**TUGAS AKHIR- KM184801**

# IMPLEMENTASI *TEXT MINING* UNTUK PENGELOMPOKAN ULASAN PELANGGAN *E-COMMERCE* BERDASARKAN TOPIK ULASAN MENGGUNAKAN ALGORITMA *HIERARCHICAL AGGLOMERATIVE CLUSTERING*

**LI'IZZA DIANA MANZIL**
**06111540000049**

**Dosen Pembimbing :**
**Prof. Dr. Mohammad Isa Irawan, M.T.**

**DEPARTEMEN MATEMATIKA**
**Fakultas Matematika Komputasi dan Sains Data**
**Institut Teknologi Sepuluh Nopember**
**Surabaya 2019**

**FINAL PROJECT– KM184801**

# *IMPLEMENTATION TEXT MINING FOR GROUPING REVIEW CUSTOMER E-COMMERCE BASED ON TOPIC REVIEW USING HIERARCHICAL AGGLOMERATIVE CLUSTERING*

**LI'IZZA DIANA MANZIL**
**06111540000049**

**Supervisor :**
**Prof. Dr. Mohammad Isa Irawan, M.T.**

*DEPARTMENT OF MATHEMATICS*
*Faculty of Mathematics, Computing, and Data Science*
**Institut Teknologi Sepuluh Nopember**
**Surabaya 2019**

**LEMBAR PENGESAHAN**

**IMPLEMENTASI *TEXT MINING* UNTUK PENGELOMPOKAN ULASAN PELANGGAN *E-COMMERCE* BERDASARKAN TOPIK ULASAN MENGGUNAKAN ALGORITMA *HIERARCHICAL AGGLOMERATIVE CLUSTERING***

***IMPLEMENTATION TEXT MINING FOR GROUPING REVIEW CUSTOMER E-COMMERCE BASED ON TOPIC REVIEW USING HIERARCHICAL AGGLOMERATIVE CLUSTERING***

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Matematika
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
LI'IZZA DIANA MANZIL
NRP. 06111540000049

Menyetujui,
Dosen Pembimbing ,

Prof. Dr. Mohammad Isa Irawan, M.T.
NIP. 19700831 199403 1 003

Mengetahui,
Kepala Departemen Matematika
FMKSD ITS

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

Surabaya, 25 Juli 2019

# IMPLEMENTASI TEXT MINING UNTUK PENGELOMPOKAN ULASAN PELANGGAN *E-COMMERCE* BERDASARKAN TOPIK ULASAN MENGGUNAKAN ALGORITMA *HIERARCHICAL AGGLOMERATIVE CLUSTERING*

**Nama** : **Li'Izza Diana Manzil**
**NRP** : **06111540000049**
**Departemen** : **Matematika FMKSD - ITS**
**Pembimbing** : **Prof. Dr. Mohammad Isa Irawan, M.T.**

**ABSTRAK**

Pesatnya perkembangan teknologi membuat masyarakat lebih mudah melakukan kegiatan apapun. Misalnya dalam kegiatan bisnis. Kini sebagian besar masyarakat lebih suka melakukan pembelian melalui internet, karena lebih mudah dalam transaksinya. Hal itu mengakibatkan banyak munculnya *e-commerce* di Indonesia. Namun, pembelian barang melalui *e-commerce* terkadang tidak sesuai dengan apa yang diinginkan, baik secara pelayanan maupun barang yang diperoleh. Oleh karena itu, banyak *e-commerce* yang memberikan fasilitas pemberian ulasan mengenai pelayanan ataupun barang. Ulasan yang diberikan pelanggan sangat beragam. Dengan demikian, perlu adanya suatu penelitian dengan tujuan mengelompokkan ulasan untuk memudahkan penjual dan calon pelanggan *e-commerce* dapat mengetahui kualitas barang melalui aspek yang diulas oleh pelanggan-pelanggan sebelumnya. Metode yang digunakan pada penelitian ini adalah *clustering* dengan algoritma *Hierarchical Agglomerative Clustering* (HAC). Untuk menentukan jumlah *cluster* pada ulasan dilakukan pemotongan jarak *cluster* pada titik *threshold* tertentu. *Threshold* yang digunakan pada proses *clustering* adalah 0.94, 0.95 dan 0.96 dengan masing-masing memiliki hasil 15, 8 dan 4. Dari ketiga proses *clustering* tersebut hasil terbaik dimiliki proses *clustering* dengan *threshold* 0.96 dengan nilai presisi sebesar 85.2 %.

**Kata Kunci:** ***E-commerce, HAC, Ulasan***

# *IMPLEMENTATION TEXT MINING FOR GROUPING REVIEW CUSTOMER E-COMMERCE BASED ON TOPIC REVIEW USING HIERARCHICAL AGGLOMERATIVE CLUSTER*

**Name** : **Li'Izza Diana Manzil**
**NRP** : **06111540000049**
**Department** : **Mathematics FMKSD - ITS**
**Supervisor** : **Prof. Dr. Mohammad Isa Irawan, M.T.**

## A*BSTRACT*

*The rapid development of technology has made it easier for people to carry out any activities. For example, in business activities. Now most people like to make purchases through the internet, because it is easier in the transaction. That requires many e-commerce transfers in Indonesia. However, the purchase of goods through e-commerce is issued not in accordance with what is desired, both services and goods obtained. Therefore, many e-commerce services provide facilities for reviewing goods services. Customer reviews are very diverse. Thus, there is a need for a study to classify and classify assessments for providers and prospective e-commerce customers to find out what aspects are reviewed by customers. The method that will be used in this study is cluster with the Hierarchical Agglomerative Cluster (HAC) algorithm. To determine the number of clusters in the review, the cluster distance is cut at a certain threshold point. The threshold used in the cluster process is 0.94, 0.95 and 0.96 with each having results of 15, 8 and 4. Based on the three clustering processes the best results have a clustering process with a threshold of 0.96 with a precision value of 85.2%.*

***Keywords**: E-commerce, HAC, Review*

# KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillaahirobbil'aalamiin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan limpahan rahmat, taufik dan hidayah – Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Implementasi Text Mining Untuk Pengelompokan Ulasan Pelanggan *E-Commerce* Berdasarkan Topik Ulasan Menggunakan Algoritma *Hierarchical Agglomerative Clustering*" sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMKSD Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik dan tepat waktu berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terimakasih kepada :

1. Orang tua, adik serta keluarga besar penulis yang telah banyak mendukung dan memberikan semangat dalam menjalani masa perkuliahan.
2. Bapak Dr. Imam Mukhlas, S.Si, MT sebagai Kepala Departemen Matematika FKMSD ITS sekaligus sebagai dosen penguji penulis.
3. Bapak Dr. Didik Khusnul Arif, S.Si., M. Si. Selaku Ketua Program Studi S-1 Departemen Matematika ITS dan Bapak Drs. Iis Herisman, M. Si. Selaku Sekretaris Program Studi S-1 Departemen Matematika yang selama ini sudah bekerja keras dalam membantu dan menanggapi kebutuhan penulis.

4. Bapak Drs. Daryono Budi Utomo, MSi,sebagai dosen wali penulis, yang telah mengarahkan penulis mengenai akademik selama perkuliahan.
5. Bapak Prof. Dr. Mohammad Isa Irawan, MT sebagai dosen pembimbing yang telah memberikan motivasi, bimbingan dan pengarahan dalam menyelesaikan Tugas Akhir ini.
6. Bapak Dr. Imam Mukhlash, S.Si, MT, Bapak Drs. Nurul Hidayat, M. Kom, Ibu Alvida Mustika Rukmi, S.Si, M.Si dan Ibu Dr. Rinurwati, M.Si dosen penguji Tugas Akhir penulis yang telah memberikan masukan, arahan, dan juga saran-saran yang sangat berguna bagi penulis.
7. Seluruh Bapak/Ibu dosen dan seluruh staff Departemen Matematika ITS yang selama perjalanan kuliah telah memberikan pelajaran berharga kepada penulis baik akademik maupun moral.
8. Riko dan Nenek (Ayu N) yang telah membantu penulis memahami dan menyelesaikan Tugas Akhir hingga selesai.
9. Penghuni lab ilkom yang berjuang bersama penulis menyelesaikan tugas kuliah dan tugas akhir ini (Nay, Sima, Rama, Komting, Devia)
10. Sahabat penulis yang tidak hentinya memberikan support dan doanya kepada penulis Nida, Izah, Mail, Mega, Efi, Ayu F, Rukha, Azizah.
11. Teman-teman penghuni setia rumah Aljabar Vira dan Diki yang telah membantu penulis dan meberikan support kepada penulis
12. Teman-teman Matematika angkatan 2015, yang selalu mendukung dan menjadi keluarga penulis selama kuliah.

13. Semua pihak yang telah memberikan dukungan dan ilmu kepada penulis dalam masa perkuliahan hingga penyelesian Tugas Akhir ini.

Penulis menyadari bahwa dalam Tugas Akhir ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan. Akhirnya penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi banyak pihak.

Surabaya, Agustus 2019

Penulis

# DAFTAR ISI

## DAFTAR GAMBAR

## DAFTAR TABEL

# BAB 1
# PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, pelaksanaan kegiatan, dan sistematika penulisan laporan Tugas Akhir.

## 1.1 Latar Belakang

Di era yang serba digital ini teknologi memberikan dampak pada gaya hidup masyarakat. Teknologi semakin berkembang dengan pesat. Salah satu teknologi yang banyak digunakan masyarakat yaitu internet. Internet saat ini menjadi sebuah kebutuhan primer bagi seluruh elemen di masyarakat. Perkembangan internet yang begitu pesat mengantarkan banyak kemudahan bagi masyarakat. Hampir seluruh aspek kehidupan dipengaruhi oleh internet. Seperti dalam bidang bisnis, banyak bermunculan bisnis online dan maraknya kegiatan berbelanja melalui media internet. Suatu survei yang telah dilakukan oleh APJII pada tahun 2016 menyebutkan bahwa terdapat 98,6% pengguna internet di Indonesia mengetahui internet sebagai tempat jual beli barang dan jasa. Sementara 63,5% atau sekitar 84,2 juta pengguna internet pernah melakukan transaksi online[1]. Hal itu akan sering bertambah seiring berkembangnya teknologi dan bisnis online. Hal ini ditandai dengan banyaknya *e-commerce* yang ada di Indonesia. *E-commerce* merupakan proses transaksi jual/beli atau pertukaran barang/jasa dan informasi melalui internet. Banyak masyarakat Indonesia memiliki *e-commerce* baik dalam skala kecil ataupun besar.

[1] Asosiasi Penyelenggara Jasa Internet Indonesia. 2016. **Saatnya Jadi Pokok Perhatian Pemerintah dan Industri.** Buletin APJII.

Tim publikasi Katadata mengemukakan bahwa selama 2014-2017, rata-rata pertumbuhan tahunan penjualan online di Indonesia diperkirakan mencapai 38 persen[2].

Salah satu akibat dari peningkatan jumlah transaksi melalui internet dipengaruhi oleh mudahnya proses transaksi jual/beli. Hanya dengan melakukan kegiatan sederhana seseorang sudah bisa membeli barang yang diinginkan. Namun hal itu juga mempunyai kekurangan. Orang yang akan membeli barang secara online tidak bisa mengetahui bagaimana kualitas barang yang diinginkan, sehingga tidak jarang juga pembeli yang merasa kecewa dengan barang yang telah dibeli. Oleh karena itu, beberapa *e-commerce* menyediakan fasilitas untuk memberikan komentar mengenai barang yang dibeli atau pelayanan yang telah diberikan. Komentar tersebut dinamakan ulasan. Ulasan yang diberikan pelanggan tersebut sangat berpengaruh terhadap penjualan produk *e-commerce*, karena orang yang akan membeli tentu melihat terlebih dahulu kualitas barang atau pelayanan melalui ulasan tersebut. Sehingga pelanggan sangat membutuhkan ulasan ketika ingin membeli suatu barang untuk menghindari ketidakpuasan atas barang yang akan dibelinya.

Ulasan yang diberikan oleh pelanggan *e-commerce* sangat bermacam-macam, diantaranya memuat pujian dan kritikan. Banyaknya ulasan yang diberikan oleh pelanggan mengakibatkan calon pelanggan dan penjual seringkali kesulitan dalam menyimpulkan aspek yang dibahas pada ulasan. Untuk mengolah dan memantau ulasan yang diberikan

---

[2] Tim Publikasi Katadata. 2016. **Infografik: Indonesia, Pusat E-Commerce ASEAN.** Katadata. [Online]. Tersedia : https://katadata.co.id/infografik/2016/01/04/indonesia-pusat-e-commerce-asean. (diakses 24 Januari 2019)

pelanggan bukan hal yang mudah karena jumlah ulasan yang dimuat dalam *website* umumnya sangat banyak apabila dilihat secara manual. Oleh karena itu perlu adanya suatu penelitian untuk mengelompokkan ulasan sehingga dapat memudahkan penjual mengetahui aspek apa yang diulas oleh pelanggan. Dengan demikian penjual dapat melakukan kebijakan mengenai aspek yang dikritik dan mempertahankan aspek yang mendapat pujian. Selain itu dapat memudahkan calon pelanggan dalam mengambil keputusan yang tepat dalam memilih produk.

Dalam penelitian ini dilakukan proses *text mining* untuk pengelompokan ulasan pelanggan berdasarkan aspek yang dibahas. Pengelompokan ulasan menggunakan salah satu metode pada *cluster*. Metode *clustering* yang akan digunakan algoritma *Hirarchical Agglomerative Clustering* (HAC). Proses *clustering* pada penelitian ini bertujuan untuk mengelompokkan ulasan dan menentukan topik yang sedang dibahas pada ulasan.

Beberapa penelitian mengenai ulasan yang diberikan pelanggan telah dilakukan, salah satunya yaitu penelitian yang dilakukan oleh Noor, Zarinah & Hamid di tahun 2015 [1]. Peneliti melakukan identifikasi dan ekstraksi pada dokumen untuk dijadikan input dalam suatu *software* dengan teknik LSA. Dokumen yang dihasilkan kemudian dikelompokkan dengan menggunakan algoritma K-Means dan HAC. Dari *clustering* tersebut hasil yang baik diperoleh dari *clustering* dengan menggunakan algoritma HAC.

Penelitian ini diharapkan mampu mengelompokkan ulasan berdasarkan topik yang dibahas sehingga dapat menjadikan data yang dapat diekstraksi dengan baik untuk memberikan informasi yang bermanfaat bagi pihak-pihak yang terkait.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah penelitian ini yaitu:

1. Bagaimanakah implementasi *text mining* pada ulasan produk di suatu *e-commerce* menggunakan *hierarchical agglomerative clustering*?
2. Bagiamana analisis dari hasil proses *clustering* ulasan pelanggan menggunakan *hierarchical agglomerative clustering?*

## 1.3 Batasan Masalah

Dalam penelitian tugas akhir ini terdapat beberapa batasan masalah yang digunakan, diantaranya:

1. Penelitian ini hanya menganalisa ulasan pelanggan dalam bahasa Indonesia
2. Data yang digunakan berasal dari ulasan produk fashion di *e-commerce* Tokopedia
3. Metode *hierarchical agglomerative clustering* yang digunakan adalah metode *average linkage*

## 1.4 Tujuan

Adapun tujuan yang ingin dicapai pada penelitiaan ini adalah sebagai berikut:

1. Untuk mengimplementasikan text mining pada ulasan produk menggunakan *hierarchical agglomerative clustering*
2. Untuk mengalisis hasil proses *cluster* ulasan pelanggan menggunakan *hierarchical agglomerative clustering*

## 1.5 Manfaat

Manfaat penelitian Tugas Akhir ini adalah sebagai berikut:

1. Dalam bidang akademik, diharapkan dapat mengimplementasikan ilmu matematika dalam bidang *text mining*.
2. Membantu pihak *e-commerce* dalam mengelompokkan ulasan berdasarkan topik, yang dapat dimanfaatkan lebih lanjut, misalnya untuk memudahkan pelanggan dalam pencarian ulasan mengenai topik tertentu dan evaluasi produk berdasarkan aspek-aspek tertentu.

## 1.6 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini disusun dalam lima bab, yaitu:

1. BAB I Pendahuluan
   Pada bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi penelitian, pelaksana kegiatan, tempat dan waktu pelaksanaan, serta sistematika penulisan laporan.
2. BAB II Tinjauan Pustaka
   Pada bab ini dijelaskan teori dasar yang mendukung penulisan Tugas Akhir, yaitu penelitian terdahulu, *e-commerce*, text mining, TF-IDF, LSA dengan menggunakan SVD, serta teori tentang *clustering* dengan menggunakan *Hierarchical Agglomerative Clustering* (HAC).
3. BAB III Metode Penelitian
   Bab ini menjelaskan tentang tahapan-tahapan dan metode yang digunakan disertai penjelasan dalam tiap tahapan yang dilakukan dalam menyelesaikan Tugas Akhir.

