TUGAS AKHIR – SM141501

PENGELOMPOKAN DAN KLASIFIKASI LAPORAN MASYARAKAT DI SITUS MEDIA CENTER SURABAYA MENGGUNAKAN METODE *K-MEANS CLUSTERING* DAN *SUPPORT VECTOR MACHINE*

ERIES BAGITA JAYANTI
NRP 1213 100 037

Dosen Pembimbing
Alvida Mustika Rukmi, S.Si, M.Si

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017

FINAL PROJECT – SM141501

CLUSTERING AND CLASSIFICATION THE CITIZEN REPORT IN MEDIA CENTER SURABAYA WEBSITE USING K-MEANS CLUSTERING AND SUPPORT VECTOR MACHINE METHODS

ERIES BAGITA JAYANTI
NRP 1213 100 037

Supervisor
Alvida Mustika Rukmi, S.Si, M.Si

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Science
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

PENGELOMPOKAN DAN KLASIFIKASI LAPORAN MASYARAKAT DI SITUS MEDIA CENTER SURABAYA MENGGUNAKAN METODE *K-MEANS CLUSTERING* DAN *SUPPORT VECTOR MACHINE*

CLUSTERING AND CLASSIFICATION THE CITIZEN REPORT IN MEDIA CENTER SURABAYA WEBSITE USING K-MEANS CLUSTERING AND SUPPORT VECTOR MACHINE METHODS

Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Sains
Pada Bidang Studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
ERIES BAGITA JAYANTI
NRP. 1213 100 037

Menyetujui,
Dosen Pembimbing,

Alvida Mustika Rukmi, S.Si, M.Si
NIP. 19720715 199802 2 001

Mengetahui,
Kepala Departemen Matematika

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, 1 Agustus 2017

PENGELOMPOKAN DAN KLASIFIKASI LAPORAN MASYARAKAT DI SITUS MEDIA CENTER SURABAYA MENGGUNAKAN METODE *K-MEANS CLUSTERING* DAN *SUPPORT VECTOR MACHINE*

Nama Mahasiswa	: Eries Bagita Jayanti
NRP	: 1213 100 037
Departemen	: Matematika
Dosen Pembimbing	: Alvida Mustika Rukmi, S.Si, M.Si

Abstrak

Media center merupakan sebuah sistem pelayanan terintegrasi bagi masyarakat Surabaya. Melalui media center masyarakat dapat berpartisipasi dengan memberikan laporan terkait kota Surabaya. Pengelompokan laporan masyarakat yang dilakukan oleh media center masih manual. Sehingga diperlukan sebuah alternatif lain untuk mempermudah pihak media center dalam melakukan pengelompokan tersebut. Pada tugas akhir ini penulis melakukan penelitan terkait pengelompokan dan klasifikasi laporan masyarakat dengan menerapkan metode *K-Means Clustering* dan *Support Vector Machine*. Laporan masyarakat yang digunakan pada penelitian ini belum memiliki label sehingga harus melalui pengelompokan terlebih dahulu menggunakan *K-Means Clustering* untuk memberikan label pada data berdasarkan label clusternya. Selanjutnya data yang telah berlabel tersebut dapat diklasifikasikan dengan SVM untuk membentuk model klasifikasi. Berdasarkan hasil pengelompokan yang dilakukan terhadap 1948 data laporan, diperoleh 10-*cluster* sebagai *cluster* terbaik dengan nilai koefisien *sillhoute* sebesar 0,61. Selanjutnya dengan menggunakan 1568 data *training* dan 380 data *testing* didapatkan akurasi model klasifikasi sebesar 83,42 % .

Kata Kunci : Text mining, K-Means Clustering, Support Vector Machine, Laporan Masyarakat.

CLUSTERING AND CLASSIFICATION THE CITIZEN REPORT IN MEDIA CENTER SURABAYA WEBSITE USING K-MEANS CLUSTERING AND SUPPORT VECTOR MACHINE METHODS

Name of Student	: Eries Bagita Jayanti
NRP	: 1213 100 037
Department	: Mathematics
Supervisor	: Alvida Mustika Rukmi, S.Si, M.Si

Abstract

Media center is an integrated service system for the community. Through the media center community can participate by providing related reports the city of Surabaya. Grouping reports conducted by media center still manual. So it required an alternative method to simplify media center do grouping. In this final task the author doing clustering and classification community report by applying method of K-Means Clustering and Support Vector Machine. Community reports used in this research does not yet have the label so it has to go through the process first clustering using K-Means Clustering to provide labels on the data. Further data that has been labeled based on clusters can be classified with SVM to form the model. Based on the results of clustering is done against the 1948 report data, retrieved 10-cluster as the best cluster with sillhoute coefficients of 0.61. Furthermore using 1568 data training and 380 data testing obtained accuracy classification model of 83.42%

Keywords : Text mining, K-Means Clustering, Support Vector Machine, The citizen report.

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

"PENGELOMPOKAN DAN KLASIFIKASI LAPORAN MASYARAKAT DI SITUS MEDIA CENTER SURABAYA MENGGUNAKAN METODE *K-MEANS CLUSTERING* DAN *SUPPORT VECTOR MACHINE*"

merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Drs. Lukman Hanafi, M.Sc selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
3. Alvida Mustika Rukmi, S.Si, M.Si selaku Dosen Pembimbing yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
5. Drs. Iis Herisman, M.Sc selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
7. Orang tua dan kedua adik yang senantiasa memberikan dukungan dan do'a yang tak terhingga.

8. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
9. Hanum, Nurita, dan Mila yang banyak membantu penulis dalam pengerjaan tugas akhir ini.
10. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih atas semangat dan doa yang tak henti-hentinya dihaturkan hingga terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Agustus 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN v
Abstrak vii
Abstract ix
KATA PENGANTAR xi
DAFTAR ISI xiii
DAFTAR GAMBAR xv
DAFTAR TABEL xvii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan 3
1.5 Manfaat 4
1.6 Sistematika Penulisan Tugas Akhir 4
BAB II TINJAUAN PUSTAKA 7
2.1 Penelitian Terdahulu 7
2.2 Media Center 8
2.3 *Text Mining* 10
2.4 *K-Means Clustering* 17
2.5 *Support Vector Machine* (SVM) 18
2.6 Metode Evaluasi 21
BAB III METODE PENELITIAN 23
3.1 Tahapan Penelitian 23
3.2 Diagram Alir Penelitian 25
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM 27
4.1 Perancangan Data 27
4.1.1 Data Masukan 27
4.1.2 Data Keluaran 27
4.2 Perancangan Diagram Alir Sistem 28

4.2.1 Pre-proses Teks 29
4.2.1.1 *Case Folding* dan *Tokenizing* 29
4.2.1.2 *Filtering* 30
4.2.2 Pembobotan TF-IDF dan SVD 32
4.2.3 Pengelompokan Data menggunakan *K-Means Clustering* 35
4.2.4 Klasifikasi menggunakan *Support Vector Machine* 41
4.3 Implementasi 43
4.3.1 Implementasi Pre-proses Teks 43
4.3.1.1 Implementasi *Case Folding* dan *Tokenizing* 43
4.3.1.2 Implementasi *Filtering* 44
4.3.2 Implementasi Pembobotan TF-IDF dan SVD. 44
4.3.3 Implementasi Pengelompokan Data menggunakan *K-Means Clustering* 46
4.3.4 Implementasi proses klasifikasi menggunakan *Support Vector Machine* 51
4.3.5 Implementasi Metode Evaluasi 53
4.4 Implementasi Antarmuka GUI 54
BAB V HASIL DAN PEMBAHASAN 57
5.1 Hasil Pre-proses Data 57
5.2 Pengelompokan Menggunakan *K-Means Clustering* 60
5.3 Analisa Hasil *Cluster* 63
a. *Cluster* ke-1 63
b. *Cluster* ke-4 66
c. *Cluster* ke-5 68
5.4 Klasifikasi Menggunakan SVM 73
BAB VI PENUTUP 77
6.1 Kesimpulan 77
6.2 Saran 78
DAFTAR PUSTAKA 79
LAMPIRAN 81
BIODATA PENULIS 107

DAFTAR GAMBAR

Gambar 3. 1 Diagram Alir Penelitian 25

Gambar 4. 1 Diagram Alir Sistem 28
Gambar 4. 2 Diagram Alir *Case Folding* dan *Tokenizing* 29
Gambar 4. 3 Diagram Alir *Filtering* 30
Gambar 4. 4 Diagram Alir Pembobotan 32
Gambar 4. 5 Diagram Alir *K-Means Clustering* 35
Gambar 4. 6 Diagram Alir Pembentukan Vektor Dokumen . 39
Gambar 4. 7 Diagram Alir Klasifikasi 41
Gambar 4. 8 Contoh Data Input SVM 42
Gambar 4. 9 Tab Proses Awal 55
Gambar 4. 10 Tab *Clustering* Menggunakan K-Means 55
Gambar 4. 11 Tab Tampilkan Hasil *Cluster* 56
Gambar 4. 12 Tab Klasifikasi menggunakan SVM 56

Gambar 5. 1 Grafik *Cluster* dan Koefisien *Sillhoute* 61
Gambar 5. 2 Model Klasifikasi 74

DAFTAR TABEL

Tabel 4. 1 Contoh Data Awal..... 30
Tabel 4. 2 Contoh Hasil *Casefolding* dan *tokenizing*..... 30
Tabel 4. 3 Sebelum Melalui *Filtering*..... 31
Tabel 4. 4 Setelah Melalui *Filtering*..... 31
Tabel 4. 5 Contoh Hasil Perhitungan TF-IDF..... 34
Tabel 4. 6 Contoh Hasil Perhitungan SVD..... 34
Tabel 4. 7 Contoh Hasil Pengelompokan Kata Penting..... 37
Tabel 4. 8 Contoh Dokumen dan Label *Cluster*..... 40

Tabel 5.1 Contoh data awal..... 57
Tabel 5.2 Contoh Data Hasil Pre-proses..... 59
Tabel 5.3 Jumlah *Cluster* dan Koefisien *Sillhoute*..... 60
Tabel 5.4 Jumlah laporan tiap *cluster*..... 62
Tabel 5.5 Dokumen *Cluster* ke- 1..... 63
Tabel 5.6 Dokumen *Cluster* ke-4..... 66
Tabel 5.7 Dokumen *Cluster* ke-5..... 68
Tabel 5.8 Bahasan tiap *Cluster*..... 72
Tabel 5.9 Hasil akurasi..... 76

BAB I
PENDAHULUAN

Pada bab ini akan dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang selanjutnya diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Kota Surabaya merupakan salah satu kota di Indonesia yang telah menerapkan teknologi informasi dan komunikasi dalam pemerintahan atau disebut *e-government*. Masyarakat dapat memberikan masukan terhadap kebijakan yang dibuat oleh pemerintah sehingga dapat meningkatkan kinerja pemerintah[1]. *E-government* di kota Surabaya meliputi berbagai macam bidang salah satunya media center[2].

Pemerintah Surabaya menyadari perlu adanya keterbukaan informasi kepada masyarakat dengan tidak adanya pembatasan kepada masyarakat perihal memberikan aspirasi, keluhan atau informasi terkait kota Surabaya sehingga akhirnya meluncurkan media center melalui dinas komunikasi dan informatika. Media center ini adalah sistem pelayanan terintegrasi bagi masyarakat Surabaya yang ingin berpartisipasi dalam perkembangan pembangunan kota Surabaya. Melalui media center, masyarakat juga dapat mengetahui sejauh mana tahapan pembangunan yang disusun oleh pemerintah kota Surabaya[3]. Sejak pertama kali di-*launching* pada 28 November 2011, media center telah banyak dimanfaatkan masyarakat. Berdasarkan data dari diskominfo Surabaya, laporan yang masuk pada tahun

pertama sebanyak 698 keluhan. Pada 2012, tercatat 2.717 keluhan. Tahun 2013 terdapat 4.176 keluhan dan 2014 sebanyak 4.298 keluhan[2].

Saat ini proses pengelompokan laporan masyarakat masih dilakukan secara manual oleh operator media center. Sedangkan laporan masyarakat yang masuk dari tahun ke tahun semakin banyak. Berdasarkan permasalahan tersebut diperlukan sebuah cara untuk mempermudah pihak media center dalam mengelompokkan data teks tersebut. Jumlah laporan masyarakat pada media center yang berukuran besar tersebut dapat disebut sebagai *Big data*. Menurut pengertian yang ada, *Big data* sendiri adalah data yang mempunyai jumlah dan variasi besar dan bergerak cepat[4].

Dokumen yang umumnya memiliki data dengan jumlah besar dan bervariasi dapat menyulitkan dalam membuat model klasifikasi. Sehingga dokumen-dokumen tersebut dapat dikelompokkan menurut kemiripan satu sama lain agar dapat dengan mudah diklasifikasi[5].

Oleh karena itu dalam tugas akhir ini, penulis melakukan penelitian terkait pengelompokan dan klasifikasi laporan masyarakat khususnya yang berasal dari situs media center Surabaya. Langkah yang digunakan adalah mengelompokkan menggunakan *K-Means Clustering* lalu diklasifikasi menggunakan *Support Vector Machine*(SVM). Dengan adanya pengelompokan terlebih dahulu menggunakan *K-Means Clustering* data yang awalnya masih belum memiliki label dapat terlabeli berdasarkan *cluster*. Sehingga dapat diklasifikasikan menggunakan SVM. Proses klasifikasi ini nantinya akan membentuk model klasifikasi.

Pada penelitian berikut dipilih *K-Means Clustering* dikarenakan algoritma ini memiliki efisiensi tinggi dalam

melakukan partisi data khususnya pada *dataset* yang besar[6]. Sedangkan SVM dipilih karena telah banyak digunakan dalam teknik *supervised machine learning* dekade terakhir. Diterapkan pada berbagai jenis permasalahan nyata karena memiliki nilai akurasi yang sangat baik dibandingkan teknik klasifikasi lain[6].

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam Tugas Akhir ini adalah :

1. Bagaimana mengelompokkan data laporan masyarakat di situs Media Center Surabaya menggunakan metode *K-Means Clustering*.
2. Bagaimana melakukan klasifikasi dari hasil pengelompokan data laporan masyarakat dengan menggunakan metode *Support Vector Machine* (SVM).

1.3 Batasan Masalah

Pada penelitian ini, penulis membuat batasan masalah sebagai berikut :

1. Data laporan masyarakat yang digunakan berasal dari situs Media Center Surabaya
2. Data laporan masyarakat yang digunakan dalam rentang waktu Januari 2016 hingga Desember 2016.
3. Data teks yang digunakan berbahasa Indonesia.
4. Perangkat lunak yang digunakan untuk mendukung pengerjaan Tugas Akhir ini adalah bahasa pemrograman Java dan software basis data MySQL.

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah untuk mengelompokkan data laporan masyarakat yang terdapat pada

situs Media Center Surabaya menggunakan metode *K-Means Clustering* dan melakukan klasifikasi dari hasil pengelompokan data laporan masyarakat dengan menggunakan metode *Support Vector Machine* (SVM).

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah sebagai berikut :

1. Memberikan pengetahuan bagi pembaca mengenai penggunaan metode *K-Means Clustering* dan *Support Vector Machine* (SVM) pada data teks khususnya data laporan masyarakat.
2. Memberikan alternatif baru bagi Media Center Surabaya dalam hal pengelompokan laporan masyarakat.
3. Memberikan informasi untuk digunakan dalam riset selanjutnya.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN
   Bab ini menjelaskan tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA
   Bab ini berisi tentang materi-materi yang mendukung Tugas Akhir ini, antara lain penelitian terdahulu, media center, *text mining*, *K-Means Clustering*, *Support Vector Machine* (SVM) dan Metode Evaluasi.
3. BAB III METODE PENELITIAN
   Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir.

4. BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM
   Pada bab ini akan menguraikan bagaimana perancangan dan tahapan tahapan dalam implementasi sistem, yaitu implementasi pre-proses teks, pembobotan TF-IDF dan SVD, serta pengelompokan dan klasifikasi menggunakan metode *K-Means Clustering* dan *Support Vector Machine*.
5. BAB V HASIL DAN PEMBAHASAN
   Bab ini menjelaskan mengenai hasil dari sistem yang telah dibuat disertai dengan penjelasan pembahasan.
6. BAB VI PENUTUP
   Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II`
TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Selain itu juga akan dijelaskan terkait penelitian terdahulu dan media center. Dasar teori yang dijelaskan adalah *text mining*, *K-Means Clustering*, *Support Vector Machine* (SVM) dan metode evaluasi.

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya, Ahmad Yusuf dan Tirta Priambadha[5] melakukan penelitian berjudul *Support Vector Machines* yang didukung *K-Means Clustering* dalam klasifikasi dokumen. Penulis mengusulkan sebuah metode baru untuk kategorisasi dokumen teks berbahasa Inggris dengan terlebih dahulu melakukan pengelompokan menggunakan *K-Means Clustering* kemudian dokumen diklasifikasikan menggunakan *multi-class Support Vector Machines* (SVM). Dengan adanya pengelompokan tersebut, variasi data dalam membentuk model klasifikasi akan lebih seragam. Hasil uji coba terhadap judul artikel jurnal ilmiah menunjukkan bahwa metode yang diusulkan mampu meningkatkan akurasi dengan menghasilkan akurasi sebesar 88,1% dengan parameter jumlah kelompok sebesar 5. Hasil tersebut kedepannya dapat digunakan sebagai landasan untuk pengembangan atau penelitian selanjutnya.

Selanjutnya adalah penelitian yang dilakukan oleh Chyntia Megawati [7], penulis melakukan penelitian tentang analisis aspirasi dan pengaduan di situs LAPOR! menggunakan *text mining*. LAPOR! (Layanan Aspirasi dan Pengaduan Online Rakyat) adalah *website* aspirasi dan pengaduan masyarakat yang dibangun oleh pemerintah. Penelitian ini menggunakan metode *text mining* untuk menganalisis data tekstual yang berupa opini atau keluhan

dengan mengklasifikasikannya menjadi beberapa kelas dan kemudian *data set* setiap kelas akan dikelompokkan lagi menjadi beberapa topik khusus (*cluster*). Pengklasifikasian menggunakan metode SVM (*support vector machine*) dan metode SOM (*self organizing-map*) untuk *clustering*. Hasil penelitian menunjukkan bahwa laporan terkait kemiskinan memiliki jumlah terbanyak dengan topik mayoritas yang dibahas adalah mengenai beberapa jenis bantuan sosial seperti KPS (Kartu Perlindungan Sosial) dan BLSM (Bantuan Langsung Sementara Masyarakat) yang tidak didistribusikan dengan baik atau tidak tepat sasaran.

