ANALISIS KOMPARATIF KINERJA ALGORITMA *LATENT SEMANTIC INDEXING* DAN *K-MEANS* DALAM MENGELOMPOKAN DOKUMEN TEKS PENDEK

FITRIANTO ADI SAPUTRO

5235117154

Skripsi ini Ditulis untuk Memenuhi Sebagian Persyaratan dalam Memperoleh Gelar Sarjana Pendidikan

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS NEGERI JAKARTA

2016

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
Widodo, M.Kom	.....	25-01-2016
(Dosen Pembimbing I)		
Bambang Prasetya Adhi, S.Pd., M.Kom	.....	25-01-2016
(Dosen Pembimbing II)		

PENGESAHAN PANITIAN UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Dr. Yuliatri Sastrawijaya, M.Pd	.....	25-01-2016
(Ketua Penguji)		
Drs. Bachren Zaini, M. Pd	.....	25-01-2016
(Sekretaris Penguji)		
M. Ficky D, M.Sc	.....	25-01-2016
(Dosen Ahli)		

Tanggal Lulus: 22-01-2016

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi yang telah saya buat berjudul "Analisis Komparatif Kinerja Algoritma *Latent Semantic Indexing* dan *K-Means* Dalam Mengelompokan Dokumen Teks Pendek" :

1. Asli dan belum pernah diajukkan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lainnya.
2. Murni gagasan dari penjelasan saya sendiri, rumusan dari penelitian saya sendiri dengan arahan dosen pembimbing.
3. Tidak terdapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sumber yang jelas.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila ditemukan hal-hal yang tidak sesuai dengan isi pernyataan ini, saya bersedia sanksi akademik sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2015

Yang Membuat Pernyataan

Fitrianto Adi Saputro

5235117154

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmatnya dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan baik yang berjudul "Analisis Komparatif Kinerja Algoritma *Latent Semantic Indexing* dan *K-Means* Dalam Mengelompokan Dokumen Teks Pendek".

Penulis menyadari bahwa skripsi ini jauh dari sempurna, sehingga penulis membutuhkan kritik dan saran yang bersifat membangun. Serta dalam penulisan ini penulis banyak diberikan bantuan dari berbagai pihak.

Oleh karena itu, izinkan penulis untuk mengucapkan terima kasih kepada seluruh pihak yang berpengaruh terhadap penulis selama menyusun skripsi ini, yaitu :

1. Bapak Widodo, M.Kom selaku Dosen Pembimbing I dan Bapak Bambang P. Adhi, M.Kom selaku Dosen Pembimbing II yang telah banyak meluangkan waktu untuk membimbing, memotivasi, mengarahkan, serta memberi nasihat yang berharga kepada penulis sampai terselesaikannya skripsi ini.
2. Kedua orang tua, kakak, adik, dan saudara atas segala do'a, semangat, dukungan, dan kasih sayangnya yang telah diberikan kepada penulis.
3. Para sahabat dan teman-teman yang telah memberikan do'a, bantuan, semangat, serta dukungan kepada penulis.

Semoga segala bantuan yang diberikan, sebagai amal soleh senantiasa mendapatkan ridho Allah SWT. Sehingga pada akhirnya skripsi ini dapat bermanfaat.

Bekasi, Januari 2015

Fitrianto Adi Saputro

ANALISIS KOMPARATIF KINERJA ALGORITMA *LATENT SEMANTIC INDEXING* DAN *K-MEANS* DALAM MENGELOMPOKAN DOKUMEN TEKS PENDEK

FITRIANTO ADI SAPUTRO

ABSTRAK

Di era yang serba cepat ini, pertumbuhan data digital akan semakin cepat dan akhirnya banyak penggunaan teks pendek dalam dunia digital, akibatnya dokumen tersebut menjadi tidak terorganisir, terlebih lagi dokumen teks pendek biasanya sulit di kelompokan karena sering menyebabkan ambiguitas. Kebutuhan analisis teks sangat diperlukan dalam menangani masalah tersebut. Agar dokumen teks pendek dapat terorganisir kembali maka diperlukan pengelompokan data dengan cara *text mining*. Penelitian ini dilakukan di Jurusan Teknik Elektro Universitas Negeri Jakarta dimulai dari bulan Mei 2015 hingga Desember 2015. Tujuan dari penelitian ini adalah untuk mengetahui algoritma mana yang lebih baik antara *LSI* dan *K-means* dalam melakukan pengelompokan dokumen teks pendek. Pengujian dalam peneltian ini menggunakan metode eksperimen, data teks pendek yang digunakan berasal dari *tweets* akun *twitter* @detik, yang diambil dari tanggal 7 Oktober 2015 hingga 21 Oktober 2015. Hasil dari pengolahan data kedua algoritma akan dibandingkan tingkat akurasinya. Untuk menghitung akurasi algoritma tersebut diperlukan perhitungan *Confusion Matrix*. Dari hasil pengujian, menunjukkan bahwa dengan menggunakan *K-Means Clustering* akurasinya selalu lebih tinggi dibandingkan *LSI*. Algoritma *K-Means* memiliki akurasi 49% hingga 51%, sedangkan *LSI* memiliki keakurasian antara 15% hingga 40%. Dan dapat ditarik kesimpulan bahwa Algoritma *K-Means* lebih baik dari pada algoritma *LSI* dalam pengelompokan dokumen teks pendek.

Kata kunci : *LSI*, *K-means*, Pengelompokan Teks , Dokumen Teks Pendek, *Text Mining*

COMPARATIVE ANALYSIS OF PERFORMANCE LATENT SEMANTIC INDEXING AND K-MEANS IN A SHORT TEXT DOCUMENT CATEGORIZATION

FITRIANTO ADI SAPUTRO

ABSTRACT

In this fast-paced world, the growth of digital data will be faster and ultimately a lot of use of short texts in the digital world, as a result of the document becomes disorganized, especially short text documents are usually difficult grouped as they often lead to ambiguity. Text analysis needs is indispensable in addressing such issues. In order for the short text documents can be organized back then required grouping data by text mining. This research was conducted in the Department of Electrical Engineering, State University of Jakarta starting from May 2015 to December 2015. The purpose of this study was to determine which one is better algorithms between LSI and K-means clustering of text documents in conducting short. Testing in this research used experimental method, short text data used comes fromdetik tweets twitter account, taken from the date of October 7, 2015 to October 21, 2015. The results of the second data processing algorithms will be compared to the level of accuracy. To calculate the required calculation algorithm accuracy Confusion Matrix. From the test results, showed that by using the K-Means Clustering accuracy is always higher than the LSI. K-Means algorithm has an accuracy of 49% to 51%, while LSI has accuracy between 15% to 40%. And it can be concluded that the K-Means algorithm is better than the LSI algorithm in a short text document grouping.

Keywords : LSI, K-means, Clustering Document , Short Text Document, Text Mining

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan informasi teks digital telah tumbuh sangat cepat. Bisa dilihat saat ini terlalu banyak teks digital. Karena terlalu banyaknya penggunaan teks digital, maka dokumen tersebut menjadi tidak terorganisisir. Kebutuhan analisis teks sangat diperlukan dalam menangani teks yang tidak terorganisir tersebut. Agar dokumen dapat terorganisir kembali maka diperlukan pengelompokan data dengan cara *text mining*. Pengelompokan data dengan *text mining* saat ini memiliki berbagai cara pendekatan antara lain pendekatan *probabilistic, support vector machine, artificial neural network,* atau *decision tree classification, clustering.*

Sering kali ditemukan bahwa teks pendek menjadi objek vital dari sebuah topik pembahasan, misalnya saja adalah judul sebuah berita, memang judul berita terkesan pendek, akan tetapi judul itu biasanya diambil dari inti dari sebuah berita.

Menurut Timonen, dkk.(2013: 130-146), diacu dalam Dini (2015: 7) mendefinisikan dokumen pendek sebagai dokumen yang berisi tidak lebih dari 100 kata, yang sama dengan abstrak ilmiah yang sangat singkat. Penelitian ini akan menggunakan data *tweets* dari twitter. *Tweets* pada *twitter* bisa digolongkan pada dokumen teks pendek, dimana *twitter* memiliki panjang 140 karakter, *tweets* sering kali mirip dengan dokumen teks yang ditemukan sehari-hari misalkan judul berita, pesan singkat atau sms, kutipan penting.

Mungkin saat ini pengelompokan teks sudah sering dilakukan pada dokumen yang panjang, namun melakukan *text mining* dalam mengelompokan teks pada

dokumen yang pendek masih jarang, hal ini dikarenakan sulitnya dalam melakukan *text mining* pada dokumen teks pendek. Padahal pengelompokan teks pendek dapat membantu pengguna agar tidak kewalahan dengan membaca informasi, misal *tweet* dalam kondisi yang masih acak atau membantu mempercepat dalam sistem pencarian. Menurut Dini (2015: 7) Tidak seperti dokumen, teks pendek memiliki beberapa karakteristik unik yang membuatnya sulit untuk ditangani. Pertama, teks pendek tidak selalu memperhatikan sintaks dari tulisan. Kedua, teks pendek memiliki konteks yang terbatas. Mayoritas permintaan dalam pencarian menggunakan teks pendek berisi kurang dari 5 kata.

Permasalahan lain yang muncul adalah seberapa banyak dokumen yang dibutuhkan agar saat melakukan *clustering* dokumen dapat memberikan akurasi yang maksimal. Apabila jumlah dokumen yang digunakan terlalu sedikit, maka tidak akan menghasilkan tingkat akurasi yang maksimal (Lim, dkk., 2007:690-700, diacu dalam Basnur dan Sensuse, 2010:2). Misalkan dalam sebuah dokumen teks pendek dominan mengandung kata komputer, akan tetapi apakah yakin dokumen tersebut masuk kedalam kategori bahasan komputer? Sepertinya belum tentu, karena data yang dimiliki untuk diolah sangat sedikit, sehingga sulit memprekdisikan data tersebut. Sehingga perlu dicari tahu algoritma yang tepat dalam melakukan pengelompokan dokumen teks pendek. Salah satu algoritma yang akan digunakan pada penelitian ini adalah algoritma *LSI* dan *K-Means*.

Latent Semantic Indexing (*LSI*) adalah salah satu algoritma yang digunakan untuk melakukan *indexing* dengan menggunakan prinsip similaritas. Algoritma *LSI* ini sering digunakan pada bidang *information retrieval*, namun kali ini akan diuji coba kinerjanya dalam mengelompokan dokumen teks pendek. Dimana diketahui

bahwa *LSI* digunakan untuk algoritma *searching*, namun disisi lain keuntungannya dari *LSI* adalah *query* yang diproses dalam sistem pencarian biasanya adalah termasuk golongan dokumen teks pendek, sehingga secara karakteristik bisa dikatakan cocok dalam penelitian dokumen teks pendek. Meskipun begitu *LSI* juga belum diketahui hasilnya untuk pengelompokan teks pendek.

K-Means merupakan salah satu metode data clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*/kelompok. Metode ini mempartisi data ke dalam *cluster*/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. (Agusta, 2007:47)

Jadi pada penelitian ini akan membandingkan keakuratan *LSI* dan *K-means* dalam mengelompokan dokumen teks pendek, sehingga dapat diketahui mana yang algoritma yang lebih baik untuk digunakan dalam pengelompokan dokumen teks pendek.

1.2. Identifikasi Masalah

Dengan mengacu pada latar belakang masalah di atas, maka permasalahan yang akan dibahas dan diteliti adalah:

1. Terlalu banyak data dokumen teks pendek digital yang tidak terorganisir
2. Belum diketahuinya algoritma yang efektif untuk mengelompokan dokumen teks pendek
3. Keakurasian *text mining* pada dokumen teks pendek yang sulit diprediksi
4. Belum diketahui mana yang lebih baik antara *LSI* dan *K-Means* dalam pengelompokan data

1.3. Batasan Masalah

Pembatasan masalah yang diberikan penulis pada skripsi ini:

1. Dokumen teks pendek yang digunakan hanya berasal dari *twitter* dengan maksimal 140 karakter
2. Dokumen teks yang digunakan untuk pengelompokan menggunakan Bahasa Indonesia
3. Data yang digunakan berasal dari akun *twitter* @detik
4. Data yang digunakan adalah data yang memiliki kategori sport, oto, inet, dan food pada periode 7 Oktober 2015 hingga 21 Oktober 2015.
5. Pengujian hanya menjawab algoritma yang baik antara *LSI* dan *K-means* dalam pengelompokan teks pendek

1.4. Perumusan Masalah

Berdasarkan latar belakang, maka perumusan masalah yang akan dibahas dalam penelitian ini adalah:

"Algoritma manakah yang paling baik antara *LSI* dan *K-means* dalam mengelompokkan dokumen teks pendek?"

1.5. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui algoritma mana yang lebih baik antara *LSI* dan *K-means* dalam melakukan pengelompokan dokumen teks pendek.

1.6. Kegunaan Penelitian

Penelitian ini diharapkan dapat bermanfaat, diantaranya yaitu:

1. Dari hasil penelitian ini akan terlihat seberapa besar kinerja antara *LSI* dan *K-means* dalam mengelompokan dokumen teks pendek, dan dapat diketahui mana yang lebih baik dalam melakukan pengelompokan dokumen teks pendek.
2. Penelitian ini menjadi kontribusi keilmuan yang dapat dipelajari di dalam perkuliahan, guna dapat dikembangkan dalam pemanfaatan segala bidang yang memanfaatkan *text mining* dokumen pendek

BAB II
KAJIAN TEORI DAN KERANGKA BERFIKIR

2.1. *Twitter*

Salah satu cara untuk menggambarkan *Twitter* adalah sebagai layanan *microblogging* yang memungkinkan orang untuk berkomunikasi dengan singkat, pesan dengan 140 karakter yang mungkin sesuai dengan pemikiran atau ide penggunanya. Dalam hal itu, dapat dikatakan bahwa *Twitter* itu gratis, mempunyai kecepatan yang tinggi, dan layanan pesan teks mendunia. Hal ini memungkinkan setiap penggunanya dapat berkomunikasi secara mudah dan cepat dalam berbagi dan mencari sebuah informasi. Model hubungan *Twitter* memungkinkan untuk mengetahui keadaan terbaru dari pengguna lain meskipun pengguna lainnya mungkin tidak memilih untuk mengikuti pengguna lainnya. Model "*Twitter's following*" cukup sederhana namun mengeksploitasi aspek fundamental dari apa yang membuat pengguna mengeluarkan rasa ingin tahunya. Seperti berita tentang topik olahraga, kesehatan serta dalam topik politik tertentu, atau keinginan untuk berhubungan dengan seseorang yang baru, *Twitter* memberikan kesempatan tidak terbatas untuk memenuhi rasa ingin tahu. *Twitter* dapat dikatakan real-time, layanan *microblogging* yang sangat sosial memungkinkan penggunanya untuk memperbaharui sebuah status pendek disebut dengan *tweet*, yang dapat muncul di timeline. *Tweet* adalah inti dari *Twitter*, dan pendapat lain mengatakan tweet sebagai konten teks yang terdiri dari 140 karakter dan terkait dengan status *update* dari penggunanya, kurang lebih ada sedikit metadata yang dapat dilihat dari *tweet* karena hanya memuat 140 karakter (Russell, 2014: 7 – 10).

2.2. Dokumen

Dokumen menurut bahasa inggris berasal dari kata *document* yang memiliki arti suatu yang tertulis atau tercetak dan segala benda yang mempunyai keterangan-keterangan dipilih untuk di kumpulkan, disusun, disediakan atau untuk disebarkan.

2.2.1. Dokumen Teks Pendek

Dokumen teks pendek sering muncul dimanapun, bahkan terkadang menjadi hal utama dari sebuah dokumen, seperti judul dokumen, kutipan, kata-kata penting, namun didalam *text mining* sifat dokumen teks pendek sangat berbeda dengan dokumen.

Menurut Timonen, dkk. (2013: 130-146), diacu dalam Dini (2015: 7) mendefinisikan dokumen pendek sebagai dokumen yang berisi tidak lebih dari 100 kata, yang sama dengan abstrak ilmiah yang sangat singkat.

Tidak seperti dokumen, teks pendek memiliki beberapa karakteristik unik yang membuatnya sulit untuk ditangani. Pertama, teks pendek tidak selalu memperhatikan sintaks dari tulisan. Kedua, teks pendek memiliki konteks yang terbatas.

Karena *tweets* pada *twitter* hanya memiliki tidak lebih dari 140 karakter. Dengan demikian, teks pendek biasanya tidak memiliki perhitungan yang cukup untuk menunjang teknik pengolahan pengelompokan teks. Karena alasan tersebut, teks pendek menimbulkan sejumlah ambiguitas yang signifikan. Dengan popularitas teks pendek, beberapa karya yang harus muncul dalam literatur untuk mempelajari isu-isu representasi teks pendek untuk memfasilitasi pengelompokan teks pendek dan klasifikasi.

2.3. *Text Mining*

Menurut Brown (2014: 388) *text mining* adalah teknik *data mining* yang diterapkan ke dalam teks. Karena ini bergantung pada analitik dasar yang pendekatannya sama dengan analisis teks, *text mining* identik dengan analisis teks, dan terutama penggunaan istilah *mining* adalah soal gaya dan konteks.

Perbedaan mendasar antara *text mining* dan *data mining* terletak pada sumber data yang digunakan. Pada *data mining*, pola-pola diekstraksi dari basis data yang terstruktur, sedangkan di *text mining*, pola-pola diekstraki dari data tekstual (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks ditulis untuk dibaca langsung oleh manusia (Hearst, 2015).

Jadi *text mining* adalah sebuah teknik *data mining* yang diterapkan kedalam data tidak terstruktur khususnya data teks, proses *text mining* yang khas meliputi kategorisasi teks, *text clustering*, ekstraksi konsep/entitas, produksi taksonomi granular, *sentiment analysis*, penyimpulan dokumen, dan pemodelan relasi entitas (yaitu, pembelajaran hubungan antara entitas bernama).

2.3.1. *Text Preprocessing*

Tahapan awal dari *text mining* adalah *text preprocessing* yang bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahapan berikutnya.

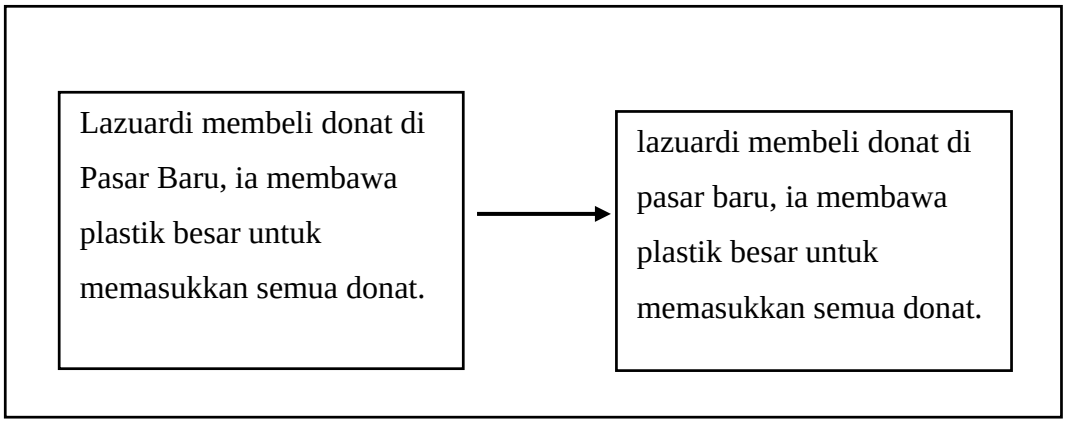
Menurut Adiwijaya, Igg (2006) diacu dalam Langgeni, dkk. (2010: 2), Teks yang akan dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terdapat *noise* pada data, dan terdapat struktur teks yang tidak baik.

Ada beberapa tahapan *preprocessing* yang dilakukan secara umum dalam *text mining* pada dokumen, yaitu *case folding*, *tokenizing*, *filtering* dan *stemming*.

Contoh jika kita memiliki kalimat “Lazuardi membeli donat di Pasar Baru, ia membawa plastik besar untuk memasukkan semua donat.”. Maka menurut Langgeni, dkk. (2010: 2) langkah langkah dalam text processing adalah sebagai berikut:

1. *Case folding*

Tahap mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. Contohnya seperti yang tertera pada gambar 2.1.