Selain itu, terdapat diagram alir dari metodologi penelitian.

4. BAB IV Perancangan dan Implementasi
   Bab ini akan membahas bagaimana sistem dirancang dan diimplementasikan. Perancangan sistem dimulai dari pengumpulan data, pre-proses data, pembobotan dengan TF-IDF, menentukan kemiripan dengan menggunakan LSA, dan melakukan *clustering* dengan algoritma *Hierachical Agglomerative Clustering.*
5. BAB V Hasil dan Pembahasan
   Pada bab ini menjelaskan mengenai hasil dari sistem yang telah dibuat disertai dengan penjelasan pembahasan.
6. BAB V Penutup
   Bab ini menjelaskan kesimpulan akhir yang diperoleh dari pembahasan dan saran untuk pengembangan penelitian selanjutnya.

# BAB II
# TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai teori-teori yang digunakan dalam penyusunan Tugas Akhir ini. Teori-teori tersebut diantaranya tentang *e-commerce, text mining*, TF-IDF, LSA, dan *clustering*. Selain itu dijelaskan penelitian terdahulu yang digunakan untuk menyelesaikan tugas akhir ini.

## 2.1 Penelitian Terdahulu

Penelitian terkait ulasan pelanggan telah dilakukan oleh Noor, Zarinah & Hamid di tahun 2015 [1]. Noor, dkk melakukan penelitian untuk mengidentifikasi dan mengektraksi suatu dokumen ulasan dari produk yang berbeda. Hasil identifikasi dan ekstraksi tersebut dijadikan input yang outputnya menjadi suatu rekomendasi suatu produk. Pada penelitiannya, Noor, dkk menggunakan metode *Latent Semantic Analysis* (LSA) dengan *Singular Value Decomposition (*SVD) untuk *feature selection* dokumen. Setelah proses *feature selection*, dilakukan perbandingan algoritma *clustering*. Algoritma yang dibandingkan yaitu K-Means dan *Hierarchical Agglomerative Clustering* (HAC). Hasil yang diberikan mengatakan bahwa algoritma HAC memberikan hasil yang lebih baik.

Chantal Fry dan Sukanya Manna pada tahun 2016 juga melakukan penelitian mengenai ulasan produk [2]. Fry dan Manna melakukan penelitian untuk mengelompokan ulasan produk berdasarkan topik. Objek yang diteliti yaitu ulasan produk yang terdapat di *website* Amazon. Pada penelitiannya, Fry dan Manna menggunakan dua metode *clustering* yang nantinya akan dibandingkan hasilnya. Metode yang digunakan

adalah K-Means dan Peak-Searching. Berdasarkan hasil penelitiannya, metode K-Means yang lebih baik jika dibandingkan dengan metode Peak-Searching.

Selain itu, terdapat penelitian lain mengenai *clustering* dokumen yang dilakukan oleh Fahrur Rozi, dkk pada tahun 2014 [3]. Para peneliti melakukan pelabelan hasil cluster dengan melakukan *clustering* teks dengan menggunakan *Hierarchical Agglomerative Clustering.* Pada penelitiannya, dilakukan perbandingan *clustering* terhadap ulasan dengan metode *single linkage*, *complete linkage* dan *average linkage*. Hasil pengujian menyatakan bahwa nilai presisi yang dihasilkan relative tinggi dimiliki oleh *cluster* dengan metode *average linkage.*

## 2.2 *E-Commerce*

*E-commerce* diartikan sebagai penggunaan internet dan website untuk bertransaksi bisnis atau bisa juga didefinisikan sebagai suatu transaksi perdagangan secara digital antar organisasi atau individu [4]. *E-commerce* didefinisikan sebagai cara untuk menjual dan membeli barang atau jasa melalui jaringan internet yang mencakup beberapa aspek [5]. *E-commerce* merupakan transaksi digital antara dan diantara organisasi atau secara individu [6]. Perusahaan *e-commerce* juga melakukan kegiatan yang mendukung transaksi pasar mereka, seperti periklanan, pemasaran, dukungan pelanggan, kemanan, pengiriman dan pembayaran [6].

Transaksi di *e-commerce* memiliki sifat lebih personal daripada transaksi di toko offline. Pedagang dapat menargetkan pesan pemasaran mereka kepada individu tertentu dengan menyesuaikan pesan ke perilaku pembeli, nama pembeli, minat pembeli, dan pembelian sebelumnya [6]. *E-commerce* memiliki

banyak informasi tentang konsumen yang dapat dikumpulkan di pasar online pada saat pembelian. Dengan peningkatan padatnya informasi, banyak informasi tentang pembelian dan perilaku konsumen sebelumnya melalui ulasan yang dapat disimpan dan digunakan perusahaan untuk mengembangkan perusahaan di masa mendatang.

Selain itu, *e-commerce* juga bersifat lebih transparan daripada toko offline. Pembeli di *e-commerce* dapat melihat harga suatu barang. Sehingga pembeli dapat membandingkan harga antar penjual online sebelum mereka ingin membeli barang tersebut. Selain membandingkan harga, pembeli juga dapat mengetahui ulasan mengenai barang atau toko yang diberikan pembeli sebelumya. Oleh karena itu, pembeli dapat mempertimbangkan untuk membeli barang tersebut atau tidak. Hal itu juga yang merupakan salah satu faktor banyaknya peminat pembeli di *e-commerce* terus meningkat.

### 2.3 *Text Mining*

Menurut Shivaprasad & Reddy, *Text mining* merupakan suatu proses menemukan dan mengekstraksi informasi dari sekumpulan sumber teks yang banyak dan tidak terstruktur [7]. Sedangkan berdasarkan pendapat Feldman, dkk *Text mining* yaitu suatu proses ekstraksi pengetahuan implisit dari data tekstual [8]. *Text mining* merupakan salah satu variasi dari *data mining* dengan menemukan pola dari sekumpulan data yang berupa teks. Sumber data yang dapat digunakan pada *text mining* yaitu artikel, paper, buku, *website*, media sosial dan lain sebagainya. Adapun langkah-langkah yang dilakukan dalam *text mining* adalah [9]:

1. Praproses data

Praproses data memiliki beberapa tahap diantaranya:

a. Tokenizing

Tokenizing merupakan proses pemisahan teks menjadi token dengan spasi atau tanda baca. Pada prosesnya, tokenizing juga melakukan penghapusan kata-kata yang menyertakan karakter khusus atau nilai numerik seperti 16%, dan token diubah menjadi karakter huruf kecil misalnya token “Manajemen” diubah menjadi “manajemen”.

*b.* *Stopword Removal*

Stopword removel yaitu suatu langkah mengambil kata-kata penting dari hasil *tokenizing*. Dalam proses *stopword removal* dapat menggunakan metode membuang kata yang kurang penting yang disebut *stoplist* atau menyimpan kata penting yang disebut *wordlist*. Contoh kata yang dibuang saat proses stoplist adalah “yang”,”di”,”dan”,”dari”, dan seterusnya.

*c*. *Stemming*

Stemming adalah tahap mencari root kata dari tiap kata hasil *stopword removal*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan untuk teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen [9]. Contohnya pada kata bersama, kebersamaan, menyamai, akan dilakukan proses *stemming* ke kata dasarnya 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 [10].

*2. Text transformation*

*Text transformation* merupakan 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*). *Text transformation* juga melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen. Pada umumnya, dokumen diwakili oleh vektor. Model vektor dibangun dari dokumen dengan mengubah token-token dalam dokumen menjadi vektor numerik yang dioperasikan berdasarkan operasi aljabar linier.

Dalam rangka mambangun model vektor, perlu dilakukan proses pembobotan yang paling banyak digunakan adalah skema *term frequency-inverse document frequency* (TF-IDF). *Term frequency* (TF) didefinisikan sebagai metode pembobotan dengan menghitung jumlah kemunculan suatu kata/istilah dalam suatu dokumen [11]. Misalnya TF pada dokumen pertama untuk kata/istilah "warna" adalah dua, karena kata/istilah tersebut muncul dua kali dalam dokumen pertama. Pada asumsi pembobotan dibalik TF-IDF, kata-kata dengan nilai TF yang tinggi mendapat bobot yang tinggi kecuali jika jumlah dokumen yang mengandung kata tersebut juga tinggi yang disebut *inverse document frequency* (IDF). Misalnya kata "yang" memiliki jumlah kemunculan yang tinggi tetapi jumlah dokumen yang mengandung kata "yang" juga tinggi, sehingga

kata tersebut memiliki bobot yang rendah. Nilai TF merupakan jumlah kata yang muncul pada data. Sedangkan persamaan IDF adalah sebagai berikut:

$$idf_j = log\left(\frac{N}{doc_j}\right) + 1 \tag{2.1}$$

dengan:

| | |
|---|---|
| $idf_j$ | : inverse document *frequency* kata *j* |
| $N$ | : jumlah dokumen yang digunakan |
| $doc_j$ | : jumlah dokumen yang mengandung kata *j* |

Sehingga untuk memperoleh bobot TF-IDF untuk kata *j* pada dokumen ke- $i$, $w_{i,j(pra)}$ adalah sebagai berikut:

$$w_{i,j(pra)} = tf_{i,j} \times idf_i \tag{2.2}$$

Normalisasi bobot TF-IDF dari hasil pada Persamaan (2.3) dapat dilakukan menggunakan Persamaan (2.4) berkut:

$$w_{i,j} = \frac{w_{i,j(pra)}}{\left|\sum_{j=1}^{n} w_{i,j(pra)}\right|} \tag{2.3}$$

dengan:

| | |
|---|---|
| $w_{i,j}$ | : bobot kata ke $j$ pada dokumen $i$ yang telah dinormalisasi |
| $w_{i,j(pra)}$ | : bobot kata $j$ pada dokumen $i$ |

Bobot TF-IDF yang diperoleh berupa elemen yang membentuk mantriks. Matriks yang didapatkan adalah matriks yang merepresentasikan dokumen dalam kolom dan token-token atau kata yang sudah dipisah-pisahkan dalam baris.

Berikut merupakan bentuk matriks yang dihasilkan dengan menggunakan Persamaan (2.3).

Tabel 2. 1 Bentuk Matriks TF-IDF

| | Kata 1 | Kata 2 | … | Kata $j$ |
|---|---|---|---|---|
| Dokumen 1 | $w_{1,1}$ | $w_{2,1}$ | … | $w_{j,1}$ |
| Dokumen 2 | $w_{2,1}$ | $w_{2,2}$ | … | $w_{j,2}$ |
| … | … | … | … | |
| Dokumen $j$ | $w_{j,1}$ | $w_{j,2}$ | … | $w_{j,j}$ |

Tabel 2.1 merupakan *document by term matrix* dengan elemen $a_{ij}$, dimana $i$ merupakan indeks dokumen sedangkan $j$ merupakan indeks kata pada dokumen.

*3. Feature Selection*

*Feature selection* adalah tahap lanjut dari pengurangan dimensi pada proses *text transformation*. Proses ini bertujuan untuk pemilihan kata-kata 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*.

*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. Dalam ilmu data mining, untuk mennetukan pola pada suatu data terdapat beberapa *task,* diantaranya *Classification, Prediction, Time-Series Analysis, Association, Clustering* dan *Summarization.* Dalam penelitian ini, dari beberapa yang telah

disebutkan, *task clustering* yang digunakan dalam pencarian *pattern discovery*.

*Clustering* adalah metode pengelompokan data yang bertujuan untuk menemukan dan menggambarkan kohesif atau homogeneus potongan data [13]. Pengelompokan adalah disiplin yang bertujuan untuk mengungkapkan kelompok atau kelompok-kelompok serupa entitas dalam data. *Clustering* memunculkan banyak informasi yang penggunaannya mengungkapkan struktur data [13]. *Clustering* merupakan suatu metode untuk membuat kelompok objek, atau cluster, sehingga obkjek dalam satu *cluster* sangat mirip dan objek dalam *cluster* berbeda memiliki sifat yang berbeda.

Algoritma *clustering* membangun sebuah model dengan melakukan serangkaian pengulangan dan berhenti ketika model tersebut sudah terpusat dan segmentasi lebih stabil. Hasil *clustering* yang baik ditentukan oleh ukuran kesamaan dan metode yang digunakan. Pendekatan dalam *cluster* dibagi menjadi dua kelompok utama yaitu [16]:

i. Metode Hirarki, yaitu metode yang membentuk *cluster* yang membagi partisi secara berulang-ulang dari atas ke bawah atau sebaliknya. Hasil dari metode hirarki berupa dendogram yang mewakili kelompok objek dan tingkat kesamaan di mana terdapat perubahan pengelompokkan.
ii. Metode Partisi, yaitu metode yang membuat inisial partisi untuk membentuk. Kemudian secara iteratif menggunakan teknik relokasi dengan mencoba berulang-ulang memindahkan objek dari satu kelompok ke kelompok lain untuk memperoleh partisi optimal.

## 2.4 *Latent Semantic Analysis*

*Latent Semantic Analysis* (LSA) merupakan metode untuk menemukan korelasi, hubungan, kemiripin, atau keterkaitan antar dokumen, penggalan dokumen, dan kata-kata yang muncul pada dokumen dengan pendekatan matematika [14]. LSA yaitu suatu teknik matematika atau statistika untuk mengekstraksi dokumen dalam menemukan hubungan dari penggunaan kata secara kontekstual.

LSA melakukan metode *Singular Value Decomposition* (SVD) pada suatu matriks. Untuk SVD akan dijelaskan pada subbab berikutnya.

## 2.5 *Singular Value Decomposition*

*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 dimana variasi yang 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 [15].

SVD adalah proses menguraikan matriks yang dideskripsikan $A_{mn}$ dengan $m$ merupakan jumlah ulasan dan $n$ adalah jumlah term hasil ekstraksi menjadi matriks matriks orthogonal berukuran $mxn$, matriks diagonal dengan ukuran $nxn$, dan matriks orthogonal $nxn$. Misalkan terdapat $m$ jumlah vektor yang berukuran $1\ x\ n$, $A = \{a_1, a_2, \dots, a_n\}$. Maka matriks $mxn$ adalah:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

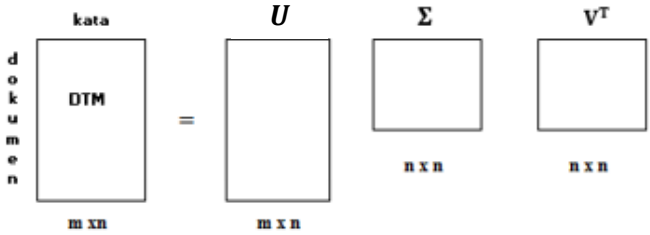
Dari matriks A dapat diperoleh $A^T$ sehingga dapat dikonstruksi $B = A^T A$ jika $m \geq n$ atau $B = AA^T$ jika $m < n$. $B$ merupakan matriks persegi dan simetri dengan ukuran $r\ x\ r$ dimana $r = \min(m, n)$. Dapat dibuktikan bahwa $B$ merupakan matriks simetri, yakni $B^T = (A^T A)^T = A^T (A^T)^T = A^T A = B$. Nilai eigen $(\lambda)$ dari matriks B berjumlah $r$ dan bilangan real positif.

Penguraian matriks A$mn$ menjadi tiga matriks dapat dilihat pada Persamaan (2.4):

$$A = U\Sigma V^T \tag{2.4}$$

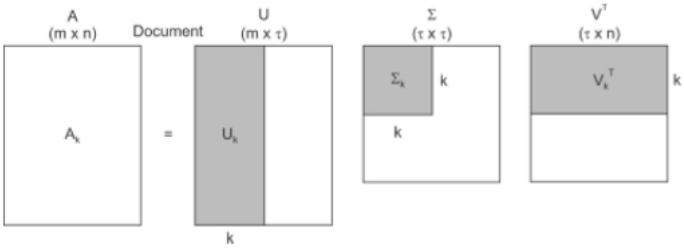
dengan $U$ merupakan $m$ eigenvector dari matriks $AA^T$ yang disusun secara vetor kolom. Kemudian $m$ nilai singular yang berasal dari akar nilai eigen, $\sigma_i = \sqrt{\lambda_i}$, sehingga $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ yang diperoleh dari akar kuadrat nilai eigen $A^T A$ yang membentuk Σ yang disusun dengan membentuk diagonal. Yang terakhir adalah $n$ eigen vektor dari matriks $AA^T$ yang membentuk matriks $V^T$ yang disusun dari vektor baris.