Penelitian berikutnya adalah penelitian yang berjudul *Using Unsupervised Clustering Approach to Train The Support Vector Machine for Text Classification* yang dilakukan oleh Niusha Shafiabady dkk [8] . Penelitian ini mendeskripsikan metodologi SOM (*Self Organizing Maps*) dan menggunakan alternatif pengelompokkan otomatis CorrCoef (*Correlation Coeficient*). *Cluster* digunakan sebagai label untuk mengarah ke SVM (*Support Vector Machine*). Percobaan dan hasil akan disajikan berdasarkan penerapan dari metode tersebut pada beberapa data teks. Kesimpulan yang didapatkan dari penelitian ini adalah pendekatan *unsupervised clustering* dapat digunakan untuk melabeli data sebelum klasifikasi menggunakan SVM.

2.2 Media Center

Transparansi informasi sudah menjadi sebuah keharusan dalam tata kelola pemerintahan di era sekarang ini. Syarat penting terwujudnya penyelenggaraan pemerintahan yang baik *(good government)* dan demokratis adalah terbukanya akses informasi publik seluas-luasnya. Transparansi informasi ini bisa berarti dibukanya kemudahan, kejelasan, dan kecepatan bagi masyarakat yang ingin mengakses informasi layanan publik. Sebagaimana diamanatkan oleh Undang-

Undang Nomor 14 Tahun 2008 tentang keterbukaan informasi publik. Pemerintah kota Surabaya melalui dinas komunikasi dan informatika menyadari kebutuhan akan keterbukaan informasi publik sebagai salah satu upaya untuk mengembangkan masyarakat informasi, dengan mendirikan media center yang di-*launching* pada 28 November 2011. Media center digagas untuk menampung partisipasi masyarakat baik dalam bentuk keluhan, informasi, maupun saran pada proses pembangunan kota yang dilaksanakan oleh pemerintah kota Surabaya. Selain itu, media center juga berfungsi sebagai fasilitator untuk menghubungkan kebutuhan masyarakat Surabaya dengan pemerintah kota Surabaya dan menjadi jembatan komunikasi antar *stakeholder* kota Surabaya untuk mencapai tujuan pembangunan kota Surabaya[3].

Media center adalah sistem pelayanan terintegrasi bagi masyarakat Surabaya yang ingin berpartisipasi dalam perkembangan pembangunan kota Surabaya. Melalui media center, masyarakat juga bisa mengetahui sejauh mana tahapan pembangunan yang disusun oleh pemerintah, dapat dilaksanakan sesuai sasaran. Media center terdapat dalam berbagai media diantaranya *facebook*, *twitter*, *instagram*, layanan sms, *whatsapp* ataupun lini telepon. Bahkan masyarakat dapat langsung datang ke kantor media center yang bertempat di Jalan Jimerto.Berdasarkan surat keputusan (SK) Walikota Surabaya No. 188.45/54/436.1.2/2013, menyebutkan bahwa masing-masing satuan kerja perangkat daerah (SKPD) menugaskan satu personel sebagai Tim Pelayanan Keluhan/Pengaduan Masyarakat (TPKPM). Keluhan warga yang diterima operator langsung dikirimkan

ke personel TPKPM yang ada di dinas terkait. Kemudian, jawaban dari tim harus diberikan kepada pelapor.

2.3 *Text Mining*

Text mining merupakan bidang interdisiplin yang mengacu pada perolehan infomasi (*information retrieval*), *data mining*, pembelajaran mesin (*machine learning*), statistik dan komputasi linguistik[9]. *Text mining* juga merupakan variasi dari *data mining* yang berusaha menemukan pola menarik dari sekumpulan data tekstual yang berjumlah besar[10]. Perbedaan mendasar antara *text mining* dan *data mining* terletak pada sumber data yang digunakan. Pada data mining, pola-pola diekstrak dari basis data yang terstruktur, sedangkan di *text mining*, pola-pola diekstrak dari data tekstual (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis, sedangkan teks tertulis untuk dibaca langsung oleh manusia [11].

Langkah-langkah yang dilakukan dalam *text mining* adalah sebagai berikut :

1. Pre-proses teks

   Langkah-langkah dalam pra-proses teks terbagi menjadi :

   a. *Case folding* dan *Tokenizing*

      Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf “a” sampai dengan “z” yang diterima. Karakter selain huruf dihilangkan. Tahap *tokenizing/parsing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Proses pemisahan teks menjadi potongan kalimat dan kata yang disebut token.

b. *Filtering*

Filtering adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak *deskriptif* yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah "yang", "dan", "di", "dari", dan seterusnya.

c. *Stemming*

Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen[11] . Sebagai contoh, kata bersama, kebersamaan, menyamai, akan di-*stem* ke *root* katanya yaitu "sama". Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan *sufiks*. Sedangkan pada teks berbahasa Indonesia, selain *sufiks*, *prefiks*, dan *konfiks* juga dihilangkan [12].

2. *Text Transformation*

Transformasi teks atau pembentukan atribut mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan representasi dokumen yang lazim digunakan oleh model "*bag of words*" dan model ruang vektor (*vector space model*). Transformasi teks sekaligus juga

melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen.Pada umumnya, dokumen akan diwakili oleh vektor[10]. Model vektor dibangun dari dokumen dengan mengubah *token-token* dalam dokumen menjadi vektor numerik yang akan dioperasikan berdasarkan operasi aljabar linear[14]. Dalam rangka membangun model vektor, perlu dilakukan proses pembobotan. Skema pembobotan yang paling banyak digunakan adalah skema *term frequency-inverse document frequency* (TF-IDF). *Term frequency* (TF) didefinisikan sebagai jumlah kemunculan suatu kata/istilah dalam suatu dokumen. Misalnya TF pada dokumen pertama untuk kata/istilah "jalan" adalah 2, karena kata/istilah tersebut muncul 2 kali dalam dokumen pertama. Pada asumsi pembobotan dibalik TF-IDF, kata-kata dengan nilai TF yang tinggi akan mendapat bobot yang tinggi kecuali jika jumlah dokumen yang mengandung kata tersebut juga tinggi (*inverse document frequency* (IDF)). Misalnya kata "yang" memiliki jumlah kemunculan yang tinggi tetapi jumlah dokumen yang mengandung kata "yang" juga tinggi, sehingga kata tersebut akan memiliki bobot yang rendah. Skema persamaan TF-IDF yang digunakan pada penelitian ini ditunjukkan oleh persamaan berikut:

$$tfidf(w) = tf(w)x\ (1 + \log N - \log df(w)) \qquad (1)$$

Keterangan:

tf (w) = *Term frequency* (jumlah kemunculan suatu kata dalam suatu dokumen)

df(w) = *Document frequency* (jumlah dokumen yang mengandung suatu kata)

N = Jumlah dokumen

Setelah melewati skema TF-IDF, akan didapatkan hasil yang berupa matriks. Matriks yang didapatkan adalah matriks yang merepresentasikan dokumen dalam kolom dan *token-token* atau kata yang sudah dipisah-pisahkan dalam baris. Selanjutnya dilakukan satu tahapan lagi yaitu *Singular Value Decomposition*(SVD).

Singular Value Decomposition (SVD) merupakan suatu metode untuk mengidentifikasi dan mengurutkan dimensi yang menunjukkan data mana yang mempunyai variasi paling banyak. Berkaitan dengan hal tersebut, SVD dapat mengidentifikasi di mana variasi muncul paling banyak, sehingga hal ini memungkinkan untuk mencari pendekatan yang terbaik pada data asli menggunakan dimensi yang lebih kecil. Oleh karena itu, SVD dapat dilihat sebagai metode pengurangan data. Proses pereduksian dengan SVD akan semakin menegaskan kemiripan data yang mirip dan menegaskan ketidakmiripan data yang tidak mirip[13].

SVD akan menguraikan sebuah matriks menjadi tiga buah matriks baru yaitu matriks vektor singular kiri, matriks nilai singular, dan vektor singular kanan. SVD dari sebuah matriks A yang berdimensi mxn adalah sebagai berikut :

$$A_{(mxn)} = U_{(mxd)} S_{(dxd)} V^{T}_{(dxn)} \quad (2)$$

Keterangan :

A_{mxn} = Matriks yang mewakili m jumlah kata pada dokumen dan n jumlah dokumen

U = Vektor eigen ortonormal dari AA^T

V = Vektor eigen ortonormal dari A^TA

S = Matriks diagonal dimana nilai diagonalnya merupakan akar dari nilai U dan V yang disusun dengan urutan menurun berdasarkan besarnya nilai (nilai singular dari A).

d = nilai akar dari jumlah dokumen

3. *Feature Selection*

Pemilihan fitur (kata) merupakan tahap lanjut dari pengurangan dimensi pada proses transformasi teks. Pemilihan hanya dilakukan terhadap kata-kata yang relevan yang benar-benar mempresentasikan isi dari suatu dokumen. Algoritma yang digunakan pada *text mining*, biasanya tidak hanya melakukan perhitungan pada dokumen saja tetapi juga pada *feature*. Terdapat empat macam feature yang sering digunakan:

- *Character*, komponen individual, bisa huruf, angka, karakter spesial dan spasi, merupakan block pembangun pada level paling tinggi pembentuk semantik *feature* seperti kata, *term* dan *concept*. Representasi *character based* jarang digunakan pada beberapa teknik pemrosesan teks.
- *Words*
- *Terms* adalah *single word* dan *multiword phrase* yang terpilih secara langsung dari corpus.

- *Concept*, merupakan *feature* yang di-*generate* dari sebuah dokumen secara manual, *rule based* atau metodologi lain.

4. *Pattern Discovery*

Tahapan ini merupakan tahap penting untuk menemukan pola dari keseluruhan teks. Dalam penemuan pola ini, proses *text mining* dikombinasikan dengan proses-proses *data mining*. Masukan awal dari proses *text mining* adalah suatu data teks dan menghasilkan keluaran berupa pola sebagai evaluasi.

Terdapat beberapa jenis kategori utama yang bisa dilakukan sebagai berikut [7] :

1. Klasifikasi/prediksi

Klasifikasi adalah bentuk analisis data yang mengekstrak model untuk menggambarkan kelas data[10]. Model yang dibangun meliputi pengklasifikasian dan prediksi kategori label kelas. Klasifikasi data mempunyai dua tahapan proses, yaitu tahap pembelajaran (*learning step*) dimana model klasifikasi dibangun berdasarkan label yang sudah diketahui sebelumnya dan tahapan klasifikasi (*classification step*) dimana model digunakan untuk memprediksi label kelas dari data yang diberikan. Klasifikasi memiliki berbagai aplikasi, termasuk deteksi penipuan, penargetan marketing, prediksi kinerja, manufaktur, diagnosis medis, dan banyak lainnya. Sebagai contoh, kita dapat membangun sebuah model klasifikasi untuk mengkategorikan apakah suatu aplikan pinjaman bank termasuk aman atau berisiko. Karena padaawal pembangunan model label kelas dari data telah

diketahui, klasifikasi juga disebut sebagai metode *supervised learning.*

2. *Clustering*

Tidak seperti klasifikasi, kelompok label kelas pada model *clustering* tidak diketahui sebelumnya dan tugas *clustering* adalah untuk mengelompokkannya. Menurut Linof & Berry, *clustering* adalah proses pengelompokan satu set data objek menjadi beberapa kelompok atau *cluster* sehingga objek dalam sebuah *cluster* memiliki kemiripan yang tinggi satu sama lain, tetapi sangat berbeda dengan objek dalam kelompok lainnya.

3. Asosiasi

Asosiasi merupakan proses pencarian hubungan antar elemen data. Dalam dunia industri retail, analisis asosiasi biasanya disebut *Market Basket Analysis*[7]. Asosiasi tersebut dihitung berdasarkan ukuran *support* (presentase dokumen yang memuat seluruh konsep suatu produk A dan B) dan *confidence* (presentase dokumen yang memuat seluruh konsep produk B yang berada dalam subset yang sama dengan dokumen yang memuat seluruh konsep produk A).

4. Analisis Tren

Tujuan dari analisis tren yaitu untuk mencari perubahan suatu objek atau kejadian oleh waktu[7]. Salah satu aplikasi analisis tren yaitu kegiatan identifikasi evolusi topik penelitian pada artikel akademis.

2.4 *K-Means Clustering*

K-means Clustering merupakan salah satu metode data *clustering* yang berusaha mempartisi N jumlah data ke dalam K jumlah kelompok/klaster. *K-means Clustering* melakukan partisi data ke dalam kelompok/klaster sehingga data yang memiliki karakteristik yang sama akan dikelompokkan ke dalam satu klaster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. *K-Means Clustering* merupakan metode pengelompokan paling sederhana yang mengelompokkan data kedalam *k* kelompok berdasar pada *centroid* masing-masing kelompok. Hanya saja hasil dari *K-Means Clustering* sangat dipengaruhi parameter *k* dan inisialisasi *centroid*. Pada umumnya *K-Means Clustering* menginisialisasi *centroid* secara acak[14]. Menurut Santosa[15]**,** langkah-langkah melakukan clustering dengan metode *K-Means Clustering* adalah sebagai berikut:

a. Pilih jumlah *cluster k*.

b. Inisialisasi *k* pusat *cluster* ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara random. Pusat-pusat *cluster* diberi nilai awal dengan angka-angka random.

c. Alokasikan semua data/ objek ke *cluster* terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data ke *cluster* tertentu ditentukan jarak antara data dengan pusat *cluster*. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling antara satu data dengan satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana. Untuk menghitung jarak semua data ke

setiap titik pusat *cluster* menggunakan teori jarak *euclidan*, dengan rumus sebagai berikut[15] :

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

Keterangan :

d(x,y) = jarak antara x dengan y

x_i = nilai bobot kata ke-i pada data.

y_i = nilai bobot kata ke-i pada data pusat *cluster*

n = jumlah atribut yang digunakan

d. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/ objek dalam *cluster* tertentu. Jika dikehendaki bisa juga menggunakan median dari cluster tersebut. Jadi rata-rata bukan satu-satunya ukuran yang bisa dipakai.

e. Tugaskan lagi setiap objek memakai pusat *cluster* yang baru. Jika pusat cluster tidak berubah lagi maka proses *clustering* selesai. Atau, kembali ke langkah c sampai pusat *cluster* tidak berubah lagi.

2.5 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) pertama kali dikembangkan oleh *Boser, Guyon*,dan *Vapnik*. Pada tahun 1992 ketika diadakan di *Annual Workshopon Computational Learning Theory*. *Support Vector Machine* (SVM) merupakan sistem pembelajaran yang pengklasifikasiannya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah

ruang fitur (*feature space*) berdimensi tinggi. Dalam konsep SVM berusaha menemukan fungsi pemisah (*hyperplane*) terbaik diantara fungsi yang tidak terbatas jumlahnya. *Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. Pada awalnya prinsip kerja dari SVM yaitu mengklasifikasi secara linear (linear classifier),kemudian SVM dikembangkan sehingga dapat bekerja pada klasifikasi non linear dengan memasukkan konsep kernel trick pada ruang kerja berdimensi tinggi.

SVM dapat digunakan untuk mengklasifikasikan data lebih dari dua kelas yang selanjutnya disebut SVM multi kelas. Salah satu metodenya yaitu *One-against-one*. Adapun untuk metode ini, akan dikontruksi sejumlah k(k-1)/2 model klasifikasi SVM dengan masing-masing model dilatih menggunakan data dari dua kelas yang berbeda. Dengan demikian, untuk training data dari kelas ke-*i* dan ke-*j*, diselesaikan persoalan klasifikasi dua kelas berikut[5].

$$\min_{w_{ij},b_{ij},\varepsilon_{ij}} \frac{1}{2} w_{ij}^T w_{ij} + C \sum_{1} \varepsilon_{ij}^1$$

bergantung pada

$$w_{ij}^T \phi(x_1) + b_{ij} \geq 1 - \varepsilon_{ij}^1, jika\ y_1 = i$$

$$w_{ij}^T \phi(x_1) + b_{ij} \leq 1 + \varepsilon_{ij}^1, jika\ y_1 = j$$

$$\varepsilon_{ij}^1 \geq 0 \qquad (4)$$

Keterangan :

- Data *training* $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$ dimana $x_i \in R^n, i = 1,2, \dots, n\ dan\ y_i \in \{1,2, \dots, k\}$

- ϕ fungsi non linear yang mengarahkan data *training* x_i ke ruang dimensi yang lebih tinggi
- C adalah parameter *penalty*
- w adalah bobot vektor kelas
- b adalah bias kelas
- ε^1 adalah variabel slack yang mempunyai indeks dari data setiap kelas sama dengan 1.

Meminimalkan $\frac{1}{2}w_{ij}^T w_{ij}$ berarti memaksimalkan $2/\|w_{ij}\|$ margin antara dua kelompok. Ketika data tidak linear dipisahkan, penalti $C\sum_1 \varepsilon_{ij}^1$ dapat mengurangi jumlah kesalahan *training*. Setelah semua fungsi pemisah k(k-1)/2 ditemukan, ada beberapa metode untuk melakukan *testing* data baru. Salah satu strategi adalah *max-voting*. Berdasarkan pada strategi ini, jika hasil dari $w_{ij}^T\phi(x) + b_{ij}$menyatakan bahwa data *x* berada di kelas *i*, maka nilai *vote* untuk kelas *i* ditambah satu. Selanjutnya, prediksi kelas dari data *x* adalah kelas dengan nilai *vote* tertinggi. Jika sebaliknya, nilai vote untuk kelas *j* ditambah satu.

SVM pada awalnya digunakan untuk klasifikasi data numerik, tetapi ternyata SVM juga sangat efektif dan cepat untuk menyelesaikan masalah-masalah data teks. Data teks cocok untuk dilakukan klasifikasi dengan algoritma SVM karena sifat dasar teks yang cenderung mempunyai dimensi yang tinggi, dimana terdapat beberapa fitur yang tidak relevan, tetapi akan cenderung berkorelasi satu sama lain dan umumnya akan disusun dalam kategori yang terpisah secara linear[14].

