pada *cluster* memiliki rentang nilai *Silhouette* antara -1 sampai 1. Semakin dekat nilai s*ilhouette* ke 1 maka semakin tinggi derajat O_j di dalam *cluster*. Pada persamaan 2.10 dan 2.11 merupakan perhitungan nilai *Silhouette (s(i))* untuk setiap dokumen.

$$b(i) = min_{cj \neq a}\, d(i, c_j) \tag{2.10}$$

$$s(i) = \frac{b(i) - a\,(i)}{max\,\{a(i), b(i)\}} \tag{2.11}$$

Dimana *a(i)* adalah jarak kedekatan dokumen *i* terhadap seluruh dokumen yang ada di *cluster* tempat *i* berada disebut juga *cluster* internal. Sedangkan *b* adalah jarak kedekatan dokumen *i* terhadap seluruh dokumen yang ada *cluster* selain *cluster* internal (*cluster* external). Selanjutnya setiap *cluster* yang telah dihitung nilai *s(i)* akan dihitung nilai rata-rata dari *s(i)*. Perhitungan ini lebih dikenal dengan nama *Average Silhouette Width* (ASW). Range nilai ASW dapat dibagi menjadi empat kriteria yaitu:

1. Sangat baik : range (0,71 ≤ ASW < 1)
2. Baik : range (0,51 ≤ ASW < 0,71)
3. Cukup baik : range (0,26 ≤ ASW < 0,51)
4. Kurang baik : range (ASW < 0,26)

2.9.2 *Adjusted Rand index* (ARI)

Adjusted rand index merupakan salah satu metode evaluasi untuk *clustering* yang mengukur kemiripan dari dua data. *Adjusted rand index* merupakan perbaikan dari metode evaluasi *rand index* yang ditemukan oleh (Rand, 1971). Nilai *Adjusted rand index* memiliki rentang antara -1 sampai 1. Jika dibandingkan dengan *rand index*, a*djusted rand index* memiliki rentang nilai yang lebih besar. Hal ini membuat ARI sebagai ukuran kinerja yang lebih bagus (Vargas, Rodrigues, & Bedregal, 2013). Semakin besar nilai ARI, semakin baik kualitas suatu *cluster*. Formula dari *Adjusted rand index* dapat dilihat pada persamaan 2.12

$$Adjusted\ rand\ index = \frac{2\ (ad-bc)}{(a+b)(b+d)\ (a+c)(c+d)} \quad (2.12)$$

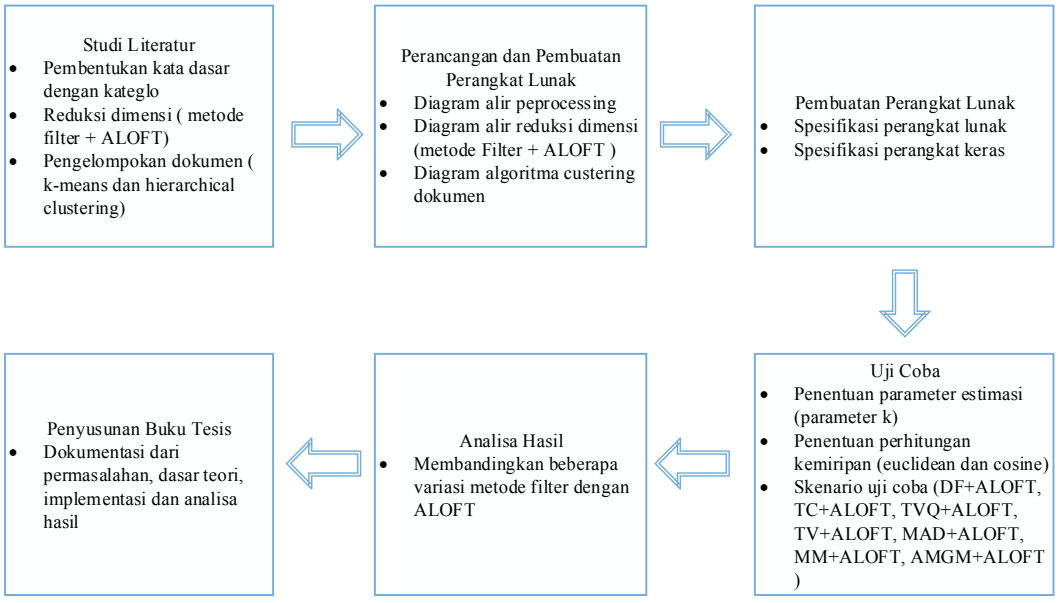
Nilai a,b,c,d didapat berdasarkan tabel kontingensi sesuai Tabel 1.1 dimana a adalah jumlah dokumen yang masuk kelompok yang sama antara *ground truth* dan hasil pengelompokan. b adalah jumlah dokumen yang masuk kelompok yang berbeda antara *ground truth* dan hasil pengelompokan. c adalah jumlah dokumen yang tidak masuk kelompok *ground truth* tapi masuk kelompok hasil pengelompokan sedangkan d jumlah dokumen yang tidak masuk kelompok *ground truth* dan tidak masuk dalam kelompok hasil pengelompokan.

Tabel 1. 1 Tabel Kontingensi

	Cluster yang sama (hasil pengelompokan)	*Cluster* berbeda (hasil pengelompokan)
Cluster yang sama (*ground truth*)	a	c
Cluster berbeda (*ground truth*)	b	d

BAB 3
METODOLOGI PENELITIAN

Pada bab ini akan dibahas metode penelitian yang digunakan dan langkah – langkah yang dilakukan dalam rangka mencapai tujuan yang diharapkan dalam penelitian ini. Tahapan – tahapan yang dilalui pada penelitian ini meliputi studi literatur untuk mempelajari permasalahan dan penelitian terkini, perancangan sistem, pembuatan perangkat lunak berdasarkan perancangan yang telah dilakukan, kemudian dilakukan uji coba terhadap perangkat lunak yang telah dibuat dan selanjutnya dilakukan analisis. Tahapan yang terakhir adalah penyusunan buku tesis. Alur tahapan tersebut dapat dilihat pada Gambar 3.1



Gambar 3. 1 Diagram Alur Metodologi Penelitian

3.1 Studi Literatur

Dalam melakukan suatu penelitian, tahapan studi literatur merupakan tahapan yang harus dilakukan dimana tahapan ini berkaitan dengan suatu pemahaman detail baik dari sisi dasar teori yang dipakai maupun teknis dari setiap tahapan suatu penelitian. Pada tahap ini dipelajari tentang informasi dari literatur yang digunakan, perkembangan, serta penelitian – penelitian sebelumnya yang berkaitan dengan konteks penelitian yang dilakukan. Informasi – informasi tersebut didapatkan dari

buku, jurnal, artikel, tesis dan sumber informasi lainnya. Adapun topik literatur yang perlu dipelajari pada penelitian ini adalah sebagai berikut :

1. *Text preprocessing* yang meliputi pembersihan data, *tokenizing*, *stopword removal*, dan *stemming* bahasa Indonesia
2. Model pencarian kata dasar berdasarkan kata turunan dari Kateglo
3. Metode seleksi fitur dengan model filter yang meliputi perhitungan relevansi dengan metode filter dan pemilihan fitur dengan menggunakan algoritma ALOFT
4. Metode *clustering* dengan algoritma *k-means* dan HAC
5. Metode evaluasi *cluster* dengan menggunakan *Silhouette coefficient*

3.2 Perancangan Sistem

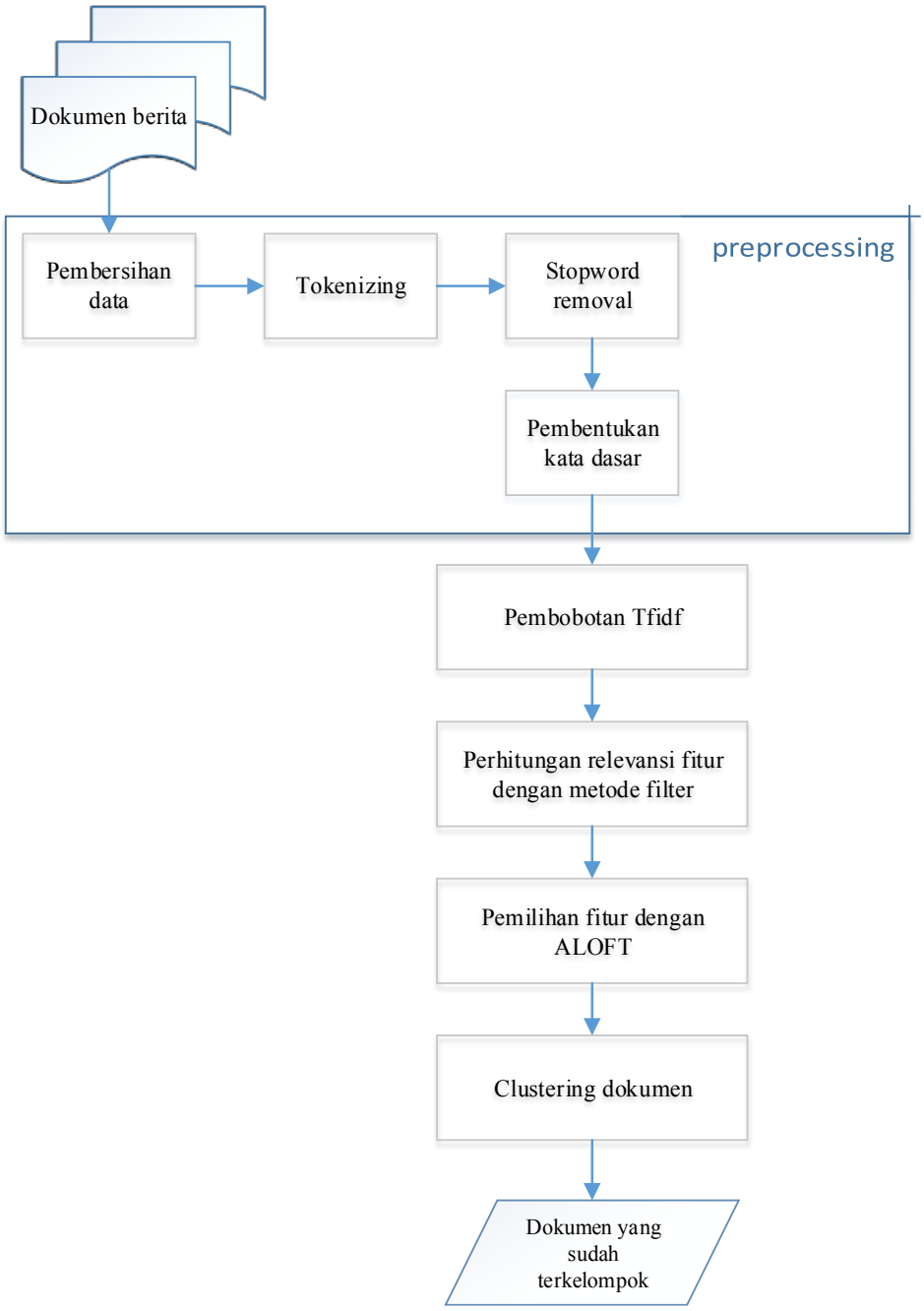
Secara umum rancangan sistem yang digunakan dapat dilihat pada Gambar 3.2. Sistem yang dibagun diuji dengan data berupa dokumen berita yang diambil dari situs berita online di Indonesia.

3.2.1 Fase *Preprocessing* Dokumen

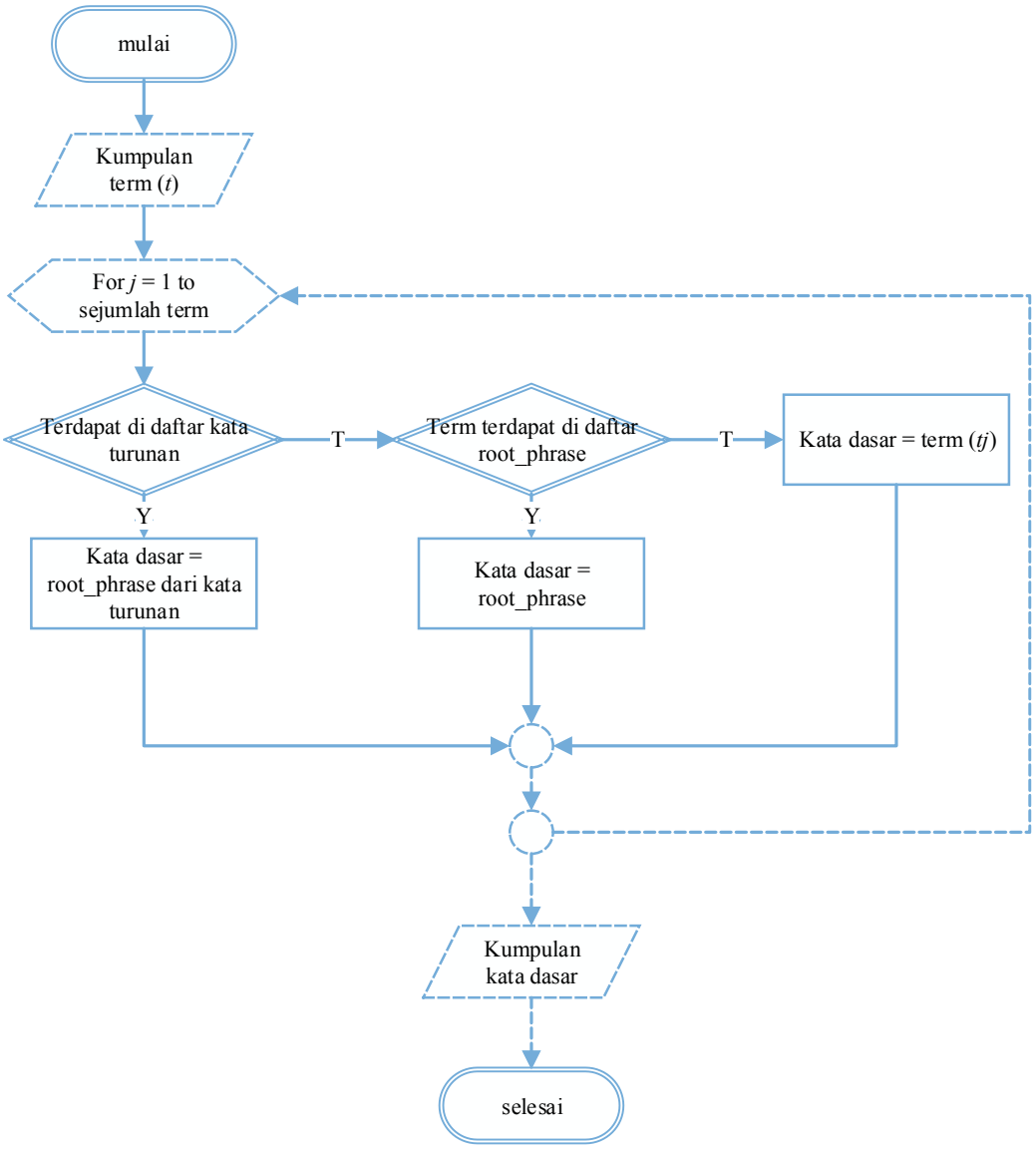
Fase ini adalah fase yang pertama kali yang harus dilakukan sebelum dokumen diproses untuk seleksi fitur dan pengelompokan dokumen. Data masukan yang digunakan adalah berupa koleksi dokumen berita dan daftar kata *stopword* kemudian dilanjutkan dengan proses pembersihan data, *tokenizing*, *stopword removal*, dan pembentukan kata dasar. Keluaran yang dihasilkan berupa koleksi term dalam bentuk kata dasar.