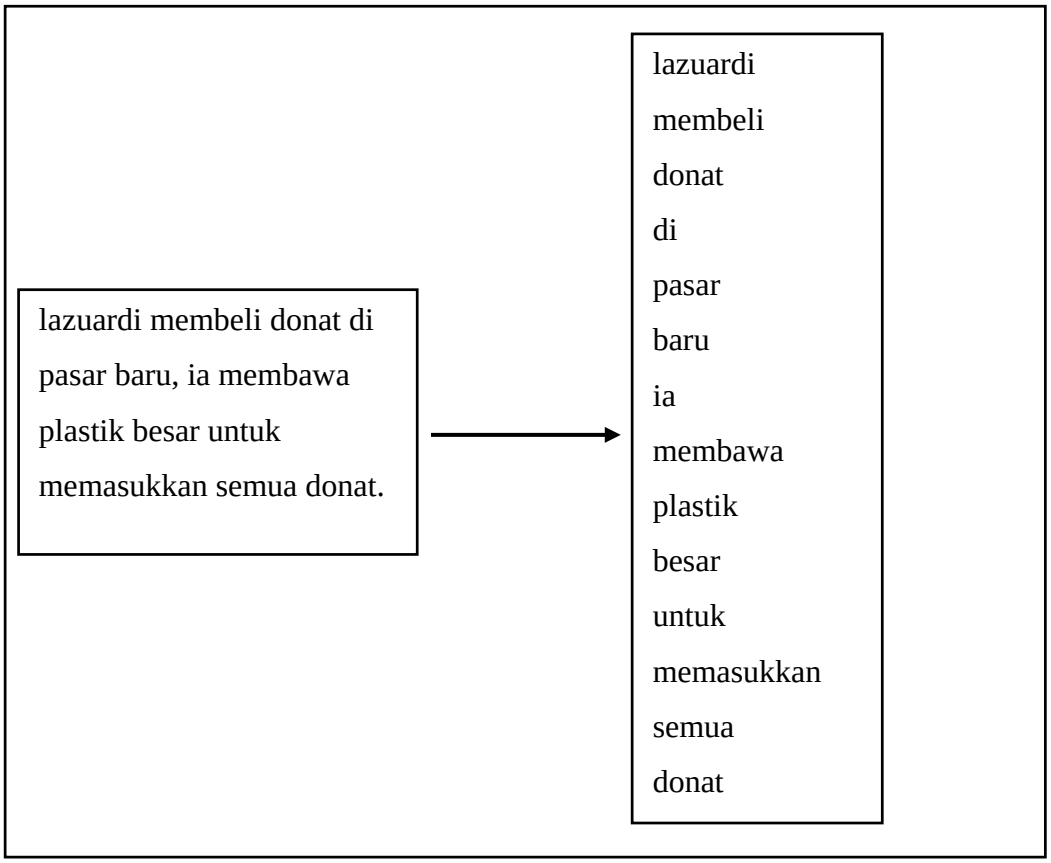


Gambar 2.1 Contoh *Case folding*

2. *tokenizing/ parsing*

Tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Secara garis besar tokenisasi adalah tahap memecah sekumpulan karakter dalam suatu teks kedalam satuan kata. Sekumpulan karakter tersebut dapat berupa karakter *whitespace*, seperti *enter*, tabulasi, spasi. Namun untuk karakter petik tunggal (‘), titik (.), semikolon (;), titik dua (:) atau lainnya, juga dapat memiliki

peran yang cukup banyak sebagai pemisah kata. Contohnya seperti yang tertera pada gambar 2.2.



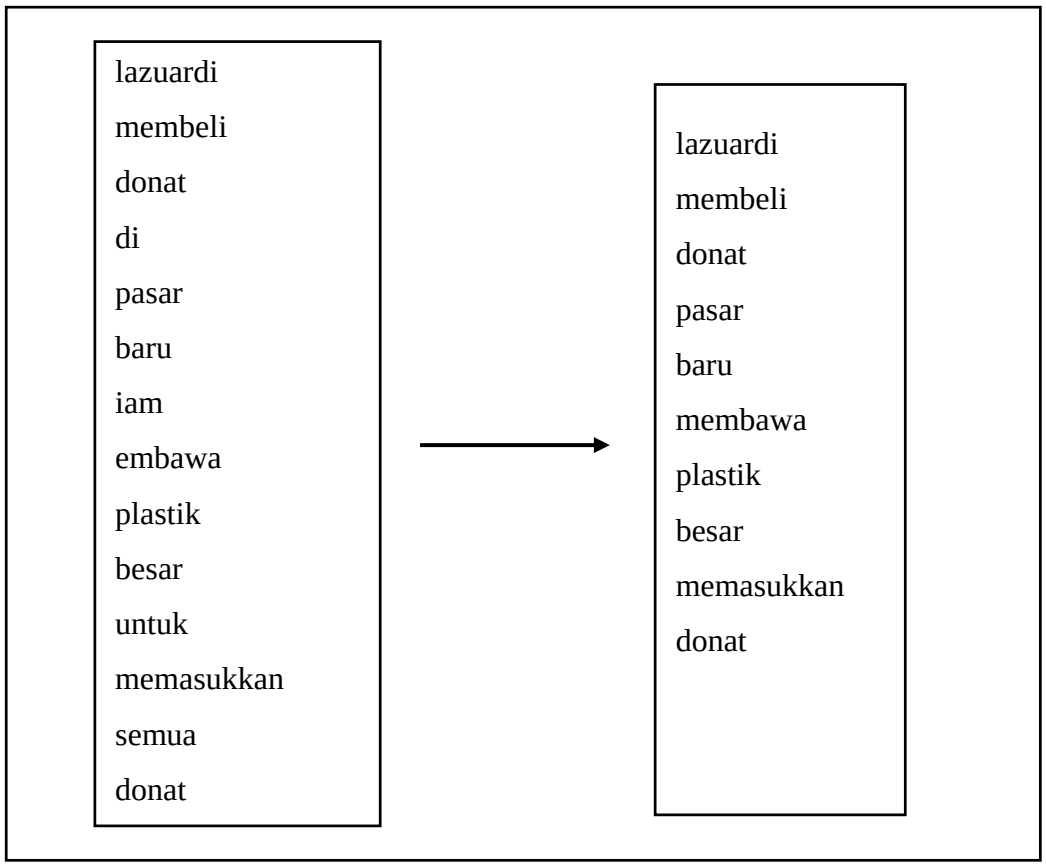
Gambar 2.2 Contoh *Tokenizing*

3. *filtering*

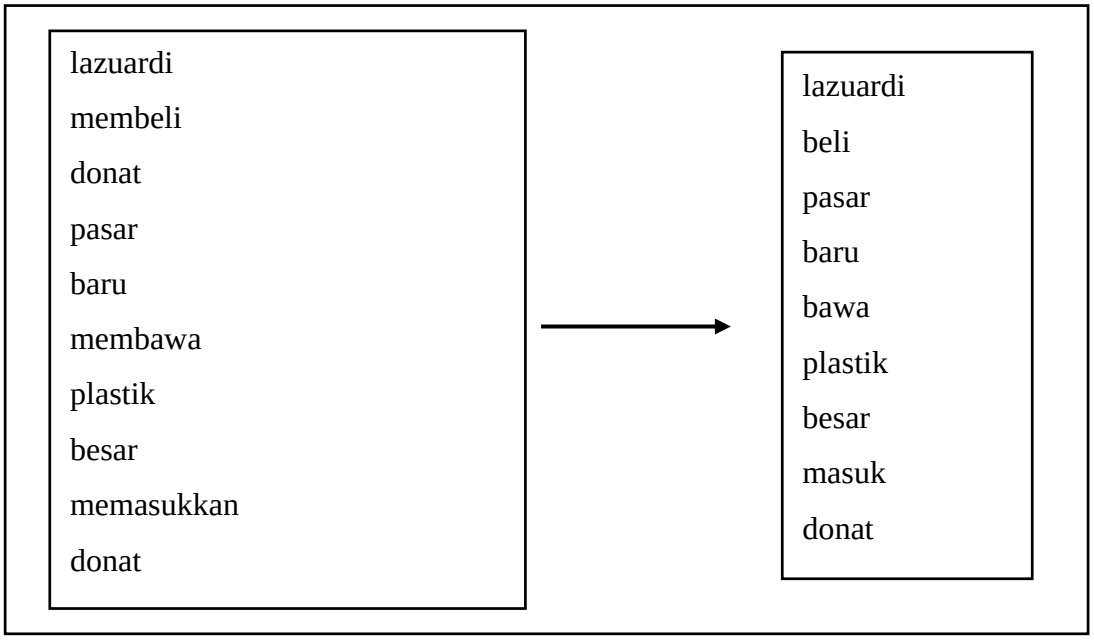
Tahap mengambil kata-kata penting dari hasil *token.* Bisa menggunakan algoritma *stoplist* (membuang kata-kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari”, dan seterusnya. Contohnya dapat dilihat pada gambar 2.3.

4. *Stemming*

Tahap mencari *root* kata atau kata dasar dari tiap kata hasil *filtering*. Penghilangan duplikat pada kata-kata di dalam dokumen juga penting. Contoh hasil proses *stemming* dapat dilihat pada gambar 2.4.



Gambar 2.3 Contoh *Filtering*



Gambar 2.4 Contoh *Stemming*

2.3.2. TF-IDF

Menurut Robertson (2004: 1) diacu dalam Suliantoro, dkk. (2012: 2) Metode *TF-IDF* merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata didalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut didalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila kata tersebut memiliki frekuensi yang tinggi didalam dokumen, frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (*database*). Menurut Suliantoro, dkk. (2012: 2), nilai *TF-IDF* didapat menggunakan rumus sebagai berikut:

$$W_{(t,d)} = TF_{(t,d)} * IDF_{(t)}$$

Dimana nilai IDF(t) didapatkan dari :

$$IDF_{(t)} = \log(|D|/DF_{(t)})$$

Keterangan :

$TF_{(t,d)}$: Jumlah kemunculan token t pada dokumen d

$IDF_{(t)}$: Nilai *IDF* token t

$DF_{(t)}$: jumlah dokumen yang memuat token t

$|D|$: jumlah dokumen dalam korpus/*database*

Berdasarkan rumus diatas, berapapun besarnya nilai $TF_{(t,d)}$, apabila |D| = $DF_{(t)}$ maka akan didapat hasil 0 (nol) untuk perhitungan *IDF*. Untuk itu dapat

ditambahkan nilai 1 (satu) pada sisi *IDF*, sehingga perhitungan bobotnya menjadi:

$$W_{(t,d)} = TF_{(t,d)} * \left(log \frac{|D|}{DF_{(t)}} + 1 \right)$$

2.4. *Clustering*

2.4.1. Pengertian *Clustering*

Clustering berkaitan dengan pengelompokan bersama objek-objek yang mirip antara satu sama lain dan berbeda dengan objek milik *cluster* lain. (Bramer, 2013:311)

Proses pengelompokan satu set objek fisik atau abstrak ke dalam kelas objek yang serupa disebut *clustering*. (Han dan Kamber, 2006:383). Dalam hal ini beliau juga menyebutkan bahwa *Clustering* juga disebut segmentasi data dalam beberapa aplikasi karena partisi pengelompokan data ke dalam kelompok sesuai dengan kesamaan mereka.

Cluster adalah kumpulan objek data yang mirip antara satu sama lain dalam kelompok yang sama dan berbeda dengan objek dalam kelompok lainnya. Sekelompok objek data dapat diperlakukan secara kolektif sebagai satu kelompok. (Han dan Kamber, 2006:383).

Clustering dapat dianggap yang paling penting dalam masalah *unsupervised learning*, karena setiap masalah semacam ini, berurusan dengan mencari struktur dalam kumpulan yang tidak diketahui datanya. Sehingga dapat didefinisikan bahwa *clustering* merupakan proses mengatur objek menjadi anggota kelompok yang hampir sama dalam beberapa cara tanpa membutuhkan informasi data acuan.

Dengan *clustering* maka obyek akan dikelompokkan ke dalam satu atau lebih *cluster* sehingga obyek-obyek yang berada dalam satu *cluster* akan mempunyai

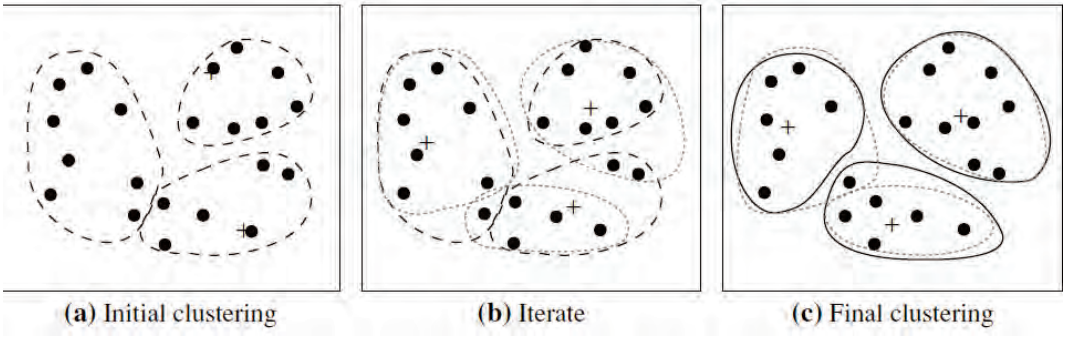
kesamaan yang tinggi antara satu dengan lainnya. Obyek-obyek dikelompokkan berdasarkan prinsip memaksimalkan kesamaan obyek pada *cluster* yang sama dan memaksimalkan ketidaksamaan pada *cluster* yang berbeda. Kesamaan obyek biasanya diperoleh dari nilai-nilai atribut yang menjelaskan obyek data yang sama.

2.4.2. Metode *Clustering*

Secara garis besar, terdapat beberapa metode *clustering*. Menurut Han, dkk. (2012: 448-450) diacu dalam Wahyuni (2014: 12-14) metode utama *clustering* dapat diklasifikasikan kedalam kategori berikut:

a. *Partitioning Method*

Membangun berbagai partisi dan kemudian mengevaluasi partisi tersebut dengan beberapa kriteria, di mana setiap partisi mewakili cluster. *Cluster* dibentuk untuk mengoptimalkan kriteria partisi tujuan, seperti fungsi perbedaan berdasarkan jarak, sehingga objek satu cluster yang sama memiliki sifat serupa, sedangkan objek *cluster* yang berbeda memiliki sifat yang berbeda. Metode *partitioning* seperti pada Gambar 2.5., merupakan contoh *clustering* dengan *Partioning Method* yaitu menggunakan algoritma *K-Means*. Algotritma yang termasuk dalam metode ini adalah *K-Means, K-Medoids* dan *CLARANS.*



Gambar 2.5 Contoh *Partitioning Method*

b. *Hierarchical Methods*

Membuat suatu penguraian secara hierarkikal dari himpunan data dengan menggunakan beberapa kriteria. Metode ini terdiri atas dua macam, yaitu *Agglomerative* yang menggunakan strategi *bottom-up* dan *Disisive* yang menggunakan strategi top-down. Metode ini meliputi algoritma *Birch*, *Cure*, dan *Chameleon*.

c. *Density-based Methods*

Metode ini berdasarkan konektivitas dan fungsi densitas. Metode ini meliputi algoritma *Dbscan*, *Optic*, dan *Denclu*.

d. *Grid-based Methods*

Metode ini berdasarkan suatu struktur granularitas multi-level. Metode clusterisasi ini meliputi algoritma *Sting*, *WaveCluster*, dan *Clique*.

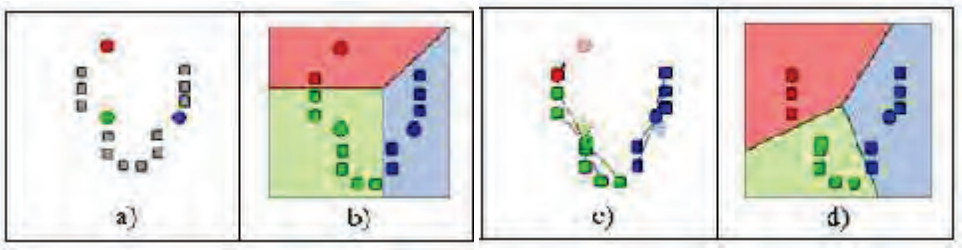
2.5. *K-Means*

K-Means merupakan salah satu metode data clustering *non hierarchical* yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*/kelompok. Metode ini mempartisi data ke dalam cluster/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam cluster yang lain. Adapun tujuan dari data clustering ini adalah untuk meminimalkan *objective function* yang diset dalam proses clustering, yang pada umumnya berusaha meminimalkan variasi di dalam suatu cluster dan memaksimalkan variasi antar cluster. (Agusta, 2007:47)

Algoritma *K-Means* memakai parameter *input*, k, dan partisi satu set objek n ke dalam k *cluster* sehingga menghasilkan tingkat similaritas yang tinggi setiap

intracluster, tetapi hasil tingkat similaritas yang rendah untuk *intercluster*. Kemiripan *cluster* diukur dengan nilai rata-rata dari objek didalam *cluster*, yang dapat dilihat sebagai *centroid cluster* atau pusat gravitasi. (Han dan Kamber, 2006:402)

Jadi *K-Means* adalah algoritma yang digunakan untuk pengelompokan berdasarkan tingkat similaritas disetiap *cluster*, tingkat similaritas tersebut dinilai berdasarkan *centroid* atau titik pusat dari sebuah *cluster*



Gambar 2.6 Contoh *K-Means Clustering*

2.5.1. Tahapan Algoritma *K-Means*

Tahapan algoritma dari *K-Means Clustering* menurut Suliantoro, dkk. (2012: 2) seperti yang digambarkan pada Gambar 2.6. lebih jelasnya sebagai berikut:

a. Pilih secara acak vektor data yang akan digunakan sebagai *centroid* awal sebanyak k.
b. Cari *centroid* yang paling dekat dari setiap data dengan cara menghitung jarak setiap data dengan setiap *centroid cluster*.
c. Hitung ulang untuk menentukan *centroid* baru dari setiap *cluster* dengan menghitung rata-rata nilai vektor semua data dalam *cluster* tersebut.
d. Lakukan langkah b dan c hingga *centroid* tidak mengalami perubahan lagi (tidak ada data yang berpindah *cluster* lagi) atau perubahan *centroid* lebih kecil dari nilai *error/threshold* yang ditetapkan. Dalam menentukan jarak antara

sebuah data dengan *centroid* sebuah *cluster*, digunakan rumus *Euclidean distance*

Rumus menghitung jarak menggunakan *euclidean distance*:

$$d(i,j) = \sqrt{(xi1 + xj1)^2 + (xi2 + xj2)^2 + \cdots + (xip + xjp)^2}$$

Keterangan:

d = titik pada data

i = data record $(xi1, xi2, \dots, xip)$

j = data centroid $(xj1, xj2, \dots, xjp)$

Untuk setiap *cluster*, menghitung rata-rata pembentukan *centroid* akan menggunakan objek yang ditugaskan untuk *cluster* pada iterasi sebelumnya. Semua objek diperbarui sebagai pusat *cluster* baru. Iterasi akan berlanjut sampai *cluster* stabil, dimana kelompok yang dibentuk di iterasi saat ini adalah sama seperti yang dibentuk di babak sebelumnya. Adapun rumus untuk penghitungan *centroid* yang baru adalah sebagai berikut:

$$C(i) = \frac{i1 + i2 + i \dots}{\sum i} \qquad C(j) = \frac{i1 + i2 + i \dots}{\sum i}$$

Keterangan :

$i/j1$ = nilai data *record* ke-1

$i/j2$ = nilai data *record* ke-2

Σ*i* / *j* = jumlah data *record*

2.6. *Latent Semantic Indexing*

Menurut Rosario (2000: 2) *Latent Semantic Indexing (LSI)* adalah sebuah penerapan teknik matematika tertentu yang disebut *Singular Value decomposition* atau *SVD* untuk pengolahan *word-by-document* matriks. Proyeksi ke dalam ruang

semantik laten dipilih sedemikian rupa sehingga representasi dalam ruang asli berubah sesedikit mungkin.

LSI (*Latent Semantic Indexing*) dibuat untuk mendukung *information retrieval* dan memecahkan masalah ketidaksesuaian antara kamus pemakai dengan penulis dokumen. Asumsi yang mendasari *LSI* adalah terdapat sebuah struktur pokok atau "*latent*" yang mempresentasikan hubungan antar kata. *LSI* menerima sebuah vektor atau matrik dari sekumpulan dokumen, dimana setiap baris mewakili satu term (bisa kata atau frase), tiap kolom mewakili satu dokumen, dan tiap selnya akan berisi nilai bobot kata terhadap dokumen. Bobot dari kata tiap dokumen dapat berisi *Term Frequency* atau juga menggunakan *TF-IDF* (dalam contoh yang akan saya tuliskan mengasumsikan penggunaan *TF*).

LSI menggunakan SVD (*Singular-Value Decomposition*) untuk memodelkan relasi asosiatif antara term. Ide dasar SVD adalah menerima kumpulan data dengan dimensi dan variabel tinggi serta menguranginya ke dalam ruang dimensi yang berukuran lebih kecil untuk menampakkan lebih jelas sub struktur dari data asli dan mengurutkannya mulai dari paling bervariasi sampai dengan tidak bervariasi.

Jadi *Latent Semantic Indexing* pada dasarnya sering digunakan untuk sistem *information retrieval*, *LSI* menggunakan SVD untuk memperkecil ukuran dimensi dan menggunakan perhitungan similaritas antar tiap dokumen untuk mengindex dokumen yang mirip dengan kueri pencarian, namun dengan sedikit perubahan pada bagian perhitungan similaritas antar dokumen, bisa dimungkinkan untuk dijadikan sistem pengelompokan.

LSI dalam kegunaannya ada dua hal, menurut Zelkovitz (2010: 2) LSI bisa digunakan sebagai berikut:

a. LSI untuk *Retrieval*

Ketika *LSI* digunakan untuk pencarian, *query* direprentasikan sebagian kecil yang sama dengan koleksi dokumen yang direprentasikan didalamnya. Hal ini dilakukan dengan mengalikan transpose vektor dari query dengan matriks T dan S1. Setelah *query* diwakili cara ini, jarak antara *query* dan dokumen dapat dihitung dengan menggunakan metrik kosinus, yang merupakan pengukuran similaritas antara dokumen. *LSI* menghasilkan jarak antara query *dan* semua koleksi dokumen. Dokumen-dokumen yang memiliki nilai jarak kosinus lebih tinggi dari pada yang lain, bisa dikatakan sebagai dokumen yang relevan dengan query

b. *LSI* untuk Klasifikasi

Kami menggunakan *LSI* untuk klasifikasi teks, sehingga kita dapat selanjutnya mengacu pada koleksi dokumen sebagai contoh pelatihan dan *query* sebagai contoh uji. Menggunakan pengukuran similaritas kosinus, *LSI* menghasilkan dokumen yang paling mirip dari contoh uji dalam ruang baru. Dan dibandingkan dengan dokumen yang mirip dengan tetangganya, sehingga yang akan dibandingkan adalah kedekatan similaritas antar dokumen satu dengan dokumen lainnya.