2.6 Metode Evaluasi

Pada penelitian ini terdapat dua metode evaluasi, yaitu evaluasi proses *clustering* dan evaluasi proses klasifikasi. Metode evaluasi yang digunakan untuk proses *clustering* adalah Koefisien *Sillhoute*. Algoritma *Sillhoute* berperan besar dalam *clustering* dan merupakan algoritma yang mengenali kualitas pembagian data. Nilai koefisien *sillhoute* berada pada rentang -1 hingga 1 dimana semakin mendekati -1 berarti hasil pembagian data tersebut buruk, dan sebaliknya. Nilai koefisien *sillhoute* didapatkan dari[16] :

$$s(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}} \qquad (5)$$

Keterangan :

i = objek data ke- *i*

a(i) = rata-rata jarak *i* terhadap objek lain pada *cluster* yang sama

b(i) = nilai minimum dari rata-rata jarak *i* terhadap objek lain pada *cluster* yang berbeda

s(i) = nilai koefisien *sillhoute*

Berikut ini ukuran nilai *sillhoute* berdasarkan kaufman dan rousseeuw :

0,7 < koefisen *sillhoute* <= 1 (*strong structure*)

0,5 < koefisen *sillhoute* <= 0,7 (*medium structure*)

0,25 < koefisen *sillhoute* <= 0,5 (*weak structure*)

koefisen *sillhoute* <= 0,25 (*no structure*)

Sedangkan untuk metode klasifikasi penulis menggunakan rumus *accuration*. Adapun pengertian dari *accuration* adalah didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai *actual* yang dirumuskan. *Accuration* digunakan untuk mengevaluasi model klasifikasi.

$$Accuration = \frac{Total\ data\ dengan\ prediksi\ yang\ benar}{Total\ data\ yang\ diuji} \qquad (6)$$

BAB III
METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Tahapan Penelitian

Langkah-langkah yang digunakan dalam menyelesaikan Tugas Akhir ini adalah sebagai berikut :

1. **Pengumpulan Data**

   Pengumpulan data yang dimaksud adalah data laporan masyarakat yang terdapat di situs Media Center Surabaya dengan rentang waktu Januari 2016 - Desember 2016.

2. **Studi Literatur**

   Studi Literatur ini dilakukan untuk identifikasi permasalahan dengan mencari referensi yang menunjang penelitian yang berupa Tugas Akhir, jurnal internasional, buku, maupun artikel yang berhubungan dengan topik Tugas Akhir ini.

3. **Pre-proses Teks**

   Pre-proses teks terdiri dari beberapa tahapan yaitu :

   a. *Case Folding* dan *Tokenizing*

   Pada tahap ini semua huruf akan diubah menjadi huruf kecil. Hanya huruf “a” sampai dengan “z” yang diterima. Karakter selain huruf dihilangkan. Lalu dilakukan tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Proses pemisahan teks menjadi potongan kalimat dan kata yang disebut token.

   b. Filtering

   Selanjutnya akan diambil kata-kata penting dari hasil token. Menggunakan algoritma *stoplist*

(membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).

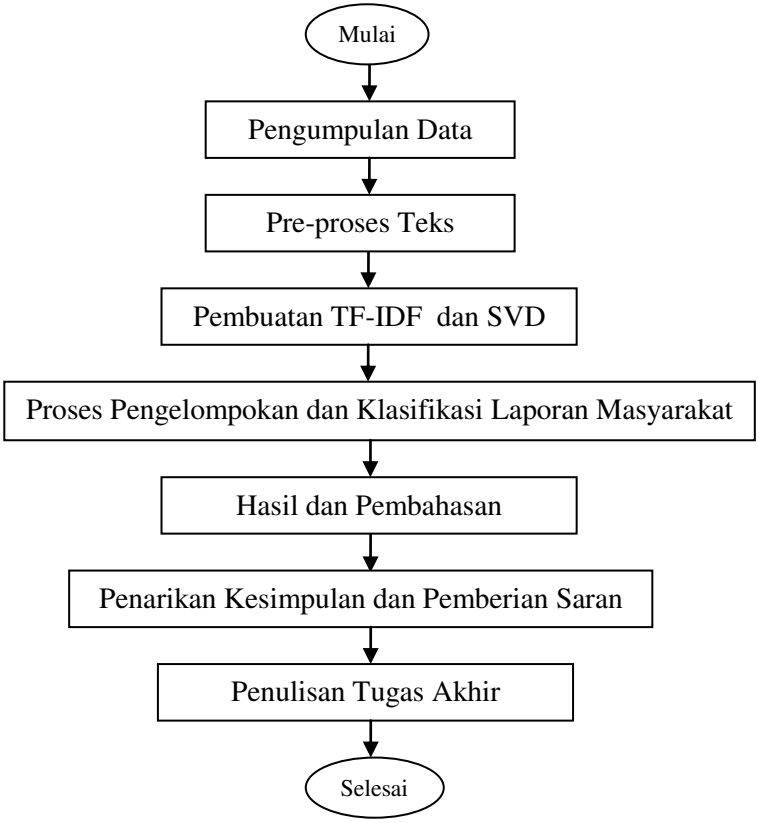
4. **Pembuatan TF-IDF dan SVD**
   Dilakukan proses pembobotan yaitu menggunakan skema term *frequency-inverse document frequency* (TF-IDF). Setelah melewati skema TF-IDF akan didapatkan hasil yang berupa matriks. Matriks tersebut selanjutnya melalui tahap *Singular Value Decomposition* (SVD).
5. **Proses Pengelompokan dan Klasifikasi Laporan Masyarakat**
   Dari hasil pra-proses dan pembuatan TF-IDF serta SVD lalu laporan masyarakat dikelompokkan dengan *K-Means Clustering*. Tujuan dari tahap pertama ini adalah untuk memberikan label pada setiap data teks berdasarkan *cluster* dimana data itu berada. Sehingga dihasilkan data teks yang sudah terlabeli. Selanjutnya laporan tersebut akan diklasifikasi menggunakan SVM (*Support Vector Machine*) yang kemudian akan didapatkan model klasifikasi.
6. **Hasil dan Pembahasan**
   Pada tahap ini akan dilakukan pembahasan hasil pengolahan data.
7. **Penarikan kesimpulan dan saran**
   Setelah dilakukan pembahasan selanjutnya akan dilakukan penarikan kesimpulan dari penelitian. Selain itu juga akan diberikan saran sebagai masukan dalam pengembangan penelitian selanjutnya.
8. **Penulisan Tugas Akhir**
   Tahap terakhir yang dilakukan dalam Tugas Akhir ini adalah penulisan Tugas Akhir dan membukukan hasil penelitian yang telah dilakukan.

Pembuatan perangkat lunak dalam tahap pre-proses teks, pembuatan TF-IDF dan SVD serta proses klasifikasi laporan

masyarakat adalah perangkat lunak sebagai *tools* pendukung pengerjaan tugas akhir ini.

3.2 Diagram Alir Penelitian

Berdasarkan uraian tersebut diatas, penelitian Tugas Akhir ini dapat dinyatakan dalam diagram alir sebagai berikut.



Gambar 3. 1 Diagram Alir Penelitian

BAB IV
PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini menjelaskan tentang perancangan yang diperlukan dan implementasi sistem. Selain itu juga akan dijelaskan mengenai tampilan antarmuka program.

4.1 Perancangan Data

Data-data yang digunakan dalam program ini dapat dibedakan menjadi tiga jenis, yaitu data masukan, data proses dan data keluaran.

4.1.1 Data Masukan

Data masukan adalah data-data yang digunakan sebagai masukan dari program. Masukan-masukan ini yang kemudian diolah oleh program melalui tahap-tahap tertentu sehingga menghasilkan keluaran yang diinginkan. Data masukan yang digunakan adalah :

1. Data laporan masyarakat yang telah disimpan dalam *database*
2. Data kumpulan *stopword* yang telah tersimpan di dalam *database*

Database laporan masyarakat memiliki dua atribut yaitu :

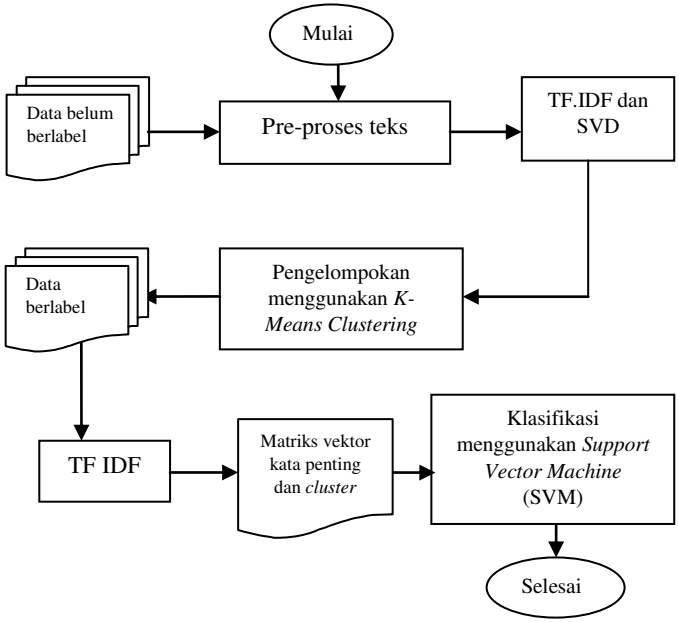
1. id merupakan nomor laporan
2. teks merupakan teks laporan masyarakat

4.1.2 Data Keluaran

Data keluaran merupakan data yang dihasilkan oleh program setelah proses-proses tertentu selesai dilakukan. Terdapat dua data keluaran pada program ini, yaitu :

1. Data teks yang sudah terlabeli atau terkelompokkan
2. Model klasifikasi

4.2 Perancangan Diagram Alir Sistem



Gambar 4. 1 Diagram Alir Sistem

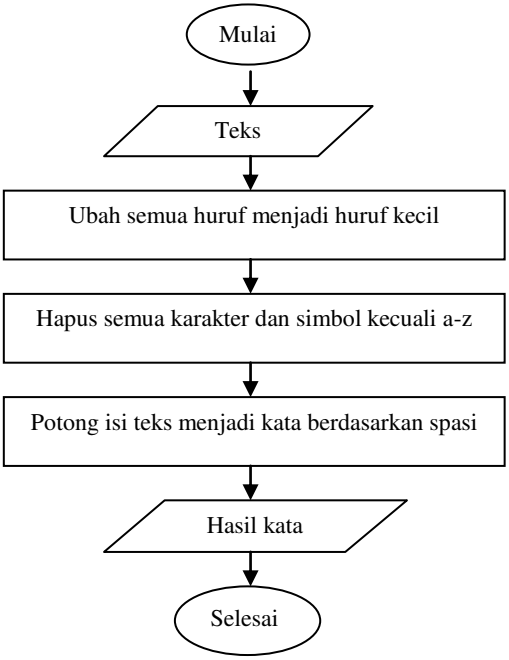
Pada diagram alir diatas menjelaskan tentang tahapan-tahapan yang akan dilakukan untuk mencapai tujuan tugas akhir. Dalam tugas akhir ini *tools* untuk komputasi menggunakan bahasa pemrograman java dan software basis data mysql.

Tahapan pre-proses data terdiri dari *case folding*, *tokenizing* dan *filtering*. Tahapan selanjutnya adalah pengelompokan menggunakan *k-means clustering* sebagai pelabelan data berdasarkan *cluster*. Kemudian data yang terlabeli akan melalui proses pembobotan kembali sehingga dapat

dilanjutkan pada proses klasifikasi menggunakan metode *support vector machine* (SVM).

4.2.1 Pre-proses Teks

4.2.1.1 *Case Folding* dan *Tokenizing*



Gambar 4. 2 Diagram Alir *Case Folding* dan *Tokenizing*

Pada tahap ini akan dilakukan proses menghilangkan karakter selain huruf, menghilangkan *url* serta mengubah huruf besar menjadi huruf kecil. Berikut adalah contoh hasil data *casefolding* dan *tokenizing* :

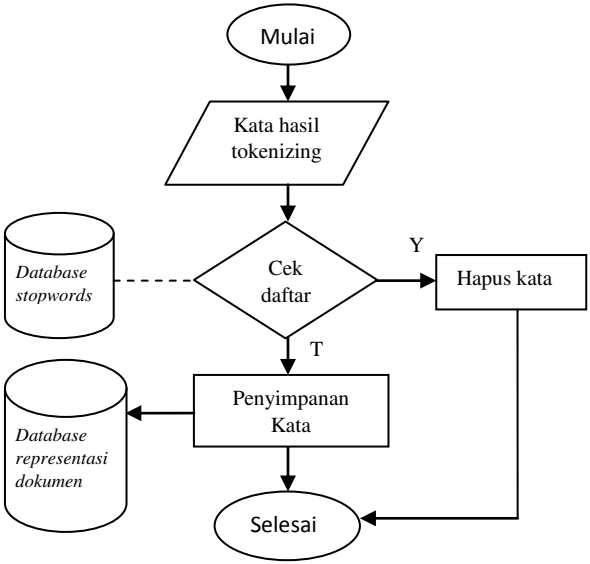
Tabel 4. 1 Contoh Data Awal

Mohon bantuan, untuk lampu PJU di jl. Pagesangan baru 7 (belakang rumah makan kauman dan biro travel adzikra) banyak yg mati. Sehingga jika malam tiba sangat gelap

Tabel 4. 2 Contoh Hasil *Casefolding* dan *tokenizing*

mohon bantuan untuk lampu pju di jl pagesangan baru belakang rumah makan kauman dan biro travel adzikra yg mati sehingga jika malam tiba sangat gelap

4.2.1.2 *Filtering*



Gambar 4. 3 Diagram Alir *Filtering*

Setelah data melalui proses *case folding* dan *tokenizing* selanjutnya token yaitu sebutan untuk hasil *tokenizing* akan di-*filter*. Berikut adalah contoh hasil data yang melalui proses *filtering* :

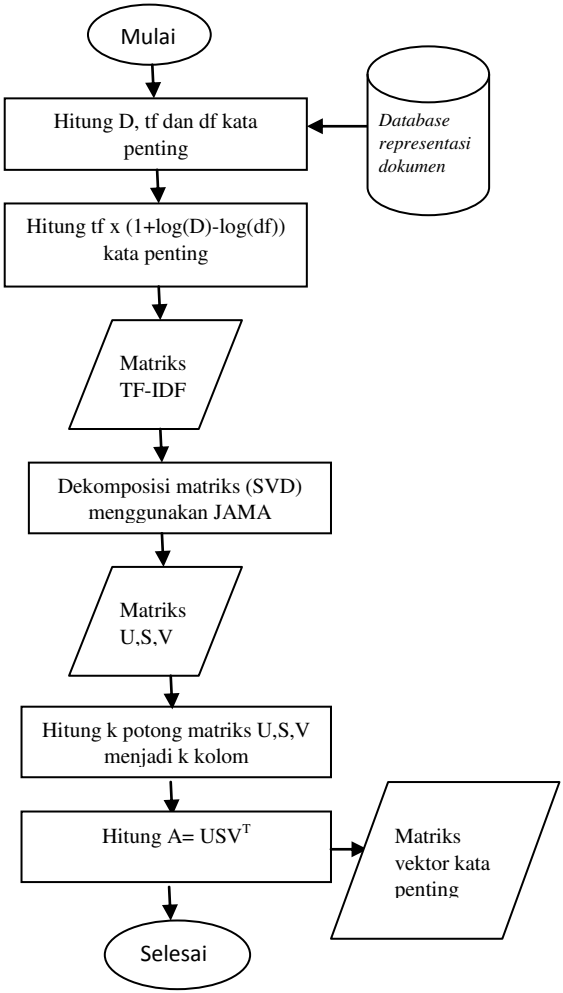
Tabel 4. 3 Sebelum Melalui *Filtering*

mohon bantuan untuk lampu pju di jl pagesangan baru belakang rumah makan kauman dan biro travel adzikra yg mati sehingga jika malam tiba sangat gelap

Tabel 4. 4 Setelah Melalui *Filtering*

mohon bantuan lampu pju jl pagesangan rumah makan kauman biro travel adzikra yg mati malam gelap

4.2.2 Pembobotan TF-IDF dan SVD



Gambar 4. 4 Diagram Alir Pembobotan

Pada proses TF-IDF dan SVD ini tahapan-tahapan yang dilalui adalah membangun matriks kemunculan tiap kata penting pada tiap dokumen, menghitung nilai *term frequency(tf)* dan *invers document frequency (idf)* dan mengaplikasikan metode *Singular Value Decomposition (SVD)* pada matriks yang telah terbentuk. Pada proses ini penulis menggunakan library JAMA dalam perhitungan SVD.

Langkah-langkah :

1. Menghitung frekuensi kemunculan kata penting pada dokumen (tf)
2. Menghitung jumlah koleksi dokumen yang ada (D)
3. Menghitung jumlah dokumen yang mengandung kata penting tersebut (df)
4. Menghitung tf x (1+log(D)-log(df)) yang merupakan nilai *inverse document frequency* (idf) setiap kata
5. Melakukan dekomposisi matriks hasil TF-IDF tersebut menjadi matriks U, matriks S, serta matriks V. Matriks U merupakan vektor kata penting, matriks S merupakan vektor sigma dan matriks V merupakan vektor dokumen
6. Menghitung k = akar jumlah kolom matriks yang merupakan nilai batas pemotongan kolom pada matriks U,S dan V
7. Menghitung $A = USV^T$
8. Menyimpan seluruh kata beserta matriks barisnya sebagai matriks vektor kata penting.