Koleksi dokumen *D* yang menjadi masukan berupa dokumen berita berbahasa Indonesia yang di ambil dari situs berita online Kompas (www.kompas.com) pada tanggal 16-03-2009 sampai 20-09-2015 . Data yang dimanfaatkan adalah judul berita dan isi dari berita. Pada fase *preprocessing* dokumen akan dilakukan proses pembersihan data untuk menghilangkan tag html, gambar, tanggal terbit, nama penulis, dan editor. Kemudian proses *tokenizing* dilakukan untuk memotong string dokumen input berdasarkan pemisah kata yaitu spasi. Selanjutnya pada proses *stopword removal* akan dilakukan penghapusan kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna misalnya yang, di, ke, dengan, dan sebagainya. Kamus daftar kata *stopword* yang digunakan merupakan

kumpulan bahasa Indonesia yang didapatkan pada appendix sebuah penelitian (Tala, 2003). Pada tahap terakhir adalah proses pencarian kata dasar dengan menggunakan Kateglo, yaitu dengan cara mencari *root phrase* dari kata turunan.



Gambar 3. 2 Arsitektur Sistem



Gambar 3. 3 Diagram Alir Pembentukan Kata Dasar

Gambar 3.3 merupakan diagaram alir pembentukan kata dasar dengan Kateglo. Masukan dari proses ini adalah kumpulan term yang sudah dilakukan proses *stopword removal*. Term – term ini kemudian akan dilakukan pengecekan apakah term tersebut terdapat di dalam daftar kata turunan. Jika term (t_j) tersebut terdapat didalam daftar kata turunan, maka *root phrase* dari kata turunan tersebut merupakan kata dasar. Jika ternyata t_j tidak terdapat di dalam daftar kata turunan maka dilakukan pengecekan terhadap daftar *root phrase.* Apabila t_j tiidak terdapat di dalam daftar kata turunan dan daftar *root phrase* maka term t_j tersebut dianggap sudah berupa kata dasar.

3.2.2 Pembobotan Term

Setiap term yang dihasilkan dari proses preprocessing merupakan representasi dari fitur, kemudian term / fitur ini akan dilakukan perhitungan bobot dengan menggunakan *tfidf*. *Tfidf* merupakan pembobotan term berdasarkan jumlah frekuensi yang terdapat pada dokumen dari seluruh koleksi dokumen. Penjelasan lengkap tentang *tfidf* berserta formula yang digunakan pada penelitian ini dapat dilihat pada persamaan 2.1

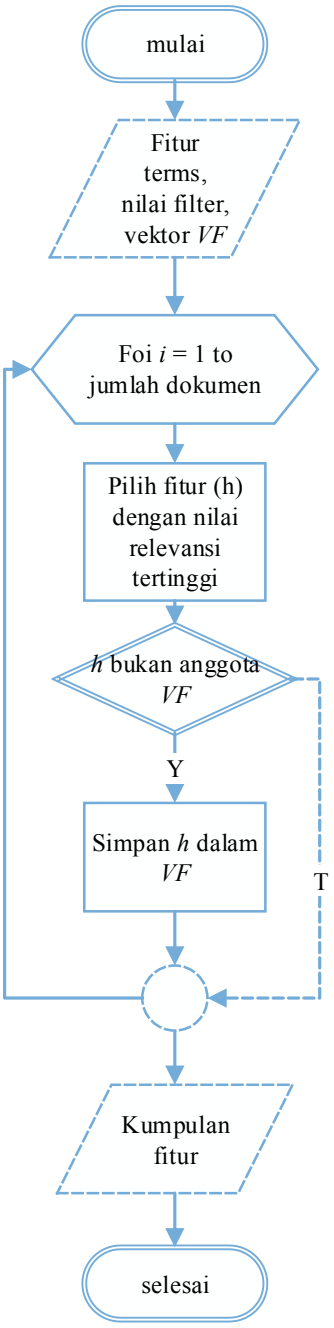
3.2.3 Relevansi Fitur dengan Filter

Fase ini adalah fase awal sebelum dilakukan pemilihan fitur untuk mengurangi dimensi dari vektor dokumen. Tujuan dari pengurangan dimensi ini adalah agar dapat meningkatkan performa dari algoritma *cluster*ing. Proses ini bertujuan untuk menentukan skor relevansi dari sebuah fitur. Semakin tinggi nilai filter dari sebuah fitur maka semakin relevan fitur tersebut, begitu juga sebaliknya. Perhitungan nilai filter dilakukan terhadap seluruh fitur tanpa adanya batasan bobot tertentu, sehingga meskipun nilai bobot *tfidf* dari sebuah fitur kecil akan tetap dilakukan perhitungan nilai filter. Hal ini dilakukan karena besar kecilnya nilai *tfidf* untuk setiap dokumen dapat berbeda-beda, pada dokumen yang berbeda dapat menghasilkan bobot *tfidf* yang berbeda. Sedangkan nilai filter untuk sebuah fitur dalam keseluruhan dokumen hanya akan menghasilkan satu nilai. Pada penelitian ini digunakan tujuh penilaian fitur yaitu *Document Frequency* (DF), *Term Contributtion* (TC), *Term Variance Quality* (TVQ), *Term Variance* (TV), *Mean Absolute Difference* (MAD), *Mean Median* (MM), *Arithmetic Mean Geometric Mean* (AMGM). Penjelasan secara rinci dari metode ini disajikan pada Bab 2.

3.2.4 ALOFT

Himpunan fitur yang akan digunakan untuk pengelompokan dokumen akan dipilih dengan menggunakan metode ALOFT. Dengan algoritma ini dipastikan bahwa setiap dokumen akan berkontribusi untuk pemilihan akhir himpunan fitur. Setidaknya terdapat satu fitur yang mewakili dokumen. Masukan dari proses ini adalah himpunan fitur yang berupa term beserta nilai relevansinya masing – masing. Selanjutnya dari fitur – fitur tersebut akan dilakukan pemilihan himpunan fitur akhir dengan menggunakan ALOFT. Keluaran dari proses ini adalah

sekumpulan fitur terbaik dari setiap dokumen. Diagram alir dari ALOFT dapat dilihat pada Gambar 3.4



Gambar 3. 4 Diagram Alir ALOFT

Proses pertama yang dilakukan pada ALOFT adalah mengosongkan vektor fitur (*VF*), kemudian untuk setiap dokumen akan dicari fitur (*h*) yang memiliki nilai relevansi tertinggi. Dimulai dari dokumen pertama, fitur dengan nilai tertinggi (h_1)

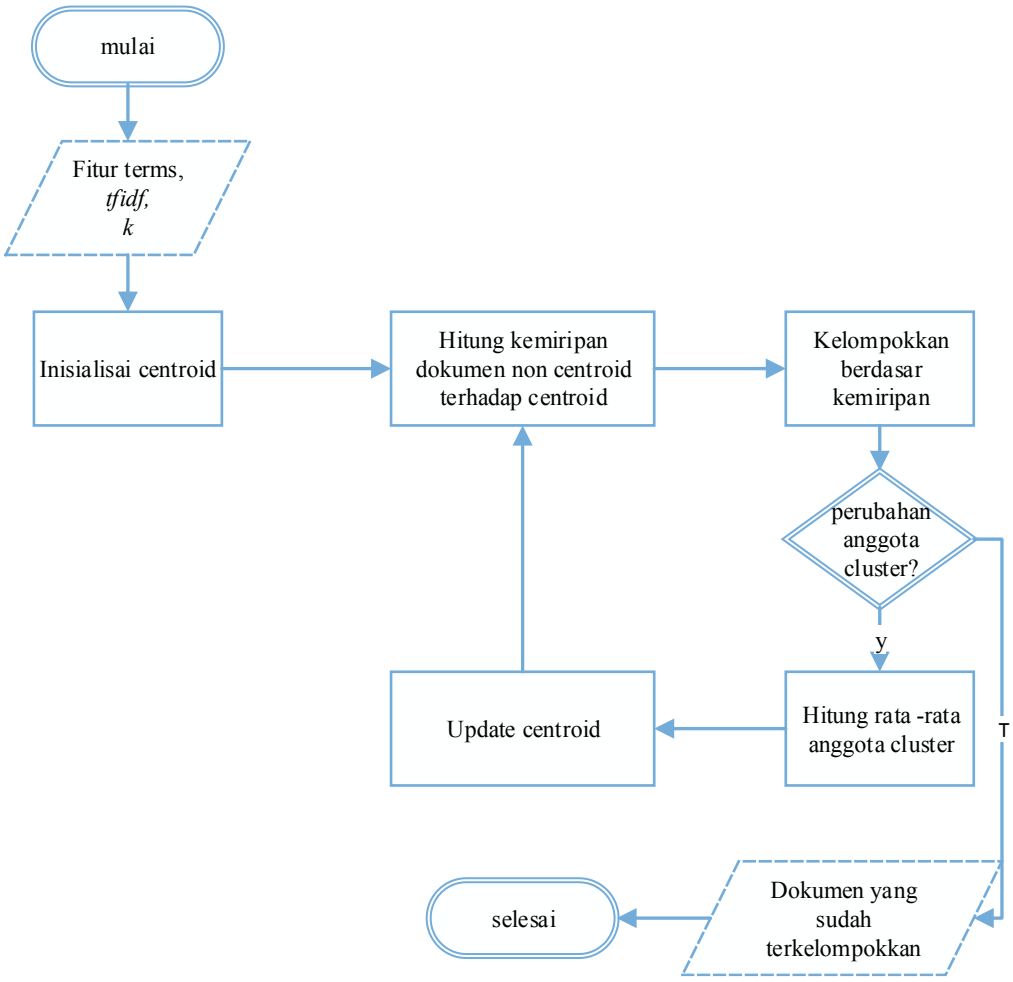
akan disimpan pada vektor fitur (*VF*) kemudian beralih ke dokumen yang kedua, fitur dengan nilai tertinggi dari dokumen 2 (h_2) akan disimpan ke dalam vektor *VF* jika vektor *VF* tidak memiliki fitur (h_2). Jika ternyata vektor *VF* memiliki fitur (h_2) maka pemilihan fitur akan berlanjut ke dokumen berikutnya. Iterasi ini akan diulang sampai dokumen terakhir.

3.2.5 Pengelompokan dokumen

Himpunan fitur yang terpilih dari proses sebelumnya akan di proses pada bagian ini untuk mendapatkan kelompok dokumen. Dokumen yang memiliki kemiripan *content* akan di terkumpul dalam satu kelompok. Metode *clustering* yang digunakan pada penelitian ini adalah *k-means* dan HAC.

K - means

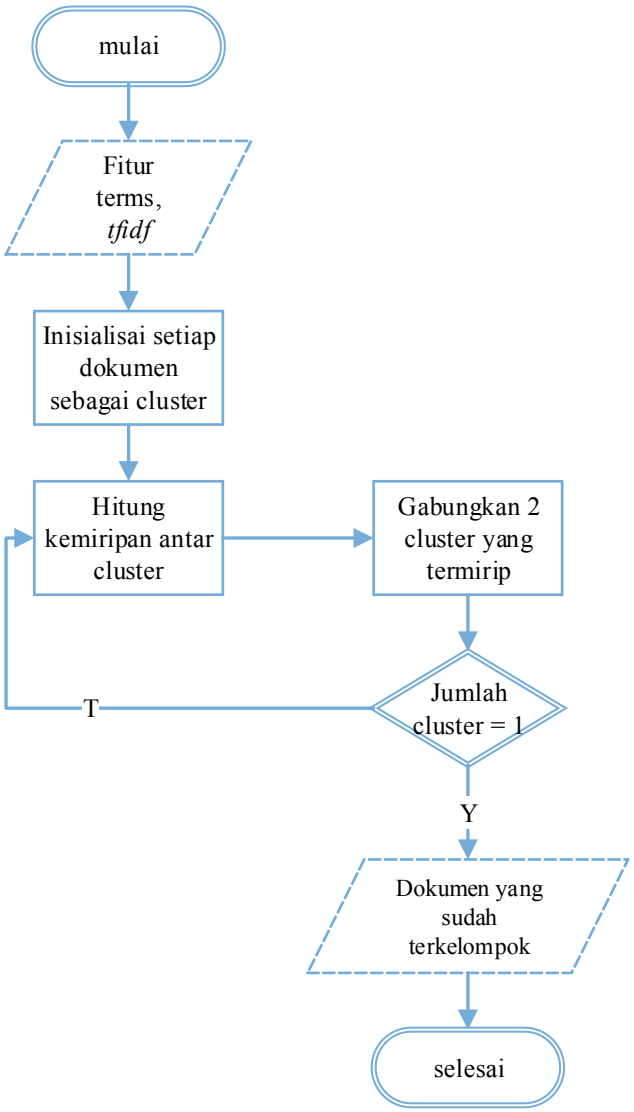
Detail proses algoritma *k-means* yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.5. Himpunan fitur hasil seleksi fitur, bobot dari fitur berupa *tfidf* merupakan masukan dari proses ini selain itu proses ini juga membutuhkan inisialisasi jumlah *cluster* yang ingin dihasilkan. Langkah pertama adalah untuk memilih sejumlah *k* centroid secara acak sesuai *cluster* yang akan dibentuk. Pada *cluster*ing dokumen maka yang dijadikan centroid adalah dokumen. Selanjutnya dokumen selain centroid akan dihitung kemiripannya terhadap centroid. Algoritma *k* means kemudian menentukan anggota *cluster* dengan cara memilih dokumen-dokumen yang memiliki kemiripan tertinggi antara centroid dan non-centroid dokumen. Selanjutnya untuk setiap *cluster* dihitung rata-rata dari anggota *cluster* untuk menentukan centroid baru. Iterasi berlanjut sampai tidak ada perubahan anggota *cluster*.



Gambar 3. 5 Diagram Alir *K – Means Clustering*

Hierarchical Agglomerative Clustering

Sama seperti halnya metode *clustering* lainnya, metode ini juga membutuhkan masukan berupa himpunan fitur hasil seleksi fitur, bobot dari term berupa *tfidf*. HAC yang digunakan pada penelitian ini adalah dengan kriteria pengukuran kemiripan *minimum similarity* yaitu *complete – linkage*. Secara detail proses algortima HAC dapat dilihat pada Gambar 3.6



Gambar 3. 6 Diagram Alir *Hierarchical Agglomerative Clustering*

3.3 Pembuatan Perangkat Lunak

Pada tahap ini dilakukan implementasi rancangan sistem ke dalam kode program sehingga dapat dimengerti oleh komputer. Lingkungan implementasi dilakukan pada penelitian ini adalah sebagai berikut :

1. Spesifikasi perangkat lunak
   a. Sistem operasi Microsoft Windows 8
   b. Apache *Web server*
   c. *Database Management System* (DBMS) MySQL

d. Bahasa pemrograman python.

2. Spesifikasi perangkat keras

   Spesifikasi komputer yang digunakan pada penelitian ini adalah komputer dengan prosesor Intel(R) Core (TM) i5-3230M CPU @ 2.60GHz RAM 4.00GB.

3.4 Skenario uji coba

Uji coba dilakukan untuk menguji sistem dengan beberapa parameter yang ada pada metode. Parameter – parameter yang digunakan akan diestimasi dan di ubah – ubah untuk mendapatkan nilai yang optimal sehingga memberikan hasil pengujian yang terbaik. Selain itu pada bagian ini juga akan dijelaskan mengenai skenario uji coba yang akan dilakukan pada penelitian ini. Parameter yang akan diestimasi pada penelitian ini dapat dilihat pada Tabel 3.1

Tabel 3. 1 Parameter Estimasi

Parameter	Keterangan
Nilai *k*	Jumlah *cluster* yang akan dibentuk dalam proses perhitungan *k-means* dan HAC

Skenario uji coba yang akan dilakukan untuk menguji reduksi dimensi fitur menggunakan variasi metode filter pada algoritma ALOFT terdiri dari beberapa skenario. Setiap variasi metode filter pada ALOFT akan dilakukan pengelompokan dokumen dengan menggunakan dua buah metode kemiripan yang berbeda yaitu *cosine similarity* dan *euclidean distance*, selanjutnya hasil *cluster* akan dievaluasi dengan menggunakan metode *silhouette* untuk mengetahui kualitas dari hasil pengelompokan dokumen. Nilai s*ilhouette* merupakan nilai kualitas *cluster* yang menunjukkan derajat atau tingkat kedekatan objek di dalam sebuah *cluster*.