2.6.1. Tahapan *LSI*

Meunurut Garcia (2006: 1-4) langkah langkah algoritma *LSI* adalah sebagai berikut:

1. Hitung *term weight* dan buatlah *term* dokumen matriks A serta query matriks.
2. Dekomposisikan matriks A dan bentuklah nilai matriks U, S, V.
3. Lakukan perduksian dimensi, dalam hal ini pereduksian dimensi terbaik dibutuhkan penelitian oleh matematikawan.

4. Cari vektor koordinat dokumen yang telah direduksi menjadi ruang 2-dimensi.
5. Cari *query vector* koordinat yang telah direduksi menjadi 2-dimensi

   Rumus mencari *query vector*:

   $q = q^T U_k {S_k}^{-1}$
6. Urutkan dokumen dengan urutan nilai yang terbesar ke yang terkecil dari *query* dokumen berdasarkan hasil kosinus similaritas.

   Rumus similaritas:

   $sim(q, d) = \frac{q.d}{|q||d|}$
7. Maka didapatkan hasil similaritas, semakin besar hasilnya maka semakin mirip dokumen tersebut.

2.7. Perbandingan Kinerja Algoritma *Data Mining*

Menurut Oktafia dan Pardede (2008) pengukuran kinerja sebuah algoritma data mining dapat dilakukan berdasarkan beberapa kriteria antar lain akurasi, kecepatan komputasi, robustness, skalabilitas dan interpretabilitas. Kinerja algoritma akan terukur berdasarkan keakuratan dan eror yang dihasilkan. Semakin besar keakuratan, kinerja algoritma semakin baik. Semakin kecil eror, menunjukkan bahwa kinerja algoritma tersebut semakin baik. Untuk mengukur akurasi dari sebuah algoritma data mining maka bisa dihitung menggunakan teknik *Confusion Matrix.*

2.7.1. *Confusion Matrix*

Confusion matrix membandingkan kategori per kategori (kelas per kelas) hubungan antara data sebenarnya atau *Reference Class* dengan data hasil klasifikasi atau *Mapped Class* (Setioharjo dan Harjoko, 2014:184).

Menurut Han dan Kamber (2006), diacu dalam Andriani (2013:165-166) Evaluasi dengan confusion matrix menghasilkan nilai accuracy, precision, dan recall. Nilai accuracy merupakan persentase jumlah record data yang diklasifikasikan secara benar oleh sebuah algoritma dapat membuat klasifikasi setelah dilakukan pengujian pada hasil klasifikasi tersebut.

Menurut Indriani (2014:6) bahwa True Posstive (TP), yaitu jumlah dokumen dari kelas 1 yang benar dan diklasifikan sebagai kelas 1. True Negative (TN), yaitu jumlah dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0. False Positive (FP), yaitu jumlah dokumen dari kelas 0 yang salah diklasifikasikan sebagai kelas 1. False Negative (FN) yaitu jumlah dokumen dari kelas 1 yang salah diklasifikasikan sebagai kelas 0. Seperti yang tertera pada tabel 2.1.

		Prediksi	
		Positif	**Negatif**
Kenyataan	**Positif**	**TP**	**FN**
	Negatif	**FP**	**TN**

Gambar 2.7 C*onfusion Matrix*

a. *Recall* atau *true positive rat*e (*TP*) adalah proporsi kasus positif yang diidentifikasi secara benar. Untuk rumus *recall* adalah sebagai berikut

$Recall = \frac{\text{TP}}{(\text{TP+FN})}$

b. *Precision* adalah proporsi kasus dengan hasil positif diidentifikasi secara benar. Rumusnya adalah $Precision = \frac{\text{TP}}{(\text{TP+FP})}$

c. *Accuracy* adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus. Rumusnya adalah $accuracy = \frac{(\text{TP+TN})}{(\text{TP+FN+FP+TN})}$

2.8. Penelitian Relevan

Proses pengelompokkan teks menggunakan *LSI* maupun *K-Means Clustering* sudah pernah dibuat dan digunakan dalam berbagai penelitian, namun untuk data teks pendek masih jarang. Berikut ini akan dipaparkan beberapa penelitian yang telah dibuat berkaitan dengan pengelompoka teks, yaitu:

1. Penerapan Algoritma K-Means untuk *Clustering* Dokumen E-Jurnal STMIK GI MDP
    a. Penelitian ini dilakukan oleh Ernie Kurniawan, Maria Fransiska, Tinaliah, dan Rachmansyah dari Program Studi Teknik Informatika, STMIK GI MDP tahun 2013.
    b. Tujuan dari penelitian ini adalah menggunakan algoritma *K-means Clustering* untuk mengelompokan dokumen E-jurnal, pengelompokan tersebut digunakan agar mempercepat proses *query* pencarian.
    c. Data yang digunakan adalah dokumen e-jurnal dari STMIK GI MDP.
    d. Teknik disini adalah menggunakan algoritma K-means untuk mengelompokan dokumen E-jurnal yang ada pada web STMIK GI MDP, dan memanfaatkan *stemming* sebagai pemercepat proses pengelompokan

e. Pengujian dilakukan dengan cara membandingkan hasil yang diperoleh dari aplikasi menggunakan algoritma K-Means dengan klasifikasi judul yang ada pada *database* aplikasi dimana dilakukan proses pencarian dengan cara memasukkan 5 *query* yang sama ke dalam masing-masing aplikasi.

f. Hasil percobaan menunjukan bahwa nilai akurasi hasil pengelompokan clustering hanya mencapai 50% saja, pengujian dengan menggunakan stemming akan mempercepat waktu pengolahan data, sehingga ditarik kesimpulan bahwa Algoritma K-Means dapat melakukan pengelompokan dokumen dalam jumlah yang banyak akan tetapi belum efisien dalam mengelompokan dokumen secara tepat. Penentuan centroid (titik pusat) pada tahap awal Algoritma K-Means sangat berpengaruh pada hasil cluster. Steming akan mempercepat proses pengelompokan data. Semakin sedikit dokumen yang dipakai, maka semakin sulit untuk membedakan cluster antara stemming dan non-stemming.

2. Integrasi Pembobotan *TF-IDF* pada Metode *K-Means Clustering* untuk *Clustering* Dokumen Teks.

   a. Penelitian ini dilakukan oleh Deddy Wijaya Suliantoro, Irya Wisnubhadra dan Ernawati dari Magister Teknik Informatika, Universitas Atma Jaya Yogyakarta tahun 2012.

   b. Tujuan dari penelitian ini adalah menggabungkan dan mengevaluasi kinerja perpaduan metode *K-Means Clustering* dan *TF-IDF* dalam proses *clustering* dokumen teks ke dalam suatu aplikasi *clustering* dokumen teks digital.

c. Data yang digunakan dalam penelitian ini adalah jenis dokumen teks berita yang diunduh dari situs berita kompas yang disalin ke dalam *file plain text* (*.txt*).

d. Teknik yang dilakukan adalah mengintegrasikan pembobotan *TF-IDF* ke dalam *K-Means Clustering* dilakukan dengan cara menggunakan nilai bobot *token* yang didapat dalam perhitungan *TF-IDF* sebagai vektor atau parameter dalam proses *clustering* menggunakan *K-Means Clustering*, sehingga banyaknya vektor data akan didapat dari jumlah token unik didalam kamus token (*lexicon*) seluruh dokumen dalam *database.*

e. Pengujian akurasi dilakukan dengan parameter *precision* dan *recall.* Nilai *precision* didapat dari melihat berapa persen dokumen dalam suatu *cluster* yang benar, sedangkan nilai *recall* didapat dari beberapa persen dokumen yang seharusnya masuk ke dalam satu *cluster* benar-benar berada di *cluster* tersebut.

f. Tingkat *precision* dan *recall* yang didapat dari hasil percobaan cukup tinggi yaitu, diatas 50%. Hasil percobaan menunjukkan bahwa semakin besar jumlah dokumen dalam *database*, semakin tinggi pula rata-rata *precision* dan *recall* dari hasil *clustering* yang dilakukan.

3. Transductive LSI for Short Text Classification Problems

a. Penelitian ini dilakukan oleh Sarah Zelikovitz, The College of Staten Island of CUNY.

b. Tujuan dari penelitian ini adalah melakukan klasifikasi teks dokumen menggunakan algoritma *LSI.*

c. Data yang digunakan dalam penelitian ini terdapat 5 set, dimana data yang diambil dari sebuah web, set tersebut berisi tentang *physic*, *netVet*, *business*, *news*, *thesaurus.*

d. Teknik yang dilakukan adalah dengan menggunakan algoritma *LSI* sebagai alat klasifikasi dokumen, dan membandingkan keakurasian *LSI* yang menggunakan data train dengan *LSI* yang tidak menggunakan *data train.*

e. Pengujian akurasi dilakukan dengan menghitung keakurasian hasil data menggunakan *data train* dengan yang tidak menggunakan *data train*, keakurasian dihitung berdasarkan perbandingan kesamaan hasil data olahan dengan klasifikasi data mentahnya.

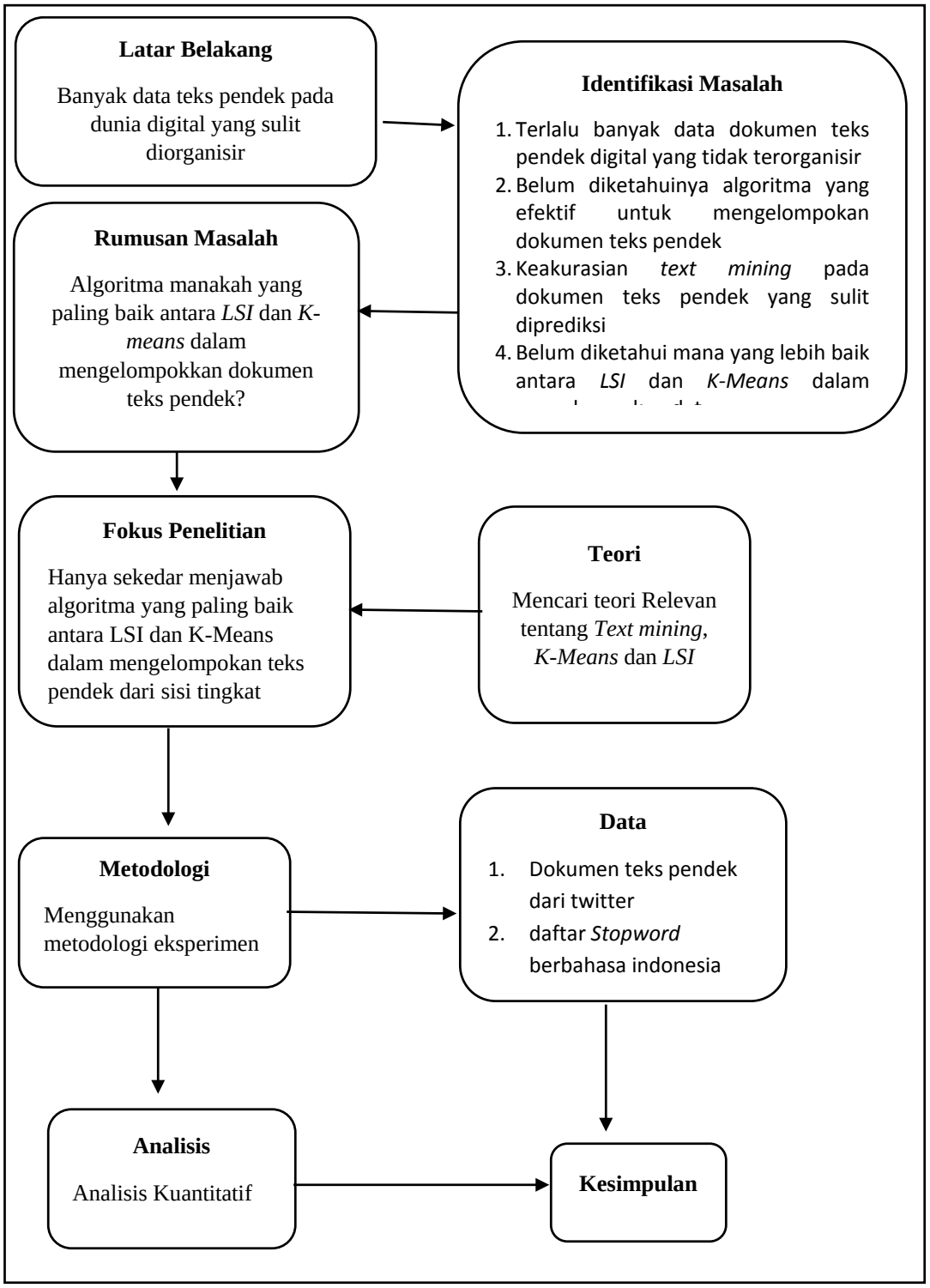
f. Hasil dari perbandingan akurasi adalah untuk dokumen *physic* yang menggunakan *data test* menghasilkan akurasi sebesar 87% sedangkan tanpa *data test* keakuratannya adalah 85%, untuk dokumen *netVet* yang menggunakan *data test* menghasilkan akurasi sebesar 52% sedangkan tanpa *data test* keakuratannya adalah 45%, untuk dokumen *business* yang menggunakan *data test* menghasilkan akurasi sebesar 19% sedangkan tanpa *data test* keakuratannya adalah 14%, untuk dokumen *news* yang menggunakan *data test* menghasilkan akurasi sebesar 92% sedangkan tanpa *data test* keakuratannya adalah 85%, untuk dokumen *thesaurus* yang menggunakan *data test* menghasilkan akurasi sebesar 23% sedangkan tanpa *data test* keakuratannya adalah 21%, dari hasil penelitian ini bisa disimpulkan LSI yang menggunakan *data test* hasilnya selalu

mendapatkan peningkatan walaupun tidak signifikan, hanya sekitar 2% sampai dengan 7%.

2.9. Kerangka Berpikir

Dokumen saat ini sudah menjadi bagian vital, terlebih lagi dokumen teks pendek, salah satu teks pendek sering ditemukan pada media sosial. Media social yang saat ini menggunakan teks pendek adalah *Twitter*, disebut demikian karena hanya dapat memuat karakter sebanyak 140 dan rata-rata hanya dapat memuat 11 kata. Mengolah teks tentu bukan pekerjaan yang mudah, terlebih lagi teks pendek yang dokumennya sulit ditebak. Oleh karena itu diperlukan sebuah metode yang baik untuk dapat mengorganisir dokumen secara otomatis, sehingga dapat mempermudah dalam pencarian informasi yang relevan.

Permasalahan yang dikaji dalam tulisan ini adalah bagaimana hasil perbandingan antara metode *K-Means Clustering* dengan *LSI* dalam pengelompokkan teks pendek yang diperoleh dari media sosial *Twitter*. Dengan diketahui hasilnya dari kenerja algoritma *LSI* dan *K-Means* dalam mengelompokan teks pendek maka di periode selanjutnya berguna untuk mengetahui mana algoritma yang paling tepat untuk pengolahan teks pendek, sehingga mempermudah dan mempercepat dalam pencarian informasi yang dibutuhkan. *LSI* disini akan sedikit dimodifikasi dari tujuannya sebagai *information retrieval* menjadi *text mining*, dengan mengubah parameter similaritas maka diharapkan LSI dapat digunakan untuk text mining, sedangkan untuk *K-Means* dipenelitian ini hanya diimplementasikan saja untuk pengelompokan teks pendek dan tidak ada modifikasi algoritma. Bagan dari kerangka berfikir dapat dilihat pada gambar 2.8.



Gambar 2.8 Bagan Kerangka Berpikir

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta. Waktu penelitian dilaksanakan pada perkuliahan semester 102/103.

3.2. Metode Penelitian

Metode penelitian yang digunakan adalah metode eksperimen. Menurut Jaedun (2011: 5) menyatakan bahwa penelitian eksperimen adalah penelitian yang dilakukan terhadap variabel yang data-datanya belum ada sehingga perlu dilakukan proses manipulasi melalui pemberian *treatment* atau perlakuan tertentu terhadap subjek penelitian yang kemudian diamati atau diukur dampaknya (data yang akan datang). Metode eksperimen dalam penelitian ini merupakan metode untuk melihat perbandingan kineja algoritma *K-Means* dan *LSI* pada teks pendek berupa kumpulan data tweet yang diambil dari akun *twitter* "detik.com". Data penelitian di ambil atau diekstrasi dari *twitter* melalui website allmytweets.com. Selanjutnya data-data tersebut akan dikelompokan menggunakan kedua algoritma tersebut dan dilihat seberapa akurat algoritma *LSI* dan *K-Means* dalam pengelompokan teks pendek dari *tweet*.

3.3. Instrumen Penelitian

Berikut ini adalah instrumen yang digunakan dalam penelitian:

1. Perangkat Keras untuk *K-means*

   a. *Processor* AMD® A6-6310 @ 2.4 GHz.

   b. *Memory RAM* 6 GB DDR3

2. Perangkat Keras untuk algoritma *LSI*

   a. *Processor* AMD® Athlon @ 2.8 Ghz

   b. *Memory RAM* 3 GB DDR2

3. Perangkat Lunak

   a. MATLAB 2014a untuk memproses algoritma *K-Means*

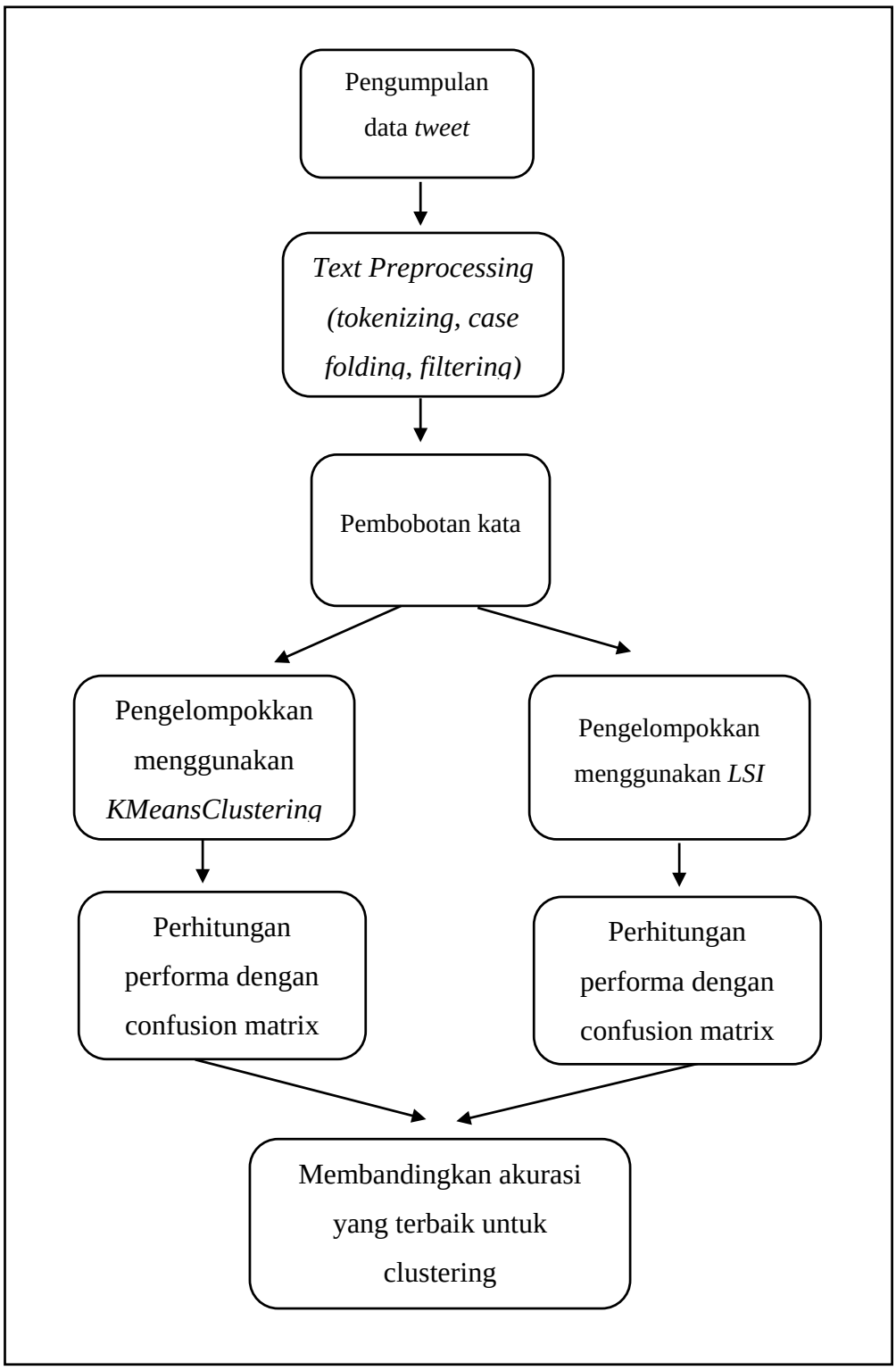
   b. MATLAB 2010b untuk memproses algoritma *LSI*

   c. Fungsi strjoin.m sebagai tambahan di *MATLAB* 2010b

   d. *website* penyedia *scrapping twitter* www.allmytweets.com

3.4. Rancangan Penelitian

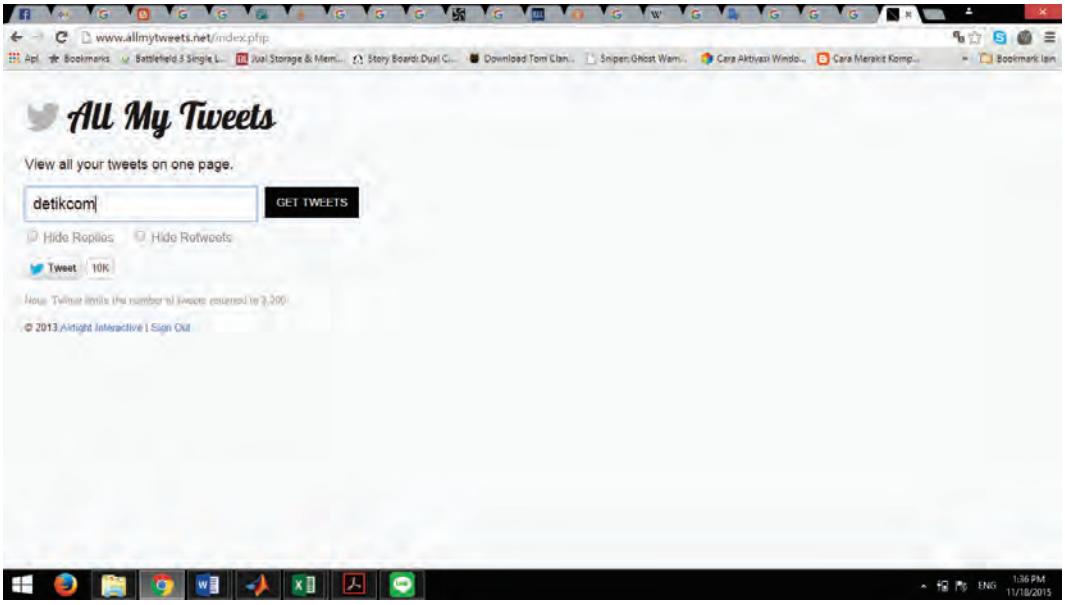
Langkah-langkah untuk pengelompokkan teks pendek yaitu, awal akan dilakukan pengumpulan data dari *web scrapper twitter*, lalu dilakukan *text processing* untuk memilah kata kata yang dibutuhkan, dan setelah itu dilakukan dokumen tersebut dilakukan pembobotan kata, dilakukan clustering dengan menggunakan algoritma *LSI* ataupun *K-Means*, lalu dihitung tingkat keakurasian hasil *clustering* dengan data mentahnya, setelah itu membandingkan tingkat keakurasian hasil clustering antara *LSI* dengan *K-Means*, dan pengambilan kesimpulan algoritma yang terbaik untuk digunakan sebagai *clustering* dokumen teks pendek yang bersumber dari *twitter*. Untuk lebih jelasnya dapat dilihat pada gambar 3.1 yang berisi bagan rancangan penelitian.