Berikut merupakan contoh dari hasil perhitungan TF-IDF :

Tabel 4. 5 Contoh Hasil Perhitungan TF-IDF

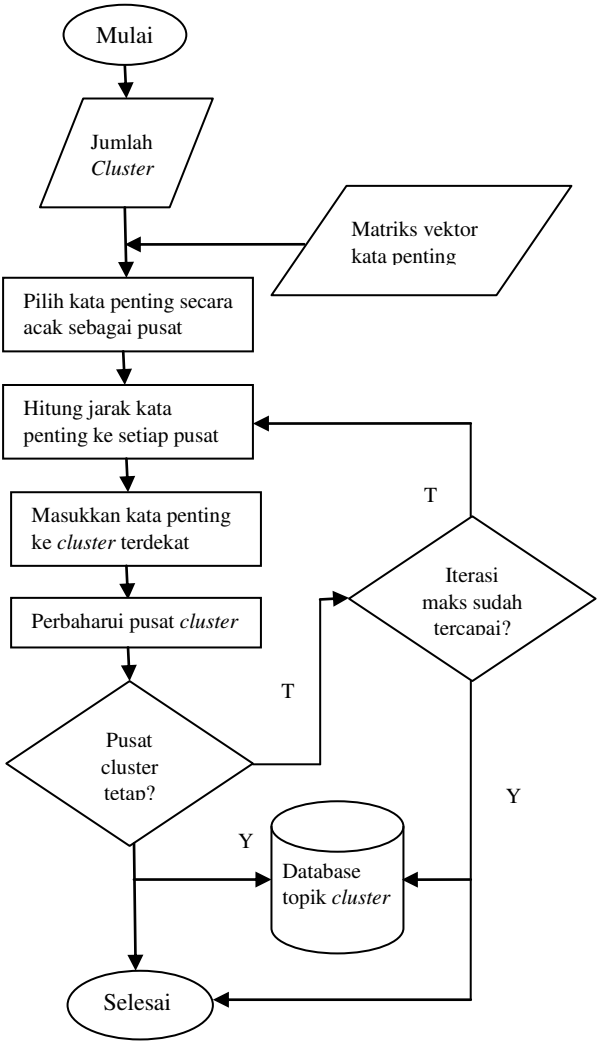
Dok. / Kata	1	2	3	4	5
air	2.7959	0.0000	0.0000	1.3979	0.0000
apotik	0.0000	0.0000	1.6990	0.0000	0.0000
area	0.0000	0.0000	0.0000	0.0000	1.6990
…	..	..	…	…	…

Berikut merupakan contoh dari hasil perhitungan SVD :

Tabel 4. 6 Contoh Hasil Perhitungan SVD

2.95	-0.00	-0.00	0.90	-0.00
0.00	0.00	1.69	0.00	0.00
0.00	0.85	0.00	0.00	0.85
….	….	…..	….	…..

4.2.3 Pengelompokan Data menggunakan *K-Means Clustering*



Gambar 4. 5 Diagram Alir K-Means Clustering

Pada tahapan ini hasil pembobotan TF-IDF dan SVD akan melalui proses pengelompokan menggunakan *K-Means clustering*. Proses ini bertujuan untuk mengelompokkan kata-kata penting ke dalam beberapa *cluster*. Langkah-langkah dalam tahapan ini adalah :

1. Memasukkan *input* berupa jumlah *cluster* yang diinginkan.
2. Memilih kata penting secara acak sebagai pusat *cluster*
3. Menghitung jarak setiap kata penting ke setiap pusat dengan menggunakan persamaan *Euclidean distance*
4. Jika jarak suatu kata penting terhadap pusat *cluster* lebih kecil daripada ke pusat *cluster* yang lain, maka kata penting dimasukkan ke *cluster* tersebut
5. Memperbaharui nilai pusat *cluster* dengan mencari nilai rata-rata vektor dari seluruh anggota *cluster*.
6. Menghitung kembali jarak tiap kata penting ke pusat *cluster* untuk mengalokasikan kembali tiap kata ke dalam *cluster*
7. Mengulangi tahap 5 dan 6 sampai pusat tidak berubah atau nilai iterasi maksimal sudah dicapai
8. Menyimpan seluruh *cluster* beserta kata penting yang ada di dalamnya ke dalam *database*.

Berikut merupakan contoh hasil pengelompokan kata penting menggunakan *K-Means Clustering* :

Tabel 4. 7 Contoh Hasil Pengelompokan Kata Penting

Id *Cluster*	Kata Penting
0	ktp, kecamatan,kelurahan,kk,…
1	jalan,warga,dispendukcapil
2	anak,dhuafa,kemiskinan,program,…
3	kriminal,tol,longsor,gempa,kebakaran,..
4	air,pengurusan,pdam,sertifikat,…
5	pju,perbaikan,padam,proyek,lampu,…
6	penanaman, perawatan,pohon,…
7	pkl,pedagang,usaha,..
8	kota,ruang, terbuka, hijau,…
9	rt,rw,sampah,parkir,rumah,…

Setelah kata-kata penting terkelompokan pada *cluster* yang sesuai. Selanjutnya akan dibentuk vektor *cluster* dari dokumen. Pembentukan ini bertujuan untuk mengukur kedekatan suatu dokumen terhadap kata-kata penting dari setiap *cluster*. Proses ini dilakukan dengan menghitung jumlah kata yang sama dengan kata pada setiap *cluster*. Nilai tersebut dinormalisasi cara membagi setiap nilai dengan jumlah kata total yang ada pada dokumen tersebut. Berikut tahapan pembentukan vektor *cluster* dokumen :

Input

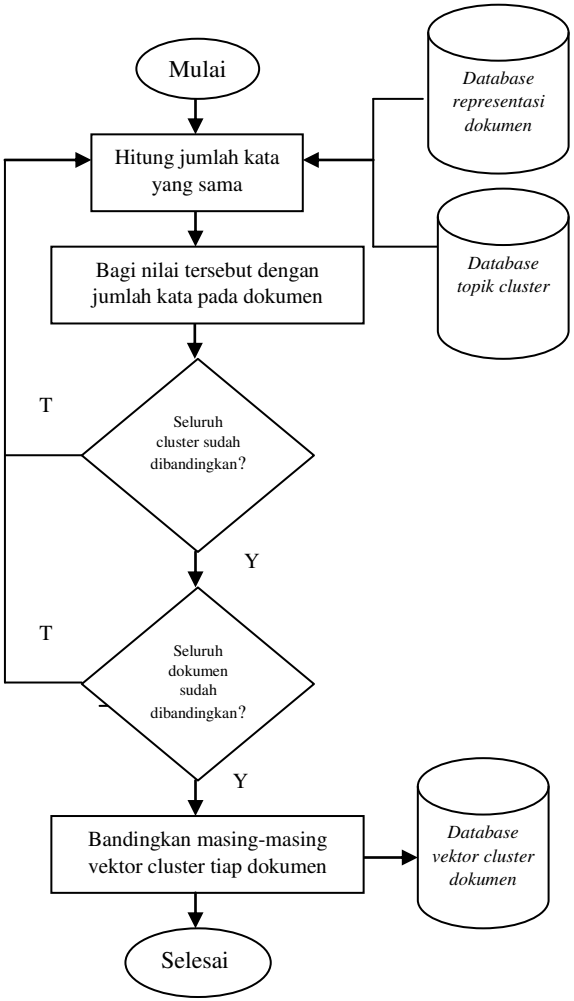
- Dokumen dan *cluster*

Output

- Vektor *cluster* dokumen

Langkah-langkah

1. Membaca dokumen beserta kata representasinya dari *database*
2. Membaca *cluster* beserta kata penting yang termasuk dalam *cluster* tersebut
3. Menghitung jumlah kata penting yang sama
4. Menormalisasi nilai tersebut dengan cara membagi dengan jumlah kata pada dokumen tersebut
5. Mengulangi langkah 2,3 dan 4 untuk setiap cluster
6. Mengulangi langkah 1,2,3 dan 4 untuk setiap dokumen
7. Membandingkan masing-masing vektor tiap *cluster* dalam dokumen
8. Menyimpan seluruh vektor cluster dokumen dan nilai *cluster* dokumen ke dalam *database* .



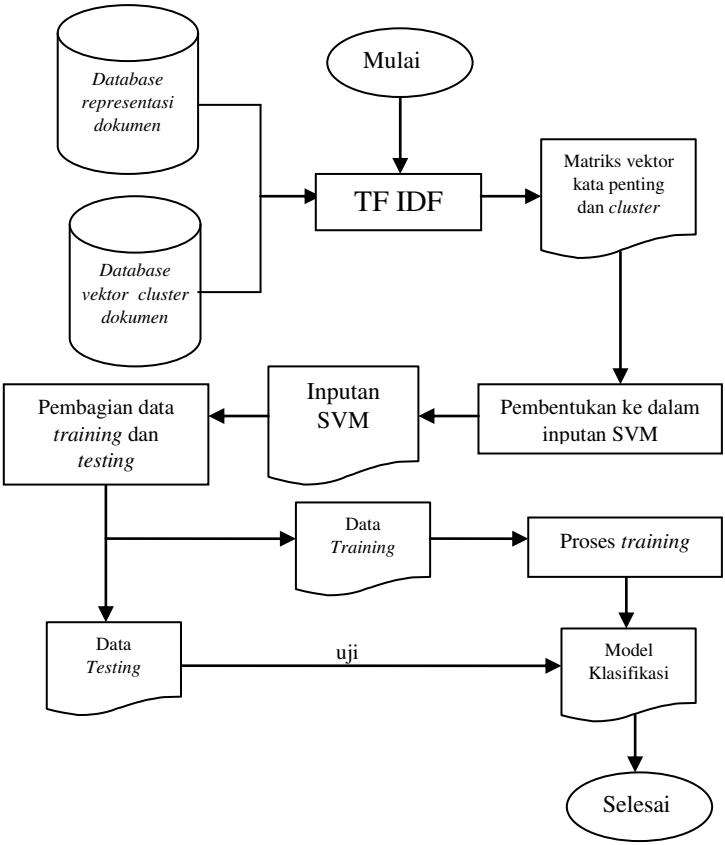
Gambar 4. 6 Diagram Alir Pembentukan Vektor Dokumen

Berikut merupakan contoh tabel data hasil pengelompokan :

Tabel 4. 8 Contoh Dokumen dan Label *Cluster*

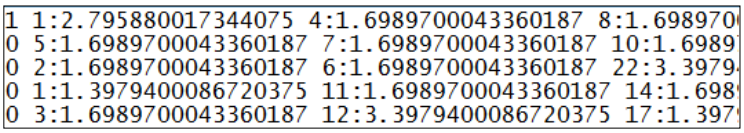
Teks	*Cluster*
Banyak jalan berlubang sepanjang jalan dharmahusada kearah kertajaya Dan sebaliknya disitu menuju arah kantor asked dharmahusada.. Lubangnya cukup besar2 krn di daerah test kalo hujan sering tergenang air terimakasih	1
Tolong jalan tembaan jalan banyak yg bergelombang, Tolong respon untuk diperbaiki. Terima Kasih	1
mohon dengan sangat jalan jalan yg rusak dan berlubang di surabaya di perbaiki terutama di daerah osowilangun,perak barat dan timur,kenjeran kapasan tolong dengan sangat untuk dinas terkait di perhatikan keselamatan warga surabaya terima kasih	1
Mohon bantuan, untuk lampu PJU di jl. Pagesangan baru 7 (belakang rumah makan kauman dan biro travel adzikra) banyak yg mati. Sehingga jika malam tiba sangat gelap	5
Air PDAM didaerah balong sari dan sekitar sdh sehari semalam mati. Tolong SEGERA dilakukan perbaikan. Kita semua butuh air. Terima Kasih.	4

4.2.4 Klasifikasi menggunakan *Support Vector Machine*



Gambar 4. 7 Diagram Alir Klasifikasi

Setelah data laporan masyarakat terkelompokkan maka tahap selanjutnya yaitu melakukan proses klasifikasi. Sebelum memasuki metode klasifikasi sendiri, proses pembobotan *term* akan dilakukan kembali, agar sesuai dengan *input* yang diinginkan oleh *support vector machine*. Berikut adalah contoh bentuk data *input* SVM :



Gambar 4. 8 Contoh Data *Input* SVM

Kemudian setelah didapatkan bentuk data inputan yang sesuai, data akan dipecah menjadi 80 % untuk *training* dan 20 % untuk *testing*. Pembagian ini dilakukan secara acak. Dengan adanya *training* akan didapat model klasifikasi. Model tersebut akan dilakukan pengujian melalui proses *testing*.

4.3 Implementasi

Penelitian ini menggunakan bahasa pemrograman JAVA pada platform Development Kit 7 dan Netbeans IDE 8.1 dengan *library* tambahan adalah JAMA dan LIBSVM. *Software* basis data yang digunakan adalah MySQL.

4.3.1 Implementasi Pre-proses Teks

Pre-proses teks merupakan proses awal dari penelitian ini. Data yang sebelumnya berisi angka, *link*, huruf yang tidak seragam kapital dan tidaknya serta kata-kata yang tidak penting akan diubah menjadi data yang hanya terdiri dari huruf bukan kapital dan diambil kata-kata penting saja.

4.3.1.1 Implementasi *Case Folding* dan *Tokenizing*

Proses penyeragaman huruf menjadi huruf bukan kapital, penghapusan *link* dan angka dilakukan pada proses *case folding*. Setelah proses *case folding*, dilakukan pemotongan kalimat menjadi potongan-potongan kata yang disebut proses *tokenizing*. Berikut merupakan implementasi dari proses *case folding* dan *tokenizing* :

```
private String prosescasefolding (String teks){

String URL =
"((www\\.[\\s]+)|(https?://[^\\s]+))";

String LAIN = "@([^\\s]+)";

        teks = teks.toLowerCase();

        teks = teks.replaceAll(URL,"");

        teks = teks.replaceAll(LAIN,"");

        teks = teks.replaceAll("[^a-z]"," ");

        return teks; }
```

```
String Dok = dokumen.getteks();
        String hasil = prosescasefolding(Dok);
        String [] words = hasil.split(" ");
        for (String word : words) {
        wordsList.add(word);
```

4.3.1.2 Implementasi *Filtering*

Setelah didapat potongan-potongan kata kemudian dicek apakah kata yang sudah ada termasuk kata penting atau bukan. Jika kata tersebut terdapat dalam *list stopwords* maka kata tersebut akan dihapus. Sedangkan untuk kata yang merupakan kata penting akan disimpan kedalam *database*. Berikut adalah implementasi dari *filtering* :

```
for (int j = 0; j < daftarstop.length; j++){
for (int i = 0; i < wordsList.size() ; i++) {
if (daftarstop[j].contains(wordsList.get(i))) {
        wordsList.remove(daftarstop[j]);
}}}
```

4.3.2 Implementasi Pembobotan TF-IDF dan SVD

Kata-kata penting yang telah disimpan dalam database akan dilakukan pembobotan menggunakan TF-IDF. Tahap ini dimulai dengan membentuk matriks kemunculan setiap kata penting pada dokumen. Setelah frekuensi kemunculan kata penting pada tiap dokumen dihitung maka langkah selanjutnya adalah menghitung nilai *inverse document frequency* kata penting menggunakan method tfidfIndexer().

Berikut merupakan implementasi dari pembobotan TF-IDF :

```
public Matrix tfidfIndexer(Matrix matrix){

int n = matrix.getColumnDimension();

for(int j = 0; j < matrix.getColumnDimension();
j++){ for(int i = 0; i <
matrix.getRowDimension(); i++){double
matrixElement = matrix.get(i, j);
if(matrixElement > 0.0D){ double dm =
countDocsWithWord(

matrix.getMatrix(i, i, 0,
matrix.getColumnDimension() - 1));

matrix.set(i, j, matrix.get(i, j) * (1 +
Math.log10(n) - Math.log10(dm)));

}}} return matrix;}
```

Tahap berikutnya adalah menerapkan metode *Singular Value Decomposition*. Tahap ini dimulai dengan mendekomposisi matriks kata penting (U), matriks sigma (S), serta matriks dokumen(V). Seluruh matriks tersebut kemudian di potong pada indeks kolom ke k dengan nilai k merupakan nilai akar dari jumlah kolom matriks awal. Matriks U,S,V yang telah direduksi dipotong kemudian dikembalikan ke bentuk matriks semula dengan mengalikan matriks U,S, dan V^T . Berikut merupakan implementasi dari SVD :

```
public Matrix lsiIndexer(Matrix matrix){

//tahap 1: SVD

            SingularValueDecomposition svd = new
SingularValueDecomposition(matrix);

            Matrix wordVector = svd.getU();

            Matrix sigma = svd.getS();

            Matrix documentVector = svd.getV();
```

```
//Menghitung nilai k(ie where to truncate)

            int k = (int)
Math.ceil(Math.sqrt(matrix.getColumnDimension()));

            Matrix reducedWordVector =
wordVector.getMatrix( 0,
wordVector.getRowDimension() - 1, 0, k - 1);

            Matrix reducedSigma =
sigma.getMatrix(0, k - 1, 0, k - 1);

    Matrix reducedDocumentVector =
documentVector.getMatrix(0,
documentVector.getRowDimension() - 1, 0, k - 1);

            Matrix weights =
reducedWordVector.times(reducedSigma).times(reduce
dDocumentVector.transpose());

            return weights;         }
```

4.3.3 Implementasi Pengelompokan Data menggunakan *K-Means Clustering*

Proses pengelompokan dilakukan pada kelas *Clustering*. Kelas *Clustering* memiliki parameter *iteration* yang merupakan jumlah iterasi maksimal untuk melakukan proses *clustering*. Selain itu terdapat parameter yang diinputkan yaitu jumlah *cluster* yang diinginkan. Input proses ini berupa kumpulan *Point*. *Point* tersebut merupakan kata penting dengan vektor yang dimiliki, sedangkan outputnya berupa kumpulan *cluster* beserta kata penting di dalamnya. Berikut merupakan implementasi dari pengelompokan data menggunakan *K-Means Clustering* :

```
public Clusters[] getClusters(ArrayList<Point> data,
int clusterMetode) {

    ArrayList<Point> centers = new ArrayList<>();

    double prevconvergen = Double.MAX_VALUE;

    centers = getRandCentres(data);
```

```
          Clusters[] clusters = reallocation(centers,
data);

    ArrayList<Point> prevCenter = centers;

for (int i = 0; i < iterations; i++) {

        double convergence = 0.0;

        centers = getCenters(clusters);

        clusters = reallocation(centers, data);

        convergence = countThreshold(prevCenter,
centers);

System.out.println(convergence);

        if(convergence <= 0.001 && prevconvergen <=
0.001){

            break;

        }

        prevconvergen = convergence;

        prevCenter = centers;

    }

  return clusters;

 }
```

Pada kode di atas terdapat beberapa *method*, yaitu :

1. getRandCentres()
   Method ini berfungsi untuk memilih awal pusat *cluster* secara acak.
2. getCenters()
   Method ini berfungsi untuk memperbaharui pusat *cluster* selama iterasi

3. reallocation()
   Method ini berfungsi untuk mengelompokkan ulang konsep ke dalam masing-masing *cluster*.