Implementasi LSA pada ini dimisalkan *document-term matrix* (DTM) tersusun atas $A_{mn}$ dengan $m$ merupakan jumlah dokumen dan $n$ menunjukkan banyaknya kata. Kemudian ditentukan $k$ topik dengan $1 < k < n$. Dari matriks B dilakukan pemecahan matriks menjadi $U\Sigma V^T$ seperti pada Gambar 2.1.




Gambar 2. 1 Ilustrasi Matriks SVD

Pereduksian matriks SVD menjadi *k* komponen dengan $1 < k < r$ dibentuk matriks *principal components*, yaitu matriks $U_k \times \Sigma_k$. $U_k$ adalah matriks singular kiri yang sudah direduksi dengan mengambil *k* kolom pertama dan semua baris matriks, kemudian $\Sigma_k$ merupakan matriks dengan elemen-elemennya merupakan nilai singular yang telah direduksi dengan mengambil $k$ baris pertama dan $k$ kolom pertama. Sedangkan, $V_k$ merupakan matriks singular kanan yang telah direduksi dengan mengambil $k$ baris pertama pada semua kolom. Berikut digambarkan ilustrasi pereduksian SVD pada Gambar 2.2




Gambar 2. 2 Ilustrasi Truncated SVD

Tampak pada Gambar 2.2, matriks $U_k$ berukuran $m \times k$ dan $\Sigma_k$ dengan ukuran $k \times k$. Hasil reduksi dengan *truncated SVD*

menjadi *principal components* ($U_k\Sigma_k$) akan menghasilkan matriks yang direpresentasikan dalam Tabel 2.2 berikut.

Tabel 2.2. Tabel *Principal Components*

| | PC1 | PC2 | …. | PC K |
|---|---|---|---|---|
| Ulasan 1 | $b_{1,1}$ | $b_{1,2}$ | … | $b_{1,k}$ |
| Ulasan 2 | $b_{2,1}$ | $b_{2,2}$ | … | $b_{2,k}$ |
| … | … | … | … | … |
| Ulasan m | $b_{m,1}$ | $b_{m,2}$ | … | $b_{m,k}$ |

Tabel 2.2 menunjukkan bobot ulasan dalam sebuah *principal components*. Penentuan jumlah $k$ pada interval tertentu dalam suatu kumpulan ulasan dengan LSA dapat ditentukan dengan menganalisis perbandingan rasio dari nilai eigen $\sigma_i^2$ sesuai pada Persamaan (2.5) berikut:

$$rasio = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \tag{2.5}$$

dengan:

$\sigma_i$ : nilai singular ke-$i$

**2.6 *Cosine Similarity***

Metode *Cosine Similarity* merupakan metode untuk mencari jarak antar dokumen untuk mengetahui tingkat kesamaannya [1]. Cosine similarity merupakan ukuran *cosinus* dari sudut diantara dua dokumen yang direpresentasikan dengan vektor tak nol positif. Metode *cosine similarity* ini menghitung kesamaan antara dua buah ulasan yang dinyatakan dalam dua vektor. Didefinisikan ulasan 1 dengan $d_1$ dan ulasan 2 dengan

$d_2$, dimana $d_1 = |f_{11}\ f_{12} \dots f_{1j}|$ dan $d_2 = |f_{21}\ f_{22} \dots f_{2j}|$. Untuk memperoleh *cosine similarity* dapat menggunakan persamaan berikut:

$$Cos(d_1, d_2) = \frac{d_1 . d_2}{||d_1||||d_2||} \quad (2.6)$$

dengan:

$\cos(d_1, d_2)$ : nilai $cosine\ similarity$ antara dokumen 1 dan dokumen 2

Nilai yang dihasilkan dari perhitungan menggunakan Persamaan (2.6) menhasilkan nilai pada interval -1 sampai 1. Hal ini bertentangan dengan *bag of word* yang nilainya berada pada rentang 0 hingga 1 karena vektor frekuensi tidak pernah negative. Sehingga Jarak antara dua dokumen dapat menggunakan persamaan berikut:

$$dis(d_1, d_2) = 1 - \cos(d_1, d_2) \quad (2.7)$$

dengan:

$dis(d_1, d_2)$ : jarak antara dokumen 1 dan dokumen 2

## 2.7 *Hierarchical Agglomerative Cluster*

*Hierarchical clustering* merupakan salah satu algoritma pada proses *clustering* yang mengelompokkan berdasarkan tingkatan tertentu. Hierarchical algoritma diantara *top-down* atau bottom-up [1]. Algoritma *bottom-up* memperlakukan setiap dokumen yang awalnya ter-*cluster* tunggal yang kemudian secara berturut-turut menggabungkan (atau mengelompokkan) pasangan *cluster* sampai semua *cluster* telah bergabung menjadi satu *cluster* yang berisi semua dokumen.

Algoritma *bottom-up* ini kemudian disebut dengan *Hierarchical Agglomerative Cluster* (HAC). Sedangkan *cluster* top-down membutuhkan metode untuk memisahkan cluster. HAC lebih sering digunakan daripada algoritma *top-down cluster*.

Suatu HAC divisualisasikan dengan suatu grafik sebagai tree yang disebut dendogram. Dendrogram menggambarkan proses penggabungan dari setiap cluster-*cluster* yang ada. Cabang-cabang pada tree menyajikan cluster. Kemudian cabang-cabang bergabung pada node yang posisinya sepanjang sumbu jarak.

Kemiripan antar dokumen ditentukan dengan mengukur jarak antar dokumen. Dua dokumen yang mempunyai kemiripan paling tinggi dan dikelompokkan ke dalam satu *cluster* yang sama. Sebaliknya dua dokumen yang mempunyai jarak paling besar dikatakan mempunyai kemiripan yang paling rendah, dan dimasukkan ke dalam *cluster* yang berbeda.

Berikut merupakan algoritma *Hierarchical Agglomerative Clustering* untuk mengelompokkan N objek [1]:

i. Mulai dengan *N* cluster, setiap *cluster* mengandung entity tunggal dan sebuah matriks simetri dari jarak $D = \{d_{ik}\}$ dengan tipe matrik adalah $NxN$.
ii. Mencari matriks jarak untuk pasangan *cluster* yang terdekat, yaitu dengan mencari jarak terbesar. Misalkan jarak antara *cluster* $d_{ab}$.
iii. Gabungkan *cluster* A dan B menjadi label AB yang baru. Update inputan pada matrik jarak dengan cara:
    a. Hapus baris dan kolom yang bersesuaian *cluster* A dan B

b. Tambahkan baris dan kolom yang memberikan jarak-jarak antara *cluster* (AB) dan cluster-*cluster* yang tersisa

Beberapa metode *hierarchical clustering* yang sering digunakan yaitu *single linkage*, *complete linkage*, *average linkage, ward's linkage, centroid linkage*. Metode-metode tersebut dibedakan berdasarkan perhitungan tingkat kemiripan atau jarak antar kelompok.

### 2.7.1 *Average Linkage*

*Average linkage* merupakan proses pengelompokan dokumen berdasarkan pada jarak rata-rata antar objeknya. Average linkage memperlakukan jarak antara dua *cluster* sebagai jarak rata-rata antara semua pasangan item-item dimana satu anggota dari pasangan tersebut merupakan kelompok *cluster* tersebut. Proses *clustering* dimulai dengan mencari matriks jarak antara dua objek, $p$ dan $p^0$ untuk memperoleh objek-objek yang paling mirip. Untuk menghitung jarak antara dua objek menggunakan perssmaan (2.8) berikut [17]:

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i p^0 \in C_j} |p - p^0| \tag{2.8}$$

dengan:

| | |
|---|---|
| $dist_{avg}(C_i, C_j)$ | : rata-rata jarak antara cluster $i$ dengan cluster $j$ |
| $n_i$ | : banyaknya objek dalam cluster $i$ |
| $n_j$ | : banyaknya objek dalam cluster $j$ |
| $\|p - p^0\|$ | : jarak antara objek $p$ dan $p^0$ |

# BAB III
# METODE PENELITIAN

Pada bab ini terdapat penjelasan mengenai objek dan penelitian, peralatan penunjang, metode penelitian dan diagram alir penelitian.

## 3.1 Objek dan Aspek Penelitian

Pada penelitian ini, objek yang digunakan adalah *scrapping website* www.tokopedia.com dengan mengambil data ulasan yang diberikan oleh pelanggan. Sedangkan aspek penelitian ini adalah mengelompokan ulasan pelanggan pada suatu produk di *website* tersebut dengan algoritma *hierarchical agglomerative clustering.*

## 3.2 Peralatan Penunjang

Penelitian ini menggunakan perangkat keras dan perangkat lunak sebagai alat penunjang pengerjaan. Adapun perangkat keras dan perangkat lunak yang digunakan ditunjukkan pada Tabel 3.1 berikut:

Tabel 3. 1 Peralatan yang Digunakan

| | |
|---|---|
| Perangkat keras | 1. Laptop HP 14". HDD 1TB<br>2. Sistem Operasi Windows 10 Education 64 bit<br>3. Prosesor AMD A9-9425 Radeon R5. 3,1 GHz<br>4. RAM 8 GB |
| Perangkat Lunak | 1. Anaconda Navigator dengan aplikasi Jupyter Notebook 5.0.0 dan Spyder dengan bahasa pemrograman Python<br>2. Microsoft Office (Word,Excel) |

### 3.3 Tahapan Penelitian

Pada penelitian Tugas Akhir ini dilakukan penelitian berdasarkan langkah-langkah sebagai berikut:

a. Pengumpulan data

Pada tahap ini dilakukan penelitian data-data ulasan pelanggan Tokopedia terhadap suatu produk. Data ulasan tersebut diperoleh dari hasil *scraping* dari halaman situs *website* www.tokopedia.com. Hasil *scraping web* akan dirubah formatnya menjadi file .csv untuk dapat dilakukan tahap selanjutnya.

b. Praproses data

Pada tahap ini dilakukan praproses data yang akan dilakukan dalam berbagai langkah, yaitu:

i. Tokenisasi, dengan menggunakan data yang telah diperoleh dari proses sebelumnya, kata-kata yang diperoleh kemudian dirubah dalam bentuk huruf kecil. Pada tahap ini juga dihilangkan karakter numerik dan tanda baca.

ii. *Stopword removal,* menghilangkan kalimat, kata atau karakter yang kurang penting sehingga hasil *clustering* lebih baik.

iii. *Stemming,* menghilangkan awalan dan imbuhan, sehingga dihasilkan kata dasar

c. Pembuatan TF-IDF

Pembuatan TF-IDF memberikan bobot terhadap data yang dihasilkan diproses sebelumnya dengan menggunakan Persamaan (2.1) dan Persamaan (2.2). Langkah pertama yang harus dilakukan yaitu membuat skema TF-IDF. Dari proses tersebut dihasilkan suatu

matriks dokumen-kata atau *document by-term matrix* (DTM).

d. Identifikasi kemiripan dokumen

Identifikasi kemiripan dokumen merupakan salah satu kegiatan *feature selection*. Proses ini bertujuan untuk mengetahui adanya kemiripan atau hubungan antar dokumen. Tahap ini menggunakan metode LSA untuk memodelkan matriks dengan memanfaatkan matriks dari langkah (c). Dalam implementasinya, matriks DTM direduksi menjadi matriks $U\Sigma V^T$ , kemudian dilakukan *truncated SVD* dengan menggunakan nilai $k$ yang dikehendaki. Dari proses tersebut dihasilkan kata kunci untuk beberapa topik dokumen yang diinginkan.

e. Proses *clustering*

Tahap selanjutnya data akan dilakukan peosws *clustering* menggunakan algoritma *Hierarchical Agglomerative Clustering* dengan metode *average linkage* untuk mengelompokkan ulasan-ulasan yang memiliki kesamaan. Proses *clustering* dimulai dengan menghitung jarak antar dokumen dengan menggunakan Persamaan (2.7). Hasil perhitungan jarak kemudian digunakan oleh sistem untuk mencari kedekatan ulasan untuk dijadikan satu kelompok.

f. Pelabelan Hasil Cluster

Dalam menentukan label tiap *cluster* digunakan tiga *keyword* dengan nilai tf-idf tertinggi untuk dijadikan topik *cluster*.

g. Analisis hasil *clustering*

Pada tahap ini, akan dilakukan analisis mengenai hasil *clustering* untuk mengetahui keoptimalan nilai *threshold* yang digunakan.

h. Penarikan Kesimpulan

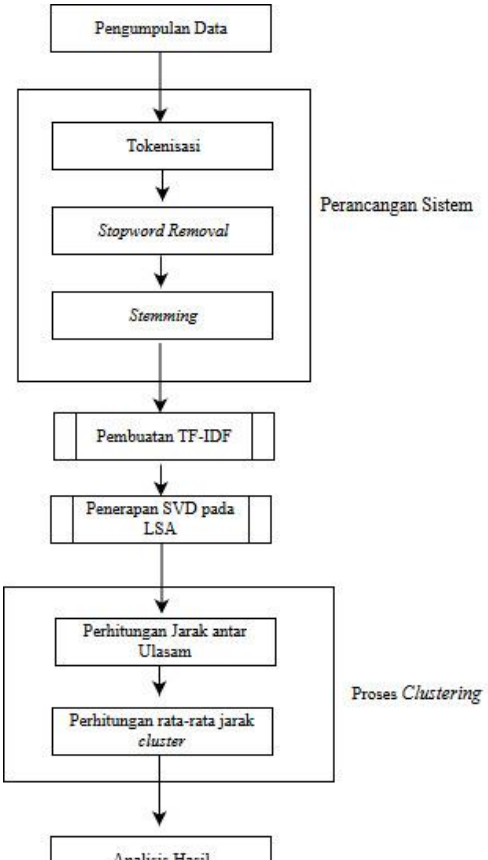
Pada tahap ini dilakukan penarikan kesimpulan dari penelitian yang telah dijalankan, yakni mengenai implementasi *text mining* dalam pengelompokan ulasan dan hasil analisis proses *clustering* yang diperoleh dari penelitian Tugas Akhir.

i. Pembuatan laporan Tugas Akhir

Setelah penelitian dilakukan dengan dihasilkan jawaban dari permasalahan yang diberikan, maka dibuatlah suatu laporan Tugas Akhir yang berisi tahapan, hasil, analisa dan kesimpulan dari penelitian.

### 3.4 Diagram Alir Penelitian

Diagram alir penelitian ini adalah sebagai berikut:



# BAB IV
# PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini dijelaskan tentang perancangan dan implementasi sistem yang digunakan pada Tugas Akhir ini. Penjelasan implementasi sistem berikut menggunakan data contoh untuk memudahkan dalam interpretasi.

## 4.1 Pengumpulan Data

Pada tahap awal perancangan sistem dilakukan pengumpulan data sebagai data masukan untuk tugas akhir ini. Data yang digunakan merupakan data ulasan suatu produk dengan tipe data berupa teks. Data diambil dengan menggunakan proses *scraping website* suatu situs *e-commerce* yang populer di Indonesia yaitu www.tokopedia.com. Pengambilan data menggunakan python dengan bantuan *library requests*. Setiap data mengandung atribut nama dan kalimat ulasan. Dataset yang terkumpul tersebut disimpan dalam bentuk *file csv* agar mudah dalam penyimpanan dan penggunaan. Agar dataset mudah digunakan, maka perlu dipisahkan antara atribut nama dan atribut kalimat ulasan. Untuk proses selanjutnya, data dengan atribut kalimat ulasan yang digunakan dalam sistem ini.

Adapun contoh ulasan yang diberikan pelanggan yang diambil dari ulasan Tokopedia adalah sebagai berikut:

Tabel 4. 1 Data Contoh Ulasan Pelanggan

| Index | Ulasan |
|---|---|
| 0 | pengiriman cepat dan barang sesuai pesanan |
| 1 | produk sih baik tapi respon penjual lama |
| 2 | aku produk nya d packing dgn rapi baik pengiriman dan respon cepat !!! |
| 3 | Barang recomended dan pengiriman cepaattt |
| 4 | Sesuai pesanan dan pengiriman tepat waktu |
| 5 | Packing rapi, pengiriman cepat dan barang sesuai pesanan |
| 6 | Cukup puas respon cepat, meskipun salah model rekomen packing rapi |

Data diatas digunakan sebagai data contoh untuk mempermudah dalam interpretasi proses berjalannya program Tugas Akhir ini. Terlihat dari data contoh tersebut, banyak data yang penulisannya tidak sesuai dengan ejaan bahasa Indonesia dengan benar.

**4.2 Praproses Data**

Setelah data terkumpul, terlebih dahulu dilakukan pra-proses data, karena data yang diperoleh dari proses sebelumnya belum bisa digunakan secara langsung. Hal itu dikarenakan masih terdapat penulisan ulasan yang tidak baku. Oleh sebab itu perlu dilakukan praproses data agar data dapat digunakan secara mudah untuk digunakan ke proses selanjutnya.