Gambar 3.1 Bagan Rancangan Penelitian

3.4.1 Pengumpulan Data

Data yang diambil adalah data dari twitter berbahasa indonesia. Data yang diambil berupa teks pendek kumpulan *tweet* pada akun Twitter "detik.com", data diambil menggunakan jasa penyedia *scrapping* data *tweets* yaitu http://www.allmy*tweets*.net/, cara menggunakannya cukup dengan memasukan akun twitter yang akan dilakukan *scrapping tweet* maka hasil dari *scrapping data tweets* akan menampilkan seperti yang tertera pada gambar 3.3 . Situs ini mampu melakukan *scrapping data tweets* sebanyak 3200 *tweets* dalam sekali pengambilan data, jumlahnya sama persis sesuai dengan yang tertera pada *Twitter API* yang dibatasi hanya 3200 *tweets.* Setelah *data tweets* terkumpul di http://www.allmy*tweets*.net/ lalu disimpan kedalam format (.xls).



Gambar 3.2 Halaman Utama All My Tweets

Gambar 3.3 Contoh Saat Melakukan Scraping Data *Twitter*

3.4.2 *Text Preprocessing*

Setelah data terkumpul langkah selanjutnya adalah *preprocessing*, isinya ada tiga tahapan yaitu:

1. *Case folding*

Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*.

2. *Tokenizing*

Tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Secara garis besar tokenisasi adalah tahap memecah sekumpulan karakter dalam suatu teks kedalam satuan kata. Dalam hal ini saya menggunakan garis baru sebagai memecah sekelompok kumpulan karakter sebagai satu dokumen, dan memecah setiap kata menggunakan spasi.

3. *Filtering*

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*.

Pada tahap ini menggunakan algoritma *stoplist* (membuang kata yang tidak penting) atau *stopword. Stoplist* atau *stopword* adalah kata-kata yang tidak deskriptif yang dibuang dalam pendekatan *bag-of-word.* Contoh dari *stopword* adalah “yang”, “dan”, “di”, “dari”, dan seterusnya.

3.4.3 Pembobotan Kata

Pembobotan kata dilakukan setelah tahap *text preprocessing*, metode yang digunakan adalah metode *TF-IDF* yang terdiri dari tahapan, yaitu:

1. Menghitung *Term Frequency*

Menghitung *term frequency* digunakan untuk memberi nilai pada sebuah kata yang ada pada dokumen, caranya adalah dengan menghitung jumlah masing-masing kata pada tiap dokumen.

2. Menghitung *DF* dan *IDF*

Tahap selanjutnya adalah menghitung *document frequency* (*DF*), *DF* adalah jumlah masing-masing kata pada keseluruhan dokumen, sedangkan *inverse document frequency* (*IDF*) untuk mengurangi bobot suatu *term* jika kemunculannya banyak tersebar di seluruh koleksi dokumen, menghitung *IDF* dapat menggunakan rumus berikut:

$$IDF_{(t)} = \log \frac{Total\ document}{Document\ frequent}$$

3. Menghitung *TF-IDF* (bobot)

Setelah didapat hasil *IDF* maka dilakukan perhitungan pembobotan dari setiap kata dengan menggunakan cara sebagai berikut:

$$W_{(t,d)} = TF_{(t,d)} * IDF_{(t)}$$

Setelah dilakukan perhitungan pembobotan maka data sudah siap untuk dilakukan perhitungan data mining, maka dapat dilakukan perhitungan clustering baik menggunakan algoritma *LSI* ataupun *K-Means*.

3.4.4 Perhitungan Clustering Menggunakan K-Means

Setelah didapatkan total nilai bobot dari tiap kata dalam satu dokumen teks pendek, maka langkah selanjutnya adalah mengelompokan menggunakan algoritma *K-Means Clustering*. Langkah-langkah *clustering* dengan algoritma *K-Means Clustering* sebagai berikut:

1. Kita harus menentukan jumlah *cluster* awal yang akan digunakan
2. Menentukan *centroid* untuk awal iterasi yang akan digunakan, biasanya *centroid* bisa dilakukan secara acak, namun perlu diingat berdasarkan penelitian sebelumnya menurut Ernie Kurniawan, dkk. (2013: 9) bahwa penentuan *cluster* awal sangat mempengaruhi hasil *cluster*
3. Menghitung jarak antara titik *centroid* dengan titik tiap objek. Untuk menghitung jarak kedekatan antar titik tersebut dapat menggunakan rumus *Euclidean Distance*, yaitu:

   $$d(i,j) = \sqrt{(xi1 + xj1)^2 + (xi2 + xj2)^2 + \cdots + (xip + xjp)^2}$$

   Dimana $d(i,j)$ adalah *Euclidean Distance*, xi adalah *data record*, xj adalah *data centroid*
4. Untuk menentukan anggota *cluster* dapat dilakukan dengan memperhitungkan jarak minimum objek. Jika sebuah objek lebih cenderung dekat dengan salah satu *centroid*, maka objek tersebut termasuk pada *cluster* sementara yang sama dengan *centroid* tersebut.

5. Hitung *centroid* baru dengan cara merata-ratakan anggota masing-masing *cluster* yang sama.
6. Ulangi lagi langkah nomor 3, untuk mendapatkan hasil anggota *cluster* yang baru, apabila hasil *cluster* iterasi sebelumnya dengan *cluster* iterasi saat ini tidak ada perubahan posisi *cluster* maka iterasi *k-means clustering* terhenti sampai disini.

3.4.5 Perhitungan Clustering Menggunakan *LSI*

Disini kita juga akan menguji menggunakan *LSI*, dimana algoritma *LSI* disini ada sedikit perubahan dari rumus aslinya, perubahan tersebut terjadi ketika menghitung similaritas, berikut langkah-langkahnya:

1. Lakukan penghitungan pembobotan dokumen, namun disini hanya menghitung pembobotan dokumen saja tanpa menghitung bobot dari *query*.
2. Dekomposisikan matriks A dan bentuklah nilai matriks U, S, V.
3. Lakukan perduksian dimensi, dalam hal ini pereduksian dimensi terbaik dibutuhkan penelitian oleh matematikawan.
4. Cari vektor koordinat dokumen yang telah direduksi menjadi ruang 2-dimensi.
5. Cari *query vector* koordinat yang telah direduksi menjadi 2-dimensi

   Rumus mencari *query vector*:

   $q = q^T U_k {S_k}^{-1}$
6. Membandingkan similaritas antara dokumen dengan anggota kelompok

   Rumus similaritas:

   $sim(q, d) = \frac{q.d}{|q||d|}$

7. Langkah selanjutnya disini kita menggunakan *threshold* sebagai pembatas perhitungan similaritas antara dokumen dengan kelompok dokumen, Dokumen pertama akan menjadi acuan awal kelompok pertama, jika hasil similaritas antara dokumen selanjutnya dibanding kelompok pertama lebih besar dari *threshold* maka dokumen kedua ini akan masuk kedalam anggota kelompok pertama, tapi jika hasil similaritasnya lebih kecil dari *threshold* maka dokumen ini akan menghitung similaritas terhadap kelompok selanjutnya, tapi jika pada akhirnya dokumen tersebut tidak menemukan kelompok yang hasil similaritasnya lebih tinggi dari *threshold* maka dokumen tersebut akan membentuk anggota kelompok yang baru.

3.4.6 Perhitungan dengan *Naïve Bayes*

Perhitungan dengan menggunakan *Naïve Bayes* hanya dilakukan jika terjadi penumpukan prediksi kelompok data yang dominan pada salah satu kelompok. Dengan menggunakan *Naïve bayes* ini akan ditentukan jumlah dan anggota masing-masing *predictive cluster* yang dominan.

Berikut langkah-langkah penggunaan naïve bayes didalam pemilihan *cluster:*

1. Pilih anggota data yang paling banyak jumlah datanya untuk dilakukan perhitungan awal
2. Hitung setiap masing-masing *predictive cluster* dari data dominan tersebut dengan rumus:

$$Predictive\ cluster\ A = \frac{Data\ dominan\ A}{Jumlah\ data\ cluster\ A}$$

3. Hasil perhitungan *predictive cluster* yang jumlahnya paling paling besar maka menjadi penentu *predictive cluster* kelompok tersebut

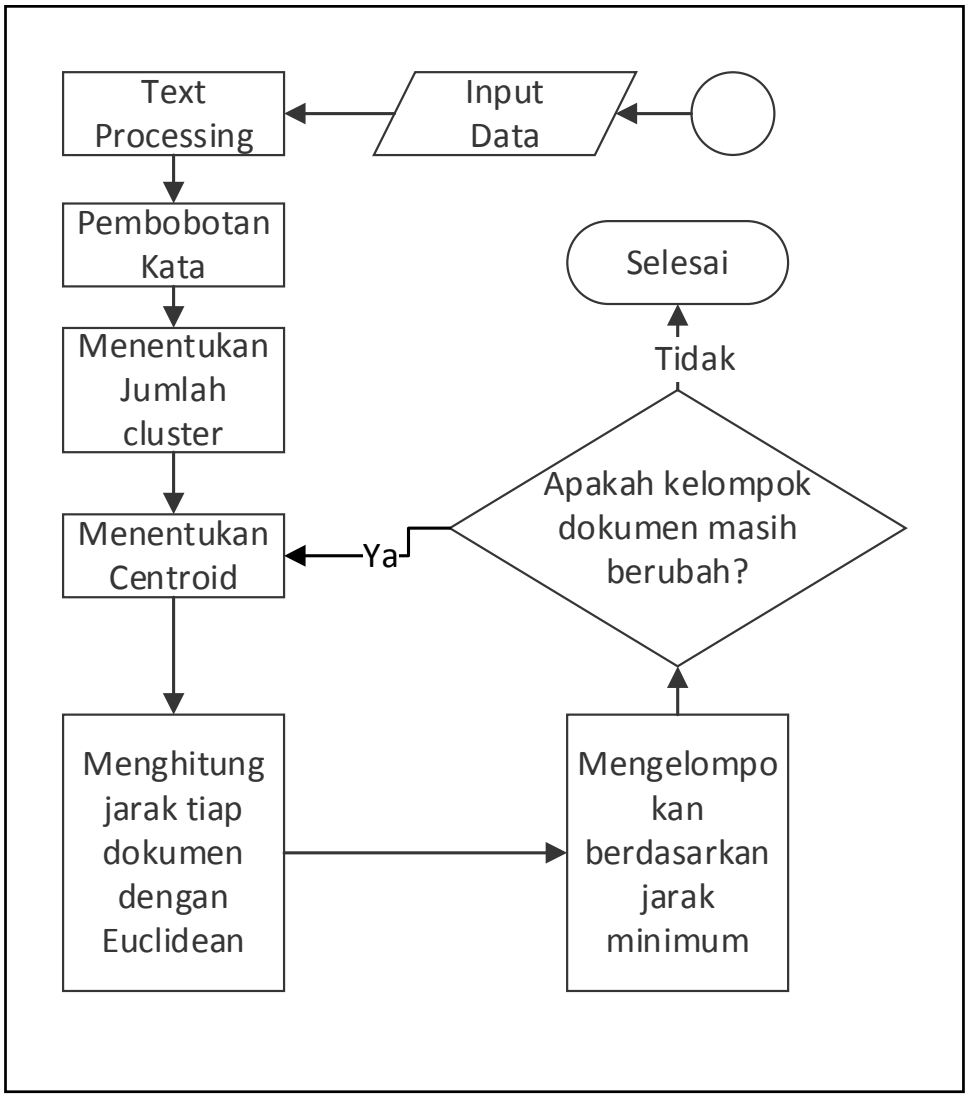
4. Selanjutnya lakukan penghitungan kembali untuk menentukan kelompok selanjutnya, jumlahkan sisa anggota kelompok lainnya tanpa menghitung kelompok yang sudah memiliki *predictive cluster* yang jelas.
5. Pilih kembali anggota yang jumlahnya paling banyak untuk perhitungan
6. Ulangi langkah ke-2 dan ke-3 untuk mendapatkan *predictive cluster* kelompok selanjutnya.
7. Setelah didapat kelompok selanjutnya mendapat predictive cluster yang jelas, ulangi langkah ke-4 sampai langkah ke-6 hingga seluruh kelompok memiliki *predictive cluster* yang jelas.

Dengan penggunaan Naïve bayes ini nanti akan didapatkan hasil predictive cluster yang jelas dari tiap kelompok yang menumpuk, dengan hasil *naïve bayes* ini akan memecah kebuntuan penumpukan data dan dapat dilanjutkan kedalam perhitungan confusion matrix untuk mencari tingkat performa algoritma berdasarkan *accuracy*.

3.5. Rancangan Program Bantu

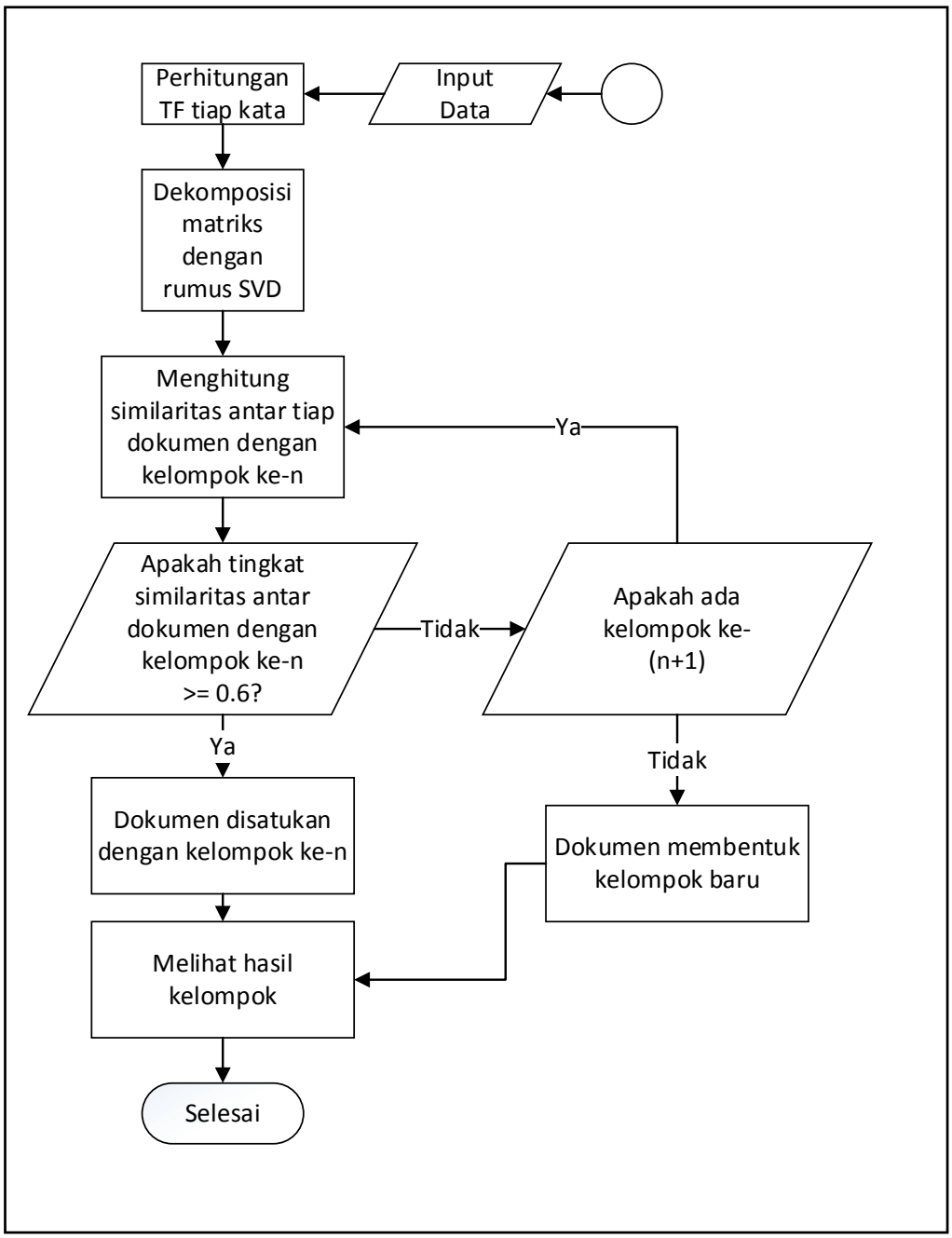
Rancangan program bantu disini dipergunakan hanya untuk memproses pengelompokan data mentah saja. Sedangkan secara pengambilan data tetap dilakukan manual, dalam melakukan perhitungan data hingga terbentuk kelompok yang sesungguhnya tetap dilakukan secara manual dengan metode *confusion matrix* menggunakan excel. Berikut pada gambar 3.4 dan pada gambar 3.5 merupakan rancangan *flowchart* program bantu untuk algoritma *K-Means* dan *LSI*.

Program bantu yang dibuat hanya berupa *console* tanpa *GUI*, hasil keluarannya bisa disimpan dalam format .mat.



Gambar 3.4 *Flowchart* Program Bantu Algoritma *K-Means*

Pada gambar 3.4 terlihat bahwa sistem yang dibuat dimulai dari *input* data, data yang diperlukan berupa data *tweet* berformat .txt, program bantu ini tidak dilengkapi dengan penentuan titik awal centroid sehingga hasil yang dikeluarkan bergantung titik awal penentuan, nantinya data ini akan diolah sehingga terbentuk *cluster*, dan data output dimuat dalam bentuk .mat yang dapat diolah kedalam excel untuk dilakukan perhitungan kinerjanya



Gambar 3.5 *Flowchart* Program Bantu Algoritma *LSI*

Pada gambar 3.5 terlihat bahwa sama seperti *K-means*, terlihat bahwa *input* data disini juga menggunakan format .txt, data yang dipergunakanpun sama, namun disini terjadi perubahan algoritma dimana yang dibandingkan similaritasnya adalah antar tiap dokumen yang ada.

3.6. Pengujian dengan *Confusion Matrix*

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining. *Confusion matrix* berisi informasi tentang actual dan prediksi yang diselesaikan dengan sistem klasifikasi.