Kebenaran hasil *cluster* dari fitur yang telah direduksi dievaluasi dengan menggunakan metode evaluasi *adjusted rand index*. Tujuan lain dari uji coba menggunakan metode evaluasi *adjusted rand index* adalah untuk mengetahui

apakah himpunan fitur akhir yang sudah terpilih dapat mewakili dokumen aslinya . Beberapa skenario yang digunakan adalah sebagai berikut :

Uji coba 1 : Pengujian untuk mengetahui pengaruh metode perhitungan kemiripan

Pada uji coba 1 dilakukan untuk mengetahui pengaruh metode kemiripan yang digunakan yaitu *cosine similarity* dan *euclidean distance*. Selain itu pada pengujian ini juga dilakukan pada data yang dilakukan proses pencarian kata dasar menggunakan produk Kateglo. Skenario yang dilakukan pada uji coba ini terdiri dari 14 skenario seperti berikut ini :

- Skenario 1 : Filter DF + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 2 : Filter DF + ALOFT + HAC (dengan beberapa variasi nilai *k*)
- Skenario 3 : Filter TC + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 4 : Filter TC + ALOFT + HAC (dengan beberapa variasi nilai *k*)
- Skenario 5 : Filter TVQ + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 6 : Filter TVQ + ALOFT + HAC (dengan beberapa variasi nilai *k*)
- Skenario 7 : Filter TV + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 8 : Filter TV + ALOFT + HAC (dengan beberapa variasi nilai *k*)
- Skenario 9 : Filter MAD + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 10 : Filter MAD + ALOFT + HAC (dengan beberapa variasi nilai *k*)
- Skenario 11 : Filter MM + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 12 : Filter MM + ALOFT + HAC (dengan beberapa variasi nilai *k*
- Skenario 13 : Filter AMGM + ALOFT + *K-means* (dengan beberapa variasi nilai *k*)
- Skenario 14 : Filter AMGM + ALOFT + HAC (dengan beberapa variasi nilai *k)*

Uji coba 2 : Pengujian dengan Tanpa Kata Dasar

Pada pengujian tanpa kata dasar dilakukan untuk pembanding nilai rata – rata *silhouette* (ASW) dari uji coba 1, perhitungan kemiripan yang digunakan pada uji

coba 2 adalah perhitungan kemiripan yang memiliki nilai ASW terbaik dari uji coba 1 . Sehingga nantinya akan diketahui seberapa besar pengaruh penggunaan Kateglo terhadap kualitas *cluster* yang dihasilkan. Seperti halnya uji coba 1 yang memiliki 14 skenario pengujian, uji coba 2 juga memiliki 14 skenario pengujian yang sama seperti uji coba 1.

3.5 Penyusunan Buku Tesis

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, rancangan, implementasi yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tesis. Buku Tesis yang akan disusun bertujuan untuk memberikan gambaran dari pengerjaan tesis dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini dipaparkan hasil uji coba penelitian yang telah dilakukan terkait dengan pengelompokan dokumen berita online dengan melakukan reduksi dimensi fitur menggunakan variasi metode filter pada ALOFT.

4.1 Spesifikasi Sitem

Metode dalam penelitian ini diaplikasikan dengan didukung oleh perangkat keras dan perangkat lunak dengan spesifikasi seperti Tabel 4.1

Tabel 4. 1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Nama	Spesifikasi
Pocessor	Intel(R) Core (TM) i5-3230M CPU @ 2.60GHz
Memori (RAM)	4.00 GB
Sistem Operasi	Microsoft Windows 8 Pro 64-bit
Bahasa Pemrograman	Python 2.7
Tools pemrograman	PyCharm 5.03
Database Management System	MySQL versi 5.1.41

4.2 Implementasi Metode

Penelitian ini dibangun dengan menggunakan lima fase yang harus dilalui yaitu *preprocessing*, pembobotan term dengan menggunakan *tfidf*, perhitungan relevansi fitur dengan metode filter, selanjutnya dilakukan pemilihan himpunan fitur dengan menggunakan algoritma ALOFT dan terakhir adalah proses pengelompokan dokumen.

4.2.1 Implementasi *Preprocessing* Dokumen

Fase ini bertujuan untuk membuat dokumen masukan lebih konsisten dan mempermudah representasi teks. Proses yang dilakukan pada fase ini terdiri dari tiga tahap yaitu proses pembersihan data, *tokenizing*, *stopword removal,* dan pembentukan kata dasar dengan produk Kateglo. Data yang digunakan dalam peneltian ini berjumlah 1000 dokumen dan di ambil dari situs berita online Kompas (www.kompas.com). Data tersebut memiliki format *xml* hanya beberapa *tag* saja

yang digunakan yaitu *tag* judul dan isi berita. Setelah didapatkan data tersebut selanjutnya data tersebut dimasukkan ke dalam database dan siap untuk diproses pada tahap selanjutnya. Gambar 4.1 merupakan contoh format dokuman yang digunakan dalam penelitian ini setelah dilakukan pembersihan data.

```
Ahok Undang Jokowi Hadiri Pembangunan Kampung Atlet
Gubernur DKI Jakarta Basuki Tjahaja Purnama (Ahok) mengatakan peletakan batu pertama atau groundbreaking pembangunan kampung atlet untuk Asian Games 2018 dilaksanakan pada 15 September 2015. Basuki meminta kehadiran Presiden Joko Widodo untuk meresmikan sarana penunjang yang terletak di Kemayoran, Jakarta Pusat itu.  "Kalau Presiden ada waktu, kami mau minta datang untuk groundbreaking di Kemayoran. Kampung Atlet ini untuk menampung hampir 15.000 atlet Asian Games," kata Basuki di Balai Kota, Selasa (8/9/2015).  Rencananya kampung atlet seluas 11 dan 5 hektar itu akan dibangun oleh PT Jakarta Propertindo. Untuk membangun kampung atlet itu, Pemprov DKI akan mengalokasikan sejumlah anggarannya sebagai penyertaan modal pemerintah (PMP) PT Jakpro. Sementara kampung atlet siap dibangun, lahan lapangan golf yang akan diubah menjadi lapangan futsal belum diserahkan pengelolaannya ke Pemprov DKI Jakarta. "Kami ingin ubah lapangan golf itu kayak di Victoria Park. Nah kami ingin membuat lapangan golf jadi lapangan bola kaki segitu banyak," kata Basuki.
```

Gambar 4. 1 Contoh Dokumen Berita sebagai Dataset Uji Coba

```
1   def token(text_news):
2       text_lower = text_news.lower()
3       remove_number = re.sub(r'\d+', '', text_lower)
4
5       #tokenizing
6       tokenizer = RegexpTokenizer(r'\w+')
7       token = tokenizer.tokenize(remove_number)
8
9       #stopword removal
10      #isi finnish diganti dengan stopword bhs indonesia
11      stops = set(stopwords.words("finnish"))
12      meaningful_words = [w for w in token if not w in stops]
13      return meaningful_words
```

Gambar 4. 2 Potongan Kode Proses Tokenizing dan Stopword Removal

Setelah didapatkan data bersih selanjutnya dilakukan proses *tokenizing* yang didahului dengan menghilangkan seluruh tanda baca dan angka sehingga di dalam dokumen berita tersebut hanya tersisa huruf saja. Selanjutnya proses *tokenizing* dilakukan dengan memecah masing – masing kata dengan berdasarkan pemisah kata yaitu spasi. Setelah didapatkan term – term untuk masing – masing dokumen

dilakukan proses *stopword removal* dengan cara menghapus term yang terdapat dalam daftar *stopword*. Terdapat sebanyak 758 daftar *stopword* bahasa Indonesia yang digunakan, daftar *stopword* yang digunakan didapat dari appendix sebuah penelitian (Tala, 2003). Untuk proses *tokenizing* dan *stopword removal* dilakukan dengan potongan kode program seperti pada Gambar 4.2 dan dihasilkan term - term dalam bentuk token yang kemudian disimpan dalam database.

Tabel 4. 2 Contoh Kata Turunan pada Kateglo

Kata Dasar	Kata Turunan
sidik	menyidik
sidik	penyidik
sidik	penyidikan
jelajah	menjelajah
jelajah	menjelajahi
jelajah	menjelajahkan
jelajah	penjelajah
jelajah	penjelajahan
sapu	disapu
sapu	menyapu
sapu	menyapukan
sapu	penyapuan
sapu	sapuan
sapu	tersapu

```
1   SELECT
2   c_tf.id_doc,c_df.kata_dasar,(c_tf.nilai_TF*(LOG10(N/c_df.DF)+1))
3   AS TfIdf
4   FROM
5          (SELECT  COUNT(DISTINCT  id_doc)  AS  DF,kata_dasar  FROM
6          `result_kadas` WHERE `kata_dasar` IN
7          (SELECT DISTINCT kata_dasar FROM `result_kadas`
8          )GROUP BY kata_dasar) c_df,
9          (SELECT id_doc,terms,`nilai_TF` FROM `bobot_tf`)c_tf,
10         (SELECT MAX(id_doc) AS N FROM result_kadas) n
11  WHERE c_df.kata_dasar=c_tf.terms
```

Gambar 4. 3 Potongan Kode Pencarian Kata Dasar

Proses selanjutnya adalah pencarian kata dasar dengan menggunakan produk Kateglo dimana didalam Kateglo disediakan kamus kata turunan. Dari kata turunan ini dapat dicari *root phrase* yang merupakan kata dasar. Terdapat 40.144 kata turunan dan 9.369 *root phrase* (kata dasar) pada Kategelo. Tabel 4.2 merupakan contoh data kata turunan yang disediakan oleh Kateglo. Jumlah term sebelum dilakukan proses pencarian kata dasar adalah sebanyak 13.859 term, setelah

dilakukan proses pencarian kata dasar jumlah term yang terbentuk berkurang menjadi 12.045 term. Untuk mendapatkan kata dasar dari term digunakan potongan kode seperti Gambar 4.3.

4.2.2 Implementasi Pembobotan

Perhitungan pembobotan yang dilakukan pada penelitian ini dengan menggunakan *tfidf* dimana setiap *term* / fitur yang dihasilkan dari proses *preprocessing* akan dilakukan perhitungan bobot. Bobot *tfidf* yang didapatkan dari penelitian ini memiliki rentang nilai antara 1,379 – 148,961. Gambar 4.4 merupakan potongan kode yang digunakan untuk menghitung bobot *tfidf*.

```
1       SELECT
2  nwt.`id_doc`,nwt.`content_term`,COALESCE(t.parent_name,nwt.`conten
3  t_term` ) AS kata_dasar, nwt.urutan
4       FROM  turunan t RIGHT JOIN `news_token` nwt
5       ON t.`child_name`=nwt.`content_term`
6       ORDER BY urutan
```

Gambar 4. 4 Potongan Kode Pembobotan *tfidf*

4.2.3 Implementasi Metode Filter

Metode Filter yang digunakan pada penelitian ini adalah metode filter *Unsupervised* dimana metode ini tidak membutuhkan label kelas. Terdapat tujuh buah metode yang digunakan yaitu *Document Frequency* (DF), *Term Contributtion* (TC), *Term variance quality* (TVQ), *Term Variance* (TV), *Mean Absolute Difference* (MAD), *Mean Median* (MM), dan *Arithmetic Mean Geometric Mean* (AMGM).

Penggunaan metode filter bertujuan untuk mencari nilai relevansi dari keseluruhan fitur yang terdapat di dokumen. Semakin besar nilai filter dari sebuah fitur / term maka semakin relevan fitur tersebut. Penjelesan mengenai masing – masing metode filter telah dijelaskan pada Sub-bab 2.7 dan implementasi dari setiap metode dapat dilihat pada Gambar 4.5

```
1   /*FILTER DF*/
2   SELECT id_doc,kata_dasar,COUNT(DISTINCT id_doc) AS DF FROM
3   `result_kadas` WHERE `kata_dasar` IN
4         (SELECT DISTINCT kata_dasar FROM `result_kadas`
5         )GROUP BY kata_dasar
6   ORDER BY DF DESC
7
8   /*FILTER TC*/
9   SELECT tabel.termsa,SUM(tabel.kali)
10        FROM
```

```
11          (SELECT a.id_doc AS iddoca,a.terms AS termsa ,
12          a.`nilai_TfIdf` AS tfidfa, b.`id_doc`,  b.`terms`,
13          b.`nilai_TfIdf`, a.`nilai_TfIdf`*b.`nilai_TfIdf` AS  kali
14          FROM `bobot_tfidf` a, `bobot_tfidf` b
15          WHERE a.`terms`=b.`terms`
16          AND a.id_doc != b.id_doc
17          )tabel
18      GROUP BY tabel.termsa
19
20      /*FILTER TV*/
21      SELECT c_tf2.terms terms ,SUM(POW(c_tf2.nilai_TF-op.rata,2)) AS
22      TV
23      FROM
24          (SELECT AVG(c_tf.nilai_TF) AS rata, c_tf.terms terms
25          ,c_tf.id_doc iddoc
26          FROM
27          (SELECT id_doc,terms,`nilai_TF` FROM `bobot_tf`)c_tf
28          GROUP BY terms)op,
29          (SELECT id_doc,terms,`nilai_TF` FROM    `bobot_tf`)c_tf2
30      WHERE c_tf2.terms = op.terms
31      GROUP BY terms
32
33      /*FILTER TVQ*/
34      SELECT op.kadas,op.dp-((1/N)*op.bl)AS TVQ
35          FROM
36          (
37          SELECT SUM(POW(c_tf.tf,2)) AS dp,COUNT(c_tf.id_doc)  AS
38      N, SUM(c_tf.tf) AS bl ,c_tf.kata_dasar kadas  ,c_tf.id_doc iddoc
39          FROM
40          (SELECT COUNT(kata_dasar) AS tf,kata_dasar,id_doc    FROM
41      `result_kadas`
42          GROUP BY kata_dasar,id_doc ) c_tf
43          GROUP BY kata_dasar
44          )op
45
46      /*FILTER MAD*/
47      SELECT c_tf2.terms terms , (1/n.N) *
48      SUM(ABS(c_tf2.`nilai_TfIdf`- op.rata)) AS MAD
49          FROM
50          (SELECT AVG(c_tf.tfidf) AS rata, c_tf.terms terms
51          ,c_tf.id_doc iddoc
52          FROM
53          (SELECT id_doc,terms,`nilai_TfIdf` AS tfidf FROM
54          `bobot_tfidf`)c_tf
55          GROUP BY terms)op,
56          (SELECT id_doc,terms,`nilai_TfIdf` FROM
57          `bobot_tfidf`)c_tf2,
58          (SELECT MAX(id_doc) AS N FROM result_kadas) n
59      WHERE c_tf2.terms = op.terms
60      GROUP BY terms
61
62      /*FILTER MM*/
63      SELECT kadas AS term,`mean_median`(a.kadas)
64          FROM
65          (SELECT    DISTINCT    kata_dasar    AS    kadas    FROM
66          `result_kadas`)a
67
68      /*FILTER DF*/
69      SELECT c.term,c.atas/c.product
70      FROM (
71      SELECT ((1/nkcl.N)* SUM(c_tf2.tfidf)) AS atas,
72          POW(EXP(SUM(LOG(c_tf2.tfidf))), (1/nkcl.N)) AS
73          product, terms AS term
74          FROM
```

```
75          (SELECT COUNT(c_tf.id_doc) AS N,c_tf.terms kadas
76          FROM
77          (SELECT `nilai_TfIdf` AS tfidf,terms,id_doc
78          FROM `bobot_tfidf`) c_tf
79          GROUP BY c_tf.terms) nkcl,
80          (SELECT b.id_doc,b.terms,(b.tfidf-
81          a.mintif)/(a.maxtif-a.mintif) AS tfidf
82          FROM
83          (SELECT id_doc,terms,nilai_tfidf AS
84          tfidf,MAX(nilai_TfIdf) AS maxtif, MIN(`nilai_TfIdf`)
85          mintif FROM bobot_tfidf
86          )a,
87          (SELECT id_doc,terms,nilai_TfIdf AS tfidf FROM
88          bobot_tfidf) b)c_tf2
89          WHERE nkcl.kadas=c_tf2.terms
90    GROUP BY terms
```