Tahap pertama pada proses pengelompokan kata adalah menentukan pusat awal. Pusat awal dipilih dari kata penting yang ada secara acak. Berikut *source code* untuk *method* getRandCentres :

```
public ArrayList<Point> getRandCentres(ArrayList<Point>
data){

    Random random = new Random();

    ArrayList<Point> centers=new ArrayList<>();

    ArrayList<Integer> randomedNum = new ArrayList<>();

    for(int i=0;i<numOfClusters;i++){

        Integer rand = random.nextInt(data.size());

        if(i == 0){

            centers.add(data.get(rand));

            randomedNum.add(rand);

        }else if(!randomedNum.contains(rand)){

            centers.add(data.get(rand));

            randomedNum.add(rand);

        } else{ i = i - 1;}

    } return centers; }
```

Setelah pusat awal diperoleh maka, langkah selanjutnya adalah mengelompokkan kata ke dalam pusat awal. Pengelompokan dilakukan dengan cara menghitung jarak tiap kata ke masing-masing pusat cluster, kemudian kata dimasukkan pada *cluster* dengan jarak pusat *cluster* terdekat.

Pengelompokan ini dilakukan menggunakan method reallocation(). Berikut *source code* untuk *method* reallocation:

```
public Clusters[] reallocation(ArrayList<Point> centers,
ArrayList<Point> data) {

       Clusters[] clus = new Clusters[numOfClusters];

       for (int i = 0; i < numOfClusters; i++){

            clus[i] = new Clusters();

       }

       for (int i = 0; i < data.size(); i++) {

            Double dis = Double.MAX_VALUE;

            int ind = 0;

            Double temp = 0.0;

for (int j = 0; j < centers.size(); j++) {

                temp = distance(data.get(i),
centers.get(j));

                //System.out.println("Distance data ke-" +
i + " terhadap pusat ke-" + j + " = " + temp);

                if (temp < dis) {

                   dis = temp;

                   ind = j;}   }

            //System.out.println("Data masuk pusat ke-" +
ind);clus[ind].clus.add(data.get(i));}return clus;

      }
```

Pada method ini terdapat *method* distance() yang berfungsi untuk menghitung jarak tiap kata ke pusat *cluster* menggunakan *euclidean distance*.

```
public Double distance(Point x, Point y) {

    Double distance = 0.0;

    for(int i = 0; i < x.getVektor().size(); i++){

        distance += Math.pow(x.getVektor().get(i) -
y.getVektor().get(i), 2);

    }

    distance = Math.sqrt(distance);

    return distance;

  }
```

Tahap berikutnya adalah melakukan perulangan untuk memperbaharui anggota cluster menggunakan method reallocation() seperti yang telah diimplementasikan sebelumnya, sedangkan untuk memperbaharui pusat cluster digunakan method getCenters(). Proses untuk memperbaharui pusat cluster dilakukan. Berikut *source code* untuk *method* getCenters:

```
public ArrayList<Point> getCenters(Clusters[] arr) {

       ArrayList<Point> centers = new ArrayList<>();

       Double vek;

       for (int i = 0; i < numOfClusters; i++) {

            ArrayList<Double> vektor = new
ArrayList<>(Collections.nCopies(arr[0].clus.get(0).getVekto
r().size(), 0.0));
```

```
            for (int j = 0; j < vektor.size(); j++) {

                vek = 0.0;

                if(arr[i].clus.size() != 0){

                    for(int k = 0; k < arr[i].clus.size();
k++){vek += arr[i].clus.get(k).getVektor().get(j);}}
vektor.set(j, vek); } if (arr[i].clus.size() > 0) {for(int a
= 0; a < vektor.size(); a++){

                            vektor.set(a,
vektor.get(a)/arr[i].clus.size());

                }centers.add(new Point("",vektor));

            } }

       return centers;

      }
```

4.3.4 Implementasi proses klasifikasi menggunakan *Support Vector Machine*

Berikut merupakan implementasi dari proses klasifikasi. Tahap pertama, data hasil pembobotan TF-IDF kedua akan diubah kedalam bentuk data masukkan SVM. Setelah itu akan melalui proses *training*. Berikut merupakan *source code* proses *training*:

```
svm_problem prob=new svm_problem();
        int
numTrainingInstances=featuresTraining.keySet().size();
        prob.l=numTrainingInstances;
        prob.y=new double[prob.l];
        prob.x=new svm_node[prob.l][];

        for(int i=0;i<numTrainingInstances;i++){
               HashMap<Integer,Double>
       tmp=featuresTraining.get(i);
            prob.x[i]=new svm_node[tmp.keySet().size()];
            int indx=0;
```

```
for(Integer id:tmp.keySet()){
                svm_node node=new svm_node();
                node.index=id;
                node.value=tmp.get(id);
                prob.x[i][indx]=node;
                indx++;
            }

            prob.y[i]=labelTraining.get(i);
        }

        svm_model model=svm.svm_train(prob,param);
        String name = "model.txt";
        try {
            svm.svm_save_model(name, model);
        } catch (IOException ex) {
        }
```

Keluaran dari proses *training* adalah model klasifikasi. Selanjutnya akan dilakukan proses *testing* untuk menguji model klasifikasi tersebut. Berikut merupakan *source code* proses *testing* :

```
for(Integer testInstance:featuresTesting.keySet()){
            //int act = simpan[0];
            HashMap<Integer, Double>
tmp=featuresTesting.get(testInstance);
            int numFeatures=tmp.keySet().size();
            svm_node[] x=new svm_node[numFeatures];

            int featureIndx=0;
            for(Integer feature:tmp.keySet()){
                x[featureIndx]=new svm_node();
                x[featureIndx].index=feature;
                x[featureIndx].value=tmp.get(feature);
                featureIndx++;

            }

            double d=svm.svm_predict(model, x);
```

4.3.5 Implementasi Metode Evaluasi

Pada tugas akhir ini penulis menggunakan evaluasi *sillhoute coefisient* untuk menentukan jumlah *cluster* terbaik dengan melakukan beberapa kali percobaan mengganti inputan k *cluster*. Berikut merupakan *source code* metode evaluasi *sillhoute coefisient* :

```
        public Double
countSilhouetteCoeff(Clusters[] cl) {

         Double SillhouetteCoeff = 0.0;

         Double SillhouetteX = 0.0;

    int jumData = 0;

         for(int i = 0; i < cl.length; i ++){

       jumData += cl[i].clus.size();}

for(int i = 0; i < cl.length; i++){

             for(int j = 0; j < cl[i].clus.size();
j++){

                  Double aX = countAX(cl[i].clus,
cl[i].clus.get(j));

                  Double bX = countBX(i,
cl[i].clus.get(j), cl);



                  SillhouetteX = (bX - aX) /
Math.max(aX, bX);

                  SillhouetteCoeff += SillhouetteX;

      }

        }

    SillhouetteCoeff = SillhouetteCoeff/jumData;

         return SillhouetteCoeff;

    }
```

Selanjutnya untuk menguji model klasifikasi menggunakan rumus akurasi. Berikut merupakan implementasi dari rumus *accuration* :

```
double accuration = correct*100;

        double result = accuration/data_testing;
```

4.4 Implementasi Antarmuka GUI

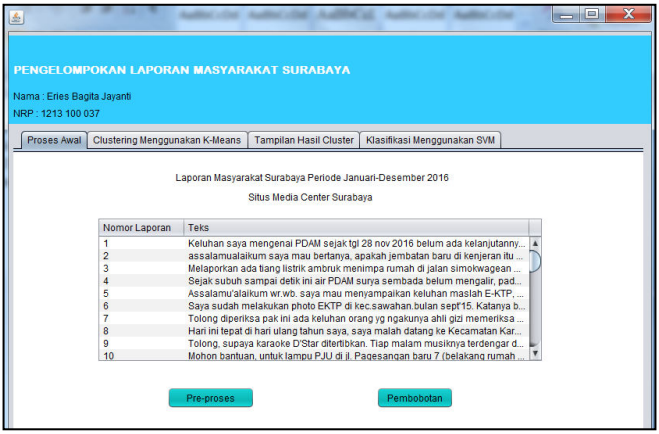
Antarmuka program ini dibagi menjadi 4 tab, yaitu tab "**Proses Awal**", "**Clustering menggunakan K-Means**", "**Tampilkan hasil cluster**" dan "**Klasifikasi menggunakan SVM**". Pada bagian **proses awal** penulis menampilkan laporan masyarakat Surabaya pada periode Januari-Desember 2016.

Langkah awal adalah melakukan pre-proses teks dengan menekan tombol Pre-proses. Selanjutnya adalah melakukan pembobotan dengan menekan tombol Pembobotan . Pada proses ini akan dikerjakan pembobotan TF-IDF dan SVD.

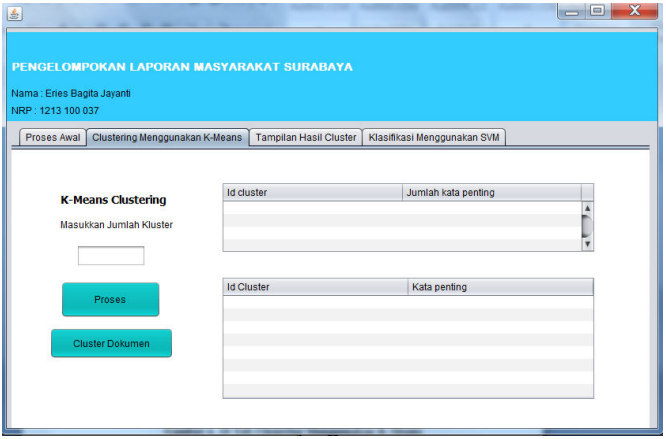
Pada tab selanjutnya yaitu **Clustering menggunakan K-Means**. Jumlah K ***cluster*** akan diinputkan pada tab ini melalui tombol Proses. Setelah itu akan muncul data pada tabel id_cluster dan kata penting, serta jumlah kata penting. Selanjutnya dilakukan proses membentuk *cluster* vektor melalui tombol *Cluster* Dokumen. Setelah proses ini juga akan didapatkan dokumen-dokumen yang terkelompokkan berdasarkan *cluster*nya.

Pada tab **tampilkan hasil *cluster*** terdapat tombol tampilkan hasil *cluster*. Penulis akan menampilkan dokumen hasil pre-proses beserta *cluster* pada tabel di tab ini.

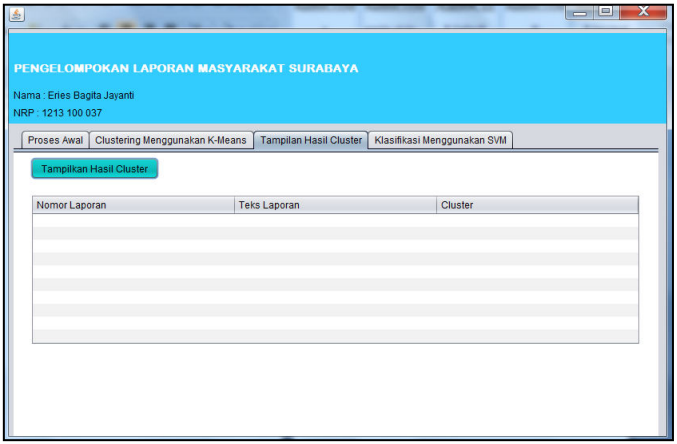
Tab terakhir adalah **Klasifikasi menggunakan SVM**. Pada bagian ini akan dilakukan proses pengubahan bentuk inputan ke inputan SVM serta proses t*raining* dan *testing*. *Training* akan menghasilkan model, yang kemudian akan diuji menggunakan data *testing*.



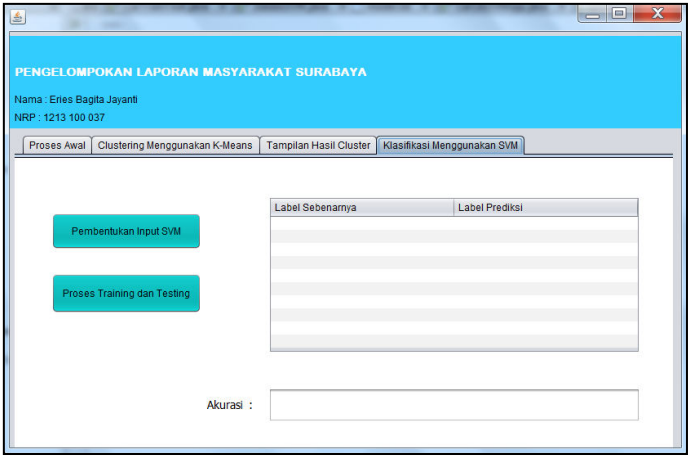
Gambar 4. 9 Tab Proses Awal



Gambar 4. 10 Tab *Clustering* Menggunakan K-Means



Gambar 4. 11 Tab Tampilkan Hasil *Cluster*



Gambar 4. 12 Tab Klasifikasi menggunakan SVM

BAB V
HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan tentang hasil dan pembahasan dari program yang telah dibuat.

5.1 Hasil Pre-proses Data

Pada penelitian ini data yang digunakan adalah data laporan masyarakat Surabaya dengan rentang waktu Januari-Desember 2016. Data tersebut berasal dari *website* media center Surabaya dan berjumlah 1948 laporan. Berikut merupakan contoh beberapa data laporan :

Tabel 5.1 Contoh data awal

No. Laporan	Teks Laporan
1	Keluhan saya mengenai PDAM sejak tgl 28 nov 2016 belum ada kelanjutannya. Dan di hari terakhir thn 2016 air tidak keluar sama sekali sejak pagi sampai saya menulis keluhan ini. PDAM hanya menanyakan data2 saya saja tapi tidak ada kelanjutannya. Jadi apa gunanya data2 yg saya berikan kalau air tetap tidak lancar. Kemana saya harus melapor lagi krn melalui media center pun keluhan saya seakan2 sia2 saja.
2	assalamualaikum saya mau bertanya, apakah jembatan baru di kenjeran itu belum boleh dilewati kendaraan bermotor?? kok setiap saya kesana pasti ditutup dan disuruh parkir oleh warga sekitar dan dikenakan biaya parkir sebesar 5rb per sepeda motor mohon - penjelasan terimakasih

No. Laporan	Teks Laporan
3	Melaporkan ada tiang listrik ambruk menimpa rumah di jalan simokwagean Surabaya. Kejadian tgl 30 desember 2016. pukul 20.45 Wib. Sudah lapor PLN 123 sdh dibetulkan tapi tiang posisi masih miring, dan rumah belum dibetulkan. belum ada tanggapan dari petugas. No pengaduan PLN U0OT77Z. Mohon bantuan pihak berwenang bisa segera ditindaklanjuti. Karena rawan roboh kembali.
4	Sejak subuh sampai detik ini air PDAM surya sembada belum mengalir, padahal satu2nya sumber air di komplek Perumahan Otoritas Bandara Wilayah III, Jemur Andayani I No.74. Mohon segera diperbaiki, terima kasih
5	Assalamu'alaikum wr.wb. saya mau menyampaikan keluhan maslah E-KTP, saya sudah bolak balik ke kelurahan dan Dispenduk di siola tp akhirnya yg saya dapatkan harus menunggu lagi 2 bulan karena data double dan harus dihapus dulu, yg saya permasalahkan kenapa harus menunggu 2 bulan hanya untuk menghapus data, saya sangat kecewa dengan pelayanan ini, sejak dari 2 tahun lalu selalu harus bolak balik dispenduk-kelurahan dan selalu hasilnya mengecewakan, harus kembalikembali dan kembali lagi
…	…

Data tersebut selanjutnya dilakukan pre-proses teks yang terdiri dari *casefolding, tokenizing* dan *filtering* . Berikut merupakan contoh beberapa data laporan masyarakat setelah melalui pre-proses teks :

Tabel 5.2 Contoh Data Hasil Pre-proses

Id Laporan	Teks Laporan
1	keluhan pdam air keluar menulis keluhan pdam data gunanya data air lancar melapor media keluhan
2	jembatan kenjeran dilewati kendaraan bermotor ditutup parkir warga dikenakan biaya parkir sepeda motor penjelasan
3	melaporkan tiang listrik ambruk rumah jalan kejadian lapor pln tiang posisi miring rumah petugas pengaduan pln u z bantuan berwenang ditindaklanjuti rawan roboh
4	detik air pdam mengalir sumber air komplek perumahan bandara wilayah jemur i diperbaiki
5	keluhan e ktp kelurahan dispenduk menunggu data dihapus menunggu menghapus data kecewa pelayanan dispenduk kelurahan mengecewakan
…	…

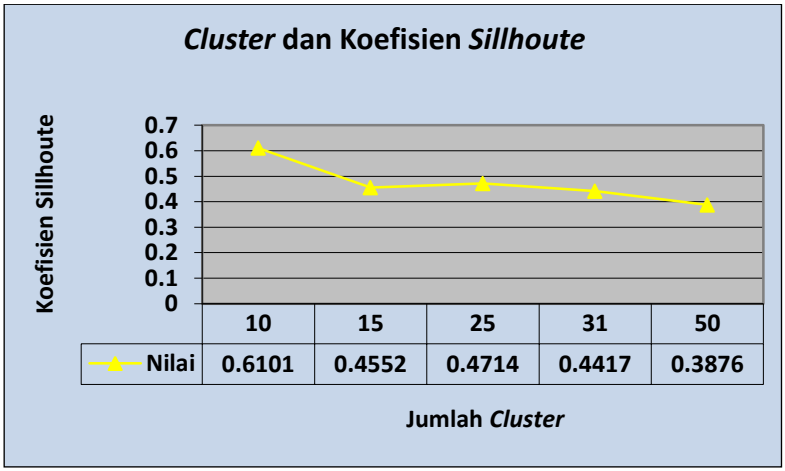
5.2 Pengelompokan Menggunakan *K-Means Clustering*

Pada proses ini dilakukan uji pengelompokan kata penting. Uji ini memiliki tujuan untuk mengelompokkan kata penting yang memiliki kemiripan ke dalam beberapa *cluster*. Performa dan kemampuan pengelompokan dilakukan dengan menghitung nilai koefisien *sillhoute*. Penulis melakukan uji coba dengan lima buah *cluster* yaitu **10, 15, 25, 31** dan **50**. Uji coba ini digunakan untuk mengetahui parameter jumlah *cluster* terbaik dalam melakukan pengelompokan. Berikut merupakan tabel jumlah *cluster* dan nilai koefisien sillhoutenya :

Tabel 5.3 Jumlah *Cluster* dan Koefisien *Sillhoute*

Jumlah *Cluster*	Nilai Koefisien *Sillhoute*
10	0.6101
15	0.4552
25	0.4714
31	0.4417
50	0.3876

Representasi dari jumlah *cluster* dan koefisien *sillhoute* dapat ditampilkan seperti berikut ini :



Gambar 5. 1 Grafik *Cluster* dan Koefisien *Sillhoute*

Pada grafik tersebut saat k = 10, nilai koefisien *sillhoute* sebesar 0.6101. Nilai tersebut lebih tinggi dibandingkan dengan nilai koefisien *sillhoute* pada saat k = 15,25,31 atau 50 yang masing-masing koefisien sillhoutenya bernilai 0.4552, 0.4714, 0.4417, 0.3876. Sehingga berdasarkan hasil tersebut penulis menetapkan jumlah *cluster* sebanyak 10 untuk membagi kelompok data, yang selanjutnya dianalisa bahasan dari masing-masing *cluster*. Pembagian 10 kelompok tersebut nantinya juga akan berlanjut ke dalam proses klasifikasi menggunakan SVM.