		Prediksi	
		Positif	**Negatif**
Kenyataan	**Positif**	**TP**	**FN**
	Negatif	**FP**	**TN**

Gambar 3.6 *Confusion Matrix*

Recall atau *true positive rat*e (TP) adalah proporsi kasus positif yang diidentifikasi secara benar. Untuk rumus *recall* adalah sebagai berikut $Recall = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}$

Precision adalah proporsi kasus dengan hasil positif diidentifikasi secara benar. Rumusnya adalah $Precision = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})}$

Accuracy adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus. Rumusnya adalah $accuracy = \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{TN})}$

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Penyajian Data

Data yang diproleh dalam penelitian ini diambil dari akun *Twitter* "detikcom". Detik.com merupakan portal situs yang menyajikan berita setiap detiknya. Setiap *tweets* yang ada di akun *Twitter* "detikcom" terhubung dengan domain yang berada di situs detik.com. Hal itu dapat dilihat dari *link* yang disertai dalam *tweets* yang di *posting* bersama dengan isi *tweets.* Pengelompokkan dengan cara manual dapat di lakukan berdasarkan *link* yang terhubung ke domain situs detik.com tersebut. Misalnya hal tersebut dapat dilihat pada Tabel 4.1

Tabel 4.1 Contoh Data Twitter yang Telah Diperoleh

No.	Tweets	Topik	Domain
1	Ronaldo Yakin Kariernya Masih Panjang, Mimpikan Pensiun di Madrid http://t.co/WL58LLfxOq via @detiksport http://t.co/yBpswbPn5L Oct 14, 2015	detiksport	*sport*.detik.com
2	Suzuki Escudo alias Grand Vitara Model Anyar https://t.co/zb8oCuS8pF via @detikoto https://t.co/5pg97aFghV Oct 20, 2015	detikoto	oto.detik.com
3	Muhaimin: Jangan Ada Lagi Konflik Antar Umat Beragama http://t.co/GUf1kamO2I Oct 14, 2015	detiknews	news.detik.com
4	Melihat Lebih Dekat PLTA Terbesar di Indonesia yang Dibangun di Perut Bumi http://t.co/QjsJuCViaO via @detikfinance Oct 14, 2015	detikfinance	finance.detik.com
5	Clemmons: Ketika Ayam Goreng Tepung Bertemu dengan Strawberry http://t.co/U7Z0BbOlGO http://t.co/4UpZ7oxE7g Oct 14, 2015	detiknews	news.detik.com

6	Yummy! Magnum Infinity Raspberry Kini Sudah Bisa Dinikmati Pencinta Es Krim https://t.co/hkxHOzUZKy via @detikfood Oct 20, 2015	detikfood	*food*.detik.com
7	Darmin Akui Impor Beras Vietnam dan Thailand, Kementan: Ada Surplus 11 Juta Ton http://t.co/bcrfKo30FP via @detikfinance Oct 14, 2015	detikfinance	finance.detik.com
8	'Kapal Nabi Nuh' di Tanzania http://t.co/p5pChQeCCd Oct 14, 2015	detiknews	news.detik.com
9	Kembali ke Rumah, Tosan Dapat Pengawalan Ketat 24 Jam http://t.co/KXVwBCKGTR http://t.co/Y9cn23AQA4 Oct 14, 2015	detiknews	news.detik.com
10	1.000 Pengguna Xperia Z3 Keroyokan Garap Android Marshmallow https://t.co/gfRThD5YtF via @detikinet Oct 20, 2015	detikinet	inet.detik.com
11	Besok, PLN Akan Resmikan Pembangkit Listrik Matahari Terbesar di RI http://t.co/osyiN2HRx5 Oct 14, 2015	detiknews	news.detik.com
12	Dikawal Ketat, Tosan Kini Sudah Kembali ke Kampungnya di Lumajang http://t.co/8iumjyeDdc Oct 14, 2015	detiknews	news.detik.com
13	Kapolsek Muda Dhayita Dikenal Sebagai Pemimpin Sopan yang Disiplin http://t.co/v7Fj84EbPD Oct 14, 2015	detiknews	news.detik.com
14	10 Hal yang Perlu Diperhatikan ketika Memperbaiki Mobil Sendiri http://t.co/PFmuZbYOWY http://t.co/Mn3JxjfCEd Oct 14, 2015	detiknews	news.detik.com
15	Ridwan Kamil: Kalau Persib Menang, Ada Bonus Rp 500 Juta http://t.co/haevTjeVEc via @detiksport http://t.co/f33CeCTh4f Oct 14, 2015	detiksport	*sport*.detik.com
16	Kabur Saat Ikuti Pelatihan, Napi Kasus Curanmor Ditangkap di Probolinggo http://t.co/0dDoO73Xy6 Oct 14, 2015	detiknews	news.detik.com
17	Rupiah Diperkirakan Kembali Melemah Esok Hari http://t.co/nj6KwgfXRc via @detikfinance Oct 14, 2015	detikfinance	finance.detik.com

18	Hingga Malam Hari, Tim SAR Gabungan Masih Terus Sisir Danau Toba http://t.co/lwMNV3g3iA Oct 14, 2015	detiknews	news.detik.com
19	Penyelenggara Final Piala Presiden Disarankan Sebar Maksimal 70 Ribu Tiket http://t.co/kT4zbDOf4L via @detiksport http://t.co/9RKqaDDyli Oct 14, 2015	detiksport	*sport*.detik.com
20	Wakil Ketua MKD DPR: Bila Kop Presiden Palsu, Pemerintah Pasti Bereaksi http://t.co/wQXZDKwhNs http://t.co/x9y7aw0N4K Oct 14, 2015	detiknews	news.detik.com

Hasil dari *crawling* data *tweets* ini secara keseluruhan ada 14 topik. Namun dalam penelitian ini hanya digunakan 4 kelompok topik. Topik yang akan di prediksi, yaitu sebagai berikut:

1. detikfood, topik ini berisi informasi tentang resep makanan dan kuliner.
2. detikinet, Memuat informasi teknologi informasi.
3. detikoto, Memuat informasi mengenai otomotif.
4. detiksport, Berisi info olahraga termasuk sepakbola.

Selanjutnya hilangkan atribut *tweets* yang tidak diperlukan, data teks yang mengandung URL, *username*, dan tanggal terlebih dahulu dihilangkan. Contohnya seperti yang tertera pada gambar 4.1. Sehingga data yang nanti digunakan hanya tersisa teks pendek yang mengandung isi topik tertentu beserta judul kelompok topiknya.

No	Tweets
1	Ronaldo Yakin Kariernya Masih Panjang, Mimpikan Pensiun di Madrid http://t.co/WL58LLfxOq via @detiksport http://t.co/yBpswbPn5L Oct 14, 2015
2	Djokovic, Wawrinka Melaju ke Babak Ketiga http://t.co/pnQtK2FvAa via @detiksport Oct 14, 2015
3	Muhaimin: Jangan Ada Lagi Konflik Antar Umat Beragama http://t.co/GUf1kamO2I Oct 14, 2015
4	Melihat Lebih Dekat PLTA Terbesar di Indonesia yang Dibangun di Perut Bumi http://t.co/QjsJuCViaO via @detikfinance Oct 14, 2015

5	Clemmons: Ketika Ayam Goreng Tepung Bertemu dengan Strawberry http://t.co/U7Z0BbOlGO http://t.co/4UpZ7oxE7g Oct 14, 2015

No	Tweets	Kelompok
1	Ronaldo Yakin Kariernya Masih Panjang, Mimpikan Pensiun di Madrid	@detiksport
2	Djokovic, Wawrinka Melaju ke Babak Ketiga	@detiksport
3	Muhaimin: Jangan Ada Lagi Konflik Antar Umat Beragama	News
4	Melihat Lebih Dekat PLTA Terbesar di Indonesia yang Dibangun di Perut Bumi	@detikfinance
5	Clemmons: Ketika Ayam Goreng Tepung Bertemu dengan Strawberry	News

Gambar 4.1 Contoh Proses Penghilangan Atribut yang Tidak Diperlukan

Setelah itu data tweets tersebut harus dipindahkan kedalam format .txt, karena program bantu yang dibuat hanya menerima file .txt, didalam matlab karena menggunakan

4.2. Pengujian

Data masukkan untuk melakukan *clustering* dalam periode satu bulan sebanyak 1585 *tweets.* Data merupakan bobot dari tiap dokumen teks pada tanggal 7 Oktober 2015 sampai 21 Oktober 2015. Data yang digunakan hanyalah data yang memiliki kategori detik *food*, detik *sport*, detik oto, dan detik inet.

4.2.1. Pembobotan TF-IDF

4.2.1.1. Hasil *Stopword removal* dan Menghitung *Term Frequency*

Di tahap ini akan awal mula akan dibuang kata-kata yang tidak memiliki makna, kata-kata yang tidak bermakna tersebut sudah dikumpulkan dalam *list stopword removal*, *list stopword removal* dapat dilihat pada bagian lampiran.

Setelah itu akan dilakukan penghitungan *term frequency*, hasilnya dari perhitungan term frequency dapat dilihat pada tabel berikut:

Tabel 4.2 Hasil Term Frequency dan Stopword Removal

No	Term	Dt 1	Dt 2	Dt 3	Dt 4	Dt 5	Dt 6	Dt 7	Dt 8	Dt ….	Dt 1583	Dt 1584	Dt 1585
1		1	1	1	1	1	0	0	0	....	0	0	0
2	a	0	0	0	0	0	0	0	0	....	0	0	0
3	abaikan	0	0	0	0	0	0	0	0	....	0	0	0
4	abbey	0	0	0	0	0	0	0	0	....	0	0	0
5	abcdefghijklmnopqrstuvwxyz	0	0	0	0	0	0	0	0	....	0	0	0
6	abeng	0	0	0	0	0	0	0	0	....	0	0	0
7	absen	0	0	0	0	0	0	0	0	....	0	0	0
8	abu	0	0	0	0	0	0	0	0	....	0	0	0
9	academy	0	0	0	0	0	0	0	0	....	0	0	0
10	ace	0	0	0	0	0	0	0	0	....	0	0	0
11	aceh	0	0	0	0	0	0	0	0	....	0	0	0
12	acer	0	0	0	0	0	0	0	0	....	0	0	0
13	achmad	0	0	0	0	0	0	0	0	....	0	0	0
14	ada	0	0	0	0	0	0	0	1	....	0	0	0
15	adakan	0	0	0	0	0	0	0	0	....	0	0	0
16	adaptasi	0	0	0	0	0	0	0	0	....	0	0	0
17	adapter	0	0	0	0	0	0	0	0	....	0	0	0
18	adegan	0	0	0	0	0	0	0	0	....	0	0	0
19	ademnya	0	0	0	0	0	0	0	0	....	0	0	0
20	adil	0	0	0	0	0	0	0	0	....	0	0	0
21	adu	0	0	0	0	0	0	0	0	....	0	0	0

No	Term	Dt 1	Dt 2	Dt 3	Dt 4	Dt 5	Dt 6	Dt 7	Dt 8	Dt ….	Dt 1583	Dt 1584	Dt 1585
22	adware	0	0	0	0	0	0	0	0	....	0	0	0
23	afrika	0	0	0	0	0	0	0	0	....	0	0	0
24	again	0	0	0	0	0	0	0	0	....	0	0	0
25	agama	0	0	0	0	0	0	0	0	....	0	0	0
26	agar	0	0	0	0	0	0	0	0	....	0	0	0
27	agen	0	0	0	0	0	0	0	0	....	0	0	0
28	agenda	0	0	0	0	0	0	0	0	....	0	0	0
29	agendakan	0	0	0	0	0	0	0	0	....	0	0	0
30	agresif	0	0	0	0	0	0	0	0	....	0	0	0
31	aguero	0	0	0	0	0	0	0	0	....	0	0	0
32	ahmad	0	0	0	0	0	0	0	0	....	0	0	0
33	ahmed	0	0	0	0	0	0	0	0	....	0	0	0
34	ahok	0	0	0	0	0	0	0	0	....	0	0	0
35	ahsan	0	0	0	0	0	0	0	0	....	0	0	0
36	air	0	0	0	0	0	0	0	0	....	0	0	0
37	ajak	0	0	0	0	0	0	0	0	....	0	0	0
38	ajang	0	0	0	0	0	0	0	0	....	0	0	0
39	ajari	0	0	0	0	0	0	0	0	....	0	0	0
40	ajukan	0	0	0	0	0	0	0	0	....	0	0	0
41	akal	0	0	0	0	0	0	0	0	....	0	0	0
42	akar	0	0	0	0	0	0	0	0	....	0	0	0
43	akibat	0	0	0	0	0	0	0	0	....	0	0	0
44	akibatnya	0	0	0	0	0	0	0	0	....	0	0	0

No	Term	Dt 1	Dt 2	Dt 3	Dt 4	Dt 5	Dt 6	Dt 7	Dt 8	Dt ….	Dt 1583	Dt 1584	Dt 1585
45	akrab	0	0	0	0	0	0	0	0	….	0	0	0
46	aksi	0	0	0	0	0	0	0	0	….	0	0	0
47	akui	0	0	0	0	0	0	0	0	….	0	0	0
48	akuisisi	0	0	0	0	0	0	0	0	….	0	0	0
49	ala	0	0	0	0	0	0	0	0	….	0	0	0
50	alaba	0	0	0	0	0	0	0	0	….	0	0	0
…..	….	….	…..	…..	…..	…..	….	….	…..	…..	…..	….	…..
3295	windows	0	0	0	0	0	0	0	0	….	0	0	0
3296	windroid	0	0	0	0	0	0	0	0	….	0	0	0
3297	wine	0	0	0	0	0	0	0	0	….	0	0	0
3298	wireless	0	0	0	0	0	0	0	0	….	0	0	0
3299	wisma	0	0	0	0	0	0	0	0	….	0	0	0
3300	wisman	0	0	0	0	0	0	0	0	….	0	0	0
3301	wolfsburg	0	0	0	0	0	0	0	0	….	0	0	0
3302	wolves	0	0	0	0	0	0	0	0	….	0	0	0
3303	works	0	0	0	0	0	0	0	0	….	0	0	0
3304	wouw	0	0	0	0	0	0	0	0	….	0	0	0
3305	wsbk	0	0	0	0	0	0	0	0	….	0	0	0
3306	wujud	0	0	0	0	0	0	0	0	….	0	0	0
3307	wujudkan	0	0	0	0	0	0	0	0	….	0	0	0
3308	x	0	0	0	0	0	0	0	0	….	0	0	0
3309	xavi	0	0	0	0	0	0	0	0	….	0	0	0
3310	xbox	0	0	0	0	0	0	0	0	….	0	0	0

No	Term	Dt 1	Dt 2	Dt 3	Dt 4	Dt 5	Dt 6	Dt 7	Dt 8	Dt ….	Dt 1583	Dt 1584	Dt 1585
3311	xc	0	0	0	0	0	0	0	0	….	0	0	0
3312	xenia	0	0	0	0	0	0	0	0	….	0	0	0
3313	xiaomi	0	0	0	0	0	0	0	0	….	0	0	0
3314	xl	0	0	0	0	0	0	0	0	….	0	0	0
3315	xperia	0	0	0	0	0	0	0	0	….	0	0	0
3316	xuerui	0	0	0	0	0	0	0	0	….	0	0	0
3317	xueuri	0	0	0	0	0	0	0	0	….	0	0	0
3318	y	0	0	0	0	0	0	0	0	….	0	0	0
3319	yamaha	0	0	0	1	0	0	0	0	….	0	0	0
3320	yaya	0	0	0	0	0	0	0	0	….	0	0	0
3321	yellow	0	0	0	0	0	0	0	0	….	0	0	0
3322	yogya	0	0	0	0	0	0	0	0	….	0	0	0
3323	yogyakarta	0	0	0	0	0	0	0	0	….	0	0	0
3324	yordan	0	0	0	0	0	0	0	0	….	0	0	0
3325	yos	0	0	0	0	0	0	0	0	….	0	0	0
3326	youth	0	0	0	0	0	0	0	0	….	0	0	0
3327	youtube	0	0	0	0	0	0	0	0	….	0	0	0
3328	yuk	0	0	0	0	0	0	0	0	….	0	0	0
3329	yummy	0	0	0	0	0	0	0	0	….	0	0	0
3330	yunani	0	0	0	0	0	0	0	0	….	0	0	0
3331	yunus	0	0	0	0	0	0	0	0	….	0	0	0
3332	z	0	0	0	0	0	0	0	0	….	0	0	0
3333	zaitun	0	0	0	0	0	0	0	0	….	0	0	0

No	Term	Dt 1	Dt 2	Dt 3	Dt 4	Dt 5	Dt 6	Dt 7	Dt 8	Dt ….	Dt 1583	Dt 1584	Dt 1585
3334	zarco	0	0	0	0	0	0	0	0	....	0	0	0
3335	zargari	0	0	0	0	0	0	0	0	....	0	0	0
3336	zenit	0	0	0	0	0	0	0	0	....	0	0	0
3337	zero	0	0	0	0	0	0	0	0	....	0	0	0
3338	zidane	0	0	0	0	0	0	0	0	....	0	0	0
3339	zola	0	0	0	0	0	0	0	0	....	1	0	0
3340	zombie	0	0	0	0	0	0	0	0	....	0	0	0
3341	zucchini	0	0	0	0	0	0	0	0	....	0	1	0
3342	zulham	0	0	0	0	0	0	0	0	....	0	0	1
3343	zx	0	0	0	0	0	0	0	0	....	0	0	0
3344	zzr	0	0	0	0	0	0	0	0	....	0	0	0

4.2.1.2. Hasil *DF* dan *IDF*

DF didapat dari jumlah masing-masing term dari keseluruhan dokumen, sedangkan *IDF* didapat dari perhitungan *DF* dengan jumlah dokumen yang ada menggunakan rumus *IDF*, hasil dari *DF* dan *IDF* dapat dilihat pada tabel berikut:

Tabel 4.3 Hasil *DF* dan *IDF*

No	Term	DF	IDF		No	Term	DF	IDF		No	Term	DF	IDF
1		1	3.200029		35	ahsan	4	2.597969		3312	Xenia	1	3.200029
2	a	10	2.200029		36	air	3	2.722908		3313	Xiaomi	2	2.898999
3	abaikan	1	3.200029		37	ajak	2	2.898999		3314	Xl	2	2.898999
4	abbey	1	3.200029		38	ajang	1	3.200029		3315	Xperia	2	2.898999

No	Term	DF	IDF		No	Term	DF	IDF		No	Term	DF	IDF
5	Abcdefghijklm nopqrstuvwxyz	1	3.200029		39	ajari	1	3.200029		3316	Xuerui	2	2.898999
6	abeng	1	3.200029		40	ajukan	3	2.722908		3317	xueuri	1	3.200029
7	absen	11	2.158637		41	akal	1	3.200029		3318	y	1	3.200029
8	abu	2	2.898999		42	akar	1	3.200029		3319	yamaha	23	1.838301
9	academy	2	2.898999		43	akibat	1	3.200029		3320	yaya	2	2.898999
10	ace	1	3.200029		44	akibatnya	2	2.898999		3321	yellow	1	3.200029
11	aceh	2	2.898999		45	akrab	2	2.898999		3322	yogya	2	2.898999
12	acer	5	2.501059		46	aksi	1	3.200029		3323	yogyakarta	1	3.200029
13	achmad	1	3.200029		47	akui	5	2.501059		3324	yordan	1	3.200029
14	ada	1	3.200029		48	akuisisi	4	2.597969		3325	yos	1	3.200029
15	adakan	3	2.722908		49	ala	3	2.722908		3326	youth	1	3.200029
16	adaptasi	1	3.200029		50	alaba	1	3.200029		3327	youtube	2	2.898999
17	adapter	1	3.200029		...	…..	….	…		3328	yuk	5	2.501059
18	adegan	1	3.200029		3295	windows	7	2.354931		3329	yummy	1	3.200029
19	ademnya	1	3.200029		3296	windroid	1	3.200029		3330	yunani	1	3.200029
20	adil	3	2.722908		3297	wine	2	2.898999		3331	yunus	1	3.200029
21	adu	1	3.200029		3298	wireless	1	3.200029		3332	z	3	2.722908
22	adware	1	3.200029		3299	wisma	1	3.200029		3333	zaitun	4	2.597969
23	afrika	1	3.200029		3300	wisman	1	3.200029		3334	zarco	1	3.200029
24	again	1	3.200029		3301	wolfsburg	1	3.200029		3335	zargari	1	3.200029
25	agama	1	3.200029		3302	wolves	1	3.200029		3336	zenit	1	3.200029
26	agar	1	3.200029		3303	works	1	3.200029		3337	zero	1	3.200029
27	agen	3	2.722908		3304	wouw	1	3.200029		3338	zidane	1	3.200029

No	Term	DF	IDF		No	Term	DF	IDF		No	Term	DF	IDF
28	agenda	1	3.200029		3305	wsbk	1	3.200029		3339	zola	1	3.200029
29	agendakan	1	3.200029		3306	wujud	2	2.898999		3340	zombie	1	3.200029
30	agresif	2	2.898999		3307	wujudkan	1	3.200029		3341	zucchini	1	3.200029
31	aguero	6	2.421878		3308	x	9	2.245787		3342	zulham	1	3.200029
32	ahmad	2	2.898999		3309	xavi	3	2.722908		3343	zx	1	3.200029
33	ahmed	1	3.200029		3310	xbox	2	2.898999		3344	zzr	1	3.200029
34	ahok	5	2.501059		3311	xc	1	3.200029					

4.2.1.3. Hasil *TF-IDF*

Berikut merupakan hasil dari *TF-IDF*:

Tabel 4.4 Hasil *TF-IDF*

No	Term	Wt 1	Wt 2	Wt 3	Wt 4	Wt 5	Wt 6	Wt 7	Wt 8	Wt…	Wt 1583	Wt 1584	Wt 1585
1		3.200029	3.200029	3.200029	3.200029	3.200029	0	0	0	…..	0	0	0
2	a	0	0	0	0	0	0	0	0	…..	0	0	0
3	abaikan	0	0	0	0	0	0	0	0	…..	0	0	0
4	abbey	0	0	0	0	0	0	0	0	…..	0	0	0
5	Abcdefghijklmnopqrstuvwxyz	0	0	0	0	0	0	0	0	…..	0	0	0
6	abeng	0	0	0	0	0	0	0	0	…..	0	0	0
7	absen	0	0	0	0	0	0	0	0	…..	0	0	0
8	abu	0	0	0	0	0	0	0	0	…..	0	0	0
9	academy	0	0	0	0	0	0	0	0	…..	0	0	0
10	ace	0	0	0	0	0	0	0	0	…..	0	0	0

No	Term	Wt 1	Wt 2	Wt 3	Wt 4	Wt 5	Wt 6	Wt 7	Wt 8	Wt…	Wt 1583	Wt 1584	Wt 1585
11	aceh	0	0	0	0	0	0	0	0	…..	0	0	0
12	acer	0	0	0	0	0	0	0	0	…..	0	0	0
13	achmad	0	0	0	0	0	0	0	0	…..	0	0	0
14	ada	0	0	0	0	0	0	0	3.200029	…..	0	0	0
15	adakan	0	0	0	0	0	0	0	0	…..	0	0	0
16	adaptasi	0	0	0	0	0	0	0	0	…..	0	0	0
17	adapter	0	0	0	0	0	0	0	0	…..	0	0	0
18	adegan	0	0	0	0	0	0	0	0	…..	0	0	0
19	ademnya	0	0	0	0	0	0	0	0	…..	0	0	0
20	adil	0	0	0	0	0	0	0	0	…..	0	0	0
21	adu	0	0	0	0	0	0	0	0	…..	0	0	0
22	adware	0	0	0	0	0	0	0	0	…..	0	0	0
23	afrika	0	0	0	0	0	0	0	0	…..	0	0	0
24	again	0	0	0	0	0	0	0	0	…..	0	0	0
25	agama	0	0	0	0	0	0	0	0	…..	0	0	0
26	agar	0	0	0	0	0	0	0	0	…..	0	0	0
27	agen	0	0	0	0	0	0	0	0	…..	0	0	0
28	agenda	0	0	0	0	0	0	0	0	…..	0	0	0
29	agendakan	0	0	0	0	0	0	0	0	…..	0	0	0
30	agresif	0	0	0	0	0	0	0	0	…..	0	0	0
31	aguero	0	0	0	0	0	0	0	0	…..	0	0	0
32	ahmad	0	0	0	0	0	0	0	0	…..	0	0	0
33	ahmed	0	0	0	0	0	0	0	0	…..	0	0	0
34	ahok	0	0	0	0	0	0	0	0	…..	0	0	0