Pada tahap pra-proses data ini dilakukan beberapa tahap, yaitu:

a. *Load* ulasan ke software python

Proses ini bertujuan untuk membuka file yang disimpan dalam *.csv* ke python. Hal itu dilakukan karena dalam

pengolahan ulasan-ulasan tersebut menggunakan python. Berikut merupakan *syntax* yang digunakan untuk *load* data:

```
with open('coba.csv') as f:
    ulasan=f.readlines()
ulasan
```

b. *Cleaning Data*

*Cleaning data* dilakukan untuk menghilangkan karakter, simbol, atau angka-angka yang terdapat pada ulasan. Proses ini menggunakan *syntax* yang ada pada python. Adapun *syntax* yang diguakan adalah sebagai berikut:

```
def remove_punctuation(text):
    for punctuation in string.punctuation:
        text = text.replace(punctuation, "")
    return text.strip(' ')

#removes URLs
def removes_url(text):
    text =
re.sub(r'(?i)\b((?:https?://|www\d{0,3}[.]|[a-
z0-9.\-]+[.][a-
z]{2,4}/)(?:[^\s()<>]+|\(([^\s()<>]+|(\([^\s()
<>]+\)))*\))+(?:\(([^\s()<>]+|(\([^\s()<>]+\))
)*\)|[^\s`!()\[\]{};:\'".,<>?«»“”‘’]))', '',
text)
    return text.strip(' ')

#removes # and @ in the beginning of each
word. ex: #Good -> Good
def remove_hashtag(text):
    new_text = ""
    for words in text.split():
        if words.startswith('#') or
words.startswith('@'): #remove @ amd #
            new_text += words[1:]
            new_text += ' '
        else:
            new_text += words
            new_text += ' '
```

```
    return new_text.strip(' ')

#removes # and @ even between words. ex:
#life#is#good -> life is good
def remove_hash_symbol(text):
    to_be_removed = ['#', '@']
    for prohibited_symbol in to_be_removed:
        text = text.replace(prohibited_symbol,
' ')
    text =' '.join(text.split())
    return text.strip(' ')
```

c. Tokenisasi

Pada tahap tokenisasi ini dilakukan suatu proses pemecahan tiap kalimat ulasan menjadi kata-kata yang terpisah satu dengan yang lainnya. Pemisahan kata-kata tersebut berdasarkan spasi dari setiap kata. Sebelum dilakukan pemisahan kata, dilakukan terlebih dahulu perubahan semua kata ke dalam bentuk huruf kecil sebagai berikut:

```
def lowerCase(kalimat):
    lowCase=[]
    for kata in kalimat:
        kata=kata.lower()
        lowCase.append(kata)
    return lowCase
kataLow=[]
for row in test_result:
    kataLow.append(lowerCase(row))
    kalimat_kecil=lowerCase(test_result)
print(kalimat_kecil)
```

Kemudian dilakukan proses tokenisasi dengan menggunakan *syntax* sebagai berikut:

```
#proses tokenisasi
tokenize=[word_tokenize(i)for i in kalimat_kecil]
for i in tokenize:
    print(i)
```

Tabel 4.2 berikut menunjukkan perbandingan kalimat sebelum dan sesudah dilakukan proses tokenisasi. Hasil dari proses berupa kata-kata yang terpisah berdasarkan spasi.

Tabel 4. 2 Contoh Data Sebelum dan Sesudah Tokenisasi

| Sebelum | Sesudah |
|---|---|
| pengiriman cepat dan barang sesuai pesanan | pengiriman, cepat, dan, barang, sesuai, pesanan |
| produk sih baik tapi respon penjual lama | produk, sih, baik, tapi, respon, penjual, lama |
| aku produk nya d packing dgn rapi baik pengiriman dan respon cepat | aku, produk, nya, d, packing, dgn, rapi, baik, pengiriman, dan, respon, cepat |
| Barang recomended dan pengiriman cepaatt | barang, recomended, dan, pengiriman, cepaatt |
| Sesuai pesanan dan pengiriman tepat waktu | sesuai, pesanan, dan, pengiriman, tepat, waktu |
| Packing rapi, pengiriman cepat dan barang sesuai pesanan | packing, rapi, pengiriman, cepat, dan, barang, sesuai, pesanan |
| Cukup puas respon cepat, meskipun salah model rekomen packing rapi | Cukup, puas, respon, cepat, meskipun, salah, model, rekomen, packing, rapi |

d. *Stopword Removal*

Proses *stopword removal* menggunakan data hasil proses tokenisasi. Pada proses ini, dilakukan proses penghilangan kata-kata yang tidak penting seperti kata “yang”, “dan”, “dengan”, “agak”, “walaupun”, ”dari”.

Karena data ulasan yang dihasilkan menggunakan kata yang tidak baku, oleh karena itu dilakukan penggantian kata

yang tidak baku ke bentuk yang baku. Berikut merupakan proses perubahan kata yang tidak baku:

```
  d={}
  with open ("dict.txt") as text:
      for line in text:
          if line.strip():
              key, val=line.split(None, 1)
              d[key]=val.split()
  print (d.get('banget'))
  def mencaritypo(kata):
      for key in d:
          list1=d.get(key)
          if kata in list1:
              return key
      return kata
  def replacetypo (tupel):
      temp_data=[]
      for kalimat in tupel:
          temp_kalimat=[]
          for kata in kalimat:
              lit=mencaritypo(kata)
              temp_kalimat.append(lit)
          temp_data.append(temp_kalimat)
return temp_data
```

Kemudian dilakukan proses *stopword* dengan menggunakan *library* Sastrawi dengan *syntax* sebagai berikut:

```
from
Sastrawi.StopWordRemover.StopWordRemoverFactory
import  StopWordRemoverFactory,  StopWordRemover,
ArrayDictionary

# Ambil Stopword bawaan
stop_factory                                    =
StopWordRemoverFactory().get_stop_words()
more_stopword            =            ['diatur',
'perjodohan','lah','yang','dengan','nya']

# Merge stopword
data = stop_factory + more_stopword
dictionary = ArrayDictionary(data)
```

```
str = StopWordRemover(dictionary)

def stopWord(tupel):
    temp_data=[]
    for kalimat in tupel:
        temp_kalimat=[]
        for kata in kalimat:
            lit=str.remove(kata)
            temp_kalimat.append(lit)
        temp_data.append(temp_kalimat)
    return temp_data
```

Hasil dari *syntax* diatas akan ditunjukkan pada Tabel 4.3. Pada tabel tersebut ditunjukkan data sebelum dan sesudah proses stemming supaya terlihat perbedaannya.

Tabel 4. 3 Contoh Data Sebelum dan Sesudah *Stopword* Ulasan

| **Sebelum** | **Sesudah** |
|---|---|
| pengiriman, cepat, dan, barang, sesuai, pesanan | pengiriman, cepat, barang, sesuai, pesanan |
| produk, sih, baik, tapi, respon, penjual, lama | produk, baik, respon, penjual, lama |
| aku, produk, nya, d, packing, dgn, rapi, baik, pengiriman, dan, respon, cepat | produk, pengemasan, rapi, baik, pengiriman, respon, cepat |
| barang, recomended, dan, pengiriman, cepaatt | barang, rekomendasi, pengiriman, cepat |
| sesuai, pesanan, dan, pengiriman, tepat, waktu | sesuai, pesanan, pengiriman, tepat, waktu |
| packing, rapi, pengiriman, cepat, dan, barang, sesuai, pesanan | packing, rapi, pengiriman, cepat, barang, sesuai, pesanan |

| | |
|---|---|
| Cukup, puas, respon, cepat, meskipun, salah, model, rekomen, packing, rapi | Cukup, puas, respon, cepat, salah, model, rekomendasi, packing, rapi |

Data yang dihasilkan proses ini digunakan untuk proses *stemming.*

e. *Stemming*

Pada proses ini dilakukan perubahan setiap kata menjadi kata dasar. Proses ini menggunakan *library* Sastrawi, sehingga kata yang dihasilkan dari proses ini merupakan kata dasar yang ada di kamus bahasa Indonesia.

Sehingga diperoleh perbandingan hasil *stemming* pada Tabel 4.4 berikut:

Tabel 4. 4 Contoh Data Sebelum dan Sesudah *Stemming*

| **Sebelum** | **Sesudah** |
|---|---|
| pengiriman, cepat, barang, sesuai, pesanan | Kirim, cepat, barang, sesuai. pesan |
| produk, baik, respon, penjual, lama | produk, baik, respon, jual, lama |
| produk, packing, rapi, baik, pengiriman, respon, cepat | produk, kemas, rapi, baik, kirim, respon, cepat |
| barang, rekomendasi, pengiriman, cepat | barang, rekomendasi, kirim, cepat |
| sesuai, pesanan, pengiriman, tepat, waktu | sesuai, pesan, kirim, tepat, waktu |
| packing, rapi, pengiriman, cepat, barang, sesuai, pesanan | packing, rapi, kirim, cepat, barang, sesuai, pesan |

| | |
|---|---|
| Cukup, puas, respon, cepat, salah, model, rekomendasi, packing, rapi | Cukup, puas, respon, cepat, salah, model, recomendasi, packing, rapi |

Data contoh yang sudah dilakukan *preprocessing* data memudahkan untuk digunakan di proses selanjutnya.

### 4.3 Pembuatan TF-IDF

Pada proses ini dilakukan proses tranformasi dari kata ke matriks agar lebih mudah pengolahannya. Berdasarkan proses sebelumnya diperoleh kata-kata yang terpilih untuk dihitung bobotnya. Kata-kata tersebut adalah:

| |
|---|
| [‘baik’,’cepat’,’kemas’,’kirim’,’pesan’,’produk’,’rapi’, ’rekomendasi’,’respon’, dan ‘sesuai’] |

Pembuatan TF-IDF ini dimulai dengan pembuatan matriks *document-by-term* untuk menunjukkan frekuensi tiap kata pada setiap ulasan. Pembentukan matriks tersebut dilakukan dengan bantuan *CountVectorizer* dari *library sklearn.* Gambar 4.1 merupakan matriks *document-by-term* yang dihasilkan.

| | baik | cepat | cukup | jual | kirim | lama | model | packing | pesanan | produk | puas | rapi | rekomen | respon | salah | sesuai | tepat | waktu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 1 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 |
| 5 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 6 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 |

Gambar 4. 1 Matriks *Document-by-term*

Elemen matriks pada Gambar 4.1 adalah $A_{ij}$ dimana $i$ merepresentasikan indeks ulasan dan $j$ merupakan indeks term atau kata penting. Untuk mempermudah penjelasan perhitungan, digunakan ulasan pertama pada indeks-0 dalam contoh perhitungan tf-idf. Pada Tabel 4.5 terdapat *term* dan tf dari ulasan pertama.

Tabel 4. 5 Perhitungan TF pada Ulasan Pertama

| ***Term*** | ***Tf*** |
|---|---|
| Baik | 0 |
| Cepat | 1 |
| Kirim | 1 |
| Kemas | 0 |
| Pesan | 1 |
| Produk | 1 |
| Rapi | 0 |
| Rekomendasi | 0 |
| Respon | 0 |
| Sesuai | 1 |

Kemudian berdasarkan term yang dihasilkan maka diperoleh nilai df pada setiap term pada Tabel 4.6 berikut:

Tabel 4. 6 Hasil Perhitungan IDF

| ***Term*** | ***Df*** |
|---|---|
| Baik | 2 |
| Cepat | 5 |
| Kirim | 5 |
| Kemas | 3 |
| Pesan | 3 |
| Produk | 5 |
| Rapi | 3 |
| Rekomendasi | 2 |
| Respon | 3 |
| Sesuai | 3 |

Kemudian melakukan perhitungan IDF dari data pada Tabel 4.5 dan Tabel 4.6 dengan menggunakan Persamaan (2.2) yang terbentuk menjadi matriks sebagai berikut:

Tabel 4. 7 Hasil Perhitungan TF-IDF

| ***Term*** | ***Idf*** |
|---|---|
| Baik | 2,252763 |
| Cepat | 1,336472 |
| Kirim | 1,336472 |
| Kemas | 1,847298 |
| Pesan | 1,847298 |
| Produk | 1,336472 |
| Rapi | 1,847298 |
| Rekomendasi | 2,252763 |
| Respon | 1,847298 |
| Sesuai | 1,847298 |

Normalisasi bobot $w_{pra}$ dengan menggunakan Persamaan (2.3) untuk menghitung pembagi nilai bobot awal sebagai berikut:

$$\sum_{j=1}^{n} w_{i,j} = 0 + 1{,}336472 + 1{,}336472 + 0 + 1{,}847298 + 1{,}336472 + 0 + 0 + 0 + 1{,}847298 = 7{,}704012$$

Sehingga hasil TF-IDF setelah dilakukan normalisasi adalah sebagai berikut:

Tabel 4. 8 Hasil Pembobotan TF-IDF pada Ulasan Pertama

| ***Term*** | $\boldsymbol{tf - idf}$ | $\boldsymbol{w_{i,j(pra)}}$ |
|---|---|---|
| Baik | 0 | 0 |
| Cepat | 1,336472 | 0,173477 |
| Kirim | 1,336472 | 0,173477 |
| Kemas | 0 | 0 |
| Pesan | 1,847298 | 0,239784 |
| Produk | 1,336472 | 0,173477 |
| Rapi | 0 | 0 |
| Rekomendasi | 0 | 0 |
| Respon | 0 | 0 |
| Sesuai | 1,847298 | 0,239784 |

Perhitungan TF-IDF dapat dilakukan dengan python dengan *syntax* berikut:

```
from sklearn.feature_extraction.text import
TfidfVectorizer

vectorizer = TfidfVectorizer(
max_features= 1000, norm='l1', # keep top 1000
terms
min_df = 2.0,
smooth_idf=True)

X = vectorizer.fit_transform(hasilPreprocessing)
print(X)
X.shape
```

Untuk hasil TF-IDF secara keseluruhan dapat dilihat pada Gambar 4.2 berikut:

| | baik | cepat | kemas | kirim | pesan | produk | rapi | rekomendasi | respon | sesuai |
|---|---|---|---|---|---|---|---|---|---|---|
| **Ulasan 1** | 0.000000 | 0.173477 | 0.000000 | 0.173477 | 0.239784 | 0.173477 | 0.000000 | 0.000000 | 0.000000 | 0.239784 |
| **Ulasan 2** | 0.414375 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.245832 | 0.000000 | 0.000000 | 0.339793 | 0.000000 |
| **Ulasan 3** | 0.190846 | 0.113221 | 0.156497 | 0.113221 | 0.000000 | 0.113221 | 0.156497 | 0.000000 | 0.156497 | 0.000000 |
| **Ulasan 4** | 0.000000 | 0.213420 | 0.000000 | 0.213420 | 0.000000 | 0.213420 | 0.000000 | 0.359741 | 0.000000 | 0.000000 |
| **Ulasan 5** | 0.000000 | 0.000000 | 0.000000 | 0.265644 | 0.367178 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.367178 |
| **Ulasan 6** | 0.000000 | 0.117249 | 0.162063 | 0.117249 | 0.162063 | 0.117249 | 0.162063 | 0.000000 | 0.000000 | 0.162063 |
| **Ulasan 7** | 0.000000 | 0.146364 | 0.202308 | 0.000000 | 0.000000 | 0.000000 | 0.202308 | 0.246712 | 0.202308 | 0.000000 |

Gambar 4. 2 Hasil Pembobotan *Document-by-term*

## 4.4 Identifikasi Kemiripan Kata

Proses selanjutnya yaitu mengidentifikasi kata-kata yang mirip untuk dapat dijadikan pemodelan topik. Proses ini menggunakan teorema *latent semantic analysis*. Dalam melakukan pemodelan topik pada ulasan tersebut, matriks yang disahilkan dari pross TF-IDF diuraikan dengan *Singular Value Decomposition* menjadi matriks $U, \Sigma, V^T$. $U$ merupakan vektor kata penting, Σ merupakan vektor sigma dan $V^T$ merupakan vektor dokumen. Asumsikan matriks pada Gambar 4.4 merupakan matriks $A_{7x10}$, sehingga transpose matriks A

memiliki orde $10\ x\ 7$. Dengan menggunakan Persamaan (2.4), $B = AA^T$, diperoleh matriks $B$ sebagai berikut:

$$B = \begin{bmatrix} 0.21 & 0.04 & 0.06 & 0.11 & 0.22 & 0.14 & 0.03 \\ 0.04 & 0.35 & 0.16 & 0.05 & 0.0 & 0.03 & 0.07 \\ 0.06 & 0.16 & 0.15 & 0.07 & 0.03 & 0.09 & 0.11 \\ 0.11 & 0.05 & 0.07 & 0.27 & 0.06 & 0.08 & 0.12 \\ 0.22 & 0.0 & 0.03 & 0.06 & 0.34 & 0.15 & 0.0 \\ 0.14 & 0.03 & 0.09 & 0.08 & 0.15 & 0.15 & 0.08 \\ 0.03 & 0.07 & 0.11 & 0.12 & 0.0 & 0.08 & 0.21 \end{bmatrix}$$