Gambar 4. 5 Potongan kode untuk Metode Filter

4.2.4 Implementasi ALOFT

Fase ALOFT merupakan fase yang berfungsi untuk memilih satu himpunan fitur akhir yang nantinya akan digunakan untuk pengelompokan dokumen. Proses ini dilakukan setelah setiap fitur memiliki nilai relevansi dari perhitungan dengan setiap metode filter. Sehingga dari setiap metode filter akan memiliki satu himpunan fitur akhir. Gambar 4.6 merupakan potongan kode yang digunakan untuk memilih fitur dengan metode ALOFT untuk filter DF. Pada potongan kode tersebut mulai dari dokumen pertama akan dicari fitur yang memiliki nilai DF tertinggi selanjutnya fitur tersebut akan disimpan ke dalam tabel fitur. Berikutnya diulangi untuk dokumen kedua dan seterusnya dengan syarat bahwa fitur yang memiliki nilai DF tertinggi belum ada di dalam tabel fitur.

```
1     .........
2         SET p1 = 1;
3         SELECT MAX(id_doc)+1 INTO max_id FROM result_kadas;
4
5         WHILE p1 < max_id DO
6            SELECT  a.`id_doc` ,a.`terms`  ,b.`urutan` INTO
7            var_id_doc,var_fitur,var_urutan
8            FROM filter_df a, result_kadas b
9            WHERE `nilai_DF` =
10           (
11           SELECT MAX(nilai_DF) max_DF
12           FROM `filter_df` f, `result_kadas`rk
13           WHERE rk.`id_doc` = p1 AND rk.`kata_dasar`=f.`terms`
14           )
15           AND b.id_doc = p1
16           AND a.`terms`=b.`kata_dasar`
17           AND a.`terms`  NOT IN (SELECT DISTINCT `fitur_term`  FROM
18    fitur)
19           ORDER BY b.`urutan`
20           LIMIT 1
21           ;
22
```

```
23          IF (var_id_doc IS NOT NULL  AND var_fitur IS NOT     NULL
24    AND var_urutan IS NOT NULL)
25          THEN
26                INSERT INTO fitur (id_doc,fitur_term,urutan)
27                VALUES (var_id_doc,var_fitur,var_urutan);
28          END IF;
29
30          SET var_id_doc = NULL;
31          SET var_fitur = NULL;
32          SET var_urutan = NULL;
33
34    ........
```

Gambar 4. 6 Potongan Kode untuk Algoritma ALOFT

4.2.5 Implementasi Pengelompokan Dokumen

Pengelompokan dokumen berita dilakukan setelah terpilih fitur – fitur yang dianggap penting setelah melalui proses pemilihan fitur dengan algoritma ALOFT. Pada penelitian ini digunakan dua buah metode clustering yaitu *k-means* dan *Hierarchical Agglomerative Clustering*. Selanjutnya setelah dilakukan pengelompokan dilakukan perhitungan rata- rata *silhouette coefficient* (ASW) untuk mengetahui kualitas *cluster* yang dihasilkan. Keseluruhan proses ini dapat dilihat pada Gambar 4.7

```
1     #Kmeans
2     def kmeans_clustering(vsm_doc,num_cluster):
3           km = KMeans(n_clusters=num_cluster,init='k-means++')
4           km.fit_predict(vsm_doc)
5           centroids = km.cluster_centers_
6           list_centroids = centroids.tolist()
7           clusters = km.labels_.tolist()
8           sil_coef = metrics.silhouette_score(vsm_doc,km.labels_)
9           return (clusters,sil_coef,list_centroids)
10    #HAC
11    def hac_clustering (vsm_doc,num_cluster):
12        cluster = AgglomerativeClustering( n_clusters=num_cluster,
13        compute_full_tree=True, affinity='euclidean',
14        linkage='complete')
15        cluster.fit_predict(vsm_doc)
16        cluster_label = cluster.labels_.tolist()
17        #print cluster.labels_
18        sil_coef = metrics.silhouette_score(vsm_doc,
19        cluster.labels_, metric="euclidean")
20        return cluster_label,sil_coef
```

Gambar 4. 7 Potongan Kode Program untuk Proses Pengelompokan Dokumen

4.3 Hasil dan Uji Coba

Pada bab ini dibahas mengenai hasil uji coba terhadap metode yang diusulkan. Tujuan dari tahap ini adalah untuk mengetahui performa dari metode

yang diusulkan terhadap hasil dari pengelompokan dokumen. Dalam penelitian ini akan dilakukan uji coba dengan beberapa parameter yang berbeda pada metode usulan. Sesuai pada Sub-bab 3.4 parameter *k* yang digunakan diestimasi dan di ubah – ubah untuk mendapatkan nilai yang optimal sehingga memberikan hasil pengujian yang terbaik. Selain itu pada uji coba ini juga dilakukan skenario uji coba dimana setiap skenario dilakukan dengan cara setiap fitur yang telah dihitung nilai relevansinya dengan metode filter dan dilakukan pemilihan fitur akhir dengan algoritma ALOFT yang kemudian dilakukan pengelompokan dokumen, selanjutnya hasil *cluster* akan dievaluasi dengan menggunakan metode evaluasi *silhouette* untuk mengetahui kualitas dari hasil pengelompokan dokumen. Pada penelitian ini digunakan tujuh buah filter yang berbeda yaitu *Document Frequency* (DF), *Term Contributtion* (TC), *Term variance quality* (TVQ), *Term Variance* (TV), *Mean Absolute Difference* (MAD), *Mean Median* (MM), dan *Arithmetic Mean Geometric Mean* (AMGM).

4.3.1 Uji Coba 1 : Pengujian dengan Kata Dasar

Tujuan dari uji coba ini adalah untuk mengetahui pengaruh setiap metode filter dengan pemilihan fitur menggunakan algoritma ALOFT terhadap kualitas *cluster* yang dihasilkan oleh algoritma *k-means clustering* dan algoritma HAC. Jumlah fitur dari keseluruhan dokumen sebelum dilakukan proses reduksi dimensi adalah sebanyak 12.045 fitur. Hasil jumlah fitur akhir yang telah diseleksi dengan algoritma ALOFT untuk masing-masing metode filter dapat dilihat pada Tabel 4.3. Sedangkan untuk mengetahui himpunan fitur akhir setelah dilakukan reduksi dimensi dapat dilihat pada Lampiran 1.

Uji coba 1 bertujuan untuk mengukur kualitas *cluster* yang dihasilkan jika pada tahap *preprocessing* dilakukan proses pencarian kata dasar dengan menggunakan Kateglo. Selain itu dari uji coba ini nantinya juga akan terlihat kualitas *cluster* dari metode yang diusulkan yaitu dengan menggunakan variasi metode filter pada algoritma ALOFT. Uji coba ini dilakukan dengan menggunakan dua buah algoritma *clustering* yang berbeda yaitu *k-means* dan HAC dan dengan menggunakan dua buah pengukuran kemiripan yang berbeda yaitu dengan menggunakan *cosine similarity* dan *euclidean distance*. Tujuan dari penggunaan

jarak yang berbeda ini adalah untuk mengetahui metode perhitungan kemiripan yang sesuai dan menghasilkan kualitas *cluster* yang optimal.

Tabel 4. 3 Jumlah Fitur pada Masing - masing Metode Filter dengan Pencarian Kata Dasar

No.	Metode Filter	Jumlah Fitur
1	*Document Frequency* (DF) + ALOFT	19
2	*Term Contribution* (TC) + ALOFT	16
3	*Term Variance* (TV) + ALOFT	15
4	*Term Variance Quality* (TVQ) + ALOFT	16
5	*Mean Absolute Difference* (MAD) + ALOFT	15
6	*Mean Median* (MM) + ALOFT	168
7	*Arithmetic Mean Geometric Mean* (AMGM) + ALOFT	119

Salah satu permasalahan yang mempengaruhi hasil pengelompokan adalah penentuan jumlah *cluster* (*k)* yang ingin didapat. Dalam penelitian ini nilai *k* yang digunakan dimulai dari *k* = 2 kemudian bertambah sampai nilai *k* yang menunjukkan nilai *silhouette* yang memiliki kecenderungan menurun sehingga didapat *k* = 2 - 25. Hasil *cluster* untuk setiap nilai *k* akan dihitung validasinya dengan menggunakan *silhouette coefficient* sesuai dengan persamaan (2.10) dan (2.11) selanjutnya akan dihitung nilai rata – rata *silhouette* (ASW) penjelasan lengkap dari proses ini dapat dilihat pada Sub-bab 2.9.

Pada pengujian ini dilakukan dengan menggunakan dua buah algoritma *clustering* yaitu algoritma *k-means* dan *Hierarchical Agglomerative Clustering*. Pada algoritma *k-means* pengujian dilakukan dengan cara 50 kali percobaan dengan inisial centroid secara acak. Kemudian dari 50 kali percobaan yang dilakukan diambil inisial centroid yang memiliki nilai rata-rata *silhouette* (ASW) tertinggi. Berbeda dengan pengujian *k-means* yang membutuhkan inisial centroid pada algoritma HAC tidak diperlukan inisialisasi centroid karena pada algoritma ini setiap dokumen dianggap sebagai cluster tunggal dan kemudian berturut – turut digabungkan menjadi satu *cluster*. Sehingga hasil *cluster* untuk HAC dengan parameter *k* yang sama akan tetap meskipun berulang kali dijalankan.

Tabel 4. 4 Uji Coba Parameter Menggunakan *Cosine Similarity – K-means*

k	Rata – rata *Silhouette Coefficient* (ASW) *Cosine Similarity K-means*						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	0,350	0,401	0,401	0,411	0,415	0,258	0,120
3	**0,487**	**0,534**	**0,534**	**0,544**	**0,553**	0,280	0,145
4	0,478	0,503	0,503	0,544	0,526	0,238	0,141
5	0,399	0,501	0,501	0,511	0,526	0,245	0,131
6	0,428	0,511	0,428	0,527	0,542	0,256	0,143
7	0,386	0,445	0,445	0,417	0,476	0,265	0,151
8	0,401	0,330	0,330	0,416	0,363	0,272	0,158
9	0,409	0,335	0,335	0,439	0,374	0,278	0,159
10	0,419	0,331	0,334	0,435	0,369	0,285	0,164
11	0,425	0,336	0,340	0,447	0,380	0,294	0,173
12	0,410	0,344	0,343	0,455	0,390	0,296	0,174
13	0,419	0,350	0,351	0,463	0,391	0,298	0,182
14	0,406	0,353	0,353	0,470	0,392	0,310	0,187
15	0,423	0,356	0,359	0,475	0,400	0,315	0,195
16	0,368	0,352	0,344	0,480	0,393	0,263	0,191
17	0,412	0,359	0,343	0,484	0,363	0,269	0,204
18	0,405	0,355	0,352	0,460	0,372	0,277	0,207
19	0,367	0,339	0,355	0,447	0,371	0,272	0,214
20	0,409	0,337	0,339	0,463	0,364	0,279	0,213
21	0,399	0,327	0,320	0,454	0,369	0,290	0,218
22	0,385	0,331	0,329	0,435	0,333	0,252	0,220
23	0,379	0,338	0,321	0,447	0,367	0,252	0,222
24	0,355	0,335	0,307	0,445	0,355	0,311	0,223
25	0,363	0,321	0,310	0,443	0,339	**0,361**	**0,224**

Hasil dari pengujian dengan menggunakan algoritma *k-means* dengan perhitungan kemiripan *cosine similarity* dapat dilihat pada Tabel 4.4 sedangkan hasil pengujian dengan menggunakan algoritma *k-means* dan menggunakan perhitungan kemiripan *euclidean distance* dapat dilihat pada Tabel 4.5. Berdasarkan hasil dari Tabel 4.4 dapat dilihat bahwa pada perhitungan kemiripan dengan *cosine similarity* memiliki nilai *k* dengan nilai rata-rata *silhouette* (ASW) terbaik pada $k = 3$ untuk metode filter DF, TC, TV, TVQ, dan MAD. Pada metode filter DF nilai ASW pada $k = 3$ termasuk dalam kriteria nilai "cukup baik".

Sedangkan untuk metode filter TC, TV, TVQ, dan MAD memiliki nilai ASW yang lebih tinggi dibandingkan dengan metode filter DF dengan kriteria nilai "Baik". Pada Metode filter MM dan AMGM memiliki nilai rata-rata *silhouette* (ASW) terbaik pada $k = 25$, jika nilai k diperbanyak maka MM dan AMGM akan memiliki nilai rata-rata *silhouette* (ASW) yang lebih tinggi. Hal ini terjadi karena jumlah fitur yang dihasilkan oleh MM dan AMGM terlalu banyak.

Tabel 4. 5 Uji Coba Parameter Menggunakan *Euclidean Distance – K-means*

k	Rata – rata *Silhouette Coefficient* (ASW) *Euclidean distance K-means*						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	**0,390**	**0,440**	**0,405**	**0,613**	**0,436**	**0,132**	**0,371**
3	0,327	0,422	0,365	0,419	0,412	0,085	0,338
4	0,293	0,385	0,365	0,377	0,389	0,105	0,320
5	0,310	0,322	0,291	0,383	0,335	0,131	0,301
6	0,310	0,271	0,289	0,417	0,289	0,131	0,328
7	0,340	0,271	0,313	0,425	0,306	-0,048	0,331
8	0,338	0,300	0,293	0,374	0,318	-0,042	0,254
9	0,342	0,289	0,340	0,363	0,322	-0,019	0,346
10	0,329	0,301	0,316	0,380	0,342	-0,009	0,265
11	0,331	0,288	0,307	0,338	0,357	0,002	0,277
12	0,337	0,260	0,316	0,370	0,360	0,011	0,296
13	0,240	0,279	0,310	0,392	0,372	-0,007	0,298
14	0,292	0,265	0,314	0,355	0,350	0,035	0,303
15	0,256	0,274	0,319	0,344	0,351	-0,010	0,293
16	0,264	0,279	0,321	0,359	0,357	0,031	0,309
17	0,258	0,285	0,326	0,335	0,357	0,027	0,217
18	0,308	0,281	0,322	0,341	0,367	-0,106	0,167
19	0,271	0,284	0,332	0,322	0,365	-0,086	0,102
20	0,269	0,283	0,340	0,354	0,352	-0,141	0,322
21	0,244	0,288	0,338	0,350	0,363	-0,069	0,189
22	0,268	0,283	0,339	0,352	0,371	-0,111	0,282
23	0,269	0,275	0,316	0,333	0,371	-0,100	0,182
24	0,243	0,282	0,353	0,317	0,374	-0,019	0,266
25	0,263	0,279	0,281	0,344	0,351	-0,009	0,197

Berdasarkan Tabel 4.5 dapat dilihat bahwa kualitas *cluster* menggunakan algoritma *k-means* dengan perhitungan kemiripan *euclidean distance* pada metode filter DF, TC, TV, TVQ, MAD, MM, AMGM nilai rata-rata *silhouette* (ASW) tertinggi didapat pada $k = 2$ yaitu dengan kriteria nilai "cukup baik". Akan tetapi pada nilai k yang sesuai dengan *grount thruth* yaitu pada $k = 3$ nilai rata-rata *silhouette* (ASW) tidak jauh berbeda dari nilai $k = 2$.