No	Term	Wt 1	Wt 2	Wt 3	Wt 4	Wt 5	Wt 6	Wt 7	Wt 8	Wt…	Wt 1583	Wt 1584	Wt 1585
35	ahsan	0	0	0	0	0	0	0	0	…..	0	0	0
36	air	0	0	0	0	0	0	0	0	…..	0	0	0
37	ajak	0	0	0	0	0	0	0	0	…..	0	0	0
38	ajang	0	0	0	0	0	0	0	0	…..	0	0	0
39	ajari	0	0	0	0	0	0	0	0	…..	0	0	0
40	ajukan	0	0	0	0	0	0	0	0	…..	0	0	0
41	akal	0	0	0	0	0	0	0	0	…..	0	0	0
42	akar	0	0	0	0	0	0	0	0	…..	0	0	0
43	akibat	0	0	0	0	0	0	0	0	…..	0	0	0
44	akibatnya	0	0	0	0	0	0	0	0	…..	0	0	0
45	akrab	0	0	0	0	0	0	0	0	…..	0	0	0
46	aksi	0	0	0	0	0	0	0	0	…..	0	0	0
47	akui	0	0	0	0	0	0	0	0	…..	0	0	0
48	akuisisi	0	0	0	0	0	0	0	0	…..	0	0	0
49	ala	0	0	0	0	0	0	0	0	…..	0	0	0
50	alaba	0	0	0	0	0	0	0	0	…..	0	0	0
…..	….	…..	….	…..	….	…..	….	…..	….	…..	….	…..	….
3295	windows	0	0	0	0	0	0	0	0	…..	0	0	0
3296	windroid	0	0	0	0	0	0	0	0	…..	0	0	0
3297	wine	0	0	0	0	0	0	0	0	…..	0	0	0
3298	wireless	0	0	0	0	0	0	0	0	…..	0	0	0
3299	wisma	0	0	0	0	0	0	0	0	…..	0	0	0
3300	wisman	0	0	0	0	0	0	0	0	…..	0	0	0
3301	wolfsburg	0	0	0	0	0	0	0	0	…..	0	0	0

No	Term	Wt 1	Wt 2	Wt 3	Wt 4	Wt 5	Wt 6	Wt 7	Wt 8	Wt…	Wt 1583	Wt 1584	Wt 1585
3302	wolves	0	0	0	0	0	0	0	0	…..	0	0	0
3303	works	0	0	0	0	0	0	0	0	…..	0	0	0
3304	wouw	0	0	0	0	0	0	0	0	…..	0	0	0
3305	wsbk	0	0	0	0	0	0	0	0	…..	0	0	0
3306	wujud	0	0	0	0	0	0	0	0	…..	0	0	0
3307	wujudkan	0	0	0	0	0	0	0	0	…..	0	0	0
3308	x	0	0	0	0	0	0	0	0	…..	0	0	0
3309	xavi	0	0	0	0	0	0	0	0	…..	0	0	0
3310	xbox	0	0	0	0	0	0	0	0	…..	0	0	0
3311	xc	0	0	0	0	0	0	0	0	…..	0	0	0
3312	xenia	0	0	0	0	0	0	0	0	…..	0	0	0
3313	xiaomi	0	0	0	0	0	0	0	0	…..	0	0	0
3314	xl	0	0	0	0	0	0	0	0	…..	0	0	0
3315	xperia	0	0	0	0	0	0	0	0	…..	0	0	0
3316	xuerui	0	0	0	0	0	0	0	0	…..	0	0	0
3317	xueuri	0	0	0	0	0	0	0	0	…..	0	0	0
3318	y	0	0	0	0	0	0	0	0	…..	0	0	0
3319	yamaha	0	0	0	1.838301	0	0	0	0	…..	0	0	0
3320	yaya	0	0	0	0	0	0	0	0	…..	0	0	0
3321	yellow	0	0	0	0	0	0	0	0	…..	0	0	0
3322	yogya	0	0	0	0	0	0	0	0	…..	0	0	0
3323	yogyakarta	0	0	0	0	0	0	0	0	…..	0	0	0
3324	yordan	0	0	0	0	0	0	0	0	…..	0	0	0
3325	yos	0	0	0	0	0	0	0	0	…..	0	0	0

No	Term	Wt 1	Wt 2	Wt 3	Wt 4	Wt 5	Wt 6	Wt 7	Wt 8	Wt…	Wt 1583	Wt 1584	Wt 1585
3326	youth	0	0	0	0	0	0	0	0	…..	0	0	0
3327	youtube	0	0	0	0	0	0	0	0	…..	0	0	0
3328	yuk	0	0	0	0	0	0	0	0	…..	0	0	0
3329	yummy	0	0	0	0	0	0	0	0	…..	0	0	0
3330	yunani	0	0	0	0	0	0	0	0	…..	0	0	0
3331	yunus	0	0	0	0	0	0	0	0	…..	0	0	0
3332	z	0	0	0	0	0	0	0	0	…..	0	0	0
3333	zaitun	0	0	0	0	0	0	0	0	…..	0	0	0
3334	zarco	0	0	0	0	0	0	0	0	…..	0	0	0
3335	zargari	0	0	0	0	0	0	0	0	…..	0	0	0
3336	zenit	0	0	0	0	0	0	0	0	…..	0	0	0
3337	zero	0	0	0	0	0	0	0	0	…..	0	0	0
3338	zidane	0	0	0	0	0	0	0	0	…..	0	0	0
3339	zola	0	0	0	0	0	0	0	0	…..	3.200029	0	0
3340	zombie	0	0	0	0	0	0	0	0	…..	0	0	0
3341	zucchini	0	0	0	0	0	0	0	0	…..	0	3.200029	0
3342	zulham	0	0	0	0	0	0	0	0	…..	0	0	3.200029
3343	zx	0	0	0	0	0	0	0	0	…..	0	0	0
3344	Zzr	0	0	0	0	0	0	0	0	…..	0	0	0

4.2.2. Uji Hasil *Clustering* dengan *K-Means*

4.2.2.1. Uji Menggunakan Centroid yang Sesuai dengan *Real Cluster*

Pada awal pengujian pada *K-means clustering* akan dilakukan percobaan dengan menggunakan pemilihan *centroid* dari data tweets yang sesuai mewakili anggota masing-masing kelompoknya. Uji coba yang dilakukan menggunakan data ke-4 untuk mewakili kelompok Oto, data ke-331 untuk mewakili kelompok *Sport*, data ke-865 untuk mewakili kelompok Inet, data ke-1317 untuk mewakili kelompok *Food*.

Tabel 4.5 Hasil Data Pengujian K-Means dengan Centroid yang Sesuai

		Real cluster				
		oto	*Sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	68	120	67	62	317
	B	190	707	139	187	1223
	C	8	20	7	9	44
	D	0	0	1	0	1
Hasil prediksi kelompok		B	B	B	B	

Pengelompokan sebuah *real cluster* diambil berdasarkan hasil yang paling dominan dari kesemua *predictive cluster*. Dari hasil pada tabel 4. Terlihat bahwa semua prediksi data yang dominan mengalami penumpukan pada kelompok B, dengan jumlah oto sebanyak 190, *sport* sebanyak 707, *food* sebanyak 139, dan inet sebanyak 187, dengan hasil seperti ini maka perlu dilakukan perhitungan lebih lanjut menggunakan *naïve bayes*, dengan memilih jumlah anggota kelompok prediksi yang paling banyak sebagai prioritas dari sebuah kelompok tersebut prediksi tersebut.

Tabel 4.6 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* Menggunakan Centroid Sesuai Dengan *Cluster*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.378549	A	120			
	0.578087	B	707	76	75	71
	0.454545	C	20	Oto	*food*	Inet
	0	D	0			
Oto	0.345178	A	68	8	9	
	0.333333	C	8	*food*	inet	
	0	D	0			
Inet	0.4375	C	9	*Food*		
	1	D	0	7		
Food		C	7			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* B dengan jumlah data sebanyak 707 data, kelompok Oto masuk kedalam kelompok A dengan jumlah data sebanyak 68 data, kelompok Inet masuk kedalam kelompok D namun pada kelompok Inet tidak terdapat data yang bisa dikelompokan, kelompok *food* masuk kedalam kelompok C dengan jumlah data sebanyak 7 data.

Dari perhitungan diatas sudah diketahui pengelompokan datanya, setelah itu dilakukakn pengujian menggunakan tabel *confusion matrix*. Pada tabel 4. merupakan tabel hasil *predictive* dan *real cluster* yang siap untuk dilakukan penghitungan *confusion matrix*.

Tabel 4.7 Hasil *Cluster* Setelah Ditentukan Menggunakan *Naïve Bayes* dengan *Centroid* yang Sesuai

		Predictive cluster			
		oto	*sport*	*food*	Inet
Real	Oto	68	190	8	0

	Sport	120	707	20	0
	Food	67	139	7	1
	Inet	62	187	9	0

Berikut merupakan hasil perhitungan dari *confusion matrix*

Tabel 4.8 Hasil Perhitungan *Accuracy, Precision, Recall* pada K-Means dengan Centroid yang Sesuai

Accuracy	0.493375	49%
precision oto	0.214511	21%
precision sport	0.578087	58%
precision food	0.159091	16%
precision inet	0	0%
recall oto	0.255639	26%
recall *sport*	0.834711	83%
recall *food*	0.03271	3%
recall inet	0	0%

Dari hasil tabel 4.5 terlihat bahwa akurasinya hanya 49%, namun memiliki recall dan *precision* yang rendah pada kelompok tertentu, hal ini disebabkan karena data dominan menumpuk pada salah salah satu kelompok.

4.2.2.2. Uji Menggunakan *Centroid* yang Diambil Acak Tidak Mewakil *Real Cluster*

Pengujian pada *K-means clustering* selanjutnya akan dilakukan percobaan dengan menggunakan pemilihan *centroid* dari data tweets secara acak yang tidak sesuai mewakili anggota masing-masing kelompoknya. Centroid yang digunakan pada uji coba menggunakan data ke 1, 500, 1000, dan 1500.

Tabel 4.9 Hasil Data Pengujian K-Means dengan Centroid Secara Acak

		Real cluster				
		oto	*sport*	*food*	Inet	Total *predictive cluster*
Predictive Cluster	A	252	795	204	253	1504
	B	1	0	0	0	1

	C	13	51	10	5	79
	D	0	1	0	0	1
Hasil prediksi kelompok		A	A	A	A	

Pengelompokan sebuah *real cluster* diambil berdasarkan hasil yang paling dominan dari kesemua *predictive cluster*. Dari hasil pada tabel 4. Terlihat bahwa semua prediksi data yang dominan mengalami penumpukan pada kelompok A sama persis dengan percobaan pertama, dengan jumlah oto sebanyak 252, *sport* sebanyak 795, *food* sebanyak 204, dan inet sebanyak 253, sama seperti percobaan pertama pada *K-means* maka perlu dilakukan perhitungan lebih lanjut menggunakan *naïve bayes*, untuk mendapat *predictive cluster* yang tepat bagi masing-masing kelompok.

Tabel 4.10 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* Menggunakan Centroid Secara Acak

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.378549	A	120			
	0.578087	B	707	76	75	71
	0.454545	C	20	oto	*food*	Inet
	0	D	0			
Oto	0.345178	A	68	8	9	
	0.333333	C	8	*food*	inet	
	0	D	0			
Inet	0.4375	C	9	*Food*		
	1	D	0	7		
Food		C	7			

A dengan jumlah data sebanyak 68 data, kelompok Inet masuk kedalam kelompok D namun pada kelompok Inet tidak terdapat data yang bisa dikelompokan, kelompok *food* masuk kedalam kelompok C dengan jumlah data sebanyak 7 data.

Dari perhitungan diatas sudah diketahui pengelompokan datanya, setelah itu dilakukakn pengujian menggunakan tabel *confusion matrix*. Pada tabel 4. merupakan tabel hasil *predictive* dan *real cluster* yang siap untuk dilakukan penghitungan *confusion matrix*.

Tabel 4.11 Hasil *Cluster* Setelah Ditentukan Menggunakan *Naïve Bayes* dengan *Centroid* Secara Acak

		Predictive cluster			
		oto	*sport*	*food*	inet
Real Cluster	Oto	1	252	0	13
	Sport	0	795	1	51
	Food	0	204	0	10
	Inet	0	253	0	5

Berikut merupakan hasil perhitungan dari *confusion matrix*

Tabel 4.12 Hasil Perhitungan Accuracy, Precision, Dan Recall pada K-Means dengan Centroid Secara Acak

Accuracy	0.505363	51%
precision oto	1	100%
precision sport	0.52859	53%
precision food	0	0%
precision inet	0.063291	6%
recall oto	0.003759	0%
recall *sport*	0.938607	94%
recall *food*	0	0%
recall inet	0.01938	2%

Dari hasil tabel 4. 2 terlihat bahwa akurasinya meningkat 2% menjadi 51%, namun nilai pengujian ini lebih buruk dari pada pengujian pertama dikarenakan memiliki nilai *precision* dan *recall* yang sangat buruk. Sehingga dianggap hasil

pengelompokannya terbilang buruk, hal ini disebabkan karena data dominan menumpuk pada salah salah satu kelompok.

4.2.3. Uji Hasil *Clustering* dengan *LSI*

4.2.3.1. Uji Menggunakan *Threshold* dengan Nilai 0.95 dan Menggunakan *SVD* Dengan Nilai 2

Pada awal pengujian akan dilakukan percobaan dengan menggunakan *threshold* dengan nilai 0.95 dan *SVD* dengan nilai 2, *threshold* disini diguanakan sebagai pembatas dokumen apakah dokumen tersebut masuk kedalam sebuah kelompok atau tidak, sedangkan *SVD* digunakan untuk memreduksi dimensi dari dokumen. Hasil yang didapat dari uji coba pertama sebagai berikut:

Tabel 4.13 Hasil Dari Perhitungan Menggunakan *Threshold* 0.95 Dan 2 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	55	115	50	95	315
	B	94	334	88	58	574
	C	62	197	50	44	353
	D	32	103	12	21	168
	E	23	98	14	40	175
Hasil prediksi kelompok		B	B	B	A	

Pengelompokan sebuah *real cluster* diambil berdasarkan hasil yang paling dominan dari kesemua *predictive cluster*. Dari hasil pada tabel 4. terlihat bahwa hasil *cluster* oto dominan berada di kelompok B dengan data sebanyak 94, *cluster sport* dominan berada di kelompok B dengan data sebanyak 334, *cluster food* dominan berada di kelompok B dengan data sebanyak 88, *cluster* inet dominan berada di kelompok B dengan data sebanyak 95. Maka dapat terlihat, disini hanya

terbentuk 2 *cluster* yaitu A dan B dengan penumpukan *real cluster* pada satu *real cluster*.

Karena terlalu banyak penumpukan *real cluster* pada satu kelompok *predictive cluster*. Maka perhitungan pengelompokan harus dilanjutkan dengan metode *naïve bayes*, yaitu dengan cara memasukan kelompok data yang paling besar untuk didahulukan melakukan perhitungan pemilihan *predictive cluster* berdasarkan yang paling dominan. Karena terbentuk 5 *predictive cluster* maka kelompok E akan dimasukan kedalam kelompok A. Sehingga hasil kelompok tabel menjadi berikut:

Tabel 4.14 Penggabungan *Cluster* Lain Kedalam Satu *Cluster* pada *Threshold* 0.95 dan 2 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	78	213	64	135	490
	B	94	334	88	58	574
	C	62	197	50	44	353
	D	32	103	12	21	168

Berdasarkan tabel diatas dilakukan perhitungan menggunakan *naïve bayes* dan didapat hasil perhitungan sebagai berikut:

Tabel 4.15 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* 0.95 dan 2 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.434694	A	213			
	0.581882	B	334	234	202	237
	0.558074	C	197	oto	*food*	inet
	0.613095	D	103			
Oto	0.281588	A	78	152	193	
	0.391667	B	94	*food*	inet	
	0.397436	C	62			

Inet	0.321608	A	135	*Food*		
	0.60274	B	58	64		
food		A	64			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* D dengan jumlah data sebanyak 103 data, kelompok Oto masuk kedalam kelompok C dengan jumlah data sebanyak 62 data, kelompok Inet masuk kedalam kelompok B dengan jumlah data sebanyak 58 data, kelompok *food* masuk kedalam kelompok A dengan jumlah data sebanyak 64 data.

Dari perhitungan diatas sudah diketahui pengelompokan datanya, setelah itu dilakukakn pengujian menggunakan tabel *confusion matrix*. Dengan *confusion matrix* akan dicari tiga nilai yaitu *accuracy*, *recall* dan *precision*. Berikut merupakan tabel hasil *predictive cluster* dan *real cluster*:

Tabel 4.16 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes pada Threshold* 0.95 dan 2 *SVD*

		Predictive cluster			
		oto	*sport*	*food*	inet
Real Cluster	Oto	62	32	78	94
	Sport	197	103	213	334
	Food	50	12	64	88
	Inet	44	21	135	58

Tabel 4.17 Hasil Perhitungan *Accuracy, Precision*, Dan *Recall* pada Algoritma LSI dengan *Threshold* 0.95 dan 2 *SVD*

Accuracy	0.181072555	18%
precision oto	0.175637394	18%
precision sport	0.613095238	61%
precision food	0.130612245	13%
precision inet	0.101045296	10%
recall oto	0.233082707	23%

recall *sport*	0.121605667	12%
recall *food*	0.299065421	30%
recall inet	0.224806202	22%

4.2.3.2. Uji Menggunakan *Threshold* dengan Nilai 0.90 dan dengan Menggunakan *SVD* dengan Nilai 2

Pengujian selanjutnya dicoba dengan menggunakan *threshold* dengan nilai 0.90 dan *SVD* dengan nilai 2, dengan penurunan *threshold* akan dilihat bagaimana hasil dari tingkat keakurasiannya, dan hasil percobaannya sebagai berikut:

Tabel 4.18 Hasil Dari Perhitungan Menggunakan *Threshold* 0.90 dan 2 *SVD*

		Real cluster				
		oto	*Sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	61	145	51	112	369
	B	124	426	106	83	739
	C	77	249	53	54	433
	D	4	27	4	9	44
Hasil prediksi kelompok		B	B	B	A	

Dari hasil pada tabel 4. terlihat bahwa hasil *cluster* oto dominan berada di kelompok B dengan data sebanyak 124, *cluster sport* dominan berada di kelompok B dengan data sebanyak 426, *cluster food* dominan berada di kelompok B dengan data sebanyak 106, *cluster* inet dominan berada di kelompok B dengan data sebanyak 112. Dari hasil perhitungan dapat terlihat bahwa sama seperti pada pengujian pertama yang telah terjadi penumpukan *real cluster* pada satu kelompok yang sama. Bahkan hasil pengelompokanpun tidak ada perbedaan dengan pengujian pertama.

Karena hasil jumlah *real cluster* dengan *predictive cluster* berjumlah sama maka tidak diperlukan penggabungan jumlah *predictive cluster*, sehingga bisa dilanjutkan perhitungan pengelompokan dengan *naïve bayes*. Setelah dilakukan perhitungan menggunakan *naïve bayes* dan didapat hasil perhitungan sebagai berikut:

Tabel 4.19 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* 0.90 dan 2 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.39295393	A	145			
	0.576454668	B	426	262	210	249
	0.575057737	C	249	Oto	*Food*	Inet
	0.613636364	D	27			
Oto	0.272321429	A	61	157	195	
	0.396166134	B	124	*Food*	Inet	
	0.418478261	C	77			
Inet	0.312883436	A	112	*Food*		
	0.560846561	B	83	51		
Food		A	51			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* D dengan jumlah data sebanyak 27 data, kelompok Oto masuk kedalam kelompok C dengan jumlah data sebanyak 77 data, kelompok Inet masuk kedalam kelompok B dengan jumlah data sebanyak 83 data, kelompok *food* masuk kedalam kelompok A dengan jumlah data sebanyak 51 data.