Matriks $B$ dapat dicari nilai eigennya dengan menggunakan rumus $\det(B - \lambda\, I)$. Sehingga nilai eigen dari matriks $B$ adalah tersebut digunakan untuk mendapatkan nilai singular, $\sigma_i = \sqrt{\lambda_i}$, dengan $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_7$ yang disuusn secara diagonal. Sehingga nilai Σ dari matriks $B$ adalah sebagai berikut:

$$\Sigma = \begin{bmatrix} 0.93 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.82 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.72 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.61 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.47 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.36 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 \end{bmatrix}$$

Kemudian mencari matriks singular kanan dan singular kiri dengan bantuan python, sehingga hasil perhitungannya adalah sebagai berikut:

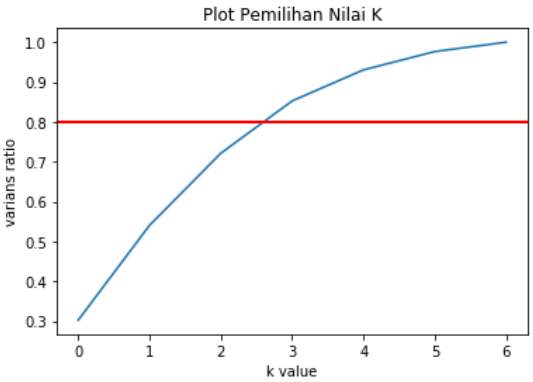
$$U = \begin{bmatrix} -0.46 & 0.28 & 0.08 & -0.18 & 0.33 & 0.58 & -0.48 \\ -0.31 & -0.62 & 0.59 & -0.25 & -0.18 & 0.18 & 0.22 \\ -0.31 & -0.33 & 0.07 & 0.29 & 0.37 & -0.60 & -0.46 \\ -0.38 & -0.13 & -0.6 & -0.64 & -0,01 & -0.22 & 0.14 \\ -0.48 & 0.54 & 0.28 & 0.05 & -0.53 & -0.34 & 0.03 \\ -0.38 & 0.12 & -0.05 & 0.38 & 0.46 & 0.07 & 0.69 \\ -0.27 & -0.32 & -0.46 & 0.52 & -0.48 & 0.32 & -0.13 \end{bmatrix}$$

Dan matriks singular kanan adalah sebagai berikut:

$$V^T = \begin{bmatrix} -0.22 & -0.33 & -0.19 & -0.43 & -0.41 & -0.37 & -0.19 & -0.24 & -0.24 & -0.41 \\ -0.48 & -0.07 & -0.14 & 0.21 & 0.42 & -0.23 & -0.14 & -0.18 & -0.49 & 0.42 \\ 0.5 & -0.35 & -0.18 & -0.07 & 0.22 & 0.06 & -0.18 & -0.64 & 0.23 & 0.22 \\ -0.13 & -0.04 & 0.57 & -0.21 & 0.1 & -0.41 & 0.57 & -0.28 & 0.18 & 0.1 \\ -0.22 & 0.37 & 0.16 & 0.05 & -0.18 & 0.49 & 0.16 & -0.55 & -0.45 & -0.18 \\ -0.31 & 0.32 & -0.13 & -0.73 & 0.21 & 0.29 & -0.13 & 0.0 & 0.24 & 0.21 \\ 0.05 & -0.64 & 0.21 & -0.26 & 0.1 & 0.45 & -0.21 & 0.29 & -0.35 & 0.1 \\ 0.5 & 0.28 & -0.37 & -0.28 & -0.02 & -0.28 & 0.37 & 0.17 & -0.41 & 0.22 \\ -0.31 & -0.17 & -0.6 & 0.17 & -0.01 & 0.17 & 0.6 & -0.1 & 0.25 & -0.11 \\ -0.11 & -0.06 & 0.02 & 0.06 & -0.72 & 0.06 & -0.02 & -0.04 & 0.09 & 0.67 \end{bmatrix}$$

Setelah matriks $A$ didekomposisi menjadi matriks-matriks tersebut, maka dilakukan *truncated SVD* dengan memilih $k$ komponen dengan $1 \leq k \leq 7$. Hasil *truncated SVD* merupakan nilai *principal components.*

Dalam penelitian ini, jumlah *principal component* memenuhi rasio pada Persamaan (2.6) yang nilainya lebih dari 0.8. Pengambilan jumlah $k$ dapat melihat grafik Gambar 4.3. Berdasarkan grafik tersebut nilai $k$ yang memenuhi adalah 3. Pada grafik tersebut dapat diketahui bahwa dengan menggunakan *threshold* 0.8 ditemukan interval nilai *principal components* sebesar 3.




Gambar 4. 3 Grafik Penentuan Jumlah *k*

Untuk data yang kecil bisa dilihat dengan menggunakan grafik, namun untuk data yang banyak jumlahnya sulit untuk ditentukan nilai *k* yang sesuai. Oleh karena itu, dengan menggunakan *syntax* berikut ditemukan jumlah *principal components.*

```
sum_s=np.sum(s)
    cumsum_s=np.cumsum(s)
    plot_var=cumsum_s/sum_s
    for i,v in enumerate (plot_var):
        if v>=rasio:
            trunc=i
         break
```

Dengan menggunakan *syntax* diatas dapat diketahui jumlah *principal components* dengan threshold 0.8 adalah 3. Kemudian untuk memperoleh matriks *principal components* dapat menggunakan *syntax* dibawah ini.

```
svd_model = TruncatedSVD(n_components= 3,
algorithm='randomized')
svdfix=svd_model.fit_transform(tfIdf)
```

Dengan menggunakan *syntax* diatas dapat dihasilkan *principal components matrix* sebagai berikut:

| | 0 | 1 | 2 |
|---|---|---|---|
| **0** | 0.39 | -0.19 | -0.04 |
| **1** | 0.27 | 0.42 | -0.30 |
| **2** | 0.27 | 0.22 | -0.04 |
| **3** | 0.33 | 0.09 | 0.31 |
| **4** | 0.41 | -0.37 | -0.14 |
| **5** | 0.33 | -0.08 | 0.03 |
| **6** | 0.24 | 0.21 | 0.23 |

Gambar 4. 4 *Principal Components Matrix*

## 4.5 Tahap Clustering

Pada tahap *clustering* ini terdapat dua tahapan, yaitu proses *clustering* dan pelabelan. Proses *clustering* merupakan proses klasterisasi ulasan untuk memperoleh kelompok ulasan berdasarkan topiknya. Kemudian dilakukan pelabelan untuk mengetahui topik pada setiap *cluster*.

### 4.5.1 Proses *Clustering*

Tahap *cluster* digunakan untuk mengelompokkan ulasan yang membahas topik yang sama dengan menggunakan metode *Average Linkage* pada *Hierarchical Agglomerative Clustering*. Matriks *principal components* yang telah dihasilkan dijadikan atribut pada tahap ini. Pada metode ini, jumlah clusternya tidak

ditentukan sebelumnya. Agar mudah dalam menganalisa, hasil *clustering* harus dipotong pada suatu titik. Pemotongan dilakukan dengan menentukan nilai *threshold* yang merupakan angka kemiripan yang dimiliki oleh dua objek.

Adapun langkah – langkah klusterisasi ulasan tersebut adalah:

1. Asumsikan setiap ulasan sebagai cluster
2. Hitung jarak antar *cluster* dengan menggunakan *cosine similarity* sesuai pada Persamaan (2.6) dan Persamaan (2.7). Adapun contoh perhitungan *cosine similarity* pada ulasan 1 dan ulasan 2 adalah sebagai berikut:

$$\cos(d_1, d_2) = (0.39\ x\ 0.27) + (-0.19x0.42) + (-0.04\ x - 0.3) + (0.07\ x0.09)\ /$$
$$\left(\sqrt{0.39^2 + (-0.19)^2 + (-0.04)^2 + 0.07^2}\right)$$
$$\left(\sqrt{0.27^2 + (0.42)^2 + (-0.3)^2 + (0.09)^2}\right)$$
$$= 0.171306$$

$$dis(d_1, d_2) = 1 - 0.171306$$
$$= 0{,}828693$$

Sehingga dengan bantuan python dapat dihitung jarak tiap ulasan dengan menggunakan *syntax* berikut:

```
from sklearn.metrics.pairwise import
cosine_distances
pd.DataFrame(cosine_distances(svdfix))
```

Dengan menggunakan *syntax* tersebut diperoleh jarak tiap ulasan yang ditunjukkan pada Gambar 4.4 berikut:

| | Ulasan 1 | Ulasan 2 | Ulasan 3 | Ulasan 4 | Ulasan 5 | Ulasan 6 | Ulasan 7 |
|---|---|---|---|---|---|---|---|
| **Ulasan 1** | 0.000000 | 0.828693 | 0.641031 | 0.497543 | 0.065093 | 0.178218 | 0.846086 |
| **Ulasan 2** | 0.828693 | 0.000000 | 0.232864 | 0.825093 | 1.010735 | 0.851127 | 0.752435 |
| **Ulasan 3** | 0.641031 | 0.232864 | 0.000000 | 0.626173 | 0.827064 | 0.371737 | 0.234861 |
| **Ulasan 4** | 0.497543 | 0.825093 | 0.626173 | 0.000000 | 0.811865 | 0.601094 | 0.463396 |
| **Ulasan 5** | 0.065093 | 1.010735 | 0.827064 | 0.811865 | 0.000000 | 0.220039 | 1.041812 |
| **Ulasan 6** | 0.178218 | 0.851127 | 0.371737 | 0.601094 | 0.220039 | 0.000000 | 0.415945 |
| **Ulasan 7** | 0.846086 | 0.752435 | 0.234861 | 0.463396 | 1.041812 | 0.415945 | 0.000000 |

Gambar 4. 5 Jarak Tiap Ulasan

3. Hitung rata-rata jarak *cluster* yang dihasilkan dengan menggunakan metode *Average Linkage.* Tahap *clustering* diawali dengan mencari jarak terkecil. Pada Gambar 4.4 dapat diketahui jarak terkecil yaitu 0,065093. Jarak tersebut merupakan jarak ulasan dengan indeks 0 dan 4. Maka data dengan indeks 0 dan 4 dijadikan objek baru dengan indeks 7 yang dapat disimbolkan dengan $d_{04}$. Sehingga objeknya menjadi 1,2,3,5,6,7. Kemudian setiap objek tersebut dihitung jaraknya. Untuk menghitung jarak objek 7 dengan objek lain dengan dicontohkan perhitungan jarak antara objek 5 dan 7 adalah sebagai berikut. Didefinisikan objek 7 dengan $d_{(04)}$ dan jarak objek 5 dan 7 adalah $d_{(04)5}$, sehingga

$$d_{(04)5} = \frac{d_{05} + d_{45}}{2} = \frac{0{,}178218 + 0.220039}{2} = 0.199129$$

Selanjutnya jarak tersebut dibandingkan dengan jarak antar dua objek lain yang kemudian diambil jarak minimum untuk

dijadikan satu cluster, begitupun seterusnya sampai semua ulasan membentuk satu cluster. Dalam hal ini, perhitungan rata-rata jarak *cluster* menggunakan python dengan *syntax* sebagai berikut:

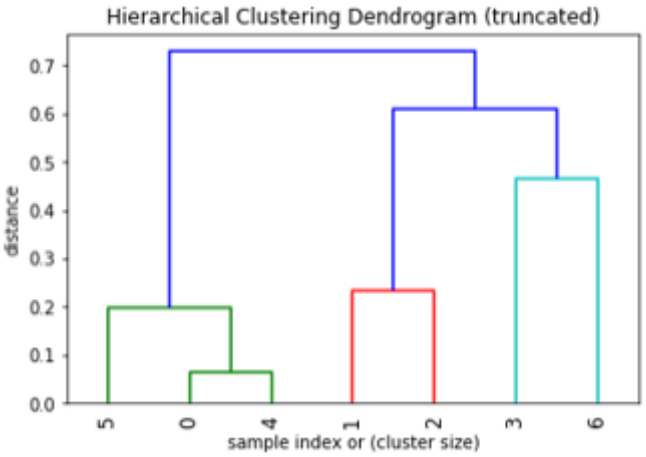
```
from scipy.cluster.hierarchy import
dendrogram, linkage
from scipy.cluster.hierarchy import
average, fcluster

Z=linkage(dist,'average')
```

Dengan menggunakan *syntax* diatas diperoleh hasil pada Gambar 4.5. Pada Gambar 4.5 terdapat empat kolom, dimana kolom 0 dan 1 merupakan kolom cluster, kolom 2 adalah rata-rata jarak dari *cluster* pada kolom 1 dan 2, dan kolom 3 merupakan banyak *cluster* yang digabungkan pada satu objek.

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| **0** | 0.0 | 4.0 | 0.065093 | 2.0 |
| **1** | 5.0 | 7.0 | 0.199129 | 3.0 |
| **2** | 1.0 | 2.0 | 0.232864 | 2.0 |
| **3** | 3.0 | 6.0 | 0.463396 | 2.0 |
| **4** | 9.0 | 10.0 | 0.609640 | 4.0 |
| **5** | 8.0 | 11.0 | 0.728728 | 7.0 |

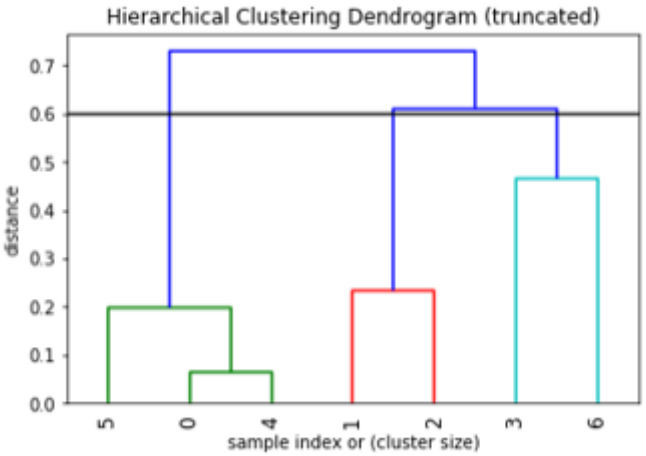
Gambar 4. 6 Rata-rata Jarak Tiap Cluster

Hasil *clustering* dapat dilihat pada gambar *dendrogram* yang ditunjukkan pada Gambar 4.7




Gambar 4. 7 Dendrogram Hasil *Cluster* Ulasan

4. Pemotongan *cluster* dengan suatu nilai *threshold* sebagai ambang batas untuk memecah hasil *clustering* menjadi beberapa *cluster*. Dalam hal ini digunakan nilai *threshold* 0.6. Pemotongan jarak *cluster* pada titik 0.6 dapat dilihat pada dendrogram pada Gambar 4.8.