Tabel 4. 6 Uji Coba Parameter Menggunakan *Cosine Similarity* – HAC

k	rata - rata *silhouette* (ASW) *cosine similarity* − HAC						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	0,104	0,082	0,260	0,107	0,204	0,102	**0,030**
3	0,004	-0,065	0,220	-0,042	0,384	-0,016	-0,014
4	-0,016	-0,004	0,129	-0,092	0,346	-0,025	-0,018
5	-0,037	-0,036	0,126	-0,094	0,426	-0,029	-0,022
6	-0,069	0,042	0,215	0,136	0,423	-0,030	-0,023
7	-0,011	0,174	0,258	0,117	0,436	-0,031	-0,036
8	0,105	0,186	0,145	0,283	**0,460**	-0,029	-0,037
9	0,106	0,260	0,184	0,323	0,387	-0,029	-0,033
10	0,225	**0,317**	0,211	0,338	0,369	-0,028	-0,042
11	0,274	0,252	0,235	**0,369**	0,351	-0,022	-0,033
12	0,285	0,252	0,303	0,348	0,270	-0,021	-0,037
13	0,299	0,244	0,295	0,360	0,278	-0,023	-0,036
14	0,309	0,256	0,301	0,359	0,271	-0,022	-0,034
15	0,346	0,243	0,303	0,319	0,273	-0,016	-0,025
16	0,352	0,246	0,302	0,306	0,278	0,176	-0,021
17	**0,358**	0,240	0,303	0,336	0,275	0,179	-0,016
18	0,358	0,237	**0,305**	0,350	0,272	**0,185**	-0,005
19	0,348	0,224	0,297	0,346	0,273	0,181	-0,001
20	0,343	0,220	0,275	0,344	0,282	0,174	-0,003
21	0,357	0,218	0,276	0,341	0,282	0,174	-0,009
22	0,347	0,220	0,280	0,343	0,283	0,173	-0,006
23	0,348	0,227	0,265	0,347	0,273	0,175	-0,013
24	0,331	0,219	0,272	0,352	0,273	0,179	-0,012
25	0,327	0,226	0,273	0,351	0,283	0,181	-0,007

Hasil uji coba dengan menggunakan algoritma HAC dapat dilihat pada Tabel 4.6 dan Tabel 4.7. Pada Tabel 4.6 merupakan hasil uji coba dengan menggunakan perhitungan kemiripan *cosine similarity,* dapat dilihat bahwa hasil nilai rata – rata

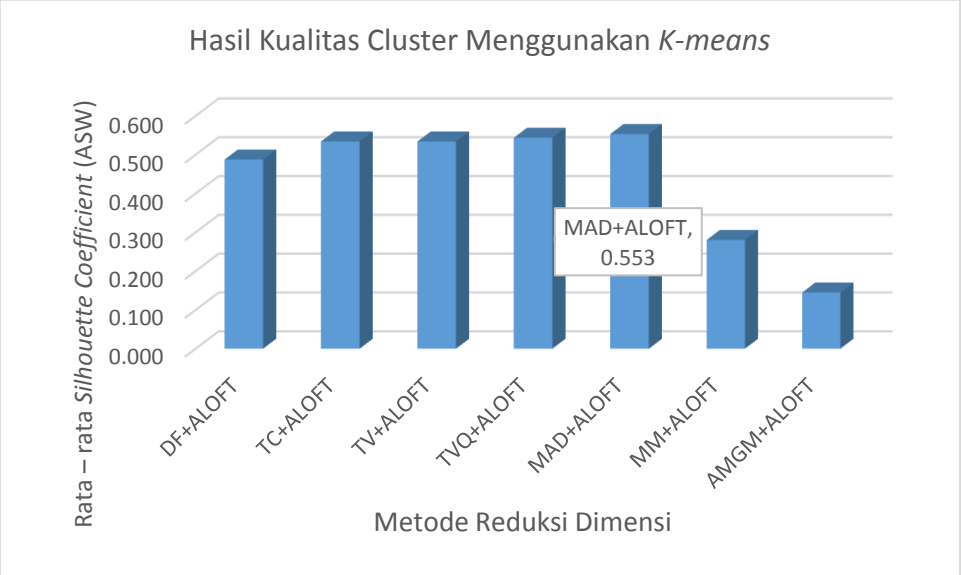
rata - rata *silhouette* (ASW) untuk semua metode filter terdapat pada *k* yang berbeda – beda. Hal ini karena proses pengelompokan menggunakan algoritma HAC menghasilkan kelompok dokumen yang berkumpul menjadi satu dan hanya beberapa dokumen yang terdapat di *cluster* yang berbeda. Sehingga didapatkan nilai *Average Silhouette Width* (ASW) yang rendah pada $k = 3$. Keseluruhan hasil pengelompokan dokumen yang terbentuk dapat dilihat pada Lampiran 2.

Tabel 4. 7 Uji Coba Parameter Menggunakan *Euclidean Distance* – HAC

k	Rata - rata S*ilhouette* (ASW) *Euclidean distance* − HAC						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	**0,253**	**0,250**	**0,208**	**0,700**	**0,232**	**0,663**	**0,505**
3	0,186	0,194	0,138	0,475	0,210	0,623	0,262
4	0,170	0,164	0,156	0,476	0,207	0,623	0,244
5	0,112	0,165	0,143	0,460	0,156	0,608	0,256
6	0,111	0,155	0,121	0,450	0,131	0,608	0,257
7	0,135	0,156	0,119	0,309	0,132	0,495	0,259
8	0,146	0,138	0,124	0,304	0,133	0,497	0,253
9	0,138	0,133	0,125	0,305	0,157	0,497	0,258
10	0,138	0,124	0,139	0,306	0,159	0,395	0,256
11	0,150	0,121	0,136	0,287	0,166	0,397	0,259
12	0,153	0,110	0,137	0,293	0,170	0,399	0,256
13	0,153	0,111	0,129	0,293	0,169	0,399	0,254
14	0,162	0,112	0,132	0,283	0,171	0,403	0,250
15	0,165	0,127	0,131	0,185	0,174	0,404	0,251
16	0,167	0,130	0,141	0,261	0,171	0,406	0,252
17	0,175	0,134	0,143	0,266	0,179	0,411	0,250
18	0,175	0,134	0,149	0,266	0,180	0,411	0,252
19	0,172	0,133	0,149	0,262	0,177	0,412	0,251
20	0,168	0,136	0,152	0,241	0,176	0,413	0,205
21	0,166	0,134	0,152	0,240	0,182	0,413	0,208
22	0,168	0,135	0,153	0,248	0,182	0,415	0,211
23	0,167	0,135	0,147	0,246	0,183	0,417	0,213
24	0,158	0,136	0,148	0,246	0,183	0,406	0,216
25	0,158	0,138	0,164	0,247	0,184	0,407	0,217

Berdasarkan Tabel 4.7 dapat dilihat bahwa pada semua metode filter memiiki nilai nilai rata-rata *silhouette* (ASW) tertinggi juga didapat pada $k = 2$. Metode filter DF dan TC memiliki nilai rata-rata *silhouette* (ASW) pada kriteria "cukup baik", metode filter TV dan MAD memiliki nilai rata-rata *silhouette* (ASW) pada kriteria "kurang baik", akan tetapi pada metode filter TVQ, MM, dan AMGM memiliki nilai nilai rata-rata *silhouette* (ASW) yang berada pada kriteria nilai "sudah baik". Dengan menggunakan perhitungan kemiripan *euclidean distance* baik menggunakan algoritma *k-means* maupun algoritma HAC didapatkan nilai ASW tertinggi pada $k = 2$. Hal ini karena *euclidean distance* menghitung kemiripan antar dokumen berdasarkan jarak antar dokumen sehingga untuk vektor yang *sparse* seperti pada vektor dokumen maka *euclidean distance* kurang efektif. Pada penelitian ini vektor baru hasil dari reduksi dimensi pada beberapa metode filter merupakan vektor yang *sparse* karena sebagian besar fitur akhir yang mewakili kseluruhan dokumen memiliki ciri khusus untuk setiap kategori. Berbeda dengan *cosine similarity* yang menghitung kemiripan dokumen berdasarkan perkalian *dot product* antar fitur yang mewakili dokumen dan hanya mempertimbangkan nilai fitur yang bukan nol sehingga *cosine similarity* lebih efisien untuk data yang *sparse*.

Berdasarkan hasil uji coba didapatkan bahwa dengan menggunakan perhitungan kemiripan dengan *cosine distance* dan menggunakan algoritma *k-means* dihasilkan kualitas *cluster* yang lebih sesuai dengan *ground truth* dimana nilai parameter k terbaik ada pada $k = 3$. Sehingga nilai rata – rata *silhouette* yang didapat dari hasil uji coba pada $k = 3$ selanjutnya digunakan untuk mencari metode filter mana yang cocok dengan Algoritma ALOFT dengan cara membandingkan nilai rata – rata *silhouette* dari masing – masing metode.



Gambar 4. 8 Grafik Perbandingan Kualitas Cluster Menggunakan *cosine similarity –K-means*

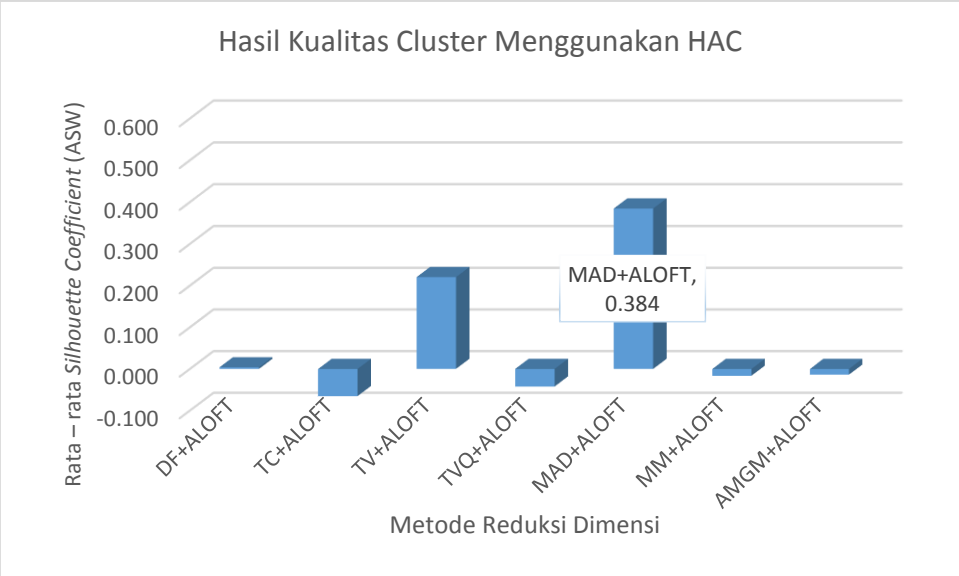
Tabel 4. 8 Himpunan Fitur Akhir

Metode Filter	Himpunan Fitur
DF	persen, indonesia, kuat, partai, gabung, dagang, poin, turun, posisi, main, harga, as, juara, nilai, hasil, rp, jakarta, capai, presiden
TC	persen, saham, rp, partai, presiden, poin, indonesia, ketua, kuat, juara, main, indeks, menang, hasil, calon, turun
TV	persen, rp, saham, partai, calon, juara, indonesia, main, ketua, kuat, nomor, serta, presiden, menteri, turun
TVQ	persen, saham, rp, partai, jokowi, juara, indonesia, main, ketua, kuat, indeks, serta, presiden, nomor, calon, turun
MAD	persen, rp, saham, partai, calon, juara, indonesia, ketua, kuat, main, indeks, menang, presiden, serta, turun

Berdasarkan Gambar 4.8 diketahui bahwa metode MAD + ALOFT memiliki nilai rata-rata *silhouette* (ASW) tertinggi dengan nilai ASW 0,553 dengan kriteria kualitas "Baik". Kriteria kualitaas *cluster* ini telah dijelaskan pada Sub-bab 2.9. Metode filter TC, TV, TVQ dan MAD memiliki nilai kualitas *cluster* yang tidak jauh berbeda hal ini karena himpunan fitur akhir yang dihasilkan oleh metode – metode ini memiliki kemiripan fitur. Tabel 4.8 merupakan hasil fitur dari metode filter TC, TV, TVQ, dan MAD telihat bahwa terdapat banyak sekali kemiripan antara himpunan fitur dari hasil metode satu dan lainnya. Sedangkan pada metode

filter MM dan AMGM nilai rata-rata *silhouette* (ASW) terdapat pada kriteria "kurang baik" karena terlalu banyaknya fitur yang dihasilkan oleh kedua metode filter ini, hasil keseluruhan himpunan fitur akhir dapat dilihat pada Lampiran 1.

Gambar 4.9 menunjukkan grafik perbandingan kualitas *cluster* dengan menggunakan algoritma HAC dan perhitungan kemiripan menggunakan *cosine similarity*. Dari grafik dapat dilihat bahwa nilai rata-rata *silhouette* (ASW) tertinggi dimiliki oleh metode MAD + ALOFT dengan nilai 0,384 dan masuk kedalam kriteria "cukup baik". Sedangkan untuk metode filter yang lain hanya memiliki kriteria "kurang baik" karena nilai rata-rata *silhouette* (ASW) dari metode filter tersebut kurang dari 0,26.



Gambar 4. 9 Grafik Perbandingan Kualitas Cluster Menggunakan *cosine similarity* −HAC

4.3.2 Uji Coba 2 : Pengujian Tanpa Pencarian Kata Dasar

Hasil dari uji coba 2 nantinya akan digunakan sebagai komparasi rata-rata *silhouette* (ASW) antara proses yang terlebih dahulu dilakukan pencarian kata dasar dengan proses yang tidak menggunakan kata dasar. Dengan komparasi ini nantinya akan diketahui pengaruh penggunaan Kateglo dalam pembentukan kata dasar terhadap kualitas *cluster* yang dihasilkan. Jumlah term / fitur yang terbentuk ketika dilakukan pencarian kata dasar adalah 12.045 fitur, sedangkan ketika proses

pencarian kata dasar tidak dilakukan terdapat penambahan fitur sebanyak 1.814 sehingga jumlah keseluruhan term ketika tidak menggunakan kata dasar adalah 13.859. Proses pengujian yang dilakukan sama seperti pengujian sebelumnya dimana fitur - fitur tersebut kemudian dilakukan perhitungan nilai relevansi dengan metode filter kemudian dilakukan pemilihan fitur akhir dengan menggunakan Algoritma ALOFT. Pada Tabel 4.9 merupakan jumlah fitur yang telah direduksi dengan masing – masing metode Filter kemudian dipilih sesuai dengan algoritma ALOFT.

Tabel 4. 9 Jumlah Fitur pada Masing - masing Metode Filter Tanpa Pencarian Kata Dasar

No.	Metode Filter	Jumlah Fitur
1	*Document Frequency* (DF) + ALOFT	23
2	*Term Contribution* (TC) + ALOFT	17
3	*Term Variance* (TV) + ALOFT	19
4	*Term Variance Quality* (TVQ) + ALOFT	20
5	*Mean Absolute Difference* (MAD) + ALOFT	16
6	*Mean Median* (MM) + ALOFT	180
7	*Arithmetic Mean Geometric Mean* (AMGM) + ALOFT	120

Pada pengujian dengan menggunakan algoritma *k-means* inisialiasi centroid yang digunakan adalah sama dengan centroid yang digunakan pada uji coba 1 dimana inisialisai centroid ini bertujuan menguragi resiko kualitas *cluster* yang buruk akibat pemilihan centroid secara acak. Akan tetapi pada uji coba 2 ini tidak dilakukan uji coba dengan menggunakan *euclidean distance* karena pada uji coba 1 dihasilkan bahwa *cosine similarity* memiliki performa yang lebih bagus sehingga hanya digunakan perhitungan kemiripan dengan *cosine similarity* untuk keseluruhan skenario pada uji coba 2.