Tabel 4.20 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* 0.90 dan 2 *SVD*

	Predictive cluster			
	oto	*sport*	*food*	inet

Real Cluster	Oto	77	4	61	124
	Sport	249	27	145	426
	Food	53	4	51	106
	Inet	54	9	112	83

Berikut merupakan hasil perhitungan *confusion matrix* dari table 4.20

Tabel 4.21 Hasil Perhitungan *Accuracy, Precision, Recall* pada Algoritma LSI dengan *Threshold* 0.90 dan 2 *SVD*

Accuracy	0.150158	15%
precision oto	0.177829	18%
precision sport	0.613636	61%
precision food	0.138211	14%
precision inet	0.112314	11%
recall oto	0.289474	29%
recall *sport*	0.031877	3%
recall *food*	0.238318	24%
recall inet	0.321705	32%

4.2.3.3. Uji Menggunakan *Threshold* dengan Nilai 0.80 dan Menggunakan *SVD* dengan Nilai 2

Pengujian selanjutnya dicoba dengan menggunakan *threshold* dengan nilai 0.80 dan *SVD* dengan nilai 2, berikut merupakan hasil percobaannya:

Tabel 4.22 Hasil dari Perhitungan Menggunakan *Threshold* 0.80 dan 2 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	76	209	63	135	369
	B	134	446	102	83	739
	C	56	192	49	40	433
Hasil prediksi kelompok		B	B	B	A	

Dari hasil pada tabel 4. terlihat bahwa hasil *cluster* oto dominan berada di kelompok B dengan data sebanyak 134, *cluster sport* dominan berada di kelompok B dengan data sebanyak 446, *cluster food* dominan berada di kelompok B dengan data sebanyak 102, *cluster* inet dominan berada di kelompok B dengan data sebanyak 135. Dari hasil perhitungan dapat terlihat bahwa sama seperti pada pengujian pertama dan kedua yang telah terjadi penumpukan *real cluster* pada satu kelompok yang sama. Namun pada kali ini terjadi pengurangan *predictive cluster*. Sehingga *real cluster* lebih banyak dari *predictive cluster*.

Karena jumlah *real cluster* lebih banyak dari *predictive cluster* maka tidak diperlukan penggabungan jumlah *predictive cluster*, sehingga bisa dilanjutkan perhitungan pengelompokan dengan *naïve bayes*. Namun dapat dipastikan ada salah satu *real cluster* yang hilang atau tidak dapat terdefinisi kedalam kelompok manapun. Setelah dilakukan perhitungan menggunakan *naïve bayes* maka didapat hasil perhitungan sebagai berikut:

Tabel 4.23 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* 0.80 dan 2 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.432712215	A	209			
	0.583006536	B	446	132	112	175
	0.569732938	C	192	Oto	*Food*	Inet
Inet	0.492701	A	135	56	49	
	0.275862	C	40	Oto	*Food*	
Oto		C	56			
Food		Tidak terdefinisi	0			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* B dengan jumlah data sebanyak 446 data, kelompok Oto masuk kedalam kelompok C dengan jumlah data sebanyak 56 data, kelompok Inet masuk kedalam kelompok A dengan jumlah data sebanyak 135 data, sedangkan kelompok *food* tidak dapat terdefinisi kedalam kelompok apapun karena hanya terdapat 3 *predictive cluster*.

Setelah itu dilakukan perhitungan kinerjanya dengan *confusion matrix*, berikut merupakan tabel hasil *predictive cluster* dan *real cluster*:

Tabel 4.24 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes pada Threshold* 0.80 dan 2 *SVD*

		Predictive cluster			
		oto	*sport*	*food*	inet
Real Cluster	Oto	56	134	0	76
	Sport	192	446	0	209
	Food	49	102	0	63
	Inet	40	83	0	135

Berikut merupakan hasil perhitungan *confusion matrix* dari table 4.24

Tabel 4.25 Hasil Perhitungan *Accuracy, Precision,* dan *Recall* Pada Algoritma LSI *d*engan *Threshold* 0.80 dan 2 *SVD*

Accuracy	0.401893	40%
precision oto	0.166172	17%
precision sport	0.583007	58%
precision food	0	0%
precision inet	0.279503	28%
recall oto	0.210526	21%
recall *sport*	0.526564	53%
recall *food*	0	0%
recall inet	0.523256	52%

4.2.3.4. Uji Menggunakan *Threshold* dengan Nilai 0.70 Dan Menggunakan *SVD* Dengan Nilai 2

Pengujian selanjutnya dicoba dengan menggunakan *threshold* dengan nilai 0.70 dan *SVD* dengan nilai 2, berikut merupakan hasil percobaannya:

Tabel 4.26 Hasil dari Perhitungan Menggunakan *Threshold* 0.70 dan 2 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	108	283	87	151	629
	B	106	402	98	71	677
	C	52	162	29	36	279
Hasil prediksi kelompok		A	B	B	A	

Dari hasil pada tabel 4.23 terlihat bahwa hasil *cluster* oto dominan berada di kelompok A dengan data sebanyak 108, *cluster sport* dominan berada di kelompok B dengan data sebanyak 402, *cluster food* dominan berada di kelompok B dengan data sebanyak 98, *cluster* inet dominan berada di kelompok A dengan data sebanyak 151. Dari hasil perhitungan dapat terlihat bahwa pengujian kali ini mirip dengan hasil dari pengujian ketiga, dimana terbentuk 3 *predictive cluster* dan terjadi penumpukan *real cluster* pada satu kelompok

Mirip dengan percobaan ketiga, dipercobaan ini data dapat langsung dilakukan perhitungan *naïve bayes* dan dapat dipastikan ada salah satu *real cluster* yang hilang atau tidak dapat terdefinisi kedalam kelompok manapun. Setelah dilakukan perhitungan menggunakan *naïve bayes* maka didapat hasil perhitungan sebagai berikut:

Tabel 4.27 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* 0.70 dan 2 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.449921	A	283			
	0.593796	B	402	160	116	187
	0.580645	C	162	Oto	*Food*	Inet
Inet	0.436416	A	151	52	29	
	0.307692	C	36	Oto	*Food*	
Oto		C	52			
Food		Tidak terdefinisi	0			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* B dengan jumlah data sebanyak 402 data, kelompok Oto masuk kedalam kelompok C dengan jumlah data sebanyak 52 data, kelompok Inet masuk kedalam kelompok A dengan jumlah data sebanyak 151 data, sedangkan sama seperti percobaan sebelumnya kelompok *food* tidak dapat terdefinisi kedalam kelompok apapun karena hanya terdapat 3 *predictive cluster*.

Setelah itu dilakukan perhitungan kinerjanya dengan *confusion matrix*, berikut merupakan tabel hasil *predictive cluster* dan *real cluster*:

Tabel 4.28 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes pada Threshold* 0.70 dan 2 *SVD*

		Predictive cluster			
		oto	*sport*	*food*	inet
Real Cluster	Oto	52	106	0	108
	Sport	162	402	0	283
	Food	29	98	0	87
	Inet	36	71	0	151

Berikut merupakan hasil perhitungan *confusion matrix* dari table 4.28:

Tabel 4.29 Hasil Perhitungan *Accuracy, Precision*, dan *Recall* pada Algoritma LSI dengan *Threshold* 0.70 dan 2 *SVD*

accuracy	0.381703	38%
precision oto	0.18638	19%
precision sport	0.593796	59%
precision food	0	0%
precision inet	0.240064	24%
recall oto	0.195489	20%
recall *sport*	0.474616	47%
recall *food*	0	0%
recall inet	0.585271	59%

4.2.3.5. Uji Menggunakan *Threshold* dengan Nilai Menurun Secara Bertahap dan Dengan Menggunakan *SVD* dengan Nilai 2

Pengujian selanjutnya dicoba dengan menggunakan *threshold* dengan cara penurunan bertingkat, *threshold* awal yang digunakan adalah 0.95 dan setiap terbentuk kelompok baru maka nilai *threshold* diturunkan sebesar 0.1 dari *threshold* kelompok sebelumnya, misal *threshold* kelompok awal adalah 0.95 lalu terbentuk kelompok baru dengan penurunan *threshold* pada kelompok baru tersebut menjadi *0.85*. Namun pada percobaan kali ini *SVD* yang digunakan tetap sama yaitu bernilai 2, berikut merupakan hasil percobaannya:

Tabel 4.30 Hasil dari Perhitungan Menggunakan *Threshold* Menurun dan 2 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	90	245	70	144	549
	B	123	426	104	76	729
	C	53	176	40	38	307

Hasil prediksi kelompok		B	B	B	A	

Dari hasil pada tabel 4. terlihat bahwa hasil *cluster* oto dominan berada di kelompok A dengan data sebanyak 108, *cluster sport* dominan berada di kelompok B dengan data sebanyak 402, *cluster food* dominan berada di kelompok B dengan data sebanyak 98, *cluster* inet dominan berada di kelompok A dengan data sebanyak 151. Dari hasil perhitungan dapat terlihat bahwa pengujian kali ini mirip dengan hasil dari pengujian ketiga dan keempat, dimana terbentuk 3 *predictive cluster* dan terjadi penumpukan *real cluster* pada satu kelompok

Data dari percobaan kali ini dapat langsung dilakukan perhitungan *naïve bayes* dan lagi-lagi dapat dipastikan ada salah satu *real cluster* yang hilang atau tidak dapat terdefinisi kedalam kelompok manapun. Setelah dilakukan perhitungan menggunakan *naïve bayes* maka didapat hasil perhitungan sebagai berikut:

Tabel 4.31 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* Menurun dan 2 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.446266	A	245			
	0.584362	B	426	143	110	182
	0.57329	C	176	Oto	*Food*	Inet
Inet	0.473684	A	144	53	40	
	0.290076	C	38	Oto	*Food*	
Oto		C	53			
Food		Tidak terdefinisi	0			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* B dengan jumlah data sebanyak 426 data, kelompok Oto masuk kedalam kelompok C dengan jumlah data sebanyak 53 data, kelompok Inet masuk kedalam kelompok A dengan jumlah data sebanyak 144 data, sedangkan sama seperti percobaan sebelumnya kelompok *food* tidak dapat terdefinisi kedalam kelompok apapun karena hanya terdapat 3 *predictive cluster*.

Setelah itu dilakukan perhitungan kinerjanya dengan *confusion matrix*, berikut merupakan tabel hasil *predictive cluster* dan *real cluster*:

Tabel 4.32 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes pada Threshold* Menurun dan 2 *SVD*

		Predictive cluster			
		oto	*sport*	*food*	inet
Real Cluster	Oto	53	123	0	90
	Sport	176	426	0	245
	Food	40	104	0	70
	Inet	38	76	0	144

Berikut merupakan hasil perhitungan *confusion matrix* dari table 4.32

Tabel 4.33 Hasil Perhitungan *Accuracy, Precision*, dan *Recall* pada Algoritma LSI dengan *Threshold* Menurun dan 2 *SVD*

Accuracy	0.39306	39%
precision oto	0.172638	17%
precision sport	0.584362	58%
precision food	0	0%
precision inet	0.262295	26%
recall oto	0.199248	20%
recall *sport*	0.502952	50%
recall *food*	0	0%
recall inet	0.55814	56%

4.2.3.6. Uji Menggunakan *Threshold* dengan Nilai Menurun Secara Bertahap dan dengan Menggunakan *SVD* dengan Nilai 3

Pengujian selanjutnya dicoba dengan menggunakan *threshold* dengan cara penurunan bertingkat, *threshold* awal yang digunakan adalah 0.95 dan setiap terbentuk kelompok baru maka nilai *threshold* diturunkan sebesar 0.1 dari *threshold* kelompok sebelumnya, misal *threshold* kelompok awal adalah 0.95 lalu terbentuk kelompok baru dengan penurunan *threshold* pada kelompok baru tersebut menjadi *0.85*. Namun pada percobaan kali ini *SVD* yang digunakan tetap sama yaitu bernilai 3, berikut merupakan hasil percobaannya:

Tabel 4.34 Hasil dari Perhitungan Menggunakan *Threshold* Menurun dan 3 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	92	194	41	124	451
	B	76	227	31	48	382
	C	88	401	133	84	706
	D	10	24	9	2	45
	E	0	1	0	0	1
Hasil prediksi kelompok		A	C	C	A	

Dari hasil pada tabel 4. terlihat bahwa hasil *cluster* oto dominan berada di kelompok A dengan data sebanyak 92, *cluster sport* dominan berada di kelompok C dengan data sebanyak 401, *cluster food* dominan berada di kelompok C dengan data sebanyak 133, *cluster* inet dominan berada di kelompok A dengan data sebanyak 124.

Data dari percobaan kali ini harus menggabungkan *predictive cluster* E kedalam *predictive cluster* yang lain, agar *predictive cluster* dengan *real cluster*

berjumlah sama, Karena terbentuk 5 *predictive cluster* maka kelompok E akan dimasukan kedalam kelompok A. Sehingga hasil kelompok tabel menjadi berikut:

Tabel 4.35 Penggabungan *Cluster* Lain Kedalam Satu *Cluster* Pada *Threshold* Menurun dan 3 *SVD*

		Real cluster				
		oto	*sport*	*food*	inet	Total *predictive cluster*
Predictive Cluster	A	92	195	41	124	451
	B	76	227	31	48	382
	C	88	401	133	84	706
	D	10	24	9	2	45
Hasil prediksi kelompok		B	B	B	A	

Setelah dilakukan perhitungan menggunakan *naïve bayes* maka didapat hasil perhitungan sebagai berikut:

Tabel 4.36 Hasil Penghitungan Penentuan *Predictive Cluster* dengan *Naïve Bayes* pada *Threshold* Menurun dan 3 *SVD*

	Hasil *Naïve* Bayes	Kelompok	Jumlah data	Sisa Data setelah *Naïve bayes*		
Sport	0.432373	A	195			
	0.594241	B	227	190	183	210
	0.567989	C	401	oto	*food*	inet
	0.533333	D	24			
Inet	0.48249	A	124	98	142	
	0.27541	C	84	oto	*food*	
	0.095238	D	2			
Food	0.60181	C	133	Oto		
	0.473684	D	9	10		
Oto		D	10			

Dari data diatas terlihat bahwa kelompok *Sport* masuk kedalam *cluster* B dengan jumlah data sebanyak 227 data, kelompok Oto masuk kedalam kelompok

D dengan jumlah data sebanyak 10 data, kelompok Inet masuk kedalam kelompok A dengan jumlah data sebanyak 124 data, kelompok Inet masuk kedalam kelompok C dengan jumlah data sebanyak 133 data.

Setelah itu dilakukan perhitungan kinerjanya dengan *confusion matrix*, berikut merupakan tabel hasil *predictive cluster* dan *real cluster*:

Tabel 4.37 Hasil Penentuan *Predictive Cluster* dengan *Naïve Bayes Pada Threshold* Menurun dan 3 *SVD*

		Predictive cluster			
		oto	*Sport*	*food*	inet
Real Cluster	Oto	10	76	88	92
	Sport	24	227	401	195
	Food	9	31	133	41
	Inet	2	48	84	124

Berikut merupakan hasil perhitungan *confusion matrix* dari tabel 4.37

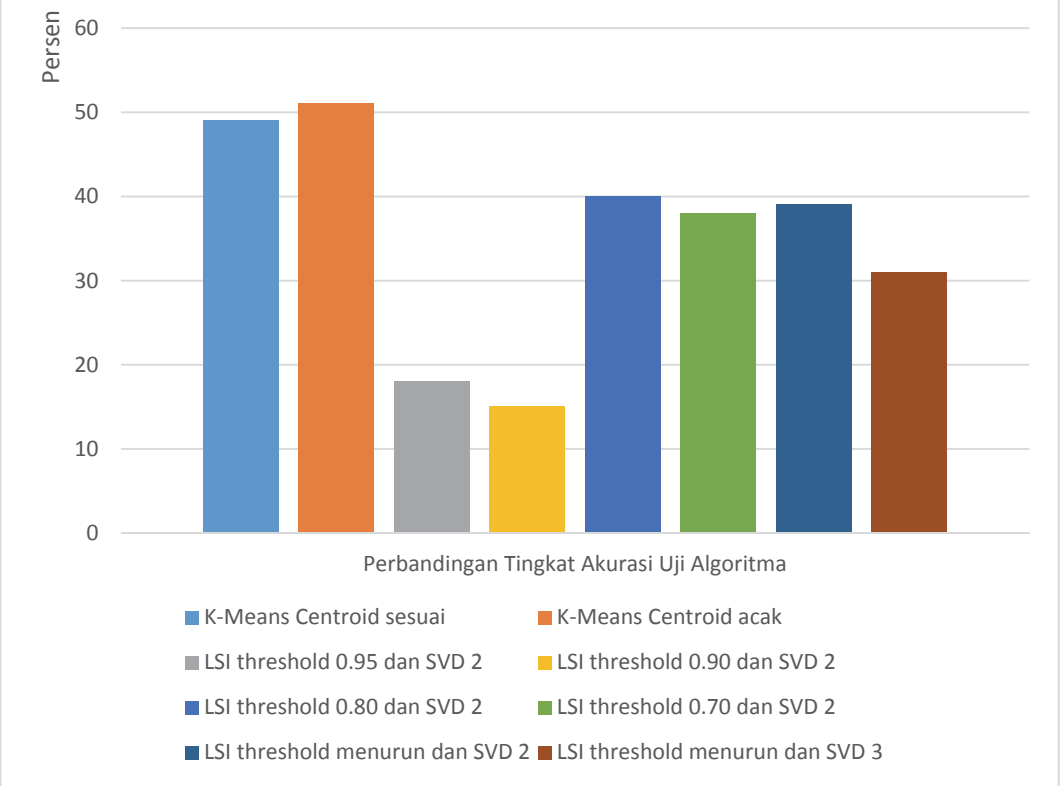
Tabel 4.38 Hasil Perhitungan Accuracy, *Precision*, Dan Recall Pada Algoritma LSI Dengan *Threshold* Menurun dan 3 *SVD*

accuracy	0.311672	31%
precision oto	0.222222	22%
precision sport	0.594241	59%
precision food	0.188385	19%
precision inet	0.274336	27%
recall oto	0.037594	4%
recall *sport*	0.268005	27%
recall *food*	0.621495	0%
recall inet	0.48062	48%

4.3. Seluruh Akurasi Hasil *Confusion Matrix*

Berdasarkan hasil perbandingan seluruh percobaan dapat dilihat bahwa percobaan pada *LSI* keakuratannya selalu dibawah *K-means*, *LSI* memiliki keakuratan sebesar 15% hingga 40%, sedangkan *K-means* memiliki keakuratan

49% hingga 51%. Untuk lebih detil dari keseluruhan uji coba dapat dilihat pada gambar 4.2.



Gambar 4.2 Perbandingan Tingkat Akurasi Dari Seluruh Percobaan

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan analisis data, implementasi, dan pengujian dapat ditarik kesimpulan bahwa algoritma *K-Means Clustering* lebih baik dari pada algoritma *LSI* dalam mengelompokan dokumen teks pendek yang diambil dari *twitter*. Berdasarkan hasil uji coba menunjukkan bahwa tingkat akurasi algoritma *K-Means Clustering* selalu lebih tinggi dibandingkan *LSI* dalam mengelompokan dokumen teks pendek. Algoritma K-Means memiliki tingkat akurasi antara 49% hingga 51%, sedangkan LSI memiliki tingkat akurasi antara 15% hingga 40%.

Dari hasil penelitian ini terdapat fenomena yang sangat mempengaruhi hasil penelitian yaitu:

1. Jika dilihat secara keseluruhan, kedua algoritma ini sering mengalami penumpukan dalam pengelompokan, salah satu penyebabnya adalah *sparse data*, *sparse data* dimana sebuah data yang ada banyak memiliki nilai 0 sehingga keterkaitan antara dokumen yang satu dengan dokumen yang lainnya sulit untuk dibedakan. Selain itu muncul pula *hapax legomenon* yaitu sesuatu yang diucapkan atau dimunculkan hanya sekali, sehingga akibat *sparse data* dan *hapax legomenon* ini dokumen menjadi sulit dikelompokan.
2. Pada penelitian LSI menunjukan bahwa semakin tinggi nilai threshold dan nilai SVD maka semakin banyak kelompok yang dibentuk, maka pada penelitian LSI diperlukan penggunaan SVD dan threshold yang tepat, agar tidak terbentuk kelompok yang terlalu banyak.

5.2. Saran

Berdasarkan hasil penelitian yang telah disimpulkan maka dapat dikemukakan beberapa saran untuk penelitian selanjutnya, yaitu:

1. Karena LSI dan K-Means terpengaruh pada salah satu dokumen yang dijadikan acuan pengelompokan, maka dalam penelitian selanjutnya disarankan untuk mencoba sistem pengelompokan dengan menggunakan supervised learning.
2. Hasil dari algoritma LSI dan K-Means masih kurang baik untuk pengelompokan, disarankan untuk mencari sistem optimasi untuk kedua algoritma ini.
3. Disarankan untuk mencari algoritma lain yang tepat untuk pengelompokan data teks pendek yang lebih baik.

DAFTAR PUSTAKA

Agusta, Y. 2007. *K-Means – Penerapan, Permasalahan Dan Metode Terkait.* Denpasar*:* STMIK STIKOM Bali.

Andriani, A. 2013. *Sistem Pendukung Keputusan Berbasis Decision Tree dalam Pemberian Beasiswa Studi Kasus: Amik "Bsi Yogyakarta"*. Jakarta: Program Studi Manajemen Informatika, AMIK BSI Jakarta.

Bramer, M. 2013. *Principles of Data Mining.* London: Springer-Verlag.

Brown, M. S. 2014. *Data Mining For Dummies*. Hoboken: John Wiley & Sons, Inc.

Darujati, C.; Gumelar, A. B. 2012. *Pemanfaatan Teknik Supervised Untuk Klasifikasi Teks Bahasa Indonesia*. Surabaya: Fakultas Ilmu Komputer, Universitas Narotama Surabaya.

Dini, E. P. 2015. *Prediksi Topik Pada Media Sosial Twitter Menggunakan K-Means Clustering Dan Naïve Bayes Classifier*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Garcia, E. 2006. *Latent Semantic Indexing (LSI) a Fast Track Tutorial*. Garcia.

Han, J. & Kamber, M. 2006. *Data mining Concept and Techniques*. Oxford: Elsevier, Inc.

Hearst, M. 2003. What is Text Mining?. [terhubung berkala] http://people.ischool.berkeley.edu/~hearst/text-mining.html. [diakses 11 September 2015, pukul 20.13].

Indriani, A. 2014. *Klasifikasi Data Forum Dengan Menggunakan Metode Naïve Bayes Classifier*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi (SNATI).