Gambar 4. 8 Dendrogram Hasil *Clustering* dengan *Treshold* 0.6

Pada Gambar 4.8 diatas dapat diketahui bahwa ulasan tersebut terbagi menjadi empat *cluster*, diantaranya *cluster* pertama digambarkan dengan garis berwarna hijau, *cluster* kedua dengan garis warna merah, *cluster* ketiga dan keempat digambarkan dengan warna biru. Untuk lebih jelasnnya, hasil *clustering* dapat dilihat pada Tabel 4.9 berikut:

Tabel 4. 9 Hasil *Clustering*

| Ulasan | Cluster |
|---|---|
| pengiriman cepat dan barang sesuai pesanan | 1 |
| produk sih baik tapi respon penjual lama | 2 |
| aku produk nya d packing dgn rapi baik pengiriman dan respon cepat | 2 |
| Barang recomended dan pengiriman cepaatt | 3 |
| Sesuai pesanan dan pengiriman tepat waktu | 1 |
| Packing rapi, pengiriman cepat dan barang sesuai pesanan | 1 |
| Cukup puas respon cepat, meskipun salah model rekomen packing rapi | 3 |

### 4.5.2 Pelabelan

Proses pelabelan ini digunakan untuk mengetahui topik dari setiap *cluster*. Label yang digunakan dari setiap *cluster* diperoleh dari *keyword* yang ada pada cluster. *Keyword* tersebut diambil dari tiga kata pada setiap *cluster* yang memiliki tf-idf tertinggi. Pengambilan tiga kata tersebut didasarkan pada percoban dengan pengambilan *keyword* dua, tiga dan empat kata dengan diperoleh hasil yang baik dengan menggunakan tiga kata. Proses perhitungan tf-idf menggunakan bigram untuk mengetahui pasangan kata yang dimaksud. Sehingga, *keyword* dari hasil *clustering* yang dihasilkan adalah sebagai berikut:

Tabel 4. 10 *Keyword* Dari Setiap Cluster

| Cluster | Banyak Ulasan | Keyword |
|---|---|---|
| 1 | 3 | ‘pesan’,’sesuai pesan’,’sesuai’ |
| 2 | 2 | ‘baik’,’respon’,’produk’ |
| 3 | 2 | ‘cepat’,’rekomendasi’,’kirim cepat’ |

Berdasarkan *keyword* yang terdapat pada Tabel 4.10 dapat ditentukan topik. Sehingga topik ulasan pada tiap cluster dapat dilihat pada Tabel 4.11 berikut:

Tabel 4. 11 Topik Ulasan Data Contoh

| Cluster | Keyword | Topik |
|---|---|---|
| 1 | ‘pesan’,’sesuai pesan’,’sesuai’ | Kesesuaian dengan pesanan |
| 2 | ‘baik’,’respon’,’produk’ | Respon penjual baik |
| 3 | ‘cepat’,’rekomendasi’,’kirim cepat’ | Pengiriman cepat dan rekomendasi |

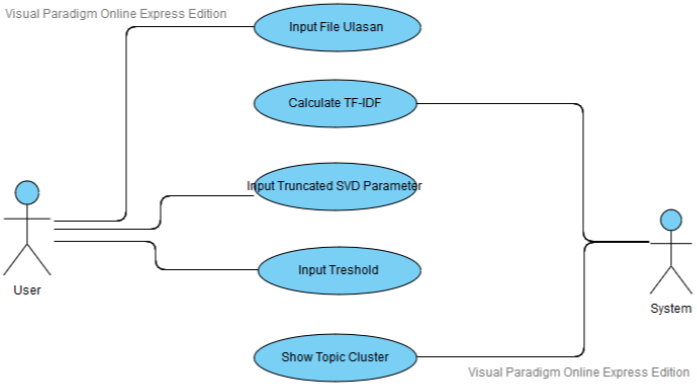
### 4.6 Analisis Sistem

Selanjutnya, sistem diimplementasikan dalam suatu program. Program yang dibuat berupa perangkat lunak yang digambarkan dalam *use case diagram* dan *activity diagram.*

#### 4.6.1 Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan kegiatan atau interaksi antara actor (pengguna) dan sistem. Sistem dimulai ketika pengguna

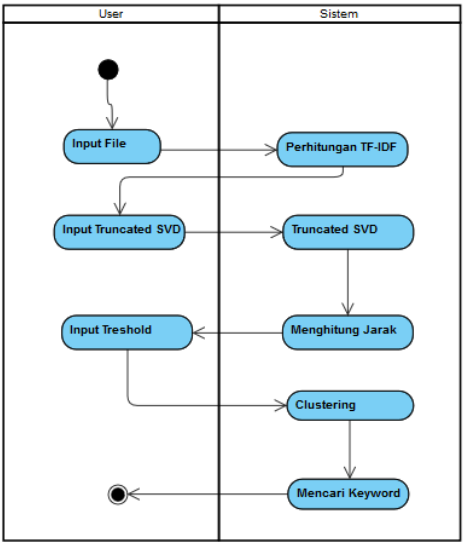
memasukkan file sebagai input. Implementasi sistem dapat dilihat pada Gambar 4.9 berikut:




Gambar 4. 9 *Use Case* Sistem

### 4.6.2 *Activity Diagram*

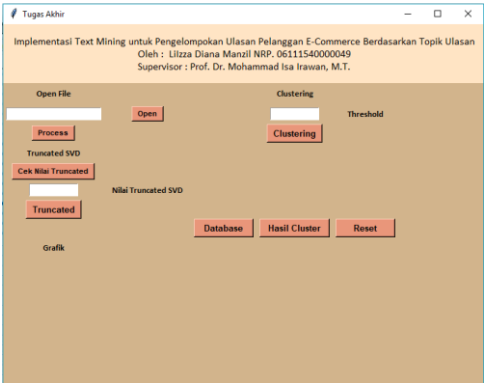
Proses sistem ini digambarkan dengan *activity diagram* pada gambar 4.10 dibawah ini:




Gambar 4. 10 *Activity Diagram*

## 4.7 Tampilan *Interface* GUI

Implementasi program menggunakan GUI untuk mempermudahkan jalannya sistem. Tampilan GUI ketika program dijalankan dapat dilihat pada Gambar 4.11.




Gambar 4. 11 Tampilan GUI Tab File

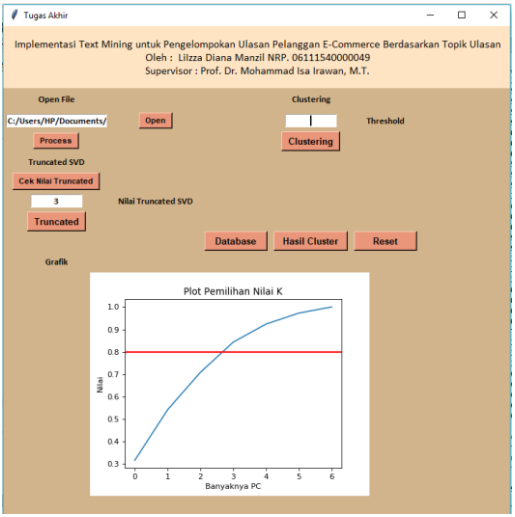
Berdasarkan Gambar 4.11 terdapat beberapa masukkan dan *task* menjalankan program. Beberapa komponen yang ada pada GUI tersebut diantaranya:

a. Open File

Pada komponen ini terdapat *entry* dan *button open* yang berfungsi untuk mencari file data dari *directory*. File yang dijadikan input merupakan file dengan format *.csv* yang mempunyai satu kolom yang bernama "text". Selain *button open*, pada komponen ini juga terdapat *button Process* yang berfungsi untuk melakukan pra-proses data dan menghitung TF-IDF sebagai bobot kata pada ulasan.

b. *Truncated SVD*

*Truncated SVD* merupakan komponen yang digunakan untuk proses Truncated SVD. Pada komponen ini terdapat *button Cek Nilai Treshold*, *entry* nilai truncated SVD yang digunakan, *button Truncated.*




Gambar 4. 12 Tampilan GUI Saat Menampilkan Plot

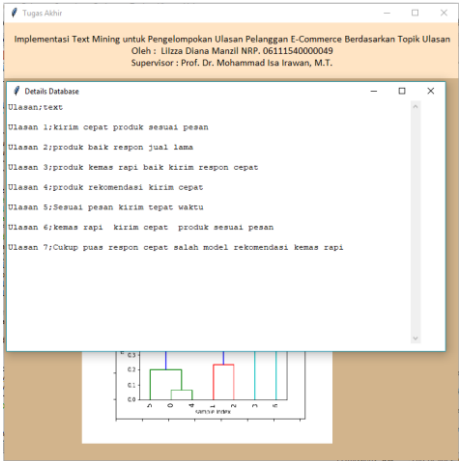
*Button Cek Nilai Treshold* menjalankan perintah untuk menampilkan plot Truncated SVD pada komponen Grafik dengan pereduksian SVD sebesar 0.8, sedangkan *button Truncated* melakukan proses pereduksian matriks sebesar nilai yang telah diinputkan pada *entry* nilai truncated. Tampilan GUI saat program menampilkan plot dapat dilihat pada Gambar 4.12.

c. *Threshold*

Pada komponen ini terdapat *entry* untuk menginputkan nilai *threshold* yang digunakan untuk melakukan *cluster* dengan menekan *button cluster.*

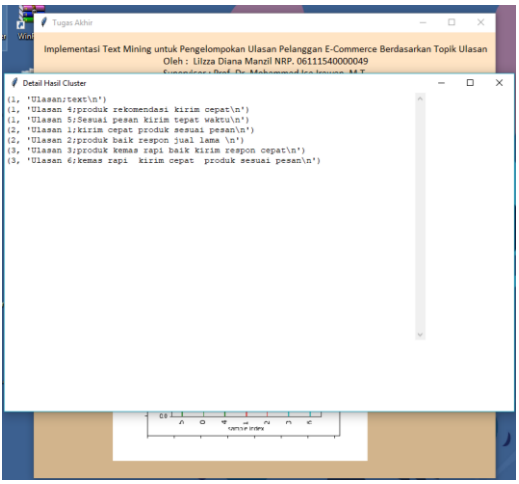
d. Lain-lain

Terdapat beberapa *button* yang tidak termasuk dalam komponen-komponen diatas, diantaranya *button Database, Hasil Cluster, Keyword* dan *Reset. Button* Database akan melihat file data yang dibuka melalui GUI tersebut. Tampilan GUI saat menampilkan file data pada Gambar 4.13



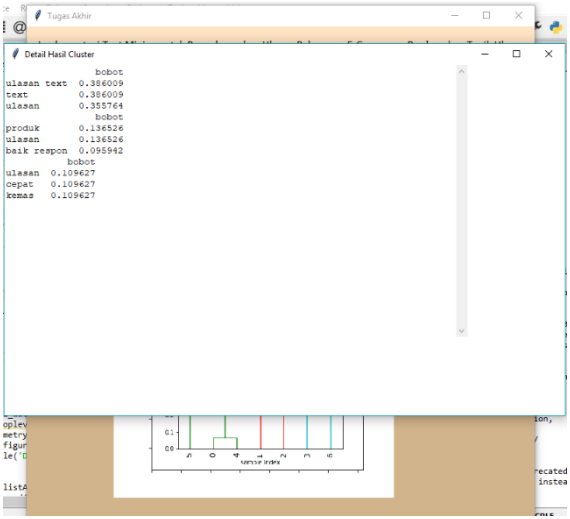

Gambar 4. 13 Tampilan GUI Menampilkan File Data

Kemudian *button Hasil Cluster* menampilkan hasil *clustering*, dimana pada panel tersebut terdapat kalimat ulasan dan angka yang mendefinisikan nomer *cluster* ulasan tersebut. Pada Gambar 4.14 dapat dilihat tampilan GUI untuk menampilkan panel jika *button hasil cluster* diklik yang dihasilkan.




Gambar 4. 14 Tampilan GUI Menampilkan Hasil *Clustering*

Pada *button keyword* GUI melakukan perintah untuk memunculkan tiga *keyword* dan bobot-bobotnya yang dihasilkan setiap *cluster*. Tampilan tersebut dapat dilihat pada Gambar 4.15 berikut:




Gambar 4. 15 Tampilan GUI Menampilkan *Keyword* Tiap *Cluster*

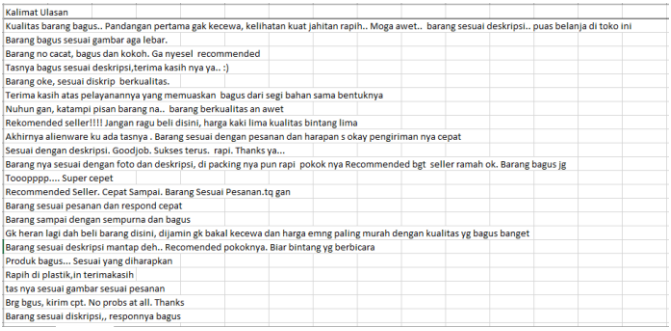
Sedangkan untuk *button reset* menampilkan GUI seperti saat pertama kali dibuka. Pada *button reset* GUI melakukan perintah untuk menghapus semua perintah dan *entry* yang telah diinputkan.

# BAB V
# Hasil dan Pembahasan

Pada bab ini dijelaskan dijelaskan mengenai uji coba sistem yang telah dibuat. Setelah itu dilakukan analisa mengenai hasil yang telah diperoleh dari uji coba.

## 5.1 Deskripsi Data

Pada uji coba ini data yang digunakan merupakan data ulasan suatu produk yang ada di www.tokopedia.com. Data diambil dengan cara *scraping* sebagaimana yang telah dijelaskan pada bab sebelumnya. Jumlah data yang diambil adalah 2501 ulasan dari produk tas. Ulasan-ulasan tersebut disimpan dalam bentuk *.csv*. Kutipan ulasan tersebut dapat dilihat pada Gambar 5.1 berikut:



| Kalimat Ulasan |
|---|
| Kualitas barang bagus.. Pandangan pertama gak kecewa, kelihatan kuat jahitan rapih.. Moga awet.. barang sesuai deskripsi.. puas belanja di toko ini |
| Barang bagus sesuai gambar aga lebar. |
| Barang no cacat, bagus dan kokoh. Ga nyesel recommended |
| Tasnya bagus sesuai deskripsi,terima kasih nya ya.. :) |
| Barang oke, sesuai diskrip berkualitas. |
| Terima kasih atas pelayanannya yang memuaskan bagus dari segi bahan sama bentuknya |
| Nuhun gan, katampi pisan barang na.. barang berkualitas an awet |
| Rekomended seller!!!! Jangan ragu beli disini, harga kaki lima kualitas bintang lima |
| Akhirnya alienware ku ada tasnya . Barang sesuai dengan pesanan dan harapan s okay pengiriman nya cepat |
| Sesuai dengan deskripsi. Goodjob. Sukses terus. rapi. Thanks ya... |
| Barang nya sesuai dengan foto dan deskripsi, di packing nya pun rapi pokok nya Recommended bgt seller ramah ok. Barang bagus jg |
| Tooopppp.... Super cepet |
| Recommended Seller. Cepat Sampai. Barang Sesuai Pesanan.tq gan |
| Barang sesuai pesanan dan respond cepat |
| Barang sampai dengan sempurna dan bagus |
| Gk heran lagi dah beli barang disini, dijamin gk bakal kecewa dan harga emng paling murah dengan kualitas yg bagus banget |
| Barang sesuai deskripsi mantap deh.. Recomended pokoknya. Biar bintang yg berbicara |
| Produk bagus... Sesuai yang diharapkan |
| Rapih di plastik,in terimakasih |
| tas nya sesuai gambar sesuai pesanan |
| Brg bgus, kirim cpt. No probs at all. Thanks |
| Barang sesuai diskripsi,, responnya bagus |

Gambar 5. 1 Kutipan Ulasan Hasil *Scraping*

## 5.2 Uji Coba Program

Data yang digunakan pada uji coba program merupakan data sebenarnya yang telah melalui tahap praproses. Tahap praproses yang telah dijelaskan di bab sebelumnya. Data hasil praproses disimpan dalam format *.csv*.

### 5.2.1 *Load* Data

Data ulasan yang telah dilakukan praproses data disimpan dalam direktori dengan file *dataTA.csv*. Hasil *load* data dapat dilihat pada Gambar 5.2 berikut:

| | Ulasan |
|---|---|
| 0 | bahan tas lumayan tebal\n |
| 1 | respon jual cepat kirim kilat banget kualitas ... |
| 2 | barang terima sesuai\n |
| 3 | layan sangat cepat packing rapi barang kondis... |
| 4 | barang bagus kirim cepat\n |
| 5 | barang terima cukup bagus cuma jahit kurang ra... |
| 6 | barang bagus ukur tepat ukur lebih besar tulis\n |
| 7 | cepat biasa bagus produk\n |
| 8 | tas udah walaupun kurir salah kirim alhamduli... |
| 9 | barang kece harga oke\n |
| 10 | barang kirim cepat trimakasih\n |
| 11 | bagus sukaa banget\n |
| 12 | tas bgus barang selamat tuju\n |
| 13 | barang terima bagus\n |
| 14 | terima kasih teirma kasih\n |
| 15 | sesuai harga\n |
| 16 | sesuai deskripsi gambar proses respon cepat\n |
| 17 | barang sesuai gambar\n |
| 18 | lumayan sesuai harga\n |

Gambar 5. 2 Load Data

### 5.2.2 Hasil TF-IDF

Setelah data berhasil diinputkan ke program, tahap selanjutnya yaitu menghitung TF-IDF. Tahap ini diawali dengan perhitungan kata pada satu dokumen. Pada perhitungan ini, digunakan minimal kemunculan kata sebanyak dua kali, sehingga untuk kata yang hanya muncul satu kali dalam data tidak diperhitungkan.