Pada Tabel 4.10 merupakan hasil uji coba menggunakan algoritma *k-means clustering.* Dari tabel dapat dilihat bahwa pada metode filter DF nilai *k* terbaik pada *k*= 3 dengan nilai ASW 0,443 sedangkan pada metode filter TC nilai *k* terbaik dimiliki oleh *k* = 5. Pada metode filter TV, TVQ, MAD, memiliki kualitas cluster

terbaik pada k = 4. Meskpun nilai ASW terbaik tidak pada k = 3 (sesuai *ground truth*) tetapi nilai ASW antara k = 3 dengan k terbaik pada pengujian jauh berbeda. Dimana nilai ASW ini masih pada rentang kriteria yang sama yaitu "Cukup Baik". Akan tetapi pada metode filter MM dan AMGM memiliki karakteristik yang mirip dengan uji coba 1 yaitu jika semakin besar jumlah k maka nilai ASW juga akan bertambah.

Tabel 4. 10 Hasil Uji Coba Parameter k tanpa Kata Dasar Menggunakan *K-means*

k	Rata - rata *silhouette* (ASW) *K-means*						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	0,335	0,396	0,394	0,388	0,406	0,259	0,112
3	**0,443**	0,513	0,502	0,491	0,525	0,279	0,144
4	0,441	0,521	**0,509**	**0,499**	**0,533**	0,243	0,146
5	0,389	**0,543**	0,509	0,465	0,525	0,255	0,147
6	0,405	0,467	0,417	0,485	0,469	0,264	0,163
7	0,364	0,356	0,356	0,374	0,362	0,267	0,163
8	0,358	0,374	0,374	0,377	0,377	0,279	0,176
9	0,369	0,401	0,391	0,378	0,401	0,292	0,179
10	0,377	0,412	0,403	0,397	0,415	0,299	0,182
11	0,383	0,428	0,415	0,408	0,427	0,307	0,190
12	0,384	0,443	0,425	0,417	0,433	0,319	0,202
13	0,379	0,447	0,432	0,432	0,449	0,323	0,209
14	0,387	0,453	0,437	0,441	0,455	0,336	0,216
15	0,391	0,460	0,450	0,455	0,464	0,303	0,209
16	0,397	0,465	0,451	0,458	0,471	0,305	0,222
17	0,355	0,469	0,465	0,469	0,480	0,313	0,225
18	0,394	0,477	0,475	0,472	0,482	0,321	0,229
19	0,381	0,484	0,475	0,479	0,462	0,326	0,248
20	0,389	0,475	0,487	0,483	0,484	0,330	0,233
21	0,355	0,481	0,478	0,491	0,466	0,336	0,241
22	0,379	0,461	0,487	0,499	0,463	0,341	0,252
23	0,350	0,465	0,469	0,457	0,418	0,342	0,254
24	0,350	0,469	0,448	0,469	0,406	0,353	**0,259**
25	0,342	0,438	0,448	0,464	0,400	**0,357**	0,257

Pada uji coba dengan tanpa kata dasar menggunakan HAC didapatkan hasil yang memiliki karakteristik yang mirip dengan uji coba dengan kata dasar menggunakan algoritma HAC. Pada Tabel 4.11 dapat dilihat bahwa nilai k terbaik

untu masing – masing metode filter tidak dominan mengarah pada satu nilai *k* ataupun mengarah pada *k* = 3 sesuai *ground truth*.

Tabel 4. 11 Hasil Uji Coba Parameter *k* tanpa Kata Dasar Menggunakan HAC

k	Rata - rata *silhouette* (ASW) HAC						
	DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
2	0,014	0,168	0,129	0,118	0,231	**0,114**	**0,049**
3	-0,115	0,036	0,006	0,198	0,088	0,103	0,022
4	-0,140	0,002	0,008	0,109	0,082	0,103	-0,021
5	-0,149	0,013	-0,003	0,103	0,082	0,102	-0,036
6	-0,142	0,026	0,006	0,167	0,008	0,110	-0,037
7	-0,143	-0,023	0,046	0,171	0,192	0,107	-0,036
8	0,047	0,041	-0,024	0,215	0,152	0,080	-0,042
9	0,052	0,183	0,143	0,376	0,189	0,079	-0,036
10	0,157	0,285	0,139	0,382	0,362	0,035	-0,033
11	0,156	0,340	0,198	0,386	0,375	0,040	-0,036
12	0,155	0,351	0,205	0,395	0,399	0,030	-0,030
13	0,158	**0,384**	0,317	0,406	**0,407**	0,043	-0,034
14	0,190	0,372	0,331	0,445	0,386	0,041	-0,032
15	0,229	0,365	0,338	0,439	0,405	0,045	-0,034
16	0,233	0,384	0,381	0,441	0,310	0,045	-0,031
17	**0,290**	0,374	0,396	0,452	0,316	0,054	-0,025
18	0,274	0,377	**0,403**	**0,453**	0,311	0,053	-0,022
19	0,279	0,374	0,372	0,335	0,328	0,053	-0,016
20	0,282	0,373	0,373	0,325	0,332	0,053	-0,021
21	0,278	0,375	0,367	0,348	0,319	0,055	-0,023
22	0,268	0,373	0,368	0,342	0,324	0,057	-0,016
23	0,269	0,383	0,353	0,346	0,362	0,064	-0,008
24	0,271	0,380	0,350	0,350	0,365	0,065	-0,004
25	0,270	0,381	0,391	0,342	0,367	0,070	0,010

4.4 Analisa dan Pembahasan

Pada proses uji coba telah dilakukan pengujian parameter *k* dengan cara nilai *k* di ubah – ubah untuk mendapatkan hasil yang terbaik. Selanjutnya untuk menentukan kehandalan dari metode yang diusulkan maka dilakukan komparasi dengan metode lain. Pada penelitian ini metode usulan dilakukan komparasi dengan metode *variable ranking* (VR) pada *k* = 3. Selain itu juga dilakukan

komparasi kualitas *cluster* antara proses yang dilakukan pencarian kata dasar menggunakan kateglo dengan yang tanpa menggunakan kata dasar.

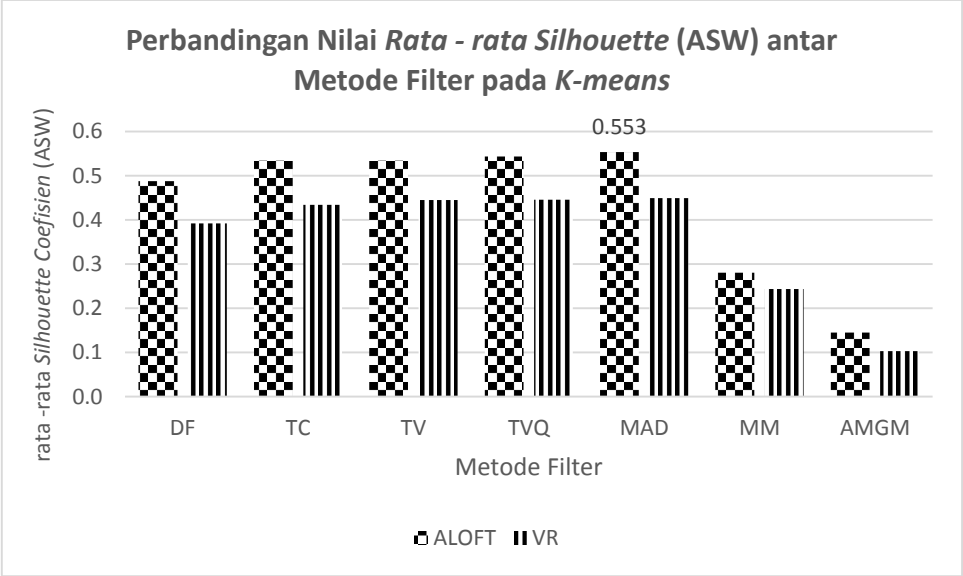
4.4.1 Perbandingan Pemilihan Fitur Menggunakan ALOFT dengan *n* Fitur Teratas (VR)

Pemilihan Fitur menggunakan algoritma ALOFT memiliki kelebihan dimana dengan menggunakan ALOFT pengguna tidak perlu memasukkan jumlah fitur yang ingin diproses. Berbeda dengan pemilihan fitur dengan metode *Variable Ranking* (VR) yang memilih fitur dengan cara memasukkan sejumlah *n* fitur teratas. nilai *n* ini menjadi sangat penting karena jumlah fitur yang berbeda mungkin akan menghasilkan kelompok dokumen yang berbeda. Karena harus melakukan estimasi jumlah *n* yang optimal metode VR menjadi sangat tidak efisien karena waktu yang akan dibutuhkan untuk memilih fitur yang tepat pasti akan lebih lama. Pada penelitian ini dilakukan perbandingan dengan metode VR untuk mengetahui kehandalan dari metode yang diusulkan. Komparasi dilakukan pada $k = 3$ dimana pada jumlah *cluster* ini adalah jumlah yang sesuai dengan *ground trurth* bahwa data yang digunakan terdiri dari tiga kategori.

Tabel 4. 12 Jumlah Fitur Menggunakan Metode VR

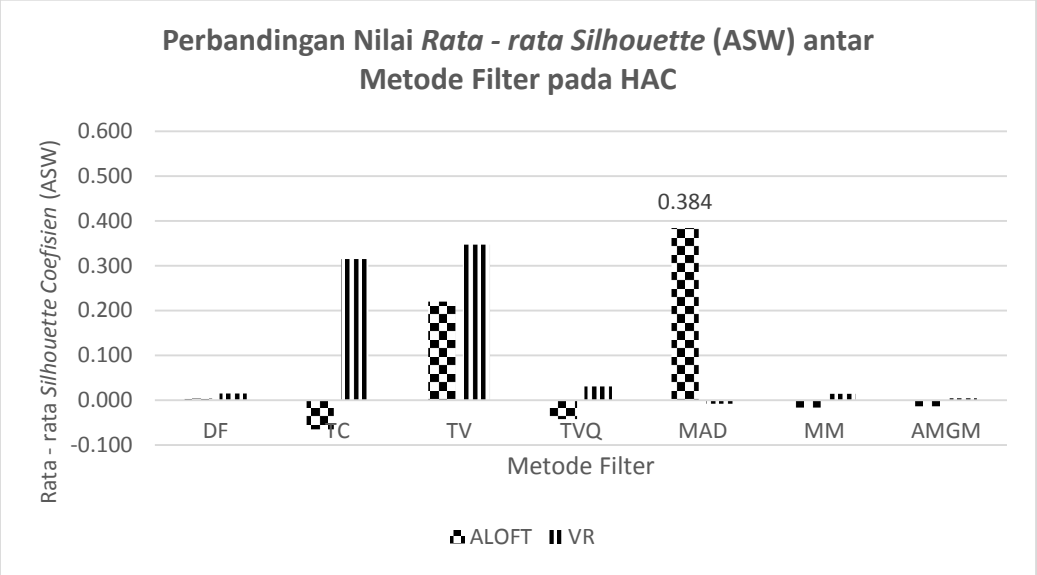
Metode	Jumlah Fitur	Jumlah Dokumen	Jumlah Fitur	Jumlah Dokumen
Document Frequency (DF) + VR	19	954	50	1000
Term Contributtion (TC) + VR	16	879	30	1000
Term Variance (TV) + VR	15	850	30	1000
Term Variance Quality (TVQ) + VR	16	931	30	1000
Mean Absolute Difference (MAD) + VR	15	997	30	1000
Mean Median (MM) + VR	168	856	450	1000
Arithmetic Mean Geometric Mean (AMGM) + VR	119	832	1150	1000

Jumlah fitur yang dihasilkan dari pemilihan fitur menggunakan metode *Variable Ranking (VR)* dapat dilihat pada Tabel 4.12 Terlihat pada Tabel 4.12 bahwa jumlah fitur untuk metode VR lebih banyak jika dibandingkan dengan metode ALOFT hal ini karena pada jumlah fitur yang sama dengan metode ALOFT, metode VR belum bisa mencapai keseluruhan dokumen. Masih terdapat beberapa dokumen yang tidak terwakili oleh fitur sehingga mengakibatkan kesulitan menentukan kemiripan dari dokumen yang tidak terwakili fitur tersebut. Oleh karena itu, jumlah fitur yang dipilih dengan metode VR adalah jumlah minimal untuk dapat mencakup keseluruhan dokumen.



Gambar 4. 10 Grafik Perbandingan Metode Usulan dengan VR pada *K-means*

Gambar 4.10 merupakan grafik perbandingan kualitas *cluster* dengan menggunakan algoritma *k-means*. Dimana pada uji coba menunjukan bahwa pemilihan fitur menggunakan variasi metode filter pada ALOFT dengan menggunakan algoritma *k-means* memiliki hasil yang lebih baik jika dibandingkan dengan algoritma HAC. Dari gambar dapat terlihat bahwa metode filter MM dan metode filter AMGM memiliki nilai *silhouette* yang lebih rendah diantara keseluruhan metode filter yang lain. Hal ini karena metode filter AMGM dan metode filter MM menghasilkan fitur yang lebih banyak dan terdapat fitur-fitur umum yang memungkinkan muncul di keseluruhan dokumen seperti fitur “sore”, “siang”, “isyarat”, “tiga”, “sisa”, “batas”, dan lain-lain.

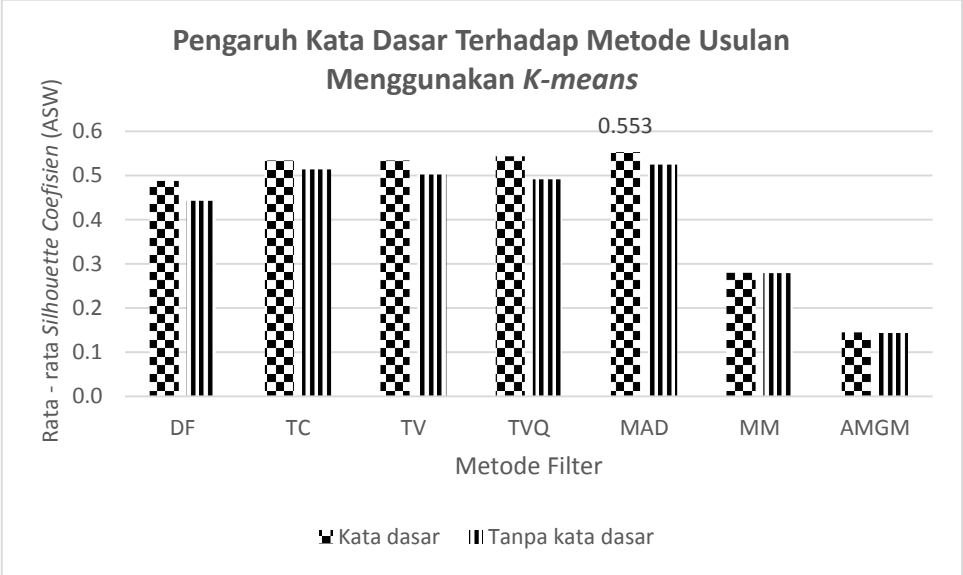


Gambar 4. 11 Grafik Perbandingan Metode Usulan dengan VR pada HAC

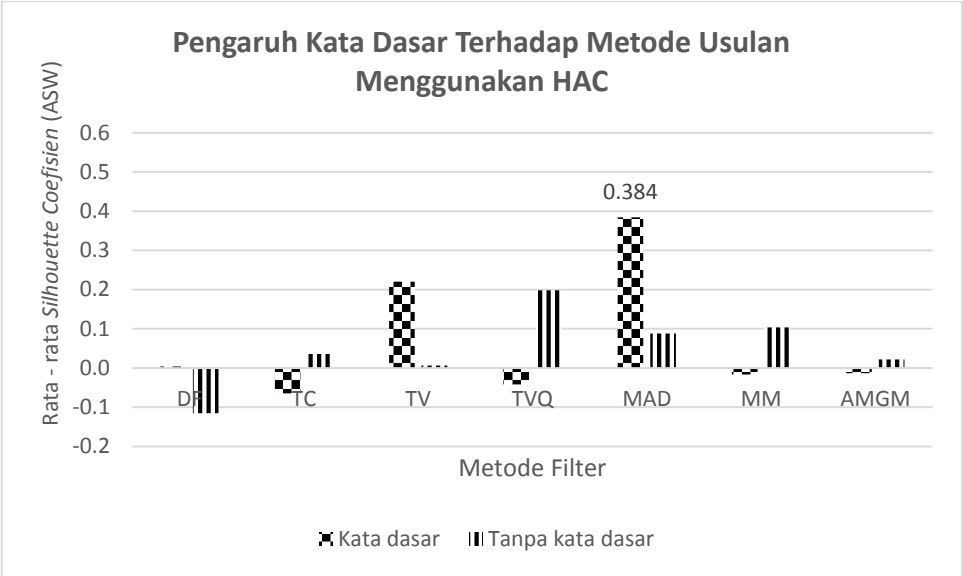
Pada Gambar 4.11 merupakan grafik perbandingan kualitas *cluster* dengan menggunakan algoritma HAC. Dengan menggunakan algoritma HAC hanya pada metode filter MAD yang memiliki nilai rata - rata silhouette yang lebih unggul dibandingkan metode VR. Pada metode filter yang lain metode VR memiliki nilai silhouette yang lebih tinggi dibandingkan metode usulan.