Jaedun, Amat. 2011. Metodologi Penelitian Eksperimen. Yogyakarta: Fakultas Teknik UNY, Ka. Puslit Dikdasmen, Lemlit UNY. [terhubung berkala] http://staff.uny.ac.id/sites/default/files/pengabdian/drs-amat-jaedun-mpd/metode-penelitian-eksperimen.pdf [diakses 6 September 2015 pukul 19:05].

Langgeni, D. P.; Baizal Z. K. A.; Firdaus, Y. 2010. *Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection*. Yogyakarta: UPN Veteran Yogyakarta.

Oktafia, D.; Pardede, D. L. C. 2008. *Perbandingan Kinerja Algoritma Decision Tree dan Naïve Bayes dalam Prediksi Kebangkrutan*. Jakarta: Sistem

Informasi Universitas Gunadarma.

Rosario, B. 2000. *Latent Semantic Indexing: An overview*. Final Paper. INFOSYS 240 Spring.

Russell, Matthew A. 2014. *Mining the Social Web*. 2nd Ed. Sebastopol: O'Reilly Media, Inc.

Setiohardjo, N. M. & Harjoko, A. 2014. *Analisis Tekstur Untuk Klasifikasi Motif Kain (Studi Kasus Kain Tenun Nusa Tenggara Timur)*. Yogyakarta: IJCCS.

Suliantoro, D. W.; Wisnubhadra, I.; & Ernawati. 2012. *Integrasi Pembobotan TF-IDF pada Metode K-Means Untuk Clustering Dokumen Teks*. Surabaya: Program Studi MMT-ITS.

The Mathwork, Inc. 2016. United States. www.mathworks.com

Tim Penyusun. 2012. *Buku Pedoman Skripsi / Komprehensif / Karya Inovatif (S1)*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Wahyuni, D. 2014 . *Implementasi Algoritma K-Means Clustering Untuk Mengetahui Bidang Skripsi Mahasiswa Multimedia Pendidikan Teknik Informatika Dan Komputer Universitas Negeri Jakarta*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.

Zelikovitz, S. 2010. *Transductive LSI for Short Text Classification Problems.* Staten Island: The College of Staten Island of CUNY.

LAMPIRAN

Lampiran 1. Contoh 100 Data *Twitter* dari 1585 Data yang telah Dikelompokkan

Tweets	Kategori
10 Kesalahan Umum Pengemudi Pemula	@detikOto
10 Mobil Terlaris di Jepang	@detikOto
2 Legenda Motor Italia Resmi Meluncur di Indonesia	@detikOto
3 Motor Yamaha Berkelir Kuning-Hitam Edisi 60 Tahun Ultah	@detikOto
4 Mobil Mewah Floyd Mayweather Terbakar	@detikOto
Makanan Manis Asin Gurih Empal Gepuk Enak Dinikmati dengan Nasi Hangat Siang Ini	@detikfood
Baidu dan Google Mau Boyong Aplikasi Indonesia ke China	@detikinet
Ada Pebalap Indonesia Naik Podium di Motegi	@detikSport
Agar Olahraga Selalu Jadi Fondasi Karakter Bangsa'	@detikSport
Angkutan Umum Masa Depan Singapura	@detikOto
Bahkan Mick Jagger Tak Bisa Beli Mobil Baru di Kuba	@detikOto
Bau Terbakar di Ruang Mesin Innova	@detikOto
Begini Akibatnya Jika Nekat Menerobos Lampu Merah	@detikOto
Crossover Toyota C-HR Terekam Kamera Dites di Jalan	@detikOto
Daihatsu Permak Xenia Jadi Baru Lagi	@detikOto
Di Bawah Guyuran Hujan, Massa Torehkan Waktu Tercepat	@detikSport
Dianggap Banyak Kelemahan? Ini Jawaban Pembuat Sarung Tangan Setang	@detikOto
Djokovic dan Nadal Tembus Semifinal	@detikSport
Esteemovers Rayakan HUT yang ke-4	@detikOto
Ferrari Edisi Ulang Tahun ke-40 Ini Gagal Dilelang	@detikOto
Goda Pecinta Motor Super, Kawasaki Poles ZZR 1400	@detikOto
Hamilton Tak Pikirkan Gelar Juara	@detikSport
Hampir Mustahil Ada Pabrik V-Kool di Indonesia	@detikOto
Hari Pertama yang Biasa-biasa Saja untuk Rossi	@detikSport
Hayden Tinggalkan MotoGP di Akhir Musim, Beralih ke WSBK	@detikSport
Honda CBR500R Bersolek	@detikOto
Hossein Askari Juara Etape 6, Zargari Kehilangan Yellow Jersey	@detikSport
Hyundai Tucson Buat Pecinta Off-Road	@detikOto
Ini Dia Honda CBR500R, Ada Beberapa Fitur Baru	@detikOto
Ini Kata Bos Yamaha soal Rossi vs Lorenzo di Dua Balapan Sisa	@detikSport
Ini Kata Rossi dan Lorenzo soal Hubungan Mereka Saat Ini	@detikSport
Ini Kota-kota di Indonesia yang Paling Aman di Jalanan	@detikOto

2015	
Ini Wujud Modifikasi Motor yang Sayang Anak	@detikOto
Ivanovic dan Muguruza Rebut Tiket Perempatfinal	@detikSport
Ivanovic Tembus Empat Besar	@detikSport
Kapan Juara Lagi, Tontowi/Liliyana?	@detikSport
Kawasaki Resmi Luncurkan Ninja ZX-10R Terbaru	@detikOto
Kecelakaan di Motegi, Alex de Angelis Kritis	@detikSport
Kia Siapkan SUV Baru, Lebih Kecil dari Kia Sportage	@detikOto
Layanan Purna Jual MINI Hadir di Surabaya	@detikOto
Lewati Ivanisevic, Ivo Karlovic Jadi Raja Ace	@detikSport
Linda Ditantang Ulang Hasil Kejuaraan Dunia di Denmark Terbuka	@detikSport
Linda Siap Tampil Habis-habisan Hadapi Li Xuerui	@detikSport
Lorenzo Berharap Dapat 'Bantuan' untuk Hadang Rossi	@detikSport
Lorenzo Kuasai Latihan Bebas Pertama	@detikSport
Lorenzo Kuasai Latihan Pertama	@detikSport
Lorenzo Masih yang Tercepat, Pedrosa Kedua	@detikSport
Lorenzo Merasa Belum Waktunya Pakai Strategi Valencia 2013	@detikSport
Lucu, Sarung Tangan Setang Alias Baramet Marak di Jalanan	@detikOto
Makin Banyak Mobil Mewah di Jalanan, Makin Makmur Negara'	@detikOto
Marquez Dukung Lorenzo atau Rossi?	@detikSport
Marquez: Rasanya Sakit, tapi Masih Bisa Ditahan	@detikSport
Mencicipi Mobil MINI Paling Tercepat, John Cooper Works	@detikOto
Menyusuri Jalanan Bali dengan Maserati Quattroporte Generasi Keenam	@detikOto
Misi Ganda Campuran Indonesia di Denmark Terbuka	@detikSport
Mobil Masa Depan Mercedes-Benz Bakal Punya 'Mata' Seperti Manusia	@detikOto
Naked Wolves Indonesia Lebarkan Sayap ke Tangsel	@detikOto
Pamela 'Duo Serigala' Joki RX-King	@detikOto
Panitia GIIAS Sempat Tak Yakin Target akan Tercapai	@detikOto
Pengguna Moge BMW di Indonesia Dimanjakan Distributor	@detikOto
Penjualan Mobil Sport Mewah di Indonesia Ikut Lesu	@detikOto
Penuhi Keinginan Orang Kaya Timur Tengah, Nissan Hadirkan SUV Ini	@detikOto
Perjuangan Tidak Makin Mudah untuk Lorenzo	@detikSport
Rabat Mundur, Johann Zarco Rebut Gelar Juara Dunia Moto2	@detikSport
Resonator 125, Konsep Motor Klasik dari Yamaha	@detikOto
Rosberg Akan Bangkit Tahun Depan'	@detikSport

Rossi Prediksikan Balapan Sulit, Pede Bisa Kejar Lorenzo	@detikSport
Rossi Punya Sentuhan Midas, Juga Selalu Beruntung'	@detikSport
Rossi: Balapannya Menyenangkan, Hasilnya Tidak	@detikSport
Sarung Tangan Motor Setang Banyak Disukai Ibu-ibu	@detikOto
Sayang, Seri 7 di Indonesia Tidak Ada Fitur Parkir Otomatis	@detikOto
Siap-siap Rakit BR-V, Honda India Kerek Produksi	@detikOto
Suzuki Ertiga, Lebih Stylish dan Nyaman	@detikOto
Terlalu Panjang, Mobil Limo Ini Susah Parkir	@detikOto
Tips Tersembunyi ketika Mencuci Kendaraan	@detikOto
Tommy Kalah, Indonesia Tanpa Gelar Juara	@detikSport
Tontowi/Liliyana Dipaksa Bermain Tiga Gim di Babak Kedua	@detikSport
Ubahan Cantik Ducati Scrambler Edisi Paul Smart	@detikOto
Wacana MotoGP di Indonesia Berlanjut	@detikSport
Yamaha Berdoa untuk Balapan Kering di MotoGP Jepang	@detikSport
Yamaha Suguhkan 20 Motor di Tokyo Motor Show 2015	@detikOto
Yamaha: V-Ixion Konsisten Kuasai Pasar Motor Laki	@detikOto
1.000 Pelari Akan Ikuti Banyuwangi International Run	@detiksport
1.000 Pengguna Xperia Z3 Keroyokan Garap Android Marshmallow	@detikinet
1.000 Prajurit Intel Garap Chip Modem iPhone 7	@detikinet
10 Hal yang Perlu Diperhatikan ketika Memperbaiki Mobil Sendiri	@detikOto
10 Jurus Menjaga Diri dari Penjahat Cyber	@detikinet
10 Kemenangan dalam 10 Laga, Hodgson: Inggris Layak Bangga	@detiksport
10 Ribu Ponsel Sony Segera Cicipi Android Marshmallow	@detikinet
13 Orang Diamankan Jelang Laga Final Piala Presiden	@detiksport
15 Ribu Orang Siap Jelajahi Ibukota di Jakarta Marathon 2015	@detiksport
2 Instruktur Indonesia Siap Tarung di Kompetisi Safety Riding	@detikoto
2 Instruktur Indonesia Siap Tarung di Kompetisi Safety Riding	@detikOto
2 Konsep Motor Listrik Terbaru Yamaha	@detikoto
2 Raksasa Storage Berebut SanDisk	@detikinet
2016, Penentuan Hidup Mati Bisnis Ponsel Sony	@detikinet
24 Jam Sibuk Bos Baidu di Indonesia	@detikinet
268 UMKM Dapat Pinjaman Modal Rp 8,8 Miliar dari Telkom	@detikinet
3 Andalan Lexus di Tokyo Motor Show	@detikoto
3 Motor Yamaha Berkelir Kuning-Hitam Edisi 60 Tahun Ultah	@detikoto

Lampiran 2. Contoh 320 dari 769 Kata *Stopword Removal*

ada	bagaimanapun	berawal	bisakah
adalah	bagi	berbagai	boleh
adanya	bagian	berdatangan	bolehkah
adapun	bahkan	beri	bolehlah
agak	bahwa	berikan	buat
agaknya	bahwasanya	berikut	bukan
agar	baik	berikutnya	bukankah
akan	bakal	berjumlah	bukanlah
akankah	bakalan	berkali-kali	bukannya
akhir	balik	berkata	bulan
akhiri	banyak	berkehendak	bung
akhirnya	bapak	berkeinginan	cara
aku	baru	berkenaan	caranya
akulah	bawah	berlainan	cukup
amat	beberapa	berlalu	cukupkah
amatlah	begini	berlangsung	cukuplah
anda	beginian	berlebihan	cuma
andalah	beginikah	bermacam	dahulu
antar	beginilah	bermacam-macam	dalam
antara	begitu	bermaksud	dan
antaranya	begitukah	bermula	dapat
apa	begitulah	bersama	dari
apaan	begitupun	bersama-sama	daripada
apabila	bekerja	bersiap	datang
apakah	belakang	bersiap-siap	dekat
apalagi	belakangan	bertanya	demi
apatah	belum	bertanya-tanya	demikian
artinya	belumlah	berturut	demikianlah
asal	benar	berturut-turut	dengan
asalkan	benarkah	bertutur	depan
atas	benarlah	berujar	di
atau	berada	berupa	dia
ataukah	berakhir	besar	diakhiri
ataupun	berakhirlah	betul	diakhirinya
awal	berakhirnya	betulkah	dialah
awalnya	berapa	biasa	diantara
bagai	berapakah	biasanya	diantaranya
bagaikan	berapalah	bila	diberi
bagaimana	berapapun	bilakah	diberikan
bagaimanakah	berarti	bisa	diberikannya

dibuat	diperlihatkan	gunakan	jawabnya
dibuatnya	diperlukan	hal	jelas
didapat	diperlukannya	hampir	jelaskan
didatangkan	dipersoalkan	hanya	jelaslah
digunakan	dipertanyakan	hanyalah	jelasnya
diibaratkan	dipunyai	hari	jika
diibaratkannya	diri	harus	jikalau
diingat	dirinya	haruslah	juga
diingatkan	disampaikan	harusnya	jumlah
diinginkan	disebut	hendak	jumlahnya
dijawab	disebutkan	hendaklah	justru
dijelaskan	disebutkannya	hendaknya	kala
dijelaskannya	disini	hingga	kalau
dikarenakan	disinilah	ia	kalaulah
dikatakan	ditambahkan	ialah	kalaupun
dikatakannya	ditandaskan	ibarat	kalian
dikerjakan	ditanya	ibaratkan	kami
diketahui	ditanyai	ibaratnya	kamilah
diketahuinya	ditanyakan	ibu	kamu
dikira	ditegaskan	ikut	kamulah
dilakukan	ditujukan	ingat	kan
dilalui	ditunjuk	ingat-ingat	kapan
dilihat	ditunjuki	ingin	kapankah
dimaksud	ditunjukkan	inginkah	kapanpun
dimaksudkan	ditunjukkannya	inginkan	karena
dimaksudkannya	ditunjuknya	ini	karenanya
dimaksudnya	dituturkan	inikah	kasus
diminta	dituturkannya	inilah	kata
dimintai	diucapkan	itu	katakan
dimisalkan	diucapkannya	itukah	katakanlah
dimulai	diungkapkan	itulah	katanya
dimulailah	dong	jadi	ke
dimulainya	dua	jadilah	keadaan
dimungkinkan	dulu	jadinya	kebetulan
dini	empat	jangan	kecil
dipastikan	enggak	jangankan	kedua
diperbuat	enggaknya	janganlah	keduanya
diperbuatnya	entah	jauh	keinginan
dipergunakan	entahlah	jawab	kelamaan
diperkirakan	guna	jawaban	kelihatan

Lampiran 3. Source Code Program Matlab Algoritma *LSI*

```
%panggil data file
file = fopen('testdikitdata.txt');
Term=textscan(file,'%s');
Term{1}=strjoin(Term{1}',' ');
Term{1}=regexp(Term{1},'[^A-Za-z]+','split')';
Term{1}=lower(Term{1}');
%memanggil data ulang untuk data perdokumen
file = fopen('testdikitdata.txt');
C = textscan(file,'%s','delimiter','');
%melakukan perhitungan TF per dokumen
Term{1}=unique(Term{1});
for i=1:length(C{1,1}(:,1)),
    C{1,1}{i,1}=regexprep(C{1,1}{i,1},'[^A-Za-z]+',' ');
    E{i} = regexp(lower(C{1,1}{i,1}),' ','split')';
    E{i}=E{i}';
    tf{i}=cell2mat(arrayfun(@(x)
sum(ismember(E{i},Term{1}(x)),2),1:numel(Term{1}),'un',0));
    %for a=1:length(tf{1,i}(1,:)),
    %    dokumen(a,i)=tf{1,i}(1,a);
    %end
end
dokumen=cell2mat(tf')';
%}

%dokumen=load('LSI.txt');

%Rumus SVD
[U,S,V] = svd(dokumen);

%taking s largest singular values
s=2;

%menghitung Uk, Sk, Vk
Uk=U(:,1:s);
Sk=S(1:s,1:s);
Vk=V(:,1:s);

%Rumus Similarity antar document
cen=1;
kel=1;
nomor=1;
length(Vk(:,1))
for i=2:length(Vk(:,1)),
    for j=1:length(cen),
        %rumus similarity pembanding

SimM=dot(Vk(cen(j),:),Vk(i,:))/((sqrt(dot(Vk(cen(j),:),Vk(cen(j),:
))))*(sqrt(dot(Vk(i,:),Vk(i,:)))))
        %pengaturan batas threeshold
        if SimM>=0.6
            nomor=nomor+1;
            kel(end+1, cen)=nomor;
            j=length(cen);
        %jika dibawah batas threeshold maka membuat kelompok baru
        elseif SimM<0.6
```

```
            if j==length(cen);
               nomor=nomor+1;
               cen(j+1)=nomor;
               j=length(cen);
               kel(end+1, j)=nomor;
            end
        end
    end
end
```

Lampiran 4. Source Code Program Matlab Algoritma *K-Means*

```
%created by rian
%panggil file stopword
file = fopen('stopwordindo.txt');
stopword=textscan(file,'%s');
stopwords_cellstring=stopword{1}';

%panggil data file
file = fopen('testdikitbagus.txt');
Term=textscan(file,'%s');
Term{1}=lower(Term{1}');

%melakukan stopword removal dengan strjoin
Term{1}=
strjoin(Term{1}(~ismember(Term{1},stopwords_cellstring)),' ');
TermStopWord{1}=regexp(Term{1},'[^A-Za-z]+','split')';

%memanggil data ulang untuk data perdokumen
file = fopen('testdikitbagus.txt');
C = textscan(file,'%s','delimiter','');

%melakukan perhitungan TF per dokumen
jmlhdoc=length(C{1,1});
TermTF=unique(TermStopWord{1}');
jmlhdoc=length(C{1,1});
E = regexp(lower(C{1,1}),'[^A-Za-z]+','split')';
for i=1:length(C{1,1}),;
    tf{i}=cell2mat(arrayfun(@(x)
sum(ismember(E{i},TermTF(x)),2),1:numel(TermTF),'un',0));
end

%df
df=lower(TermStopWord{1});
[val,idxC, idxV] = unique(df);
df = accumarray(idxV,1);

%idf
idf=log10(jmlhdoc./df);

%tfidf
for a=1:length(tf),;
    tf{a};
    tfdoc2=tf{a};
    tfidf{a}=tfdoc2'.*idf;
    tfidf{a};
end


jmlhcentroid= input('masukan jumlah centroid!');
for a=1:jmlhcentroid,;
     disp(sprintf('masukan centroid ke- %d',a));
     centroid= input('');
     centroidlama{a}=tfidf{1,centroid};
     %menghitung euclidean distance dengan pengulangan berdasar
jmlh dokumen
```

```
     for i=1:length(tfidf),;
         %D{centroid,i}adalah baris adalah 'centroid', kolom
adalah 'i'
          D1(a,i)=norm(tfidf{1,i}-centroidlama{a});
     end
end
%mencari jarak dokumen terdekat dengan centroid
[M,terdekat]= min(D1);

%iterasi mencari centroid dan mengelompokan yang baru
tfidf2=cell2mat(tfidf);
kelompok=0;
kelompok2=1;
iterasi=1;
while isequal(kelompok,kelompok2)==0;
    [k,l]=size(cell2mat(centroidlama));
    centroidbaru=zeros(k,l);
    countercentroid=zeros(1,jmlhcentroid);
    for i=1:length(tfidf2(1,:)),;
        for x=1:jmlhcentroid,;
            if terdekat(1,i)==x;
                centroidbaru(:,x)=(centroidbaru(:,x)+tfidf2(:,i));
                countercentroid(1,x)=countercentroid(1,x)+1;
            end
        end
    end
    for x=1:jmlhcentroid,;
        centroidbaru(:,x)=centroidbaru(:,x)/countercentroid(1,x);
    end
    centroidbaru=num2cell(centroidbaru,1);

    for a=1:jmlhcentroid,;
        %menghitung euclidean distance dengan pengulangan berdasar
jmlh dokumen
        for i=1:length(tfidf),;
            %D{centroid,i}adalah baris adalah 'centroid', kolom
adalah 'i'
            D2(a,i)=norm(tfidf{1,i}-centroidbaru{a});
        end
    end
    [M2,terdekat2]= min(D2);
    %centroidbaru{1}
    %equalcentroid=isequal(terdekat,terdekat2);
    kelompok=terdekat'
    kelompok2=terdekat2'
    if isequal(terdekat,terdekat2)==0
        terdekat=terdekat2;
    end
    iterasi=iterasi+1
end

hasil=zeros(1,jmlhcentroid);
for i=1:length(kelompok2)
    for y=1:jmlhcentroid
        if isequal(kelompok2(i),y)==1
            hasil(1,y)=hasil(1,y)+1;
        end
    end
```

```
end
hasil
```

TENTANG PENULIS

Penulis bernama lengkap Fitrianto Adi Saputro dilahirkan di Jakarta tanggal 22 Maret 1994 dari ayah bernama Kusmin dan ibu bernama Ruwiyati. Penulis merupakan anak sulung dari tiga bersaudara di keluarganya. Penulis menyelesaikan pendidikan dasarnya di TK Larasati pada tahun 1999, SD Negeri 3 Pengasinan Kota Bekasi pada tahun 2005, SMP Negeri 2 Kota Bekasi pada tahun 2008, dan MAN 2 Kota Bekasi pada tahun 2011.

Pada tahun 2011, penulis diterima sebagai mahasiswa Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta. Dalam menyelesaikan studinya penulis, mengadakan penelitian untuk pengerjaan skripsi dengan judul "Analisis Komparatif Kinerja Algoritma *Latent Semantic Indexing* dan *K-Means* Dalam Mengelompokan Dokumen Teks Pendek " sebagai syarat mendapatkan gelar sarjana pendidikan.