Hasil perhitungan matriks ini yang digunakan sebagai input di tahap selanjutnya. Kutipan hasil perhitungan TF-IDF dapat dilihat sebagai berikut:

| | aamiin | abis | abu | acung | ada | adik | admin | agak | agar | air | ... | yak | yakin | yalah | yang | yaw | yes | ykk | yo | you | ysampaie |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 11 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 13 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |
| 14 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 |

Gambar 5. 3 Kutipan Matriks Hasil Perhitungan TF-IDF

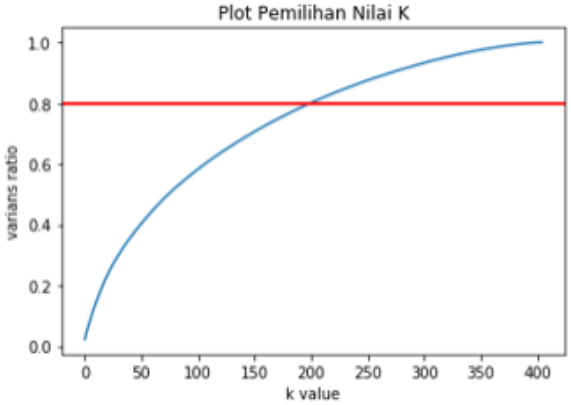
Dari Gambar 5.3 terdapat matriks dengan elemen $A_{i,j}$ dimana $i$ merupakan indeks untuk baris dan $j$ merupakan indeks kolom. Indeks baris menunjukkan indeks ulasan dan indeks kolom menunjukkan term atau kata.

### 5.2.3 Penentuan Jumlah *Principal Components*

Pada tahap selanjutnya dilakukan pemotongan matriks menjadi matriks *principal components*. Jumlah *principal components* ditentukan dengan menggunakan 0.8 sebagai titik pemilihan nilai *k* untuk *Truncated SVD.* Dengan melihat grafik pada Gambar 5.4 dapat diketahui interval jumlah *principal components.* Pada grafik tersebut dapat diketahui bahwa dengan menggunakan threshold 0.8 ditemukan interval nilai *principal components* 200 sampai 450, namun tidak diketahui angka

pastinya. Oleh karena itu, dengan menggunakan *syntax* berikut ditemukan jumlah *principal components.*

Dengan menggunakan *syntax* pada bab sebelumnya dapat diketahui jumlah *principal components* dengan threshold 0.8 200.




Gambar 5. 4 Grafik Nilai Singular

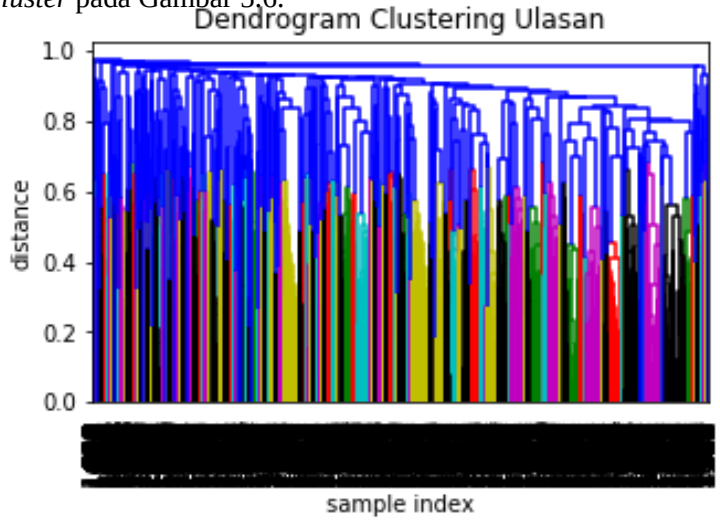
Nilai *principal components* dapat dilihat dalam skala kecil pada Gambar 5.5 berikut:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 456 | 457 | 458 | 459 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.150070 | 0.093258 | 0.088245 | -0.020949 | 0.322140 | -0.323111 | -0.227515 | -0.308825 | -0.064094 | -0.182707 | ... | 0.001008 | -0.000116 | -0.002781 | 0.005417 | - |
| 1 | 0.374674 | 0.243238 | 0.173168 | 0.033880 | -0.212082 | 0.042460 | 0.108952 | -0.039456 | 0.112370 | -0.140999 | ... | 0.004831 | 0.005384 | -0.002422 | -0.000642 | - |
| 2 | 0.133842 | 0.012491 | 0.082204 | 0.000592 | 0.035208 | 0.224360 | 0.033125 | -0.016082 | 0.093563 | -0.157672 | ... | 0.000564 | 0.004001 | -0.004383 | 0.005961 | - |
| 3 | 0.298512 | 0.013209 | 0.080277 | 0.050826 | -0.135071 | -0.091250 | -0.055875 | 0.035118 | 0.036829 | -0.139146 | ... | -0.009186 | -0.003036 | 0.004191 | -0.003624 | |
| 4 | 0.325095 | -0.281517 | 0.091983 | -0.041168 | 0.085759 | 0.222072 | -0.169401 | -0.027239 | 0.112658 | 0.293085 | ... | -0.004279 | -0.001988 | -0.003954 | -0.005048 | - |
| 5 | 0.132594 | -0.092929 | -0.084830 | 0.207767 | 0.263974 | -0.093472 | 0.195475 | 0.126765 | 0.190391 | -0.016571 | ... | -0.000190 | -0.000368 | 0.016083 | -0.002249 | - |
| 6 | 0.009777 | -0.003938 | -0.000519 | -0.009349 | 0.004663 | -0.006821 | 0.009178 | 0.003445 | -0.002160 | -0.005550 | ... | 0.001061 | -0.002046 | 0.007564 | 0.007575 | - |
| 7 | 0.526068 | -0.411902 | 0.282672 | 0.118366 | -0.362935 | -0.262392 | -0.274290 | 0.172698 | -0.113487 | -0.054765 | ... | 0.000106 | -0.000405 | 0.000445 | -0.000826 | |
| 8 | 0.304760 | -0.000961 | -0.208345 | -0.144434 | -0.118476 | 0.059597 | 0.088547 | -0.121423 | 0.074146 | -0.096181 | ... | -0.005987 | -0.017277 | 0.003719 | -0.003359 | - |
| 9 | 0.011059 | 0.004762 | 0.006528 | 0.005634 | 0.008735 | -0.013743 | 0.011087 | 0.001498 | 0.000554 | 0.013113 | ... | 0.002434 | -0.039870 | 0.005672 | 0.009950 | - |
| 10 | 0.591697 | 0.071660 | -0.169767 | -0.008426 | -0.006310 | 0.437719 | -0.182933 | 0.123234 | -0.299376 | 0.112155 | ... | 0.004326 | -0.000084 | -0.000974 | -0.002518 | - |
| 11 | 0.080267 | 0.005794 | 0.035958 | -0.058050 | 0.125360 | -0.135747 | 0.105575 | 0.079241 | -0.072190 | -0.060203 | ... | -0.006901 | -0.004180 | -0.002817 | -0.001839 | |
| 12 | 0.341668 | 0.086457 | 0.193503 | -0.064094 | -0.047100 | -0.122057 | -0.012701 | -0.072626 | 0.067352 | 0.194790 | ... | 0.001340 | -0.002633 | -0.007539 | -0.003790 | |
| 13 | 0.506885 | 0.542326 | -0.207942 | 0.050848 | -0.244445 | 0.185778 | 0.104581 | -0.010956 | 0.312586 | -0.337285 | ... | 0.001453 | -0.000050 | 0.000027 | -0.000662 | |
| 14 | 0.272399 | -0.136708 | 0.217816 | -0.040382 | -0.128070 | -0.063613 | -0.006944 | -0.090663 | -0.017232 | 0.060134 | ... | -0.002115 | 0.006270 | 0.001443 | -0.003728 | - |
| 15 | 0.497185 | -0.057117 | -0.089315 | -0.252641 | -0.216607 | -0.011793 | 0.137559 | -0.209403 | -0.060311 | 0.075885 | ... | 0.000168 | -0.000038 | 0.003242 | 0.005535 | - |

Gambar 5. 5 Kutipan *Principal Components Matrix*

### 5.2.4 Pengelompokan Menggunakan HAC

Pada proses ini dilakukan proses *clustering* pada ulasan. *Cluster* yang dihasilkan dengan menggunakan metode HAC membentuk satu *cluster* besar. Berikut ini ditunjukkan hasil *cluster* pada Gambar 5.6.




Gambar 5. 6 Dendrogram Hasil *Clustering*

Untuk menentukan kelompok-kelompok ulasan yang memiliki topik yang sama dibutuhkan titik *threshold* agar terbentuk menjadi beberapa cluster. Pada pengujian ini dilakukan *clustering* ulasan dengan menggunakan beberapa *threshold*. Banyaknya *cluster* yang dihasilkan dengan beberapa *threshold* dapat dilihat pada Lampiran 1. Setiap *threshold* yang diberikan menghasilkan masing-masing jumlah *cluster* yang berbeda.

Penentuan *threshold* berdasarkan keterkaitan topik pada setiap *cluster*. Ketika nilai *threshold* rendah, jumlah *cluster* yang dihasilkan tinggi, sehingga terdapat ulasan yang memiliki

topik yang sama terletak pada *cluster* yang berbeda. Sedangkan, jika nilai *threshold* tinggi, maka jumlah *cluster* yang dihasilkan sedikit yang dapat menyebabkan ulasan dengan topik yang berbeda tergabung pada satu *cluster* yang sama.

Kemungkinan topik yang diulas oleh pelanggan seperti pengiriman, warna, kualitas, ulasan pelanggan, pengemasan, gambar produk, pelayanan, respon, dan deskripsi.

Oleh karena itu, dalam penelitian ini *threshold* yang digunakan adalah 0.94, 0.95 dan 0.96 yang menghasilkan *cluster* sebanyak 15, 9 dan 4. Pemilihan nilai *threshold* tersebut dikarenakan dengan menggunakan *threshold* tersebut dihasilkan jumlah *cluster* yang mendekati kemungkinan topik yang diulas oleh pelanggan. Untuk nilai *threshold* 0.97 sampai *threshold* 1.00 dihasilkan satu *cluster* sehingga hal itu tidak memungkinkan untuk dijadikan pilihan *threshold.* Untuk nilai *threshold* dan banyak *cluster* dapat dilihat pada Lampiran 1. Untuk setiap *threshold* dijelaskan sebagai berikut:

a. *Clustering* dengan *threshold* 0.94

Dengan *threshold* 0.94 proses *cluster* menghasilkan 15 topik *cluster* dengan banyak ulasan dan tiga *keyword* yang dihasilkan dapat dilihat pada Tabel 5.1.

Berdasarkan Tabel 5.1 diatas mayoritas ulasan tergabung dalam *cluster* 13 dengan banyak ulasan adalah 1434. Sedangkan *cluster* dengan anggota paling sedikit adalah pada *cluster* 1 dan 3 yaitu sebanyak 2 ulasan.

Tabel 5. 1 Hasil *Clustering* dengan Threshold 0.94

| Cluster | Jumlah Ulasan | *Keyword* |
|---|---|---|
| 1 | 2 | ‘jalan’,’jalan jalan’, ‘lumayan’ |
| 2 | 8 | ‘selamat’, ’darat’, ‘bagus’ |
| 3 | 43 | ‘deskripsi’,’gambar’,’gambar deskripsi’ |
| 4 | 3 | ‘bagus’,’lanjut’,’cuma’ |
| 5 | 4 | ‘bintang’, ‘bintang bicara’,’bicara’ |
| 6 | 53 | ‘ragu’,’jual’,’sama’ |
| 7 | 5 | ‘bagus’, ‘terjangkau bagus’, ‘harga terjangkau’ |
| 8 | 48 | ‘jelek’, ‘hati’,’kualitas bahan’ |
| 9 | 11 | ‘cacat’,’mantap’,’sesal’ |
| 10 | 6 | ‘cocok’,’pas’,’mantap’ |
| 11 | 17 | ‘aman’,’cepat’,’cepat aman’ |
| 12 | 1593 | ‘sesuai’,’warna’,’pesan’ |
| 13 | 1434 | ‘sesuai’,’cepat’,’bagus’ |
| 14 | 24 | ‘resleting’,’resleting rusak’,’rusak’ |
| 15 | 55 | ‘lambat’,’kirim’,’lama’ |

Topik yang dibahas pada tiap ulasan dapat ditentukan berdasarkan keyword yang dihasilkan. Untuk topik tiap kelompok ulasan hasil *cluster* dapat dilihat pada Tabel 5.2.

Pada tabel 5.2 dapat diketahui bahwa cluster 11 dan 15 memiliki topik yang sama, yaitu pengiriman, namun memiliki sifat yang berbeda. Untuk cluster 4 disimpulkan dengan topik “Lain-lain”, hal itu dikarenakan tidak terdapat keterkaitan antar *keyword* yang dihasilkan.

Tabel 5. 2 Topik Ulasan Hasil *Clustering* dengan Threshold 0.94

| Cluster | *Keyword* | Topik |
|---|---|---|
| 1 | ‘jalan’,’jalan jalan’, ‘lumayan’ | Lumayan untuk jalan-jalan |
| 2 | ‘selamat’, ’darat’, ‘bagus’ | Barang sampai |
| 3 | ‘deskripsi’,’gambar’,’gambar deskripsi’ | Deskripsi gambar |
| 4 | ‘bagus’,’lanjut’,’cuma’ | Lain-lain |
| 5 | ‘bintang’, ‘bintang bicara’,’bicara’ | Rating |
| 6 | ‘ragu’,’jual’,’sama’ | Keraguan sama penjual |
| 7 | ‘bagus’, ‘terjangkau bagus’, ‘harga terjangkau’ | Harga Produk |
| 8 | ‘jelek’, ‘hati’,’kualitas bahan’ | Kualitas bahan produk |
| 9 | ‘cacat’,’mantap’,’sesal’ | Kekecewaan Pelanggan |
| 10 | ‘cocok’,’pas’,’mantap’ | Kesesuaian |
| 11 | ‘aman’,’cepat’,’cepat aman’ | Efektifitas pengiriman |
| 12 | ‘sesuai’,’warna’,’pesan’ | Kesesuaian warna |
| 13 | ‘sesuai’,’cepat’,’bagus’ | Kesesuaian dan kecepatan |
| 14 | ‘resleting’,’resleting rusak’,’rusak’ | Resleting produk |
| 15 | ‘lambat’,’kirim’,’lama’ | Pengiriman lambat |

b. *Clustering* dengan *threshold* 0.95

Pemotongan jarak *cluster* pada titik 0.95 menghasilkan 9 cluster. Pada *cluster* ini terdapat penggabungan *cluster* 7

dengan cluster 8 dan cluster 10 sampai *cluster* 15 pada hasil *cluster* dengan *threshold* 0.94 ke *cluster* 9. Sehingga hasil *cluster* dengan *threshold* 0.95 dapat dilihat pada Tabel 5.3 berikut:

Tabel 5. 3 Hasil *Clustering* dengan *Threshold* 0.95

| Cluster | Jumlah Ulasan | *Keyword* |
|---|---|---|
| 1 | 2 | ‘jalan’,’jalan jalan’, ‘lumayan’ |
| 2 | 32 | ‘selamat’, ’darat’, ‘bagus’ |
| 3 | 2 | ‘deskripsi’,’gambar’,’gambar deskripsi’ |
| 4 | 20 | ‘bagus’,’lanjut’,’cuma’ |
| 5 | 53 | ‘bintang’,’bintang bicara’,’bicara’ |
| 6 | 7 | ‘ragu’,’jual’,’sama’ |
| 7 | 8 | ‘bagus’, ‘harga’,’kualitas’ |
| 8 | 18 | ‘cacat’,’mantap’,’sesal’ |
| 9 | 1651 | ‘sesuai’,’cepat’,’bagus’ |

Berdasarkan Tabel 5.3 jumlah ulasan pada *cluster* 1 sampai *cluster* 6 sama dengan hasil *cluster* dengan *threshold* 0.94. Sedangkan untuk *cluster* dengan *keyword* ‘sesuai’, ‘cepat’, dan ‘bagus’ mengalami peningkatan menjadi 1651 ulasan. Untuk topik setiap ulasan dapat dilihat pada Tabel 5.4 berikut:

Tabel 5. 4 Topik Ulasan Hasil *Clustering* dengan *Threshold* 0.95

| Cluster | *Keyword* | Topik |
|---|---|---|
| 1 | ‘jalan’,’jalan jalan’, ‘lumayan’ | Lumayan untuk jalan-jalan |
| 2 | ‘selamat’, ’darat’, ‘bagus’ | Barang sampai |
| 3 | ‘deskripsi’,’gambar’,’gambar deskripsi’ | Deskripsi gambar |
| 4 | ‘bagus’,’lanjut’,’cuma’ | Lain-lain |
| 5 | ‘bintang’,’bintang bicara’,’bicara’ | Rating |
| 6 | ‘ragu’,’jual’,’sama’ | Keraguan sama penjual |
| 7 | ‘bagus’, ‘harga’,’kualitas’ | Kualitas dan harga |
| 8 | ‘cacat’,’mantap’,’sesal’ | Kekecewaan pelanggan |
| 9 | ‘sesuai’,’cepat’,’bagus’ | Kesesuaian dan kecepatan |

Pada *cluster* ini terdapat perbedaan topik dengan proses clusterimg sebelumnya. Hasil *cluster* ini terdapat ulasan yang membahas kualitas dan harga, sedangkan pada proses *cluster* sebelumnya kualitas dan harga terletak pada cluster yang berbeda. Hal itu terdapat penggabungan anggota cluster dengan cluster lainnya.

c. *Cluster* dengan *threshold* 0.96

Proses *cluster* dengan *threshold* 0.96 diperoleh hasil *cluster* sebagai berikut:

Tabel 5. 5 Hasil *Cluster* dengan *Threshold* 0.96

| Cluster | Jumlah Ulasan | *Keyword* |
|---|---|---|
| 1 | 2 | ‘jalan’,’jalan jalan’, ‘lumayan’ |
| 2 | 32 | ‘selamat’, ’darat’, ‘bagus’ |
| 3 | 2 | ‘deskripsi’,’gambar’,’gambar deskripsi’ |
| 4 | 1757 | ‘sesuai’,’cepat’,’bagus’ |

Berdasarkan Tabel 5.5 diatas, dihasilkan 4 jumlah cluster dimana setiap clusternya memiliki 2, 32, 2 dan 1757 ulasan. Pada proses *cluster* dengan threshold 0.96 cluster ke 1, 2, dan 3 memiliki hasil cluster yang sama dengan hasil *cluster* dengan *threshold* 0.96. Topik-topik cluster dapat dilihat pada Tabel 5.6 di bawah ini.