Setelah kata penting terkelompokkan pada *cluster* yang sesuai, yaitu 10-*cluster* selanjutnya dibentuk vektor *cluster* dari dokumen laporan. Sehingga tiap *cluster* memiliki anggota dokumen laporan. Berikut merupakan jumlah anggota laporan dari masing-masing *cluster* pada pembagian 10 kelompok :

Tabel 5.4 Jumlah laporan tiap *cluster*

Cluster ke-	Jumlah Laporan
0	86
1	32
2	2
3	1550
4	17
5	215
6	9
7	6
8	1
9	30

Tabel tersebut menunjukkan bahwa *cluster* ke-3 memiliki anggota laporan yang paling banyak yaitu sebesar 1550 dan *cluster* ke-8 memiliki anggota laporan yang paling sedikit yaitu hanya 1 laporan saja. Berdasarkan hasil tersebut, didapat bahwa persebaran anggota laporan di tiap *cluster* pada penelitian ini masih belum merata. Selanjutnya dari masing-

masing *cluster* tersebut akan dianalisa apa saja yang dibahas berdasarkan kesamaan kata antar dokumen laporan dalam satu *cluster*.

5.3 Analisa Hasil *Cluster*

Dokumen laporan yang sudah terkelompokkan ke dalam 10-*cluster* akan dianalisa kesamaan bahasan antara satu dokumen dengan dokumen yang lain di tiap clusternya. Hal ini juga didasarkan pada kata penting apa saja anggota *cluster*. Sehingga nantinya akan didapatkan bahasan dari masing-masing kelompok tersebut. Pada laporan tugas akhir ini penulis akan menampilkan contoh hasil analisa dari *cluster* ke-1, *cluster* ke-4, dan *cluster* ke-5 dari 10-*cluster*.

a. *Cluster* ke-1

Laporan yang merupakan anggota dari *cluster* ke-1 berjumlah 32 laporan. Namun pada analisa berikut akan ditampilkan 10 laporan saja sebagai contoh. Laporan-laporan tersebut adalah :

Tabel 5.5 Dokumen *Cluster* ke- 1

No. Laporan	Teks Laporan
75	Banyak **jalan** berlubang sepanjang jalan dharmahusada kearah kertajaya Dan sebaliknya disitu menuju arah kantor asked dharmahusada.. Lubangnya cukup besar2 krn di daerah test kalo hujan sering tergenang air terimakasih
111	Tolong **jalan** tembaan jalan banyak yg bergelombang, Tolong respon untuk diperbaiki. Terima Kasih

163	mohon dengan sangat **jalan jalan** yg rusak …terkait di perhatikan keselamatan **warga**
No. Laporan	Teks Laporan
171	Alhamdulillah penerangana **jalan** umum di botoputih gang 2 sudah bisa menyala kembali...terima kasih team penerangan jalan surabaya..terima kasih pemerintah kota surabaya..
208	sipp... e-wadul bermanfaat buat **warga** sby
359	selamat malam , saya warga surabaya yang beralamat di **jalan** wonoclo pabrik kulit jalan masuk sebelah JX Internasional yang sudah hampir 1 bulan **jalan** utama kami ditutup karena ada proyek perbaikan jalan dari pemerintah , saya berharap jalan tersebut segera di selesaikan karena sangat mengganggu aktifitas warga, seharusnya apabila memang ada perbaikan jalan pemerintah harus merencanakannya dengan baik agar lebih cepat lagi prosesnya apalagi tempat tersebut merupakan jalan utama
423	yth bu risma sy **warga** kramat ?g 1 kalau hujan d gang saya air nya sering tergenang.karena d gang saya belum ada selokonnya..trimah kasih atas perhatiannya..
610	Selamat pagi, Bu Risma... Saya adalah salah satu pengguna **jalan** di jl. A yani bu... Saya melihat di sebelah spm. Giant ada pembangunan yang masih berlangsung,. Saya sering melewati **jalan** didepannya... Saya sangat penyayangkan ada crane derek

	yang diposisikan didekat jalan... Dan sebagian dari crane itu... Yang ada …
No. Laporan	Teks Laporan
888	Alhamdulillah **jalan** bulaksari dekat smp Muhammadiyah 16 mulai dilaksanakan perbaikan, terima kasih pemkot Surabaya, semoga Surabaya semakin makmur sejahterah
919	mohon di perbaiki **jalan** bulak sari dekat smp muhammadiyah, terimakasih

Pada *cluster* ke-1 ini dari 10 laporan yang ditampilkan, 8 laporan memiliki kesamaan yaitu terdapat kata-kata "jalan" dan 2 laporan lainnya memiliki kata "warga". Oleh karena itu, berdasarkan kesamaan kata yang dimiliki dapat diperoleh informasi bahwa *cluster* ke-1 ini bahasannya berkaitan dengan jalan dan warga. "Jalan" yang dimaksud berkaitan dengan perbaikan, penerangan dan hal-hal gangguan di jalan tersebut. Gangguan yang dimaksud adalah seperti pada laporan dengan nomor 610 yang membahas tentang bahayanya *crane* untuk pengguna jalan. Sedangkan laporan yang memiliki kata "warga" membahas tentang keselamatan, kebermanfaatan aplikasi e-wadul untuk warga serta kenyamanan warga.

b. *Cluster* ke-4

Laporan yang merupakan anggota dari *cluster* ke-4 berjumlah 17 laporan. Namun pada analisa berikut akan ditampilkan 8 laporan saja sebagai contoh. Laporan-laporan tersebut adalah :

Tabel 5.6 Dokumen *Cluster* ke-4

No. Laporan	Teks Laporan
21	slamat siang ibu walikota mohon bantuannya untuk pemangkasan pohon di wilayah kami di jln donorejo selatan... kec. simokerto kel. kapasan
148	Dear bu.Risma,saya mengeluhkan tentang susahnya **pengurusan** tentang **sertifikat tanah**..pada suatu hari keluarga dari bapak mertua saya ingin mengurus sertifikat tanah.pada hari pertama keluarga dari bapak mertua ke BPN,setelah selesai dari BPN.pihak BPN menganjurkan ke kelurahan setempat utk minta surat keterangan..sesampai di kelurahan bukan yang didapat solusi tetapi selalu berbelit2...sampai sekarang keluarga dari bapak mertua saya tidak dapat mengurus sertifikat tanah karena terkendala dari pihak kelurahan...Mohon bantuan dan Solusinya serta Tindak Lanjutnya BU.RISMA..Tx
239	**Air PDAM** didaerah balong sari dan sekitar sdh sehari semalam mati. Tolong SEGERA dilakukan perbaikan… Terima Kasih.

No. Laporan	Teks Laporan
242	Sudah 1 bulan **air pdam** di jl jemur gayungan I (sebelah kantor bulog jl a.yani) tidak lancar. Air hanya keluar jam 19.00 sampai jam 05.00 pagi. Di luar jam tsb air mati atau keluar keci sekali. Mohon bantuannya agar ditindak lanjuti. Saya sudah lapor ke call center tapi tidak ada tindakan. Trm ks.
343	selamat pagi,mau tanyak kenapa **air PDAM** di daerah tambak Gringsing Baru kel.perak timur kec.pabean cantian tiap pagi hari tidak keluar,kendala ini yg saya liat sekitar 3 hari ini....terima kasih
642	**Air PDAM** tidak keluar pada tanggal 18 & 20 dini hari. Pada tanggal 18 jam 01:30 saya coba sedot **air PDAM** dengan pompa **air** selama 15 menit tapi tidak ada **air** keluar, jam 03:00 saya coba lagi 15 menit air juga tidak keluar. Tanggal 19 jam 03:00 air bisa keluar, tentu saja dengan bantuan pompa air. Pagi ini, tanggal 20 jam 03:00 saya sedot dengan pompa air selama 15 menit, air tidak keluar. Mohon bantuannya pihak **PDAM** Surabaya, dengan pompa air saja air tidak keluar, apalagi tanpa pompa air. Maturnuwun.
765	**air PDAM** di griya kebraon utara mati sejak hari senin tgl 22 agt, mohon tindak lanjutnya
777	Tolong pak informasikan ke pihak yg terkait untuk kondisi **air pdam** yg sejak senin tidak mengalir di wilayah griya kebraon karena ..

Pada *cluster* ke-4 ini terdapat kesamaan kata seperti “pengurusan”, “sertifikat”, “air” dan “pdam”. Pengurusan sertifikat terdapat pada laporan nomor 148. Sedangkan terkait air pdam berada pada semua laporan kecuali pada laporan nomor 21.

c. *Cluster* ke-5

Laporan yang merupakan anggota dari *cluster* ke-5 berjumlah 215 laporan. Namun pada analisa berikut akan ditampilkan 20 laporan saja sebagai contoh. Laporan-laporan tersebut adalah :

Tabel 5.7 Dokumen *Cluster* ke-5

No. Laporan	Teks Laporan
10	Mohon bantuan, untuk **lampu PJU** di jl. Pagesangan baru 7 (belakang rumah makan kauman dan biro travel adzikra) banyak yg mati. Sehingga jika malam tiba sangat gelap
18	tolong **lampu penerangan** jalan di area jalan ngagel mulyo **Padam** audah 2 minggu...
24	Kami mohon terhadap dinas PU,untuk memperbaikki jalanan ploso timur hingga pertigaan kalijudan,jalanan disitu sudah banyak yang **bergelombang** dan **berlubang**,jika kalau **hujan** gorong2 yang berada persis dipinggir jalan sering banjir dan menggenangi jalanan,untuk itu mohon untuk jalanannya **diperbaiki**,kalau bisa ditinggikan.TERIMAH KASIH

No. Laporan	Teks Laporan
32	Jl. Kenjeran lapisan **aspal** bagian atas sudah banyak yang terkelupas, hal ini mengganggu dan membahayakan pengguna jalan terutama roda dua. Mohon tidak lanjut dinas terkait. terima kasih.
67	Sepanjang jalan Jagir arah TL Panjang Jiwo jalannya mulai banyak yang **berlubang**, ini sangat membahayakan terutama kendaraan R2, apalagi pada waktu malam hari sangat minim penerangan..
78	2 **lampu PJU mati** pada ngagel tama tengah gang 3, tolong dicek. Terima kasih
95	Selamat Pagi Ibu Risma. Tgl 26-11-2016 kami telah melaporkan jalan **rusak** di di depan perumahan ASPOL bangkingan.Mohon segera dapat di perbaiki . mengingat keselamatan pengendara motor beresiko jatuh.Karena setiap **hujan** selalu **banjir** bagaikan lautan. Tapi statusnya SEDANG DI TINDAK LANJUTI. Kenapa TIDAK SEGERA DI PERBAIKI ?Mohon secepatnya diperbaiki. Matur nwun Ibu Risma
102	saya warga jl. lebo agung 2, ingin menginformasikan kalau **lampu.penerangan** jalan di gang saya depan rumah nomer 55 dan dan 65 padam, dibox tiang pju tertulis pemkot surabaya, terimakasih.

No. Laporan	Teks Laporan
117	Selamat malam. Di gubeng Kertajaya 9c **lampu Penerangan** jalan mati sudah 3 hari. Mohon ditindak lanjuti. Tks
126	jalan umum lebak arum tengah **berlubang** cukup dalam.sering terjadi kecelakaan terutama pengendara roda dua.mohon segera di followup dan ditindak lanjuti
127	dari arah perempatan Kapasan ke jalan Kenjeran sampai menuju ke arah pertigaan Rangkah. ada banyak sekali **lubang**. bahkan bisa saya bilang jalan tersebut Hancur berantakan, sampai-sampai warga sekitar memberikan tanaman pada jalan yang **rusak** tersebut agar tidak ada korban jiwa. Tolong segera **diperbaiki**, mohon maaf sebelumnya
128	Di sepanjang Jalan Kedung Baruk dari pertigaan Kalirungkut sampai menuju ke arah Jembatan MERR ada banyak sekali **lubang**. saya hitung semua total ada 14 Lubang kecil dan besar. Tolong Segera diperbaiki ya. Terima Kasih
130	mohon perhatiannya untuk kondiai jalan di jalan kenjeran di jalur dekat makam rangkah cukup parah,lubang jlan sampai ditutup menggunakan kayu sebagai tanda buat pengendara motor dan mobil agar menghindari jalan **berlubang** tersebut. cukup membahayakan pengguna kendaraan bermotor di malam hari utamanya jika **hujan**. mohon perhatiannya terima kasih

No. Laporan	Teks Laporan
142	Tolong **diperbaiki** jalan lakarsantri yang arah driyorejo banyak **berlubang** membahayakan pengguna jalan. Terima kasih
160	Mohon **perbaikan penerangan** jalan umum di daerah kalijudan gang 8 banyak yg **padam**. sebab tempatnya rawan..mohon perhatian pemkot
165	mohon ditindak lanjut area banyu urip **banjir** tinggi padahal **hujan** sebentar.
170	Mohon **perbaikan** jalan dari warugunung sampai karangpilang banyak jalan **berlubang**, membahayakan bagi pengendara, terutama roda dua, trima kasih
204	Tolong **diperbaiki** jalan darmo depan rs darmo (arah basuki rahmat) **berlubang**. terima kasih
205	Selamat Pagi Ibu Risma. Mohon dapat **di perbaiki** jalan Aspal yang rusak di depan perumahan ASPOL bangkingan.Mohon segera dapat di perbaiki . mengingat keselamatan pengendara motor beresiko jatuh.Karena setiap **hujan** selalu **banjir** bagaikan lautan. Matur nwun Ibu Risma
209	Mohon segera ditindak lanjuti di jln .karangan gg 4 wiyung ada dua kubangan di jln raya yg sering mengakibatkan pengendara sepeda motor jatuh.....mohon perhatiaannya dan kurangnya lampu…

Pada *cluster* ini berdasarkan laporan-laporan yang telah menjadi anggota *cluster* ke-5, membahas tentang penerangan jalan umum dan perbaikan lubang.

Analisa tiga *cluster* tersebut merupakan contoh dari analisa hasil *cluster* pada penelitian ini. Penentuan bahasan dari masing-masing *cluster* selain dianalisis secara langsung dari tiap-tiap anggota dokumen tiap clusternya, juga dicocokan dengan kata-kata penting yang sebelumnya sudah terkelompokkan pada *cluster* tersebut. Sehingga berdasarkan hal tersebut bahasan per *cluster* dapat ditampilkan seperti yang tertera pada tabel di bawah ini :

Tabel 5.8 Bahasan tiap *Cluster*

Cluster	Bahasan
0	KTP, kelurahan, kecamatan, KK, pengurusan, dispenduk, data,
1	Jalan, warga,
2	Anak, dhuafa, program, kemiskinan, sosial, hutang, investasi, sedekah,
3	Bandara, magang, upah, KUA, kriminal, hama, drainase, perindustrian, ekonomi, musim, kesehatan, pendidikan, kebersihan, polusi, ketenagakerjaan, cuaca, telekomunikasi,
4	PBB, tanah, lahan, bangunan, air, PDAM, sertifikat,
5	proyek, pju, rusak, padam, penerangan, lubang, aspal, lampu,

Cluster	Bahasan
6	Pertamanan, penghijauan, taman, pohon, keindahan,
7	Warung, PKL, pedagang, keamanan,
8	kota
9	RT, RW, parkir, rumah, sampah.,

5.4 Klasifikasi Menggunakan SVM

Klasifikasi dilakukan untuk membuat suatu model dari data hasil pengelompokan. Data laporan yang berjumlah 1948 tersebut dipisahkan menjadi data *training* dan data *testing*, yaitu sebanyak 1568 untuk *training* dan 380 untuk *testing*. Proses klasifikasi tugas akhir ini menggunakan LIBSVM dengan fungsi kernel *Gaussian Radial Basis Function*(RBF). RBF merupakan fungsi kernel yang popular karena sering digunakan dalam klasifikasi menggunakan SVM[17]. Berikut akan ditunjukkan dari model yang terbentuk :

```
svm_type c_svc
kernel_type rbf
gamma 0.5
nr_class 10
total_sv 1525
```

rho -0.9393759450808415 -0.850825737939809 -0.9774580400074893 -0.9899722756808866 -0.9824835602404814 -0.9966569850830648 -0.9933195931103351 -0.9983291889798052 -0.9990314617027483 0.592603465294865 -0.6226995590508014 -0.8298757555447002 -0.7023154484421339 -0.9432874917081288 -0.8867860258626615 -0.9716445869966045 -0.9860744607244869 -0.8483463338682162 -0.9319721823971866 -0.8809126714351264 -0.977314594686455 -0.9547560334172621 -0.98866002734346 -0.9944853048840127 -0.5396151121422963 -0.19203938092344103 -0.84610133526032 -0.6923491757720208 -0.9230521887079088 -0.9617048863981034 0.4214081335492931 -0.6664189355508948 -0.33289469376033204 -0.8332103679800905 -0.9166742469188454 -0.8071329274330188 -0.6142904728588663 -0.9035666929496378 -0.9518139845204261 0.5000000087085977 -0.5000000001822071 -0.7500000225867132 -0.7500117810150927 -0.8764604744405783 -0.5000574993900955

label 3 0 5 1 4 9 7 6 2 8

nr_sv 1198 73 179 27 12 21 4 8 2 1

SV

0.06103667751233122 0.14901540599355612 0.022460127419310083 0.009765435391059507 0.017302565289280314 0.003906249999995892 0.006831929965098848 5 9.765624637410042E-4 0.0
1360:2.927861116525003 516:4.876661207647041
868:3.6875289612146336 1142:2.4383306038235206
1494:2.6268311208610218 23:4.1763836564442895
1144:6.756487363805917 2105:4.623730694507497
1503:3.059140031164322 1855:3.444490912528339

0.06103667764452969 0.14901251270188892 ……

Gambar 5. 2 Model Klasifikasi

Svm_type menunjukkan tipe svm yang digunakan, yaitu C_SVC. Tipe ini mampu melakukan klasifikasi multi-kelas pada *dataset* dengan menerapkan pendekatan satu-lawan-satu (*one-against-one*). Kernel_type menunjukkan tipe kernel. Terdapat beberapa jenis tipe kernel yaitu linear, poli, RBF, sigmoid dan precomputed. Tipe kernel RBF adalah nilai *default*. Secara umum, RBF adalah pilihan pertama dalam menentukan tipe kernel.