4.4.2 Pengaruh Penggunaan Kata Dasar

Penggunaan aplikasi Kateglo untuk mencari kata dasar diharapkan dapat memberikan hasil yang lebih baik terhadap kualitas dari cluster yang dihasilkan. Pada uji coba sebelumnya telah dipaparkan bahwa dengan pencarian kata dasar Jumlah fitur yang terbentuk ketika dilakukan pencarian kata dasar adalah 12.045 fitur, sedangkan ketika proses pencarian kata dasar tidak dilakukan terdapat dihasilkan fitur sebanyak 13.859. terdapat selisih 1.814 yang merupakan kata turunan.



Gambar 4. 12 Grafik Pengaruh Kata Dasar Terhadap Metode Usulan Menggunakan *K-means*



Gambar 4. 13 Grafik Pengaruh Kata Dasar Terhadap Metode Usulan Menggunakan HAC.

Berdasarkan Gambar 4.12 dan 4.13 telihat bahwa telihat bahwa penggunaan kata dasar dengan menggunakan Kateglo tidak berpengaruh secara signifikan terhadap kualitas cluster yang dihasilkan, bahkan pada metofe filter TVQ dan metode filter MM dari proses yang tanpa menggunakan kata dasar memiliki nilai rata-rata *silhouette* (ASW) yang lebih tinggi. Hal ini terjadi karena sejumlah 1.814

fitur yang merupakan kata turunan tidak termasuk ke dalam fitur – fitur yang memiliki nilai relevansi tinggi sehingga tidak terpilih oleh ALOFT sebagai himpunan fitur akhir.

Alasan lain yang mendasari nilai rata – rata *silhouette* yang dimiliki beberapa metode filter tidak berpengaruh signifikan adalah karena kata turunan yang ada pada Kateglo tidak mencakup keseluruhan kata berimbuhan yang ada pada bahasa Indonesia, terdapat beberapa kata berimbuhan yang tidak terdapat pada Kateglo seperti imbuhan dengan awalan di(-) dan akhiran (–i) misalkan kata "dipakai", "ditutupi", "dijalani", "dipadati", "dipanasi" dan lain-lain. Selanjutnya kata – kata yang memiliki imbuhan lebih dari satu dan juga akhiran lebih dari satu juga tidak terdapat di daftar kata turunan kateglo seperti kata "mempermainkannya", "memperjuangkannya", dan lain lain. Kata turunan kateglo juga tidak memiliki kata – kata berisisipan seperti "jelajah", "geligi", "selidik", "melaju", dan lain-lain. Kata yang memiliki akhiran (-i) juga tidak terdapat pada kata turunan Kateglo seperti kata "sukai", "tanami", "fasilitasi", dan lain lain. Hal ini berakibat pada tidak adanya perbedaan yang signifikan antara fitur term yang dilakukan pencarian kata dasar dengan fitur term yang tidak dilakukan pencarian kata dasar karena fitur term yang tidak ditemukan pada kata turunan ataupun *root phrase* maka term tersebut dianggap sudah berupa kata dasar.

4.3.3 Analisa Hasil Cluster

Data yang digunakan pada penelitian ini merupakan data dari situs berita online Kompas dengan kategori olahraga, politik, dan ekonomi. Jumlah masing – masing data yang digunakan adalah ekonomi 350 data, politik 350 data, dan olahraga 300 data dimana dapat dikatakan bahwa *ground truth* dari data yang digunakan ada pada $k = 3$. Sehingga untuk mengevaluasi kebenaran hasil cluster dari fitur yang telah direduksi pada penelitian ini digunakan metode evaluasi *adjusted rand index*. Tujuan lain dari uji coba menggunakan metode evaluasi *adjusted rand index* adalah untuk mengetahui apakah himpunan fitur akhir yang sudah terpilih dapat mewakili dokumen aslinya.

Hasil eksperiment menggunakan metode evaluasi *adjusted rand index* dapat dilihat pada Tabel 4.13 dimana proses evaluasi dilakukan pada hasil pengelompokan pada $k = 3$ sesuai dengan *ground truth* yang ada. Hasil *cluster* yang memiliki nilai *adjusted rand index* tertinggi untuk keseluruhan metode yang diusulkan adalah pada penggunaan *k-means* sebagai algortima *clustering* dan perhitungan kemiripannya dilakukan dengan menggunakan *cosine similarity.* Hasil ini membuktikan bahwa dengan reduksi dimensi menggunakan variasi metode filter pada algoritma ALOFT untuk pengelompokan dokumen memiliki hasil yang terbaik jika menggunakan algoritma *k-means* dan perhitungan kemiripan menggunakan *cosine similarity.*

Tabel 4. 13 Hasil *Adjusted Rand Index*

		DF + ALOFT	TC + ALOFT	TV + ALOFT	TVQ + ALOFT	MAD + ALOFT	MM + ALOFT	AMGM + ALOFT
K - means	*euclidean*	0,634	0,906	0,866	0,444	0,842	0,064	0,255
	cosine	**0,897**	**0,950**	**0,979**	**0,946**	**0,970**	**0,774**	**0,846**
HAC	*euclidean*	0,104	0,498	0,139	0,048	0,517	0,002	0,020
	cosine	0,013	0,007	0,147	0,024	0,592	0,019	0,010

Jika dilihat pada hasil Lampiran 2.A saat menggunakan algoritma *k-means* dan perhitungan kemiripan menggunakan *cosine similarity* pada filter DF, terlihat bahwa dokumen sudah terkelompok berdasarkan kemiripan *content* dari dokumen tersebut, hanya terdapat beberapa dokumen yang tidak terkelompok secara benar. Tabel 4.12 merupakan contoh beberapa judul berita dari dokumen yang terkelompok pada cluster 1. Dapat terlihat bahwa beberapa data yang tidak terkelompok dengan benar pada cluster 1 seperti pada dokumen 373 dan dokumen 779. Dokumen 373 dan dokumen 779 oleh sistem dideteksi terkelompok pada cluster 1 karena pada dokumen tersebut secara content memiliki kemiripan dengan dokumen – dokumen yang ada pada kategori ekonomi. Pada dokumen 373 membicarakan barang KW yang memang biasanya bahasan barang KW ada di kategori ekomoni, akan tetapi karena yang dibicarakan adalah barang milik pejabat dan pejabat erat hubungannya dengan politik maka dokumen tersebut *ground truth* masuk kategori politik. Sedangkan pada dokumen 779 membicarakan mengenai pembelian produk dimana berita jual beli lebih sering ada pada dokumen kategori

ekomoni, akan tetapi karena yang melakukan pembelian adalah seorang petinju maka dokumen tersebut *ground truth* masuk kategori olahraga.

Tabel 4. 14 Contoh Dokumen *Cluster* 1

Dokumen	Judul	Kategori
1	Terimbas Data Manufaktur China, IHSG Ditutup Melemah 2,29 Persen	ekonomi
110	Sempat Mix Jelang Penutupan, IHSG Berakhir Menguat 0,16 Persen	ekonomi
373	Ruhut Yakin Jam Tangan Setya Novanto Bukan Barang & ;KW& ;	politik
779	Mayweather Beli Mobil Super, Hanya Ada Dua di Dunia	olahraga

LAMPIRAN

LAMPIRAN 1A

A. Himpunan fitur akhir dengan menggunakan kata dasar

Metode Filter	Himpunan Fitur
DF	persen, indonesia, kuat, partai, gabung, dagang, poin, turun, posisi, main, harga, as, juara, nilai, hasil, rp, jakarta, capai, presiden
TC	persen, saham, rp, partai, presiden, poin, indonesia, ketua, kuat, juara, main, indeks, menang, hasil, calon, turun
TV	persen, rp, saham, partai, calon, juara, indonesia, main, ketua, kuat, nomor, serta, presiden, menteri, turun
TVQ	persen, saham, rp, partai, jokowi, juara, indonesia, main, ketua, kuat, indeks, serta, presiden, nomor, calon, turun
MAD	persen, rp, saham, partai, calon, juara, indonesia, ketua, kuat, main, indeks, menang, presiden, serta, turun
MM	rp, persen, short, morgan, stanley, euro, emas, david, semester, persentase, investasi, martin, rasio, sisa, syariah, persepsi, isyarat, pdb, vs, kuota, dubes, haji, golf, tanoe, golkar, reshuffle, pan, arie, ibas, mkd, kemenlu, dpr, video, kpk, busyro, hakim, obral, daerah, kabinet, tasikmalaya, tjahjo, effendi, pdip, istana, berkas, dki, polisi, abraham, pkpi, deklarasi, debat, istri, desa, pd, tb, kubu, suryadharma, korup, nasionalisme, aktivis, ani, iklan, komite, kategori, balikpapan, pasek, tunjuk, siar, dukung, risma, hary, malang, mahasiswa, gus, hambalang, kelompok, oligarki, press, ppp, dana, ramadhan, komunikasi, medali, tvri, kompasiana, gaya, tuhan, papua, meter, edi, keluarga, suami, jabar, bacaleg, daftar, ms, yamaha, le, miller, tour, rangking, ind, wei, yu, angga, lin, ihsan, della, wimbledon, stie, fu, rcv, serena, tommy, tinju, marsheilla, renang, button, berry, fran, taipei, dian, audisi, firman, lorenzo, cilic, jorgensen, stkip, linda, upi, ahsan, kevin, formula, khan, rousey, trofi, suzuka, lap, india, mattek, radwanska, abimanyu, beregu, vita, taiwan, marin, shixian, liliyana, frost, grup, nadal, menit, halep, azarenka, piala, federer, facebook, rs, mogensen, candra, sgs, yunus, latih, remaja, aprilia, sidik, ricky, kg
AMGM	ekonomi, rupiah, juta, kawasan, menteri, lawan, menit, dukung, pelemahan, short, media, indikator, siang, dana, emas, nomor, euro, tiongkok, wib, tiga, angkat, performa, david, tekan, aksi, ii, morgan, unggul, sumbang, batas, pasang, sore, bank, test, yunani, kondisi, ekuitas, sentimen, dibuka, temu, bayu, asing, naik, negatif, jam, sisi, sisa, syariah, uji, efek, cadang, beli, kpk, jepang, sektor, dubes, golf, golkar, dki, politik, sby, republik, shamsi, yg, pan,

	nama, hakim, busyro, daerah, mk, dahlan, pkpi, istri, gita, janji, marzuki, iklan, dino, capres, meter, orang, pd, rahmad, kuota, gp, yamaha, rangking, latih, angga, kategori, linda, abraham, final, tommy, button, serta, jessica, tasikmalaya, cilic, semarang, ubl, mercedes, kevin, tinju, filipina, jorgensen, le, tim, medali, raya, atlet, trofi, serena, suami, ratchanok, azarenka, taiwan, della, berry

LAMPIRAN 1B

B. Himpunan fitur akhir tanpa menggunakan kata dasar.

Metode Filter	Himpunan Fitur
DF	persen, indonesia, perdagangan, partai, poin, jakarta, asia, posisi, pekan, juara, as, wakil, nilai, mencapai, senin, turun, harga, dunia, rp, rabu, hasil, kali, presiden
TC	persen, saham, rp, partai, presiden, poin, indonesia, ketua, pemain, juara, peserta, as, indeks, turun, ganda, jakarta, calon
TV	persen, rp, saham, partai, calon, poin, indonesia, ketua, m, pemain, juara, nomor, peserta, menit, as, presiden, orang
TVQ	persen, saham, rp, partai, jokowi, poin, indonesia, ketua, pemain, juara, peserta, as, indeks, nomor, presiden, calon, jakarta
MAD	persen, rp, saham, partai, calon, juara, indonesia, ketua, pemain, peserta, indeks, presiden, final, as, poin, jakarta
MM	rp, persen, short, morgan, stanley, euro, emas, david, semester, persentase, investasi, martin, rasio, sisa, syariah, persepsi, pdb, vs, kuota, dubes, haji, golf, tanoe, golkar, reshuffle, pan, arie, ibas, individual, mkd, kemenlu, dpr, video, kpk, busyro, hakim, janji, tahapan, kabinet, tasikmalaya, tjahjo, effendi, pdip, revisi, istana, berkas, dki, abraham, pkpi, fahri, istri, kubu, desa, jokowi, pd, tb, suryadharma, korup, nasionalisme, aktivis, debat, ani, komite, kategori, dana, pln, iklan, balikpapan, pasek, didik, gaya, politik, risma, hary, malang, kelompok, mahasiswa, gus, hambalang, badai, oligarki, press, penyisihan, ppp, ramadhan, komunikasi, medali, tvri, daerah, sumbangan, pejabat, hubungan, ari, tuhan, papua, meter, edi, mandat, keluarga, suami, jabar, jaksa, kepengurusan, ms, bacaleg, yamaha, le, miller, tour, rangking, ind, wei, yu, lin, ihsan, della, wimbledon, stie, fu, serena, tommy, angga, marsheilla, berry, fran, taipei, dian, audisi, firman, lorenzo, cilic, jorgensen, stkip, linda, ahsan, penghargaan, formula, khan, sepak, rousey, ricky, trofi, suzuka, kevin, lap, india, mattek, radwanska, filipina, abimanyu, beregu, vita, taiwan, marin, shixian, liliyana, frost, grup, nadal, menit, halep, azarenka, piala, raya, facebook, rs, mogensen, candra, sgs, yunus, remaja, aprilia, federer, rcv, adik, latihan, kg,
AMGM	uang, rupiah, juta, pertemuan, kawasan, menteri, ekonomi, kebijakan, menit, dukungan, kondisi, pelemahan, short, media, indikator, siang, emas, ii, dana, tiongkok, ketiga, performa, david, penurunan, merah, morgan, putra, rakyat, sore, wib, euro, test, yunani, sentimen, bank, dibuka, bayu, asing, negatif, jam, sisa, syariah, pemerintah, efek, kpk,

	jepang, kenaikan, sektor, bunga, nomor, dubes, hubungan, golf, golkar, dki, pan, arie, sby, republik, shamsi, yg, hakim, busyro, daerah, janji, orang, mk, pkpi, dahlan, istri, politik, gita, marzuki, pemain, dino, peserta, capres, iklan, prinsip, meter, pd, kuota, gp, rangking, kategori, linda, final, angga, tommy, jessica, kecepatan, tasikmalaya, cilic, ubl, mercedes, kepala, yamaha, penghargaan, kevin, filipina, atlet, le, liliyana, raya, tim, trofi, semarang, medali, gelar, suami, halep, ratchanok, taiwan, della, berry, rcv