Tabel 5. 6 Topik Ulasan Hasil *Cluster* Threshold 0.96

| Cluster | *Keyword* | Topik |
|---|---|---|
| 1 | ‘jalan’,’jalan jalan’, ‘lumayan’ | Lumayan untuk jalan-jalan |
| 2 | ‘selamat’, ’darat’, ‘bagus’ | Barang Sampai |
| 3 | ‘deskripsi’,’gambar’,’gambar deskripsi’ | Deskripsi gambar |
| 4 | ‘sesuai’,’cepat’,’bagus’ | Kesesuaian dan kecepatan |

Dari ketiga hasil *cluster* dengan *threshold* - *threshold* tersebut beberapa keyword yang muncul beberapa kali di beberapa cluster. Hal itu dikarenakan kata tersebut memiliki

frekuensi kemunculan kata yang tinggi yang juga dapat mempengaruhi hasil perhitungan tf-idf tiap cluster

### 5.2.5 Analisis Hasil *Clustering*

Analisis hasil *cluster* digunakan untuk mengetahui *threshold* mana yang lebih optimal diantara penggunaan *threshold* lainnya. Hal itu dilakukan dengan menghitung rata-rata *presisi* dari setiap proses *cluster*. Untuk perhitungan *presisi* pada tiap *cluster* dengan menggunakan persamaan berikut:

$$presisi_i = \frac{jumlah\ ulasan\ yang\ sesuai}{jumlah\ ulasan\ hasil\ clustering}$$

Untuk penilaian ulasan yang dianggap sesuai yaitu ulasan yang mengandung minimal salah satu *keyword* yang telah diperoleh. Proses perhitungan presisi mula-mula dilakukan dengan menghitung presisi tiap *cluster*. Kemudian hasil presisi tiap cluster dapat diperoleh rata-rata presisi yang dijadikan untuk nilai presisi pada *threshold* tersebut.

a. Perhitungan presisi pada *cluster* dengan *threshold* dapat dilihat pada Tabel 5.7 berikut:

Tabel 5. 7 Presisi Hasil *Clustering* dengan *Threshold* 0.94

| Cluster | Jumlah Ulasan | Ulasan Terdeteksi | Presisi |
|---|---|---|---|
| 1 | 2 | 2 | 100% |
| 2 | 32 | 30 | 93.8% |
| 3 | 2 | 2 | 100% |
| 4 | 20 | 13 | 65% |
| 5 | 53 | 26 | 49.1% |
| 6 | 7 | 6 | 85.7% |
| 7 | 3 | 3 | 100% |
| 8 | 5 | 5 | 100% |
| 9 | 18 | 14 | 77.8% |
| 10 | 52 | 34 | 65.4% |
| 11 | 16 | 13 | 81.3% |
| 12 | 70 | 51 | 72.9% |
| 13 | 1434 | 809 | 56.4% |
| 14 | 24 | 19 | 79.2% |
| 15 | 55 | 42 | 76.4% |
| Rata-rata | | | 80.2% |

Berdasarkan Tabel 5.4 setiap cluster memiliki nilai presisi yang beragam. Nilai presisi yang diperoleh relative tinggi. Akan tetapi, masih ditemukan cluster dengan presisi yang cukup rendah yaitu cluster 5. Jika dilihat pada Tabel 5.1 cluster 4 memiliki *keyword* “bintang” memiliki nilai presisi 49.1%. Hal itu dikarenakan dalam *cluster* tersebut banyak anggota cluster yang mengandung *keyword* “ulas”. *Keyword* “bintang” dan “ulas” beberapa kali muncul bersamaan, sehingga kalimat yang mengandung kata “ulas” namun tidak memiliki kata “bintang” tergabung pada cluster tersebut.

Dengan perhitungan tiap presisi tiap cluster dihasilkan nilai presisi untuk proses *cluster* dengan threshold 0.94 adalah 80.2%.

b. Untuk *clustering* dengan *threshold* 0.95 memiliki presisi berikut:

**Tabel 5. 8 Presisi Hasil Cluster dengan Threshold 0.95**

| Cluster | Jumlah Ulasan | Ulasan Terdeteksi | Presisi |
|---|---|---|---|
| 1 | 2 | 2 | 100% |
| 2 | 32 | 30 | 93.8% |
| 3 | 2 | 2 | 100% |
| 4 | 20 | 13 | 65% |
| 5 | 53 | 26 | 49.1% |
| 6 | 7 | 6 | 85.7% |
| 7 | 8 | 8 | 100% |
| 8 | 18 | 14 | 77.8% |
| 9 | 1651 | 1032 | 62% |
| Rata-rata | | | 79.1% |

Berdasarkan Tabel 5.8 pada proses *clustering* dengan *threshold* 0.95 memiliki nilai presisi yang relative tinggi kecuali pada cluster 5. Hal itu sama dengan proses *cluster* dengan *threshold* 0.95.

Nilai presisi untuk *cluster* dengan *threshold* 0.95 lebih rendah daripada *cluster* dengan *threshold* 0.94. Hal ini dikarenakan penggabungan ulasan pada satu cluster satu ke cluster lain. Hal itu mengakibatkan ulasan yang tidak mengandung keyword yang dimiliki cluster tersebut.

c. Perhitungan presisi pada *cluster* dengan *threshold* dapat dilihat pada Tabel 5.9 berikut:

Tabel 5. 9 Presisi Hasil Cluster dengan Threshold 0.96

| Cluster | Jumlah Ulasan | Ulasan Terdeteksi | Presisi |
|---|---|---|---|
| 1 | 2 | 2 | 100% |
| 2 | 32 | 30 | 93.7% |
| 3 | 2 | 2 | 100% |
| 4 | 1757 | 829 | 47.1% |
| Rata-rata | | | 85.2% |

Beradasrkan Tabel 5.9 nilai presisi dari proses *cluster* dengan *threshold* 0.96 adalah 85.2%. Hal itu berarti 85.2% ulasan telah tergabung pada cluster yang sesuai.

Dari perhitungan ketiga proses *cluster* tersebut dapat diketahui bahwa *cluster* dengan *threshold* 0.96 lebih tinggi dibandingkan dengan proses cluster dengan *threshold* lainnya. Hal itu dikarenakan tiap cluster memiliki ulasan yang memiliki topik yang sesuai.

# BAB VI
# KESIMPULAN DAN SARAN

## 6.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut:

1. Implementasi *text mining* untuk mengelompokan ulasan dimulai dengan melakukan praproses data. Setelah data bersih dilakukan perhitungan TF-IDF sebagai proses pembobotan kata. Hasil dari proses TF-IDF merupakan matriks *document-by-term.* Matriks tersebut kemudian diuraikan menggunakan SVD. Hasil penguraian tersebut dilakukan pereduksian dimensi matriks sebesar *k* sehingga menjadi *principal components matrix.* Kemudian melakukan *cluster* dengan algoritma *Hierarcical Agglomerative Clustering* pada *principal components matrix*. Sebelum dilakukan *clustering*, terlebih dahulu mencari jarak tiap dokumen dengan menggunakan *cosines similarity* untuk mengetahui kemiripan dokumen.
2. Berdasarkan perhitungan presisi, *clustering* dengan *threshold* 0.96 memiliki nilai presisi yang lebih tinggi daripada dengan menggunakan *threshold* yang lainnya. Nilai presisi proses *clustering* tersebut adalah 85.2%.

## 6.2 Saran

Dari penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan penelitian selanjutnya, diantaranya:

1. Terdapat penulisan kata yang beragam tapi memiliki maksud yang sama. Agar didapat hasil yang baik, sebaiknya membuat *dictionary* agar tidak mempengaruhi pembobotan dan *clustering* yang dihasilkan.

2. Dalam penentuan stopword lebih dipilah lagi, agar kata yang dihasilkan lebih baik.

## DAFTAR PUSTAKA


[1] Bakar, Noor H., Zarinah M. Kasirun, & Hamid A. Jalab. 2015. **Toward Requirement Reuse : Identifying Simlar Reqirements with Latent Semantic Analysis and *Clustering* Algorithms.** *International Journal of Advances in Computer Science & Its Application- IJCSIA.* vol. 5. no. 1. pp. 58-63.

[2] Fry, Chantal & Sukanya Manna. 2016. **Can We Group Similar Amazon Reviews : A Case Study with Different *Cluster* Algorithm.** *IEEE Tenth International Conference on Semantic Computing,* pp. 374-377.

[3] Rozi, Fahrur, dkk. 2014. **Pelabelan Klaster Fitur Secara Otomatis Pada Perbandingan Review Produk.** Jurnal Teknologi Informasi dan Ilmu Komputer. Hlm. 55-61.

[4] Griva, Anastasia, Cleopatra Bardaki, & Dimitris Papakiriakopouilos. 2018. **Retail Business Analytics : Computer Visit Segmentation using Market Basket Data.**

[5] Sarwono, J. & Prihartono, K. 2012. **Perdagangan Online: Cara Bisnis di Internet.** Elex Media Koputindo.

[6] Nugroho, Adi. 2006. **E-Commerce - Memahami Perdagangan Modern Di Dunia Maya.** Bandung: Informatika.

[7] KM, Shivaprasad & T. Hanumantha Reddy. 2016, **Text Mining : An Improvised Feature Based Model**

**Approach.** in 2*nd International Conference on Applied and Theoritical Computing and Communication Technology.*

[8] Feldman, Ronen & James Sanger. 2007. **Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data**. New York: Cambridge.

[9] Nugroho, E. 2011. **Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp.** Malang: Universitas Brawijaya.

[10] Ledy, A. 2011. **Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia**. in *Konferensi Nasional Sistem dan Informatika 2009.*

[11] Manning, Christoper D.,Prabhakar Raghavan, & Hinrich Scutze. 2008, **Introduction to Information Retrieval.** Cambridge University Press: Cambridge.

[12] Jo, Taeho. 2019. **Text Mining : Concepts, Implementation, and Bid Data Challenge.** Springer International Publishing , Switzerland.

[13] Berry, M.W, S.T. Dumais & G.W. O'Brien. 1994. **Using Linear Algebra for Intelligent Information Retrieval.**

[14] Wicaksono, D.W. 2014. **Sistem Deteksi Kemiripan Dokumen Teks Menggunakan Model Bayesian Pada Term Latent Semantic Analysis**. Tugas Akhir, Institut Teknologi Sepuluh Nopember, Surabaya

[15] Baker, K., **Singular Value Decomposition Tutorial.** 2005. [Online]. Available: davetang,org.

[16] Han, Jiawei, Micheline Kamber & Jian Pei. 2012. **Data Mining : Concepts and Techniques 3rd edition**. Waltham: Morgan Kauffman Publisher

**Lampiran 1**
**Threshold dan Jumlah Cluster**

| Threshold | Jumlah Cluster | Threshold | Jumlah Cluster |
|---|---|---|---|
| 0,01 | 1581 | 0,51 | 436 |
| 0,02 | 1580 | 0,52 | 408 |
| 0,03 | 1577 | 0,53 | 384 |
| 0,04 | 1574 | 0,54 | 364 |
| 0,05 | 1560 | 0,55 | 344 |
| 0,06 | 1540 | 0,56 | 324 |
| 0,07 | 1531 | 0,57 | 308 |
| 0,08 | 1514 | 0,58 | 290 |
| 0,09 | 1489 | 0,59 | 274 |
| 0,10 | 1471 | 0,60 | 264 |
| 0,11 | 1445 | 0,61 | 248 |
| 0,12 | 1429 | 0,62 | 233 |
| 0,13 | 1399 | 0,63 | 223 |
| 0,14 | 1380 | 0,64 | 216 |
| 0,15 | 1352 | 0,65 | 210 |
| 0,16 | 1331 | 0,66 | 202 |
| 0,17 | 1307 | 0,67 | 192 |
| 0,18 | 1282 | 0,68 | 185 |
| 0,19 | 1265 | 0,69 | 181 |
| 0,20 | 1242 | 0,70 | 177 |
| 0,21 | 1221 | 0,71 | 172 |
| 0,22 | 1187 | 0,72 | 168 |
| 0,23 | 1167 | 0,73 | 162 |
| 0,24 | 1134 | 0,74 | 155 |
| 0,25 | 1114 | 0,75 | 153 |

| Threshold | Jumlah Cluster | Threshold | Jumlah Cluster |
|---|---|---|---|
| 0,26 | 1086 | 0,76 | 148 |
| 0,27 | 1067 | 0,77 | 145 |
| 0,28 | 1028 | 0,78 | 137 |
| 0,29 | 1001 | 0,79 | 133 |
| 0,30 | 976 | 0,80 | 125 |
| 0,31 | 952 | 0,81 | 120 |
| 0,32 | 925 | 0,82 | 112 |
| 0,33 | 893 | 0,83 | 106 |
| 0,34 | 865 | 0,84 | 100 |
| 0,35 | 839 | 0,85 | 90 |
| 0,36 | 807 | 0,86 | 81 |
| 0,37 | 784 | 0,87 | 74 |
| 0,38 | 765 | 0,88 | 70 |
| 0,39 | 728 | 0,89 | 60 |
| 0,40 | 708 | 0,90 | 52 |
| 0,41 | 666 | 0,91 | 43 |
| 0,42 | 639 | 0,92 | 33 |
| 0,43 | 611 | 0,93 | 23 |
| 0,44 | 597 | 0,94 | 15 |
| 0,45 | 577 | 0,95 | 9 |
| 0,46 | 553 | 0,96 | 4 |
| 0,47 | 532 | 0,97 | 1 |
| 0,48 | 505 | 0,98 | 1 |
| 0,49 | 486 | 0,99 | 1 |
| 0,50 | 466 | 1,0 | 1 |

## Biodata Penulis

**Li'Izza Diana Manzil** atau yang akrab dipanggil Li'Izza atau Diana ini lahir di Lamongan, 12 Februari 1997. Penulis menempuh Pendidikan mulai dari TK Muslimat NU Simo (2001-2003), MI Tarbiyatul Banat Simo (2003-2009), SMP Negeri 1 Karanggeneng (2009-2012), SMA Negeri 2 Lamongan (2012-2015). Setelah itu melanjutkan studi ke jenjang S1 di Departemen Matematika ITS pada tahun 2015-sekarang melalui jalur SNMPTN dengan NRP 06111540000049.

Selama kuliah penulis aktif di Himpunan Matematika ITS (HIMATIKA ITS) sebagai staff Internal Affair Himatika ITS (2016/2017) dan Secretary of Internal Affair Himatika ITS (2017/2018). Penulis juga aktif di Lembaga Dakwah Jurusan (LDJ) Ibnu Muqlah sebagai staff Dana Usaha di periode 2016/2017. Selain itu, penulis juga pernah melakukan kerja praktik di PT. Kereta Api Indonesia (Persero) Bandung, Jawa Barat. Selama penulisan laporan Laporan Tugas Akhir ini penulis tidak lepas dari kekurangan, untuk itu penulis mengarapkan kritik dan saran mengenai laporan Kerja Praktik ini yang dapat dikirimkan melalui email manzildiana2@gmail.com.