Gamma merupakan parameter yang hanya tersedia untuk tipe kernel poli, RBF atau sigmoid. Nilai Gamma dapat memainkan peran penting dalam SVM model. Mengubah nilai Gamma dapat mengubah ketepatan model SVM yang dihasilkan. Nilai Gamma yang digunakan pada proses ini sebesar 0.5. Nr_Class menunjukkan jumlah kelas yang terdapat pada model. Pada model tersebut nr_class menunjukkan angka 10, yang artinya terdapat 10 kelas yang telah dilatih. Selanjutnya yaitu penjelasan tentang Total_sv.

Total_sv adalah total *support vector* yang dihasilkan dari proses *training*. *Support vector* tersebut berjumlah 1525. Sedangkan rho berarti bias. Rho pada model klasifikasi ini tertera seperti gambar diatas.

Penjelasan berikutnya yaitu terkait label. Label menunjukkan nama-nama label atau kelas. Seperti yang telah dijelaskan sebelumnya bahwa data telah terkelompokkan menjadi 10 kelompok, dengan masing-masing kelompok dinamakan berdasarkan nama clusternya, yaitu *cluster* ke-0 hingga *cluster* ke-9 . Nr_SV adalah banyaknya dari *support vector* . Sedangkan SV adalah *support vector* yang berhasil terbentuk.

Model klasifikasi yang telah terbentuk selanjutnya akan diuji menggunakan data *testing*. Hasil pengujian menunjukkan laporan yang benar diprediksi sebanyak 317 dari 380 data *testing*. Sehingga didapat hasil akurasi seperti berikut :

Tabel 5.9 Hasil akurasi

Cluster	**Data *Training***	**Data *Testing***	***Accuracy***
10	1568	380	83.42% (317/380)

Dari penilaian *accuracy* proses klasifikasi dapat diketahui jumlah dokumen yang benar terprediksi terhadap jumlah data *testing* atau data ujinya. Sehingga bisa didapatkan informasi bahwa model yang telah dibentuk sudah baik atau masih belum berdasarkan hasil akurasi tersebut. Akurasi menunjukkan bahwa saat nilai k atau *cluster* = 10 dengan data *training* sebanyak 1568 laporan dan data *testing* sebanyak 380 laporan didapat *accuracy* sebesar 83,42 %.

Akurasi juga menunjukkan bahwa hasil pelabelan data laporan masyarakat berdasarkan *cluster* menggunakan metode *K-Means Clustering* ini ketika diklasifikasi menggunakan *support vector machine* memiliki hasil yang baik berdasarkan akurasi model klasifikasi yang telah terbentuk.

BAB VI
PENUTUP

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Berdasarkan pembahasan hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Pada penelitian ini data laporan masyarakat telah dikelompokkan menggunakan metode *K-Means Clustering* sesuai kemiripan bahasan laporan. Tahapan prosesnya adalah data laporan yang ada dilakukan pre-proses data untuk mendapatkan kata penting. Dari kata penting tersebut akan dilakukan proses TF-IDF dan *Singular Value Decomposition* (SVD) untuk memberi bobot vektor pada setiap kata penting. Selanjutnya kata penting tersebut dilakukan proses pengelompokan menggunakan *K-Means Clustering*. Setelah kata penting terkelompokkan pada *cluster* yang sesuai selanjutnya akan dibentuk vektor *cluster* dari dokumen laporan. Sehingga tiap *cluster* memiliki anggota dokumen laporan. Hasil pengelompokan menggunakan *K-means Clustering*, dipengaruhi oleh nilai parameter K yang dimasukkan. Nilai K=10 menghasilkan pengelompokan dengan nilai koefisien *sillhoute* tertinggi sebesar 0,61.
2. Proses klasifikasi menggunakan metode *Support Vector Machine* (SVM) dilakukan dari hasil pengelompokan. Tahapannya adalah data laporan dipecah menjadi data *training* dan data *testing*. Dari data *training* akan dilakukan proses klasifikasi sehingga membentuk model. Model tersebut diuji menggunakan data *testing* dan menghasilkan nilai akurasi sebesar 83,42%.

6.2 Saran

Ada beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya :

1. Banyak terdapat singkatan pada data laporan sehingga sebaiknya dilakukan konversi singkatan tersebut ke kata yang sebenarnya pada pre-proses teks. Agar didapat teks yang lebih baik nantinya saat akan diolah ke tahapan selanjutnya.
2. Dalam penentuan *stopwords* lebih dipilah lagi, agar kata penting yang dihasilkan lebih baik.

DAFTAR PUSTAKA

[1] Dudung.(2015) . "30 Dampak Positif Dan Negatif Teknologi Informasi Dalam Bidang Pemerintahan".http://www.dosenpendidikan.com/30-dampak-positif-dan-negatif-teknologi-informasi-dalam-bidang-pemerintahan/ . Diakses tanggal 10 Februari 2017

[2] Pemerintah Kota Surabaya. www.surabaya.go.id . Diakses tanggal 11 Februari 2017

[3] Media Center Pemerintah Kota Surabaya (Dinas Komunikasi & Informatika Kota Surabaya).http://dinkominfo.surabaya.go.id/dki.php?hal=29. Diakses tanggal 11 Februari 2017

[4] James Edd Wilder.(2014)."*Defining Big Data"*.http://www.forbes.com-/sites/edddumbill/2014/05/07/ defining-big-data/. Diakses tanggal 11 Februari 2017

[5] Yusuf A., Priambadha T. (2013). ***Support Vector Machines*** **yang Didukung oleh** ***K-Means Clustering*** **dalam Klasifikasi Dokumen**. Surabaya : Institut Teknologi Sepuluh Nopember.

[6] Yao, Y., Liu, Y., Yu, Y., dkk (2013). "*K-SVM: An Effective SVM Algorithm Based K-Means Clustering"*. **Journal of Computers. VOL.8. NO.10.October 2013**.

[7] Megawati Chyntia. (2015). **Analisis Aspirasi dan Pengaduan di Situs LAPOR! Dengan Menggunakan** ***Text Mining***. Depok : Universitas Indonesia.

[8] Shafiabady, N., Lee, L,H ., Rajkumar, R., dkk (2016). "*Using Unsupervised Clustering Approach to Train The Support Vector Machine for Text Classification*". **Neurocomputing : www.elsevier.com/locate/neucom.**

[9] Jiawei, H., Kamber, M., & Pei, J, (2012).***Data Mining: Concepts and Techniques Third Edition***. Waltham, MA: Morgan Kaufmann.

[10] Feldman, R., & Sanger, J. (2007).***The Text Mining Handbook:Advanced Approaches in Analyzing Unstructured Data***. New York: Cambridge University Press

[11] Nugroho Eko. (2011). **Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp**. Malang:Universitas Brawijaya.

[12] Agusta Ledy. (2009). "Perbandingan algoritma Stemming porter dengan Algoritma nazief & adriani untuk stemming dokumen teks Bahasa indonesia"**Konferensi Nasional Sistem dan Informatika 2009**.

[13] Baker K. (2005). **Singular Value Decomposition Tutorial.** davetang.org.

[14] Zhai, C., & Aggarwal, C. C. (2012).***Mining Text Data***.New York: Springer

[15] Santosa, B. (2007). **Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis**. Yogyakarta:Graha Ilmu.

[16] JURNAL TEKNIK ITS Vol. 5, No. 2, (2016) ISSN: 2337-3539

[17] Murfi, Hendri. Support Vector Machine. Depok : Universitas Indonesia

LAMPIRAN

1. Koneksi Database

```
package tugasakhir;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class Database {
public Connection koneksi;
    public Database(){
    }
    public Connection getConn(){
        return koneksi;
    }
    public PreparedStatement prepState(String
query)throws SQLException, IllegalStateException{
        PreparedStatement prst =
koneksi.prepareStatement(query);
        return prst;
    }
    public void connectFirst() throws SQLException{
        String server = "localhost", db="bismillah",
user = "root", pass = "";
        getConnection(server, db, user, pass);
        if(isConnected()){
            System.out.println("Connected");
        }
        else{
            System.out.println("Not Connected");
        }
    }

    public boolean isConnected() {
        if(koneksi != null){
            return true;
        }
        return false;
    }
```

```
    public boolean getConnection(String server, String
db, String user, String pass)throws SQLException,
IllegalStateException{
        this.destroyConnection();
        String dbDriver = "com.mysql.jdbc.Driver";
        String dbUrl = "jdbc:mysql://" + server + "/" +
db;
        try{
            Class.forName(dbDriver);
            koneksi =
DriverManager.getConnection(dbUrl, user, pass);
            return true;
        }
        catch(ClassNotFoundException ex){

        }
        catch(SQLException ex){

        }
        return false;

    }

    public void destroyConnection()throws SQLException,
IllegalStateException{
        if(koneksi != null){
            koneksi.close();
            koneksi = null;
        }
    }

    public ResultSet executeSelect(String query) throws
SQLException, IllegalStateException {
        Statement stm;
        ResultSet rs = null;
        stm =
koneksi.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE
,
                ResultSet.CONCUR_READ_ONLY);
        rs = stm.executeQuery(query);

        return rs;

    }
```

```
    public Statement createStat() throws SQLException {
        return koneksi.createStatement();
    }

    public int executeUpdate(String query) throws
SQLException, IllegalStateException {
        Statement stmt;
        if (koneksi == null) {
            return 0;
        }
        stmt = koneksi.createStatement();
        return stmt.executeUpdate(query);
    }
}
```

2. Pre-proses

```
package tugasakhir;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

public class Prepos {
ArrayList<String> wordsList = new ArrayList<String>();
public void formRepDokumen(Dokumen dokumen) throws
SQLException{
//mengambil kamus untuk stopwords
String count ="";
Database db = new Database();
db.connectFirst();
try {

             ResultSet rs = db.executeSelect("SELECT
COUNT(*) FROM stoplist;");
             //int a = 0;
             while(rs.next()){
              count = rs.getString("COUNT(*)");
             }
           } catch (Exception e){
         }
int count2 = Integer.parseInt(count);
String[]daftarstop = new String[count2];
```

```
try {
              ResultSet rs = db.executeSelect("SELECT *
FROM stoplist;");
              int a = 0;
              while(rs.next()){
              daftarstop[a] = rs.getString("stopword");
                 // System.out.println(kamus[a]);
              a++;
              }
            } catch (Exception e){
}
String Dok = dokumen.getteks();
        String hasil = prosescasefolding(Dok);
        String [] words = hasil.split(" ");
        for (String word : words) {
        wordsList.add(word);
}

for (int j = 0; j < daftarstop.length; j++){
for (int i = 0; i < wordsList.size() ; i++) {
if (daftarstop[j].contains(wordsList.get(i))) {
        wordsList.remove(daftarstop[j]);
}}}
 for (String str : wordsList){
             if(str.length() > 0){
String query = "INSERT INTO representasi_dokumen
VALUES('"+ dokumen.getId() +"','" + str + "')";
                db.executeUpdate(query);
            }

        }
        wordsList.clear();

        db.destroyConnection();

}
//proses casefolding dan filtering
private String prosescasefolding (String teks){
String URL = "((www\\.[\\s]+)|(https?://[^\\s]+))";
String LAIN = "@([^\\s]+)";
        teks = teks.toLowerCase();
        teks = teks.replaceAll(URL,"");
        teks = teks.replaceAll(LAIN,"");
        teks = teks.replaceAll("[^a-z]"," ");
        return teks;
}
```