LAMPIRAN 2A

A. Contoh Hasil Cluster pada Metode Filter DF

Metode	Jumlah *k*	Hasil cluster
k-means , cosine similarity	*k* = 3	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

<table>
<tr><td></td><td></td><td>2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2</td></tr>
<tr><td>k-means ,
euclidean
distance</td><td>k = 3</td><td>3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 3, 1,
3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3,
3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 1, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1,
2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2,
2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1,
2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2,
2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1,
1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2,
2, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1,
2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1,
2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 2,
2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1,
2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 1,
1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2,
1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1,
2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1,</td></tr>
</table>

HAC , cosine similarity	$k = 3$	2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 1, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2
HAC , euclidean distance	$k = 3$	3, 3, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 2, 2, 3, 3, 2, 2, 3, 3, 3, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3,

		3, 2, 2, 3, 2, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 2, 3, 3, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 3, 2, 2, 3, 2, 2, 3, 3, 2, 2, 2, 3, 2, 3, 3, 3, 3, 2, 3, 2, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2, 3, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 2, 3, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 3, 2, 2, 3, 2, 3, 2, 2, 3, 2, 3, 2, 2, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 2, 3, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 3, 3, 2, 2, 3, 2, 2, 2, 3, 3, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 3, 3, 2, 3, 2, 3, 3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 3, 3, 2, 3, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 2, 2, 2, 2, 2, 3, 3, 2, 3, 2, 3, 3, 3, 2, 3, 3, 3, 2, 2, 2, 2, 2, 3, 3, 2, 3, 2, 3, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2

LAMPIRAN 2B

B. **Contoh Hasil Cluster pada Metode Filter TC**

Metode	Jumlah *k*	Hasil cluster
k-means , *cosine* *similarity*	*k* = 3	3, 1, 3, 1, 3, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

		2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
k-means , *euclidean distance*	$k = 3$	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 3

HAC , cosine similarity	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 3, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2
HAC , euclidean distance	*k* = 3	3, 2, 3, 2, 3, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 1, 1, 3, 3, 3,

		2, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 3, 3, 3, 2, 3, 3, 2, 2, 3, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 1, 3, 2, 3, 2, 3, 2, 1, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 1, 3, 2, 3, 2, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 1, 2, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 2, 3, 3, 1, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 3, 3, 2, 2, 3, 3, 3, 3, 3, 3, 2, 3, 1, 3, 3, 3, 3, 3, 2, 3, 1, 3, 3, 1, 2, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 2, 3, 1, 3, 2, 3, 2, 2, 3, 3, 3, 1, 1, 2, 3, 2, 2, 3, 2, 2, 3, 3, 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2

LAMPIRAN 2C

C. **Contoh Hasil Cluster pada Metode Filter TV**

Metode	Jumlah *k*	Hasil cluster
k-means , cosine similarity	*k* = 3	2, 3, 2, 3, 2, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

		1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
k-means , euclidean distance	*k* = 3	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 2, 2, 2, 2, 3, 3, 3, 1, 2, 3

HAC , cosine similarity	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 3, 2, 3, 3, 2, 2, 3, 1, 2, 3, 2, 3, 1, 2, 3, 3, 3, 3, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 3, 3, 3, 3, 2, 3, 2, 1, 3, 2, 2, 2, 2, 3, 3, 2, 3, 1, 2, 3, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 3, 3, 2, 3, 2, 3, 3, 3, 3, 1, 2, 3, 2, 3, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 2, 3, 2, 2, 3, 3, 3, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 3, 1, 2, 3, 3, 2, 1, 2, 2, 1, 2, 2, 3, 2, 2, 2, 3, 3, 3, 2, 3, 1, 1, 1, 3, 2, 2, 2, 2, 2, 3, 2, 3, 3, 1, 3, 2, 3, 1, 3, 2, 3, 2, 2, 3, 3, 2, 3, 3, 1, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 2, 2, 2, 2, 3, 2, 3, 3, 2, 1, 3, 2, 2, 2, 2, 2, 2, 1, 2, 3, 1, 2, 3, 3, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 1, 2, 1, 2, 1, 2, 3, 3, 2, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 3, 2, 2
HAC , euclidean distance	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

		2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 3, 3, 3, 2, 3, 3, 2, 2, 3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 3, 3, 3, 2, 3, 2, 3, 3, 3, 2, 2, 3, 2, 2, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 2, 2, 3, 3, 2, 2, 3, 3, 2, 2, 3, 2, 2, 3, 2, 3, 2, 3, 2, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3, 3, 3, 2, 3, 2, 3, 2, 2, 3, 3, 2, 2, 2, 3, 3, 3, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 2, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 2, 2, 3, 2, 2, 3, 3, 3, 2, 2, 2, 3, 3, 3, 2, 3, 2, 3, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

LAMPIRAN 2D

D. **Contoh Hasil Cluster pada Metode Filter TVQ**

Metode	Jumlah *k*	Hasil cluster
k-means , cosine similarity	*k* = 3	1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

		2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
k-means , euclidean distance	*k* = 3	2, 3, 2, 3, 2, 3, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 2, 3, 2, 3, 2, 3, 3, 3, 2, 3, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 1, 3, 2, 3, 3, 2, 3, 3, 2, 2, 3, 3, 3, 3, 3, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 1, 3, 3, 1, 2, 3, 3, 3, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 3, 2, 3, 1, 2, 3, 3, 2, 3, 2, 2, 3, 3, 2, 3, 3, 2, 3, 3, 2, 3, 3, 3, 2, 1, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 1, 3, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 2, 2, 2, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 2, 3, 2, 2, 3, 2, 2, 3, 3, 2, 3, 2, 2, 3, 3, 2, 2, 3, 3, 2, 3, 3, 2, 3, 2, 3, 3, 3, 2, 2, 2, 3, 2, 3, 2, 3, 3, 2, 2, 3, 3, 2, 2, 3, 1, 2, 3, 2, 2, 3, 3, 2, 2, 3, 2, 2, 3, 3, 2, 3, 3, 2, 3, 2, 3, 3, 2, 3, 3, 3, 1, 2, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

HAC , cosine similarity	$k = 3$	2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 1, 3, 2, 3, 3, 2, 3, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1, 3, 2, 2
HAC , euclidean distance	$k = 3$	1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 2, 1,

		2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

LAMPIRAN 2E

E. **Contoh Hasil Cluster pada Metode Filter MAD**

Metode	Jumlah *k*	Hasil cluster
k-means , cosine similarity	*k* = 3	2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 2, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

		1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
k-means , euclidean distance	*k* = 3	2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 3, 3, 1, 1, 3, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 3, 3, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 3, 3, 3, 1, 3, 1, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 3, 1, 3, 1, 1, 1, 1, 1, 1, 3, 2, 1, 3

HAC , cosine similarity	*k* = 3	2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 3, 2, 2, 3, 2, 3, 2, 3, 2, 3, 3, 3, 2, 2, 2, 3, 2, 2, 2, 3, 3, 2, 3, 3, 2, 3, 3, 3, 2, 3, 2, 2, 2, 2, 3, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 3, 2, 2, 2, 2, 3, 3, 2, 2, 2, 3, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 3, 2, 3, 2, 3, 3, 3, 2, 2, 2, 2, 3, 3, 2, 3, 3, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 2, 2, 3, 2, 3, 2, 3, 3, 3, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 2, 2, 2, 3, 2, 3, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 3, 3, 2, 2, 2, 2, 3, 2, 3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 3, 2, 3, 3, 2, 2, 3, 2, 2, 2, 3, 2, 3, 2, 3, 3, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 3, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2, 3, 3, 2, 3, 2, 2
HAC , euclidean distance	*k* = 3	2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1,

		1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 1, 1, 3, 3, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 1, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 1, 1, 3, 3, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 3, 3, 1, 3, 3, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, 3, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

LAMPIRAN 2F

F. **Contoh Hasil Cluster pada Metode Filter MM**

Metode	Jumlah *k*	Hasil cluster
k-means , cosine similarity	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, 1, 3, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 3, 1, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 1, 1, 3, 3, 3, 3, 1, 3, 1, 3, 3, 3, 1, 3, 3, 2, 3, 2, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 1, 3, 3, 3, 3, 3, 1, 3, 1, 3,

		1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3
k-means , euclidean distance	$k = 3$	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 3, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 3, 2, 3, 2, 3, 3, 3, 2, 3, 3, 3, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2

HAC , cosine similarity	*k* = 3	2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2
HAC , euclidean distance	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

		2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

LAMPIRAN 2G

G. **Contoh Hasil Cluster pada Metode Filter AMGM**

Metode	Jumlah *k*	Hasil cluster
k-means , *cosine* *similarity*	*k* = 3	3, 2, 3, 2, 3, 3, 3, 3, 1, 1, 2, 2, 2, 1, 3, 1, 1, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 2, 1, 1, 1, 3, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,

		2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2
k-means , euclidean distance	$k = 3$	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, 1, 3, 1, 1, 3, 3, 1, 1, 1, 3, 1, 3, 3, 3, 1, 3, 3, 1, 3, 1, 1, 3, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 3, 1, 1, 3, 1, 3, 3, 3, 3, 1, 3, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 3, 3, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 3, 1, 3, 1, 3, 3, 1, 1, 3, 1, 1, 3, 1, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 3, 1, 3, 3, 3, 1, 1, 1, 3, 1, 3, 3, 3, 1, 1, 3, 1, 3, 1, 1, 1, 1, 1, 3, 3, 3, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 3, 3, 1, 1, 1, 1, 1, 3, 1, 1, 3, 1, 1, 3, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1, 3, 3, 1, 3, 3, 1, 1, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 3, 3, 3, 3, 3, 1, 1, 1, 1, 3, 3, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 3, 3

HAC , cosine similarity	*k* = 3	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 3, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2
HAC , euclidean distance	*k* = 3	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

		1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1

BAB 5
Kesimpulan Dan Saran

5.1 Kesimpulan

Dalam Sub-bab ini akan dipaparkan kesimpulan yang dapat diambil berdasarkan serangkaian hasil pengujian serta analisa yang telah dilakukan terhadap metode yang diusulkan. Beberapa kesimpulan yang dapat diambil sebagai berikut :

1. Penggunaan produk Kateglo untuk proses pembentukan kata dasar dapat meningkatkan kualitas *cluster* pada beberapa metode filter, akan tetapi peningkatan kualitas *cluster* yang dihasilkan tidak terlalu signifikan.
2. Hasil uji coba pengelompokan dokumen berita online menunjukkan kualitas *cluster* pada nilai $k = 3$ memiliki kriteria "Baik" untuk filter TC, TV, TVQ, dan MAD dengan rata – rata *silhouette* lebih dari 0,5. Sedangkan untuk filter DF mimiliki kriteria "Cukup Baik" dengan rata – rata *silhouette* lebih dari 0,4.
3. Hasil uji coba pengelompokan dokumen berita online menunjukkan bahwa metode reduksi dimensi fitur menggunakan variasi metode filter pada ALOFT mendapatkan hasil yang optimal dengan menggunakan algoritma *k-means* dan Perhitungan kemiripan *cosine similarity.*

5.2 Saran

Saran yang dapat digunakan untuk pengembangan penelitian selanjutnya adalah penggunaan kombinasi dari beberapa metode filter sehingga nilai relevansi dari sebuah term tidak hanya tergantung pada satu metode filter saja.

(halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Aggarwal, C. C., & Zhai, C. (2012). *Mining text data.* Springer Science & Business Media.

Alelyani, S., Tang, J., & Liu, H. (2013). Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications, 29*.

Arifin, A. Z., Mahendra, I. P., & Ciptaningtyas, H. T. (2009). *Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language*. International Conference on Information & Communication Technology and Systems.

Bellot, P., & El-Bèze, M. (1999). *A clustering method for information retrieval.* France: Laboratoire d'Informatique d'Avignon.

Berry, M. W. (2004). Survey of text mining. *Computing Reviews, 45*(9), 548.

Bharti, K. K., & Singh, P. K. (2014). A three-stage unsupervised dimension reduction method for text clustering. *Journal of Computational Science, 5*(2), 156-169.

Bharti, K. K., & Singh, P. K. (2015). Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. *Expert Systems with Applications, 42*, 3105–3114.

Chen, C.-L., Tseng, F. S., & Liang, T. (2010). An integration of WordNet and fuzzy association rule mining for multi-label document clustering. *Data & Knowledge Engineering, 69*(11), 1208-1226.

Chen, C.-L., Tseng, F. S., & Liang, T. (2010). Mining fuzzy frequent itemsets for hierarchical document clustering. *Information processing & management, 46*(2), 193-211.

Dhillon, I., Kogan, J., & Nicholas, C. (2004). Feature selection and document clustering. (pp. 73-100). New York: Springer .

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data.* Cambridge University Press.

Ferreira, A. J., & Figueiredo, M. A. (2012). Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters, 33*(13), 1794-1804.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques.* Elsevier.

Hu, G., Zhou, S., Guan, J., & Hu, X. (2008). Towards effective document clustering: A constrained K-means based approach. *Information Processing & Management, 44*(4), 1397-1409.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR) , 31*(3), 264-323.

Kateglo. (2009). Retrieved Desember 2, 2015, from http://kateglo.com/?mod=doc&doc=README.txt

Kaufman, L., & Rousseeuw, P. (1987). Clustering by means of medoids. *Statistical Data Analysis Based on the L1 Norm and Related Methods* , 405-416.

Kummamuru, K., Dhawale, A., & Krishnapuram, R. (2003). Fuzzy co-clustering of documents and keywords. Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on.

Liu, H., Motoda, H., & eds. (2007). *Computational methods of feature selection.* CRC Press.

Liu, L., Kang, J., Yu, J., & Wang, Z. (2005). A comparative study on unsupervised feature selection methods for text clustering. (pp. 597-601). IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge: Cambridge university press.

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter, 6*(1), 90-105.

Pinheiro, R. H., Cavalcanti, G. D., Correa, R. F., & Ren, T. I. (2012). A global-ranking local feature selection method for text categorization. *Expert Systems with Applications, 39*(17), 12851-12857.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, 53-65.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*, 613–620.

Song, W., & Park, S. C. (2009). Genetic algorithm for text clustering based on latent semantic indexing. *Computers & Mathematics with Applications, 57*(11), 1901-1907.

Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence, 32*, 112-123.

Tala, F. Z. (2003). *A study of stemming effects on information retrieval in Bahasa Indonesia.* Institute for Logic, Language and Computation Universeit Van Amsterdam.

Zhao, W., He, Q., Ma, H., & Shi, Z. (2012). Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge and information systems, 30*(2), 569-587.

Zhao, Y., Karypis, G., & Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data mining and knowledge discovery, 10*(2), 141-168.

(halaman ini sengaja dikosongkan)

BIODATA PENULIS

Penulis lahir di Kediri tanggal 6 Februari 1990, merupakan anak pertama dari dua bersaudara dari pasangan Fatah dan Umikah. Penulis menempuh pendidikan formal dari SD Negeri Kandat 1 (1996-2002), MTs Negeri 2 Kediri (2002-2005), SMA Negeri 8 Kediri (2005-2008), dan Penulis menyelesaikan S1 di Program Studi Informatika Universitas Brawijaya Malang (2009-2014).

Selama perkuliahan S2, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV) karena tertarik dengan topik penelitian Data Mining dan Text Mining. Penulis dapat dihubungi melalui email: mamluatul.haniah@gmail.com