3. Pembobotan

```
package tugasakhir;

import Jama.Matrix;
import Jama.SingularValueDecomposition;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import java.util.HashMap;
import java.util.Map;
import java.util.SortedSet;
import java.util.TreeSet;
import org.apache.commons.collections15.Bag;
import org.apache.commons.collections15.bag.HashBag;


 /*
 * @author hp
 */
public class Pembobotan {

        Dokumen dokTA = new Dokumen();
        Map<Integer, String> dokTAiDMap = new
HashMap<Integer, String>();
        Map<Integer, String> termIDMap = new
HashMap<Integer, String>();
        Map<String, Bag<String>> dokKeyIDMap = new
HashMap<String, Bag<String>>();

        public Pembobotan(){

        }

 public double[][] buildMatrix(){

            int numDoc = dokTAiDMap.size();
            int numTerm = termIDMap.size();
            double[][] data = new
double[numTerm][numDoc];
```

```
            System.out.println(numDoc);
            System.out.println(numTerm);
            System.out.println(dokKeyIDMap.size());

//Membentuk matriks kemunculan tiap-tiap term pada
tiap-tiap dokumen
            for(int i = 0; i < numTerm; i++){
                for(int j = 0; j < numDoc; j++){
                    String dokName = dokTAiDMap.get(j);
                    Bag<String> dokTerm =
dokKeyIDMap.get(dokName);
                    String term = termIDMap.get(i);
                    int df = dokTerm.getCount(term);
                    data[i][j] = df;
                }
            }

            return data;
        }

//index term frekuensi
        public Matrix tfIndexer(Matrix matrix){
            for(int j = 0; j <
matrix.getColumnDimension(); j++){
                double sum = sum(matrix.getMatrix(0,
matrix.getRowDimension() - 1, j, j));
                for(int i = 0; i <
matrix.getRowDimension(); i++){
                    matrix.set(i, j, (matrix.get(i,
j)/sum));
                }
            }
            return matrix;
        }

//Mendapatkan jumlah jumlah kemunculan tiap term pada
dokumen i
//sebagai bilangan pembagi pada proses normalisasi
        private double sum(Matrix colMatrix) {
            double sum = 0.0D;
            for(int i = 0; i <
colMatrix.getRowDimension(); i++){
                sum += colMatrix.get(i, 0);
            }
            return sum;
        }
```

```
//Menghitung idf tiap term pada matrix
public Matrix idfIndexer(Matrix matrix){
            int n = matrix.getColumnDimension();
            for(int j = 0; j <
matrix.getColumnDimension(); j++){
                for(int i = 0; i <
matrix.getRowDimension(); i++){
                    double matrixElement =
matrix.get(i, j);
                    if(matrixElement > 0.0D){
                        double dm = countDocsWithWord(
                        matrix.getMatrix(i, i, 0,
matrix.getColumnDimension() - 1));
                        matrix.set(i, j, matrix.get(i,
j) * (1 + Math.log(n) - Math.log(dm)));
                    }
                }
            }

            return matrix;
        }


//Proses SVD
public Matrix lsiIndexer(Matrix matrix){
//tahap 1: SVD
            SingularValueDecomposition svd = new
SingularValueDecomposition(matrix);
            Matrix wordVector = svd.getU();
            Matrix sigma = svd.getS();
            Matrix documentVector = svd.getV();
//Menghitung nilai k(ie where to truncate)
            int k = (int)
Math.ceil(Math.sqrt(matrix.getColumnDimension()));
            Matrix reducedWordVector =
wordVector.getMatrix( 0, wordVector.getRowDimension() -
1, 0, k - 1);
            Matrix reducedSigma = sigma.getMatrix(0, k
- 1, 0, k - 1);
            Matrix reducedDocumentVector =
documentVector.getMatrix(0,
documentVector.getRowDimension() - 1, 0, k - 1);
            Matrix weights =
reducedWordVector.times(reducedSigma).times(reducedDocu
mentVector.transpose());
            return weights;
        }
```

```
//Menghitung jumlah dokumen yang mengandung term t
        private double countDocsWithWord(Matrix
rowMatrix) {
            double numDoc = 0.0D;
            for(int j = 0; j <
rowMatrix.getColumnDimension(); j++){
                if(rowMatrix.get(0, j) > 0.0D){
                    numDoc++;
                }
            }
            return numDoc;
        }

//Menampilkan Matrix ke dalam file
        public void printMatrix(String jenisMatrix,
Matrix matrix, String[] dokName, String[] terms,
PrintWriter writer ){
            writer.printf("=== %s ===%n", jenisMatrix);
            writer.printf("%15s", " ");
            for (int i = 0; i < dokName.length; i++) {
                writer.printf("%20s", dokName[i]);
            }
            writer.println();
            for (int i = 0; i < terms.length; i++) {
                writer.printf("%15s", terms[i]);
                for (int j = 0; j < dokName.length;
j++) {
                     writer.printf("%20.4f",
matrix.get(i, j));
                }
                writer.println();
            }
                writer.flush();
        }

public void printMatrix2(Matrix matrix,String[]
terms,String[] dokName, PrintWriter writer ) throws
SQLException{
             //mengambil nilai cluster
             Database db = new Database();
             db.connectFirst();
             ResultSet rs = db.executeSelect("SELECT *
FROM dokclustervektor");
             String []data = new String[3];
             int a = 0;
```

```
            //int b = 1;
            for (int i = 0; i < terms.length; i++) {}

            try {
            while (rs.next()){
            data[0] = rs.getString("cluster");
            //writer.printf("Dokumen ke-"+b);
            //writer.printf(",");
            writer.print(data[0]);

            for (int k = 0; k < terms.length; k++) {

            writer.printf(",");
            writer.print(matrix.get(k, a));
            }

            writer.println();
            a++;
            //b++;
            } }catch (Exception e){}

            writer.flush();

            }


   private String[] getDokNames() {
        String[] doks = new
String[dokTAiDMap.keySet().size()];

        for(int i = 0; i < doks.length; i++){
            String dok = dokTAiDMap.get(i);
            doks[i] = dok;
        }

        return doks;
    }

    private String[] getTerms() {
        String[] terms = new
String[termIDMap.keySet().size()];
        for(int i = 0; i < terms.length; i++){
            String term = termIDMap.get(i);
            terms[i] = term;
```

```
        }  return terms;
    }

     public Map<String, ArrayList<Double>>
prosesPembobotan1() throws SQLException,
FileNotFoundException{

            Database koneksi = new Database();
            koneksi.connectFirst();
            Database db = new Database();
            db.connectFirst();
            koneksi.executeUpdate("DELETE FROM
`list_kata_kunci`");
            Map<String, ArrayList<Double>> inputKmeans
= new HashMap<>();
            Matrix matrix;
            String query = "SELECT * FROM
representasi_dokumen ORDER BY `id_dok` ASC ";
            ResultSet rset = db.executeSelect(query);
            int mapID = 0;
            String prevDokID = "";

            /*Map<Integer, String> dokTAiDMap => list
id dokumen
              Map<String, Bag<String>> dokKeyIDMap =>
list term hasil ekstraksi lucene untuk tiap dokumen
            */
            while(rset.next()){
                String dokID =
rset.getString("id_dok");
                String key =
rset.getString("frase_rep");
                if(dokID.equals(prevDokID) == false){
                    Bag<String> dokKey = new
HashBag<String>();
                    dokKey.add(key);
                    dokTAiDMap.put(mapID, dokID);
                    dokKeyIDMap.put(dokID, dokKey);
                    mapID++;

                }
                else{
                    dokKeyIDMap.get(dokID).add(key);
                }
                prevDokID = dokID;
```

```
            }

            //query = "SELECT * FROM
`list_kata_kunci_norake` ORDER BY `kata_kunci` ASC  ";
            rset = db.executeSelect(query);
            ArrayList<String> terms = new
ArrayList<>();
            SortedSet<String> termSet = new
TreeSet<String>();
            while(rset.next()){
                String key =
rset.getString("frase_rep");
                if(!termSet.contains(key)&&key.length()
> 1){
                                termSet.add(key);
                                }

            }
            int count = 0;

            String prevTerm = "";
            for(String s : termSet){
                if(s.equals(prevTerm) == false){
                    termIDMap.put(count, s);
                    count++;
                    String query1 = "INSERT INTO
`list_kata_kunci`(`id_katakunci`,`kata_kunci`) VALUES
(\""+count+"\",\""+s+"\")";
                    koneksi.executeUpdate(query1);
                }
                prevTerm = s;

            }

            double[][] data = buildMatrix();
            //Matriks kemunculan term pada tiap dokumen
            matrix = new Matrix(data);
            //matrix = tfIndexer(matrix);
            countNonZero(matrix);
            System.out.println("Proses TF-IDF");
            matrix = idfIndexer(matrix);
            //countNonZero(matrix); //document
frequency
            System.out.println("Proses TF-IDF
selesai");
```

```
            String[] dokNames = getDokNames();
            String[] lstTerms = getTerms();

           File TF_IDF = new
File("C:\\Users\\COMPAQ\\Documents\\NetBeansProjects\\B
isa\\TF-IDF.txt");
           printMatrix("Term Frequency", matrix,
dokNames, lstTerms, new PrintWriter(TF_IDF));

            //Proses SVD
            System.out.println("Proses lsiindexer");
            matrix = lsiIndexer(matrix);
            System.out.println("Proses lsiindexer
selesai");
//
            System.out.println("Build KMeans Input");
            inputKmeans = formMap(matrix,
dokNames.length, lstTerms);
            System.out.println("Build KMeans Input
selesai");

            //savetoDB(dokTAiDMap, dokKeyIDMap);
            db.destroyConnection();
            return inputKmeans;
     }

void prosesPembobotan2() throws SQLException,
FileNotFoundException{
            Database koneksi = new Database();
            koneksi.connectFirst();
            Database db = new Database();
            db.connectFirst();

            Matrix matrix;
            String query = "SELECT * FROM
representasi_dokumen ORDER BY `id_dok` ASC ";
            ResultSet rset = db.executeSelect(query);
            int mapID = 0;
            String prevDokID = "";

            /*Map<Integer, String> dokTAiDMap => list
id dokumen
              Map<String, Bag<String>> dokKeyIDMap =>
list term hasil ekstraksi lucene untuk tiap dokumen
            */
```

```
            while(rset.next()){
                String dokID =
rset.getString("id_dok");
                String key =
rset.getString("frase_rep");
                if(dokID.equals(prevDokID) == false){
                    Bag<String> dokKey = new
HashBag<String>();
                    dokKey.add(key);
                    dokTAiDMap.put(mapID, dokID);
                    dokKeyIDMap.put(dokID, dokKey);
                    mapID++;

                }
                else{
                    dokKeyIDMap.get(dokID).add(key);
                }
                prevDokID = dokID;
            }


            rset = db.executeSelect(query);
            ArrayList<String> terms = new
ArrayList<>();
            SortedSet<String> termSet = new
TreeSet<String>();
            while(rset.next()){
                String key =
rset.getString("frase_rep");
                if(!termSet.contains(key)&&key.length()
> 1){
                            termSet.add(key);
                            }

            }
            int count = 0;
            String prevTerm = "";
            for(String s : termSet){
                if(s.equals(prevTerm) == false){
                    termIDMap.put(count, s);
                    count++;
                }
                prevTerm = s;

            }
```

```
            double[][] data = buildMatrix();

            //Matriks kemunculan term pada tiap dokumen
            matrix = new Matrix(data);
            //matrix = tfIndexer(matrix);
            countNonZero(matrix);
            System.out.println("Proses idfindexer");
            matrix = idfIndexer(matrix);
            System.out.println("Proses idfindexer
selesai");
            String[] dokNames = getDokNames();
            String[] lstTerms = getTerms();


           File INPUT_SVM = new
File("C:\\Users\\COMPAQ\\Documents\\NetBeansProjects\\B
isa\\inputSVM.txt");
            printMatrix2(matrix,lstTerms, dokNames, new
PrintWriter(INPUT_SVM));


     }

    private Map<String, ArrayList<Double>>
formMap(Matrix matrix, int length, String[] lstTerms) {
        System.out.println("Matrix size = " +
matrix.getRowDimension() + " X " +
matrix.getColumnDimension());
        System.out.println("Term size = " +
lstTerms.length);
        Map<String, ArrayList<Double>> map = new
HashMap<>();
        for(int i = 0; i < lstTerms.length; i++){
            ArrayList<Double> termVektor = new
ArrayList<>();
            for(int j = 0; j <
matrix.getColumnDimension(); j++){
                termVektor.add(matrix.get(i, j));
            }
            map.put(lstTerms[i], termVektor);
        }

        return map;
    }
```

```
    private void savetoDB(Map<Integer, String>
dokTAiDMap, Map<String, Bag<String>> dokKeyIDMap)
throws SQLException {

        Database koneksi = new Database();
        koneksi.connectFirst();
        String query = "";
        for(int i = 0; i < dokTAiDMap.keySet().size();
i++){
            String idDok = dokTAiDMap.get(i);
            Bag<String> repDok =
dokKeyIDMap.get(idDok);
            for(String s : repDok){
                query = "INSERT INTO
`list_kata_kunci`(`id_dok`, `kata_kunci`) VALUES
(\""+idDok+"\",\""+s+"\")";
                koneksi.executeUpdate(query);
            }
        }
        koneksi.destroyConnection();
    }

    private void countNonZero(Matrix matrix) {
        for(int i = 0; i < matrix.getRowDimension();
i++){
            int nonZeroval = 0;
            for(int j = 0; j <
matrix.getColumnDimension(); j++){
                if(matrix.get(i, j) > 0.0){
                    nonZeroval += 1;
                }
            }
            //System.out.println(i +" = "+ nonZeroval);
        }
    }




}
```

4. Pengelompokan menggunakan *K-Means*

```
package tugasakhir;

import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

public class Clustering {

   int iterations;
   int numOfClusters;

// default constructor : iteration =50 numOfClusters=2
public Clustering(){
      //iterations=50;
      //numOfClusters=5;
 }

public Clustering (int numOfIteration, int
numOfClusters){
      this.iterations=numOfIteration;
      this.numOfClusters=numOfClusters;
 }

public Clusters[] ProsesKmeans (ArrayList<Point> input,
int clustMetode) throws SQLException{

    System.out.println("Proses K-Means");
    Database koneksi = new Database();
    koneksi.connectFirst();
    koneksi.executeUpdate("DELETE FROM `topikcluster`");
    Clusters[] cl = getClusters(input, clustMetode);
    String query = "";
    System.out.println("Proses K-Means selesai");

    for (int i = 0; i < cl.length; i++) {
        for (int j = 0; j < cl[i].clus.size(); j++) {
            query = "INSERT INTO
`topikcluster`(`id_cluster`, `term`) "
                    + "VALUES
(\""+i+"\",\""+cl[i].clus.get(j).getTerm()+"\")";
            koneksi.executeUpdate(query);
```

```
        }
    }

    koneksi.destroyConnection();
    return cl;
}

public Clusters[] ProsesKmeansSmall (ArrayList<Point>
input, int clustMetode) throws SQLException{

    System.out.println("Proses K-Means");
    Clusters[] cl = getClusters(input, clustMetode);
    System.out.println("Proses K-Means selesai");

    return cl;
}


// untuk mendapatkan Cluster
public Clusters[] getClusters(ArrayList<Point> data, int
clusterMetode) {
    ArrayList<Point> centers = new ArrayList<>();
    double prevconvergen = Double.MAX_VALUE;
    centers = getRandCentres(data);
    Clusters[] clusters = reallocation(centers, data);

    ArrayList<Point> prevCenter = centers;
    for (int i = 0; i < 1000; i++) {

        double convergence = 0.0;
        centers = getCenters(clusters);

        clusters = reallocation(centers, data);

        convergence = countThreshold(prevCenter,
centers);

        System.out.println(convergence);
        if(convergence <= 0.001 && prevconvergen <=
0.001){
            break;
        }
```

```
        prevconvergen = convergence;
        prevCenter = centers;

    }

  return clusters;

 }

// mendapatkan pusat dari kluster
public ArrayList<Point> getCenters(Clusters[] arr) {
       ArrayList<Point> centers = new ArrayList<>();

       Double vek;
       for (int i = 0; i < numOfClusters; i++) {

            ArrayList<Double> vektor = new
ArrayList<>(Collections.nCopies(arr[0].clus.get(0).getVe
ktor().size(), 0.0));

            for (int j = 0; j < vektor.size(); j++) {
                vek = 0.0;
                if(arr[i].clus.size() != 0){

                    for(int k = 0; k <
arr[i].clus.size(); k++){
                        vek +=
arr[i].clus.get(k).getVektor().get(j);
                    }
                }

                vektor.set(j, vek);
            }

            if (arr[i].clus.size() > 0) {
                for(int a = 0; a < vektor.size(); a++){
                    vektor.set(a,
vektor.get(a)/arr[i].clus.size());
                }
                centers.add(new Point("",vektor));
            }
```

```
        }

        return centers;
       }


//Inisialisasi K-Means
public ArrayList<Point> getRandCentres(ArrayList<Point>
data){
    Random random = new Random();
    ArrayList<Point> centers=new ArrayList<>();
    ArrayList<Integer> randomedNum = new ArrayList<>();


    for(int i=0;i<numOfClusters;i++){
        Integer rand = random.nextInt(data.size());
        if(i == 0){
            centers.add(data.get(rand));
            randomedNum.add(rand);
        }
        else if(!randomedNum.contains(rand)){
            centers.add(data.get(rand));
            randomedNum.add(rand);
        }
        else{
            i = i - 1;
        }
    }

    return centers;
 }


//public Double distance(Point x, Point y) {
    Double distance = 0.0;
    for(int i = 0; i < x.getVektor().size(); i++){

if(x.getVektor().get(i).isNaN()&&y.getVektor().get(i).is
NaN()){
            distance += Math.pow(0 - 0, 2);
        }
```

```
        else if(x.getVektor().get(i).isNaN()){
            distance += Math.pow(0 -
y.getVektor().get(i), 2);
        }
        else if(y.getVektor().get(i).isNaN()){
            distance += Math.pow(x.getVektor().get(i) -
0, 2);
        }
        else{
            distance += Math.pow(x.getVektor().get(i) -
y.getVektor().get(i), 2);
        }
    }
    distance = Math.sqrt(distance);
    return distance;
  }//

public Double distance(Point x, Point y) {
    Double distance = 0.0;
    for(int i = 0; i < x.getVektor().size(); i++){
        distance += Math.pow(x.getVektor().get(i) -
y.getVektor().get(i), 2);
    }
    distance = Math.sqrt(distance);
    return distance;
  }

// regrouping of Points
public Clusters[] reallocation(ArrayList<Point> centers,
ArrayList<Point> data) {

       Clusters[] clus = new Clusters[numOfClusters];
       for (int i = 0; i < numOfClusters; i++){
            clus[i] = new Clusters();
       }

       for (int i = 0; i < data.size(); i++) {
            Double dis = Double.MAX_VALUE;
            int ind = 0;
            Double temp = 0.0;
            for (int j = 0; j < centers.size(); j++) {
```

```
temp = distance(data.get(i), centers.get(j));
                //System.out.println("Distance data ke-"
+ i + " terhadap pusat ke-" + j + " = " + temp);
                if (temp < dis) {
                   dis = temp;
                   ind = j;
                }

            }
            //System.out.println("Data masuk pusat ke-"
+ ind);
            clus[ind].clus.add(data.get(i));


       }
       return clus;

      }

private double countThreshold(ArrayList<Point>
prevCenter, ArrayList<Point> centers) {
        double convergence = 0.0;
        for(int i = 0; i < prevCenter.size(); i++){

            if(i >= centers.size()){
                convergence += 0;
            }else{
                convergence +=
distance(prevCenter.get(i), centers.get(i));
            }
        }

        convergence = convergence/centers.size();

        return convergence;
    }



}
```

5. Pengubahan TF-IDF menjadi inputan SVM

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {

        try (Stream<String> stream =
Files.lines(Paths.get("C:\\Users\\COMPAQ\\Documents\\N
etBeansProjects\\Bisa\\inputSVM.txt"));
                BufferedWriter bw =
Files.newBufferedWriter(Paths.get("C:\\Users\\COMPAQ\\
Documents\\NetBeansProjects\\Bisa\\bismillah.txt"),
StandardCharsets.UTF_8)) {
            Object[] lines = stream.toArray();
             int a =1;
            for (Object line : lines) {
                if (line instanceof String &&
(((String) line).trim().length() > 0)) {
                    String[] data = ((String)
line).split(",");
                        String dataLine = "";
                        //String cobaData = "";

                        switch (data[0]) {
                            case "0":
                                dataLine += "0";
                                //cobaData +=
data[0]+","+"0";
                                break;
                            case "1":
                                dataLine += "1";
                                //cobaData +=
data[0]+","+"1";
                                break;
                            case "2":
                                dataLine += "2";
                                //cobaData +=
data[0]+","+"2";
                                break;
                            case "3":
                                dataLine += "3";
                                //cobaData +=
data[0]+","+"3";
                                break;
```

```
                                case "4":
                                    dataLine += "4";
                                   // cobaData +=
data[0]+","+"4";
                                    break;
                                case "5":
                                    dataLine += "5";
                                    //cobaData +=
data[0]+","+"5";
                                    break;
                                case "6":
                                    dataLine += "6";
                                   // cobaData +=
data[0]+","+"6";
                                .
                                .
                                .




                                default:
                                    break;
                            }

                            for (int i = 1; i < 3069; i++)
{
                                double attributeValue =
Double.parseDouble(data[i]);
                                if (attributeValue != 0.0)
{
                                    dataLine += " " + (i)
+ ":" + attributeValue;
                                }

                            }


//System.out.println(cobaData);
                           System.out.println("Dokumen
ke-"+a+" "+dataLine);
```

```
} a++;
            }
        } catch (IOException ex) {

Logger.getLogger(GUI_TA.class.getName()).log(Level.SEV
ERE, null, ex);
         }

      double percentageInTrain = 80;
      try (Stream<String> stream =
Files.lines(Paths.get("C:\\Users\\COMPAQ\\Documents\\N
etBeansProjects\\Bisa\\bismillah.txt"));
               BufferedWriter btrain =
Files.newBufferedWriter(Paths.get("C:\\Users\\COMPAQ\\
Documents\\NetBeansProjects\\Bisa\\train.txt"),
StandardCharsets.UTF_8);
               BufferedWriter btest =
Files.newBufferedWriter(Paths.get("C:\\Users\\COMPAQ\\
Documents\\NetBeansProjects\\Bisa\\test.txt"),
StandardCharsets.UTF_8))
               {
       Object[] lines = stream.toArray();
       Integer ntotal=0;
       Integer ntrain=0;
       Integer ntest=0;

       for (Object line: lines){
           if (line instanceof String && (((String)
line).trim().length() > 0)) {
               String data = (String)line;
               ntotal++;

               if (Math.random() <
percentageInTrain/100){
                   btrain.write(data);
                   btrain.newLine();
                   ntrain++;
           } else {
                   btest.write(data);
                   btest.newLine();
                   ntest++;
               }
```

```
            }
        }

        System.out.println("Total teks laporan : " +
ntotal);
        System.out.println("Total laporan training : "
+ ntrain);
        System.out.println("Total laporan testing : " +
ntest);

    }

        catch (IOException e) {
                }        // TODO add your handling code
here:
    }
```

BIODATA PENULIS

Penulis dengan sapaan Eries dan nama lengkap Eries Bagita Jayanti bertempat tinggal di kabupaten Gresik. Anak pertama dari tiga bersaudara ini lahir tanggal 19 Februari 1996. Pernah turut aktif pada bidang organisasi diantaranya menjadi sekertaris departemen himpunan sekaligus sekertaris departemen lembaga dakwah jurusan di matematika ITS bahkan juga pernah aktif di organisasi lingkup ITS yakni JMMI ITS. Beberapa pelatihan dan seminar yang pernah diikuti yakni PKTI, *Organization Managerial Training* serta Program Studi Islam 2. Penulis juga pernah mengikuti beberapa kepanitiaan misalnya Gebyar Ibnu Muqlah, Ramadhan di Kampus serta Olimpiade Matematika ITS. Saat masa perkuliahan di jurusan Matematika ITS penulis mengambil rumpun mata kuliah ilmu komputer dan pernah melaksanakan kerja praktek di PT. DCK dengan *jobdesk* menguji ERP (*Enterprise Resources Planning*).

Jika ada yang ingin didiskusikan ataupun ditanyakan terkait tugas akhir penulis jangan ragu untuk bertanya via email : eriesbagita8@gmail.com . Terima kasih dan semoga tugas akhir ini bermanfaat.